Executive Summary

Much effort has been devoted to improving the realism of virtual body figures in order to create more and more appealing virtual characters in films and games. Within this area of virtual creation, virtual clothing modeling is a significant element in achieving character realism. For the animators and modeler, modeling clothes is time consuming and knowledge intensive work that could take 2 to 3 hours per outfit, and must include creating and sewing together multiple 2D patterns. Even to simply design a similar garment but which has different proportions in different parts of body, the modeler must repeat that similar work from scratch. As demand for the virtual characters in feature films and games has increased, the requirement for virtual clothes has also grown enormously. Therefore, virtual garment transference and fitting has been attracting broad attention among researchers.

This system was developed as a plug-in on the Maya platform; it was programmed by MEL: the Maya embedded language. Firstly, the system reads the data of the figure model in the scene or from the file. It then works out the basic garment parameters for the figure model. The calculated garment parameters were used for main garment template modeling. From the user interface, the user can combine any components they like on the garment template. The basic fitting is achieved according to the garment and figure model the user selects. This system further allows users to adjust the local shape of cloth as they wish while still preserving the fabric's design despite deformation.

The main techniques involved in this paper are virtual garment modeling and virtual garment fitting. This paper will compare the result of different methods from the previous literature and present the strengths and shortcomings of these methods. It will then propose a mesh based garment template modeling method for achieving automatic garment modeling and a new garment proportional fitting method for achieving garment fit automatically.

- The mesh based garment template automatic modeling method involves creating main bounding surface, and generating automatically the extruding components of garment based on users' needs.
- The automatic garment proportional fitting method involves feature points extracting, feature points matching and local shape deformation.

Catalogue

1. Introduction

With the development of 3D technology, shopping online is now a common occurrence in people's daily lives. We enjoy the convenience of this new tool. With just a few clicks, various kinds of clothes you are displayed right there on the screen. Some shopping systems even offer virtual fitting rooms in which customers can dress a virtual model in their favorite clothing articles. However, there are still some limitations in these online shopping systems. The greatest weakness of the online cloth shopping system is that the customer cannot choose the clothes according to their figure. Customers don't know whether the clothes suit them or not. However, body scanning technology is now becoming well understood and simplified in the Biology and Medical field. People who need to record health condition were scanned in a scanning system that scanned the body horizontally and vertically, then analyzed the key point in the graphs. A similar method can be used in the garment shopping online.

Much effort has also been devoted to improving the realism of virtual body figures, which are promoted more and more in films and games. In this area, virtual clothing modeling is a significant element in achieving character realism. For the animators and modeler, modeling clothes is time-consuming and knowledge-intensive work. It could take them 2-3 hours to generate a figure's outfit, include creating 2D pattern and sewing them together. Besides, as demand for the virtual characters in feature film and game increased, the number of virtual clothes requirement also increases. If you want to design a similar outlook garment but which has different proportions in different parts of body, the modeler has to repeat the similar work from scratch. Therefore, the virtual garment transfer or fitting receives widespread attention from researchers.

This article will first outline the strengths and weaknesses of the three major cloth modeling and fitting methods: pattern sewing modeling method, sketch-based garment modeling method and design preserving proportional scaling method. After, the significant techniques involved will be analyzed, feature points extracting technique, feature points matching technique, 3D pattern extruding method. Based on the analysis of these techniques, a new mesh based garment template modeling and pattern extruding method in modeling process is presented; the ways in which this method is more efficient and accurate than the traditional pattern sewing method will be subsequently addressed. On the basis of the feature matching method, the new proportional fitting method proposed is proven to achieve garment fitting automatically. This system was developed as a plug-in on the platform Autodesk Maya, which is 3D computer graphic software that provides mature toolsets and an environment in which the animator or programmer can develop challenging plug-in that could alleviate the workloads involved in increasing the realism of the virtual world. The focus of this project is not the simulation of cloth; there are already many mature software or plug-ins for cloth simulation, like SyFlex, NVIDIA APEX, and even the internal *ncloth* function, which focuses full attention on the simulation of cloth. Rather, the project is interested in the research on automatic cloth modeling and fitting, which is still a new area with great potential. The greatest challenge of this project was the detection of the position of feature points on figure models and virtual garments using the 3D coordinates measuring system.

1.1 Aims and Objectives

Overall the main aim of this project is to develop a Maya plug-in that could perform the clothes modeling process automatically and allow users to fit clothes to figure models by just selecting the garment style and components of the cloth in a user interface. For example, users can choose to model a short-sleeved shirt with collars and pockets. Achieving this manually would through the program would then involve no more than adjusting the vertices of the cloth along the direction of normal, which would adjust details according to the users' needs while preserving the cloth design.

The first objective is to set up the initialization process of the plug-in and embed it into the Maya interface. Specifically, first of all, the plug-in should be compatible with different operating system. The program was encapsulated into a packet with one outward interface for the user to install it. Secondly, the user interface is designed with a clear structure and layout. It should be similar to the default layout of Maya. The operation steps are simple and understandable.

The second objective is preparing the figure model. Reset the pivot of the figure model is one of the significant steps. the pivot should be in the center of the feet of the figure. The standardization of the pivot's position will not change the original characteristic of human model, but will make it easy to achieve standardization of the cloth modeling and fitting process. Besides, it's a common step for an animator before they rig the human model.

The third objective is creating the cloth template. The virtual cloth template is a mesh based template, transformed from the basic geometry. The collars, pockets, and sleeves will be formed by extruding at the position that corresponds with the cloth template. The extruding operation refers the orientation of the figure's joint. Different style of clothes modeling method stored in different functions or databases, where can be edited and expanded easily by modelers or programmers, which makes this better.

The fourth objective is fitting the created clothes to any one or several figure models automatically. This is the challenging section of this project, because it is so difficult to detect the position of the feature points on different body parts and virtual garments in the 3D coordinates measuring system.

The fifth objective is achieving local deformation. By adjusting the vertices of cloth along the direction of normal to satisfy the various kinds of shapes of the human body, from the thin frame of a teenage boy to the big stomach of pregnant woman. This local deformation method can be seen as an extension of smooth skinning. In order to preserve the style of cloth, all the vertices can only be moved along the normal and not all the vertices can be adjusted. The feature points such as the end point on the shoulder must be fixed.

This project will focus on exploring the men's shirt modeling method. Users can choose any key component on the shirt they like, such as collars, pockets, short sleeves or long sleeves. The default color of the shirt is white and the material is set to Lambert. These attributes are attached to the cloth automatically in the modeling process.

1.2 Structure

The remainder of the dissertation consists of the following sections:

Chapter 2 explains and comparing the background material of the garment modeling and fitting. My study focus on three major methods: sewing approach, sketch-based approach and design preserving garment transfer approach. Chapter 3 summarize the key techniques involve in the method discussed in chapter 2. And point out the correlation to my project. Chapter 4 presents the architecture of Maya from Dependency Graph to attributes of *nodes*. Chapter 5 presents the project specifications, which include software environment, system framework, operating flow chart, garment fitting criteria and the standard parameter of garment. Chapter 6 explains the details of methods I used in the system with the corresponding algorithm. The methods are presents in order of work flow. Results are analyzed in Chapter 7 from three different conditions or aspects. Then based on the experiment results, in chapter 8, I evaluated the completion and limitations of this design.

2. Backgrounds

There are three problems one can associate with clothing a virtual figure model: the design and modeling of the clothes, the matching of the cloth to the character, and making the final result look physically correct. The third problem is typically about cloth simulation, which is not the main focus of this research, due to its technical maturity. Although there have been a number of papers focusing on virtual garment modeling and fitting published in previous years, they have all generally had their limits. The approaches put forth by these different papers can be organized into three types of methods: the sewing approach, the sketch-based approach, and the design preserving garment transfer approach. However, as all these methods are based on the workflow of real garment manufacture process, before comparing the strengths and weaknesses of these methods, it is necessary to summarize the key techniques involved in real cloth making process.

2.1 Real Garment Manufacture Process

In general, the real clothes making process involves the clothing design, grading, fabric cutting and sewing step. Firstly, designers prepare the sketch for the garment by drawing the rough sketch in paper. It has the same demand for modelers in this step. Secondly, pattern makers develop the cloth pattern by two means: the manual method and the CAD method (Grace I. Kunz 2004). Nowadays, an increasing number of companies tend to adopt the CAD method in their garment production. The largest problems of the manual method are time-consumption and imprecise data. Clemens Groß and Arnulph Fuhrmann (2003) adopted the CAD method in his pre-positioning dressing study. Then, the aim of grading is to create patterns for multiple sizes. For the virtual garment, it is a necessary step to make the clothes fit the various shapes of a figure. In the study of Remi Brouet and Alla Sheffer (2012), the modelers use the proportional scale method to successfully transfer a cloth from one body to another. The part of the matching process in this project is based on their proportional scale method, and the cutting and sewing process is remotely related to the design at hand. The garment cutting and sewing process has been replaced by extruding method in my project, which is much easier to perform than the process of sewing two patterns together.

2.2 Garment Sewing Approach

2.2.1 Pre-positioning of 2D Pattern Sewing Method

Traditional virtual garment modeling is still done manually by modelers and animators. This is similar to the real clothes making process mention above. Clemens Groß and Arnulph Fuhrmann (2003) proposed a pre-positioning garment modeling method based on the real clothes making process. The main idea of their pre-positioning algorithm can be separated into three parts: creating bounding surfaces for the body segments, positioning cloth pattern onto bounding surfaces, and sewing the cloth pattern together.

Firstly, Arnulph Fuhrmann and Clemens Groß (2003) believe that garments can be separated into various bounding surfaces. Take the coat pictured below as an example. It includes two bounding surfaces for the arms, two bounding surfaces for the collars, and four bounding surfaces for the body (breast and back). The different parts of 2D cloth bounding surface are positioned around the figure model and sewed up afterward. These bounding surfaces correspond to the body segments like arms, legs, breast/back, hip and neck. Then, specific feature points are generated on each of these body segments. The purpose of generating these feature points is to mark the special position of the figure for bounding surfaces. Generally, one bounding surface connects to one feature point. One feature point fits one body segment. For most modelers, knowledge of tailoring is not required for them. The sewing information is attached with bounding surfaces. The figure below demonstrates the garment patterns with the sewing information.

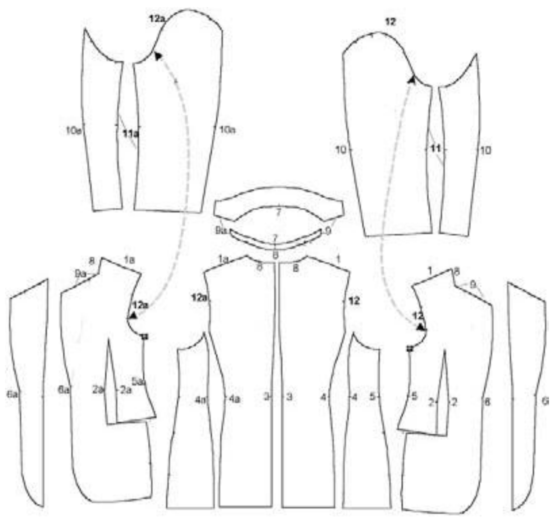


Figure 1: Cloth patterns with sewing information (Clemens Groß and Arnulph Fuhrmann. 2003)

Clemens Groß and Arnulph Fuhrmann (2003) also claim that the figure model has to face to a certain direction, such as the X axis. The posture of the figure should be similar to the one a real person would be required to assume in order to be measured by a tailor in real life. The figure has to be standing straight. The arms should stretch to the sides in a distance to the rest of the body. The legs cannot be closed together. The reason that the figure should pose in this kind of posture is for the convenience of recognizing the exact part of body and determining the positions of the feature points. It is prepared for rigging the skeleton afterwards. This kind of posture can be used for reference in this project. Figure 2 displays a figure posture example with feature points.

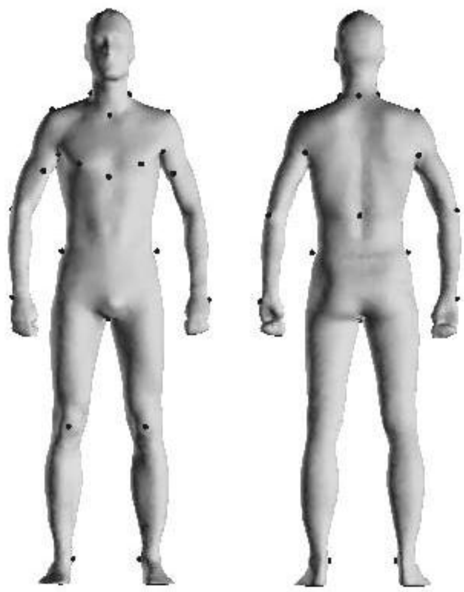


Figure 2: The feature points on figure model (Clemens Groß and Arnulph Fuhrmann.2003)

2.2.2 3D Pattern Sewing Method

In comparison to the 2D fabric pattern sewing technique, there is another 3d geometric garment modeling method that can save a large amount of time in modeling the same cloth. The 3D pattern sewing method functions without a great many patterns needing to be sewed. Typically, one simple bounding surface that designers construct fits one part of body. Taking the coat as an example again, it will need only one bounding surface for the arm, one bounding surface for the collar, and one bounding surface for the upper body. Any other style of cloth can also be separated into this kind of category. The bounding surface of upper body is the center of the other bounding surfaces, which is also the starting point of the modeling process. This method could achieve good modeling for suitable fitting cloth. However, it is hard to define the correct cloth surface when the cloth is quite loose. Volino et al (2005) and Umetaniet al (2011) provide another interactive 3D garment modeling system that allows for instant 3D feedback in answer to changes in clothes patterns by using a real-time physics-based simulation. This process still depends on the user's ability to make the right design choices in order.

2.2.3 Summary

There are several significant achievements in the garment sewing method, and an increasing number of researchers are dedicating themselves to this study. Even apart from the problems mentioned above, though, there are still many other unsolved problems in this area. For example, some studies only focus on researching in standard body size, but failed to consider the requirement of smart and personalized. There are no figures in the world with exactly the same size, and the 2D sewing method always requires some professional knowledge to design clothing pieces. It still cannot achieve automatic in garment sewing process. Based on the bounding surfaces theory (Arnulph Fuhrmann and Clemens Groß. 2003) and 3D pattern sewing technique, this paper puts forth a garment template modeling system with a 3D pattern extruding method in modeling garment component. Details of the method and its development will be described in Chapter 5.

2.3 Sketch-based Garment Modeling Approach

The typical sketch based method models 3D cloth around a 3D figure model by drawing the contour lines outlined by the user in the 2D. This method provides an intuitive way for users to design their favorite cloth patterns, and does not demand that users have professional knowledge of cloth design and cloth sewing (XU Wenpeng and Qiang Xiaohuang.2010).

Before cloth modeling, the median sagittal plane of the figure model has to be calculated (Xu Wenpeng. 2004). This is demonstrated at the left of the image in Figure 3 below. The process of garment modeling can then be divided into several key steps, the first step being the recognition of human feature points. Lu Guodong (2005) presents a feature point detecting method: judge the shape of cross section of figure model after intersecting the horizontal plane into the figure. The effect can be seen in the right image of Figure 3. However, though this feature detecting method can identify feature points faster than its corresponding alternatives in literature, it cannot detect the exact position of some non-boundary feature points when they are in the same cross section. The second step is constructing garment section lines. The figure section lines are created based on the recognized feature points. The function of these section lines is to control the section shape of cloth feature positions (M. Cani and J. Hughes. 2007). The next step is establishing silhouettes of cloth. The silhouette is constructed by connecting the key points on section line. The key points can be acquired by calculating the intersection points between the median sagittal plane and the coronal plane of figure model. The purpose of constructing the silhouettes is to control the outline of the cloth. The combination of the section line and the silhouette can form the basic cloth wireframe. Finally, the basic cloth surfaces are modeled based on the cloth wireframe with topology in each surface part (XU Wenpeng and Qiang Xiaohuang.2010).

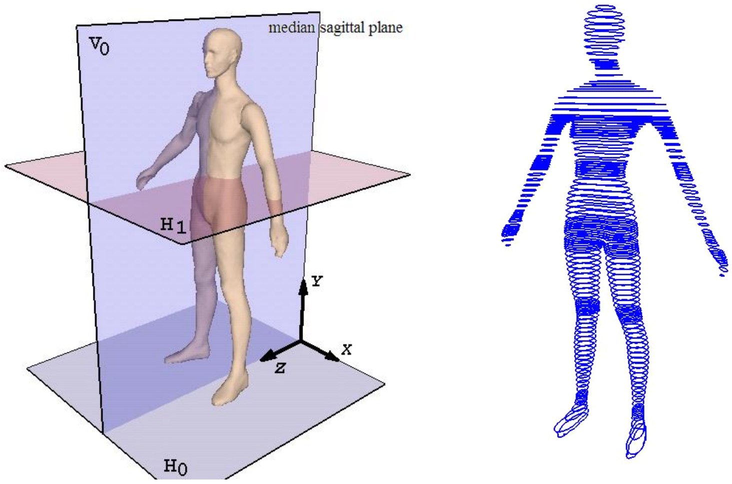


Figure 3: Median sagittal plane of figure and figure section intersected. (XU Wenpeng and Qiang Xiaohuang.2010)

2.3.1 Summary

The research based on the Sketch-based garment modeling approach presents a more appealing and more maneuverable option than the traditional pattern sewing method. Additionally, it accelerates the modeling process to a large extent. But the results achieved by much of the existing related research lack realism (Cody Robson, Ron Maharik. 2011). Specifically, the garment shape often presents an unnatural looking when the drawn sketch using a simple shape. Decaudin et al (2003) provide a method by adding procedural folds on clothes to increase realism. In this case, though, one problem is the heavy reliance on the seams lines that separate the different cloth patterns and fail to achieve meaningful results when the provided seams do not admit a developable interpolant (Rose, K, She_er, A. 2007). Another problem of sketch-based method is that it requires a large number of user input per output, which cannot achieve the automation in modeling process.

2.4 Design Preserving Garment Transfer Method

Research performed on the garment modeling section and garment fitting section separately has found these methods will not limit the users' choice in design of clothes. Due to the limits of the existing automatic garment modeling techniques, many users have their own ideas about garment design in their work. Therefore, it is necessary to develop a garment fitting method for any design of cloth. The majority of garment transferring method includes two aspects: feature points extracting and matching.

The most classical method among them is known as grading or pattern grading, which achieves the transfer required at the appropriate level of significantly specialized expertise and relies on a set of pre-computed parameter table for standard body size (Moore et al. 2001). This method is not suitable for a situation in which characters do not have the same body proportions; in other words, there is no standard garments' parameter table suitable for any kind of body shape. Another garment transferring method is skinning-type (Cordier et al. 2003; Wang et al. 2005). Cordier et al (2003) adopt skinning only to tight parts of the cloth. His method globally scales the cloth patterns for the loose areas along two major axes on the basis of changing in figure's height and circumference of hip. Such axial scaling always leads to poorly shaped clothes, and his method fails to preserve the original design of the cloth.

In 2012, Remi Brouet and Alla Sheffer presented an automatic method that could transfer a garment between two different shapes of figure model. The greatest achievement of this method was preserving the design during the transferring process. It can also handle the multiple layers of clothes, even though Remi Brouet and Alla Sheffer (2012) did not provide a way to automatically create the garment. The research significance of their project was in the alleviation of the workload of circle repeat modeling work. Because it is pervasive that diverse persons wear the same design cloth in real life or in films and games. The common relationships of these people are mother and daughter, the company's staff member, school uniform and so on. The work of Remi Brouet and Alla Sheffer can be summarized into the following steps. Their method aims at generating a proportionally scaled virtual cloth that meets the demand of location and fit criteria. This scaled one is then used for the normal preserving optimization, which creates the target transferred cloth. To perform the proportional scaling an important step is selecting the reference points. The details of the feature points extracting algorithm of Remi Brouet and Alla Sheffer (2012) will be described in Chapter 3. The feature points on the figure are calculated based on the cross-parameterization method proposed by Vladislav Kraevoy (2004). Then each feature point extracted from the garment is matched to the corresponding feature point on figure with an offset.

2.4.1 Summary

The aim of garment transferring is to make one kind of clothing fit various shapes of manikins automatically. The original pattern design should be preserved in the fitting process. The main techniques involved in garment transferring include feature points extracting and feature points matching. Compared to the existing corresponding method, Remi Brouet and Alla Sheffer (2012) propose a better solution in transferring the garment among complex figure models and proportional fitting automatically. However, one limitation of their method is that the source garment has to fit one manikin well before transferring, since the feature points on the source garment are obtained by calculating its relationship to the closest bone and skin.

3 Key Techniques

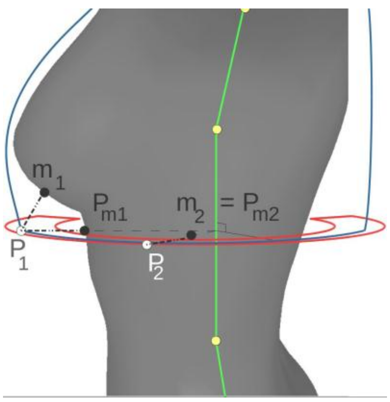
After examining the strengths and weaknesses of the existing methods of virtual garment modeling and fitting and after reviewing the study of Brouet and Sheffer, I proposed the mesh based cloth template automatic modeling method and automatic garment proportional fitting method. It is a complex project that involves several technical challenges, including feature points extraction, feature points matching and 3D garment pattern extruding. Though some of these techniques may have been addressed in Chapter 2, this chapter will summarize and compare the existing method by addressing these difficulties. Following this, the specific potential solutions that the newly proposed method in this paper possesses will then be presented.

3.1 Feature Points Extracting

Feature points extracting is a key link of garment fitting process. In most studies, both the feature points on figure model and virtual garment have to be extracted.

Zhong Li and Xiaozhang Jin (2009) extracted the feature points from the garment and figure model according to curvature property in their study; in doing so, they found that the judgement of feature points is related to the calculation of curvature values and their derivatives (Ohtake Y, Belyaev A 2004). However, for the discrete triangular mesh model, the method of Zhong Li and Xiaozhang Jin cannot explicitly compute the derivative for each vertex.

Though the work of Cordier et al and Meng et al uses the closest skin point as a reference, Remi Brouet and Alla Sheffer (2012) did not think this was good enough. They believed that it could lend to points slide along the body of virtual figure model. Therefore, the Remi Brouet's and Alla Sheffer's extracting the feature points method is better in terms of accuracy. The details of the algorithm can be summarized below:



(a)

Figure 4: (a) It can be seen in the figure that the p_1's and p_2's closest skin points m_1 and m_2 are selected as reference points. This extracting method is not the best, since the reference points can move apart on the target character, leading to an undesirable vertical displacement. Selecting the orthogonal direction, will map points to a circle around a same bone point, which could preserve the relative height. (Remi Brouet and Alla Sheffer 2012).

The reference points that Remi Brouet and Alla Sheffer (2012) obtain are computed in the following manner:

1) For each vertex $p_{g,}$ firstly its closest points p_b are computed on each bone of animation skeleton (p_b is possibly an extremity).
2) For each bone, p_m is defined as the input character point closest to p_g along the segment $[p_b, p_g]$.
3) "Selecting one or more reference points, both the distance from p_g to each of the intersections point p_m and the angle between the bone axis v_b and the vector p_g - p_b should be considered when these two vectors set to be orthogonal. The pairs (p_b, p_m), which are selected as reference for $p_{g,}$ are those that minimize a combination of these two metrics$\|p_m, p_g\| e^{\frac{-<v_b,v>^2}{\sigma^2}}$, where they set σ = 0:1." (Remi Brouet and Alla Sheffer 2012).
4) Finally, for each reference pair (p_b, p_m), each point p_g should be associated with an offset $o = \|p_m, p_g\|$.

Remi Brouet and Alla Sheffer (2012) optimized the method of feature points extracting in terms of accuracy and efficiency. Specifically, they associated vertices on the clothing with the skin and skeleton of figures in order to obtain the feature points' positions in source garments. They proposed that each feature point on the source garment was associated with an offset that preserved the distance between cloth and figure in transferring process. However, their method only focuses on garment transferring between various figures. In other words, the source garments are required to model and fit the source figure in advance. Though this method performs well in same cloth transferring between different body sizes, it cannot satisfy the demand of users for garment variety in a short time.

3.2 Feature Matching

Based on the study of Remi Brouet and Alla Sheffer (2012), the basic idea of feature matching can be summarized as making the constant offset vector from proper chosen reference points on the figure model replace the constant positions in skeleton frames. Dealing with changes in proportions of skeleton and making use of skin-based offsets allows this method to cope with changes in shape of figure, such as shoulder width. The algorithm of this method is as follows:

For every cloth point with respect to the target character, the computed reference points and offsets are used to gain the proportionally scaled locations p_g. If the proportionally scaled location p_g is certain, the first step of garment fitting is complete.
"Firstly, for every reference pairs (p_b, p_m) on the source character, the corresponding pairs ($p^c{}_b$, $p^c{}_m$) on the target are calculated by using the cross-parameterization. If the point p_g has a single pair of (p_b, p_m), the $p^c{}_g = p^c{}_m + o*v_{bm}$ where v_{bm} is a unit vector in the direction of ($p^c{}_b$, $p^c{}_m$). If there are multiple reference pairs for the cloth vertex p_g, a weighted sum of the positions dictated by each reference is used, involving non-linear weights to get a sharp transition $w_1 = \frac{\arctan(5*(t-0.5)+\frac{\pi}{2})}{\pi}$ and $w_2 = 1 - w_1$, where t is the coordinate of the projection of the garment vertex p_g on the segment between the two relevant bone reference points."(Remi Brouet and Alla Sheffer 2012).

3.3 3D Pattern Extruding

Based on 3D pattern sewing technique and the manual garment modeling process in Maya, this paper proposes a pattern extruding method that satisfies the need for flexibility in the design of garments. This is accomplished by a set of seams that indicate which part of a pattern has to be extruded from the corresponding part of another pattern. Extruding information is attached to the corresponding vertices on the surface. Extruding information includes names and positions of vertices or edges will be extruded, extruding direction, and extruding distance. The extruding direction is obtained by calculating the unit vector of the corresponding figure's joint. The extruding method is also the manual garment modeling process in Maya, which makes the whole process automatic. The next step requires the up vector and the nearby position of a feature point in order to move the pattern to the bounding surfaces. Though Clemens Groß (2003) presented a way to detect the cross section of a body part and the cloth to be wrapped around, the errors inherent in this part can be avoided using the new method being presented now.

4 Maya Architecture

Since the developed plug-in works within the framework of the Maya system, it is essential to present the structure and fundamental rules of Maya system. The architecture of Maya is quite different from other 3D applications. At a fundamental level, Maya system can be broke down into three major components; the diagram is shown in Figure 5 below.

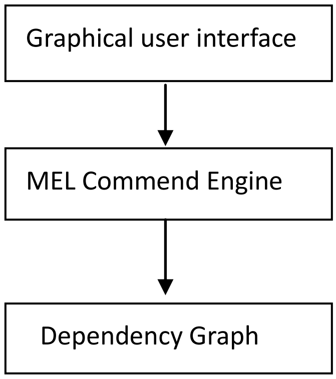


Figure 5: Maya system (David A. D. Gould. 2003)

The basic concepts in Maya which related to my project are as follows:
MEL Command Engine: When the designers interact with the user interface, Maya is actually issuing MEL commands. These commands are sent to the Command Engine directly where they are interpreted then executed.
Dependency Graph: The majority of the MEL commands operate on the Dependency Graph. The reason for this is that the dependency graph can be intuitively considered as the entire scene. In other words, the Dependency Graph is the heart of Maya, and defines what data is in the current scene.
Scene: The scene is all the important data and information that constitute the 3D world, including the objects, animation, materials, and so on.
Nodes: The data and their operations encapsulate in the Dependency Graph as *nodes*. The node is the most fundamental building block of Maya. Common 3D tasks can be completed by connecting a series of nodes together. The data is fed through the network of nodes from the very first to the last. Besides, adding new functionality in Maya means creating a new node. (David A. D. Gould. 2003). All the data flow and data structures defined in this project are based on nodes; the inner structure of node is shown in Figure 6.

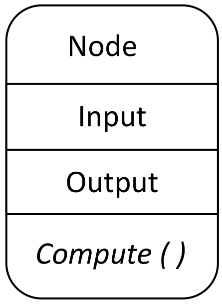


Figure 6: Generic node (David A. D. Gould. 2003). Input and output are two potential attributes of the node. All nodes include a *compute* function. The data is stored in the node's attributes. The role of the compute function is to modify the data.

Attributes in nodes: All the data stored in the node is regarded as attributes. Each attribute has a name and a type. For example, a basic polygon cube node has attributes of all vertices value, normal value, alpha value, and so on. A vertex has at least three attributes named x, y, z of type float. Attribute can allow user to store complex combination of simple data like a parent-child relationship.

***Compute* function:** The compute function is the brain of the node. It takes one or more input attributes then generates a result in an output attribute (David A. D. Gould. 2003). The computing process can be seen as a black box. The programmer can develop any function in it. The user simply needs to know what input and output attributes it has. Figure 7 presents an example of the relationship of DG nodes and DAG nodes.

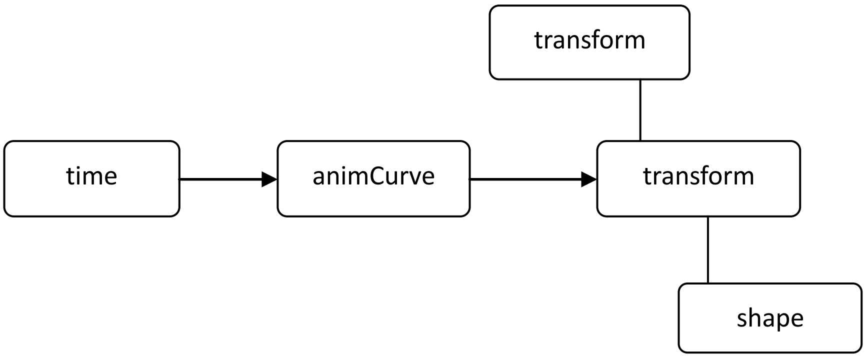


Figure 7: The relationship of time, the animCurve, and transform Nodes consist of Dependency Graph nodes. The relationship of transform and shape consist of Directed acyclic graph nodes.

5 Project Specification

Before moving on to method details, this section specifies the development environment and framework of this system and the data sources of this project.

5.1 Software Environment and Language Requirement

Maya support exports and imports certain data formats, which means the programmers can write custom data exporters and importers; these are known as translators. Since Maya provides access to the entire scene and all its data, this data can be outputted in any form. In order to achieve the function of automatic cloth modeling, some extensions had to be created and then integrated into the Dependency Graph, the heart of Maya. The functionality can be accessed in the same manner as the other standard Maya features can be accessed (David A. D. Gould. 2003).

This system can be developed using either one of Maya's two programming interfaces: MEL or C++. MEL is a Maya embedded language which is designed specifically to work inside Maya. The entire Maya graphical user interface (GUI) is written and controlled using MEL. Because of its simpler structure and syntax, it is easier and more widely accessible than the C++ programming interface. In a word, MEL can be written, debugged and tested entirely within Maya. However, MEL has a drawback: it can run a lot slower than an equivalent C++ program. Generally, MEL provided all the programming functionality necessary for this project. The C++ is typically used for specific functionality that cannot be found in the MEL interface. MEL must be the best choice for this automatic cloth modeling and fitting application (David A. D. Gould. 2003).

5.2 System Framework

Prior to discussing the concrete implementation of methods used in this system, this section will present the framework of the system, which includes data flow and architecture design. Figure 8 demonstrates the development environment and part of the fitting result in the scene.

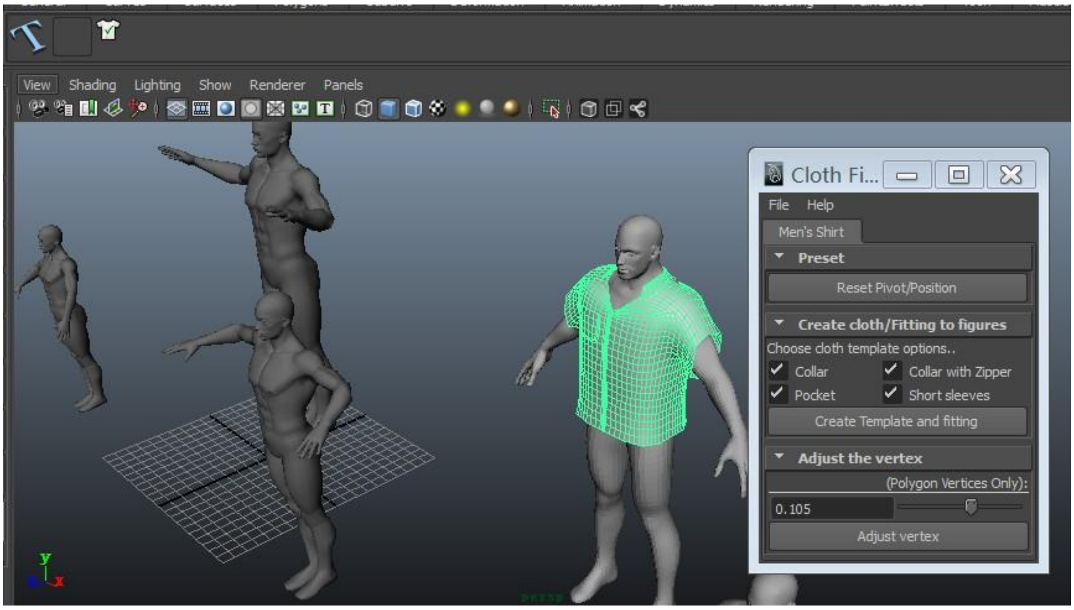


Figure 8: Development environment and user interface of the system

5.3 Operating Flow Chart

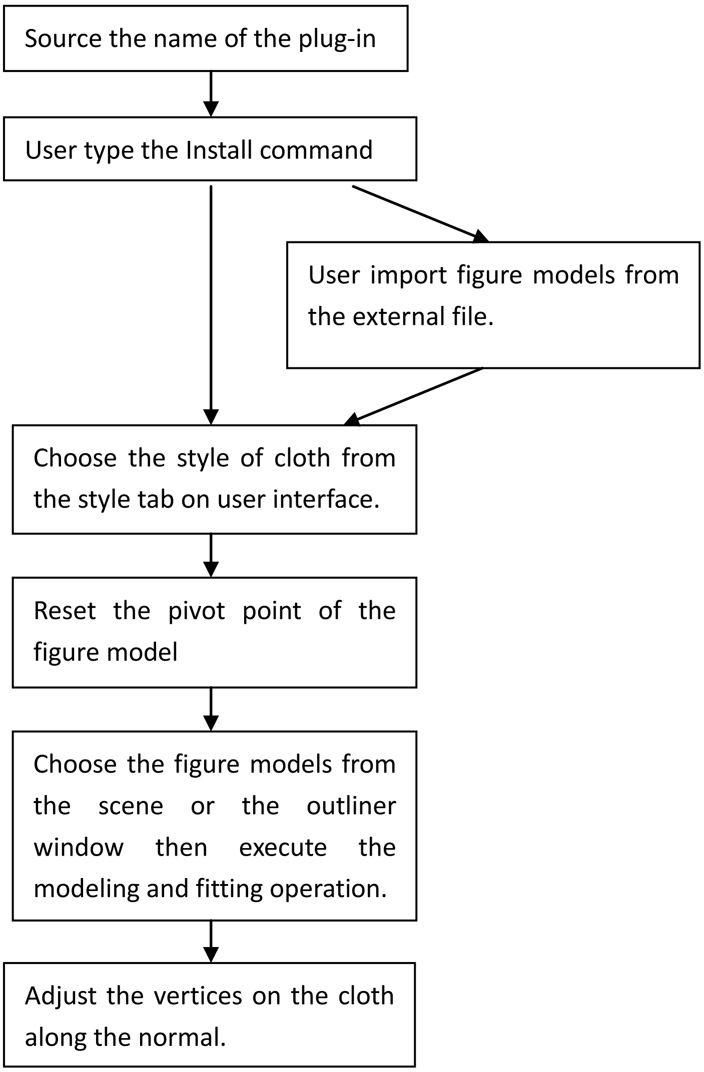


Figure 9: Operating flow chart for plug-in

The key stages of the data flow can be defined as follows:

1) **Type the Install command:** This is one of the standard processes of using the software. Encapsulating software provides an outward interface for users to type in the command, which means that this software can be used in any kind of operating system. Users do not need to worry about the compatibility problem.

2) **Source the name of the plug-in:** The source command causes MEL to compile and execute the contents of a script that is stored in a file. The argument passed to the source command can be a full path or just the name of a script (MEL Command Reference).

3) **Choose the style of cloth from the style tab on the user interface:** Different styles of clothes are included in the modeling and matching information stored in various functions in plug-in. Here, for example, I built the tab of men's shirt with a checkbox associated with collars; programmers can easily access the file and expand any new fashions and add any details that they like.

4) **Reset the pivot point of the figure model:** The pivot should be positioned in the center of the character's feet. The standardization of the pivot's position will not change the original features of figure model, but do make it easy to achieve standardization of the garment modeling and fitting process. It is also a necessary step for the animator to rig the human model.

5) **Choose the figure and execute the modeling and fitting operation:** This is the main part of the system. Figure models can be chosen from the Scene or Outliner window individually or in batches. Users can choose them one-by-one in any order. After clicking the button, clothes are computed and generated one by one on the figure.

6) **Adjust the vertices on the cloth along the normal:** Adjusting the vertices on the cloth along the normal allows the user to satisfy various kinds of human body shapes. This proportional scaling method can be considered as an extension of smooth skinning.

5.4 Architecture Design

Figure 10 features a graph that demonstrates the architecture design of the garment modeling and fitting system.

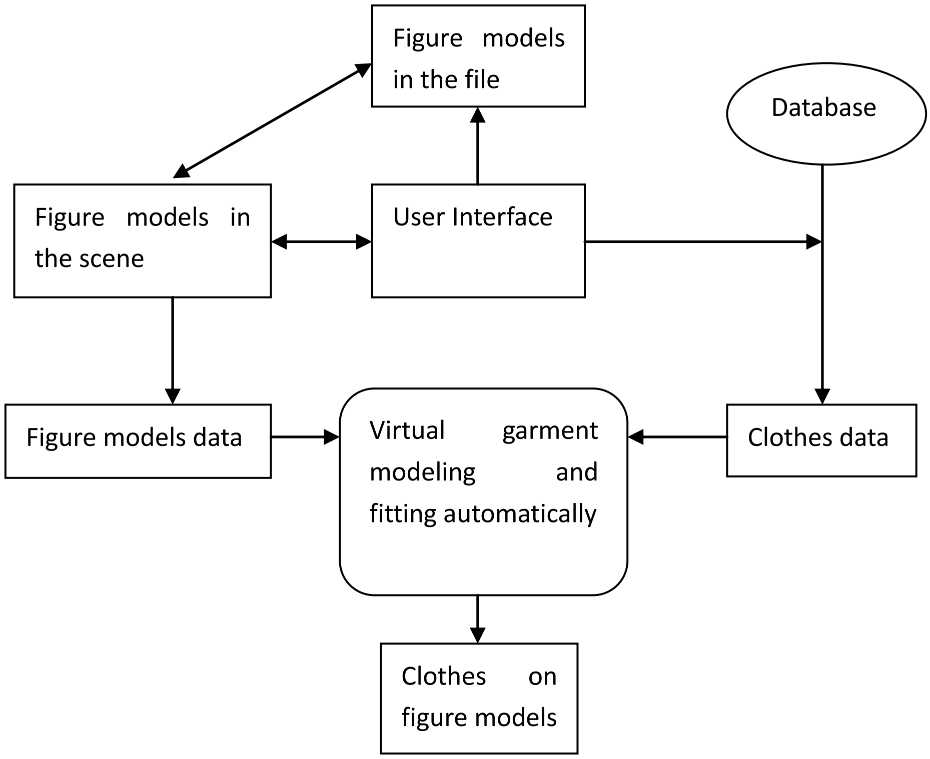


Figure 10: Architecture design diagram

5.5 Measurements of Garment

One necessary step before preparing the sketch for the garment is gathering the size of clothes. The gathered measurements in Figure 11 are the fixed parameters. These parameters can be used as reference points in garment template modeling process (in chapter 6.). Adjusting these parameters proportionally within a reasonable scale will not affect the style of garment; therefore, these parameters can be used as feature points of cloth in garment fitting process (in chapter 6). In other words, the parameters that do not appear in this table can serve as the individually changeable parameters that can be adjusted to fit different body shapes.

The man's shirt is the main object in this research. The role of the shirt is increasingly significant in the modern life of men, especially as it is a kind of undergarment that could be matched with all kinds of outerwear such as sweaters, coats and jackets. The collared shirts have been popular from the 1860. The styles and colors of shirts became varied in that period, especially in Europe and America (Yue L, XiaoGang W, 2009). Twenty shirts parameters have been collected from garment magazines and networks in order to gather statistics about the average figures. These figures are suitable for the standard body size of the male. Figure 11 presents the standard parameters of men's shirts.

Height of figure (cm)	Size (cm)	Waistline (cm)	Chest (cm)	Length of cloth (cm)	Length of sleeves (cm)	Shoulder width(cm)
165	38	98	108	78	59.5	45
170	39	102	112	79	59.5	39
175	40	106	115	79	60.5	40
180	42	113	121	81	61.5	49
185	44	119	126	82	62.5	51
185	45	122	128	82	62.5	51

Figure 11: Standard parameters of men's shirts

5.6 Mannequin

For better experiment results, three different body sizes of manikins of men were chosen from the website: http://www.maya09.cn/forum-51-1.html. In order to make comparisons regarding fitting effect, the shape of one of the manikins was proportional changed. As the method of garment fitting requires the figure body to be rigged, the upper bodies of all figure models were rigged in Maya.

5.7 Input Requirement

1) It was assumed that the figure model posed a similar posture to any one of the standard body standing postures, since it is also the standard posture for manikins need to be rigged. Standard postures do not include bowing, a handstand, lying-down, and so on.

2) The figure models are assumed have been rigged.

6 Method

Figure 12 demonstrates the connection of methods involved in this system.

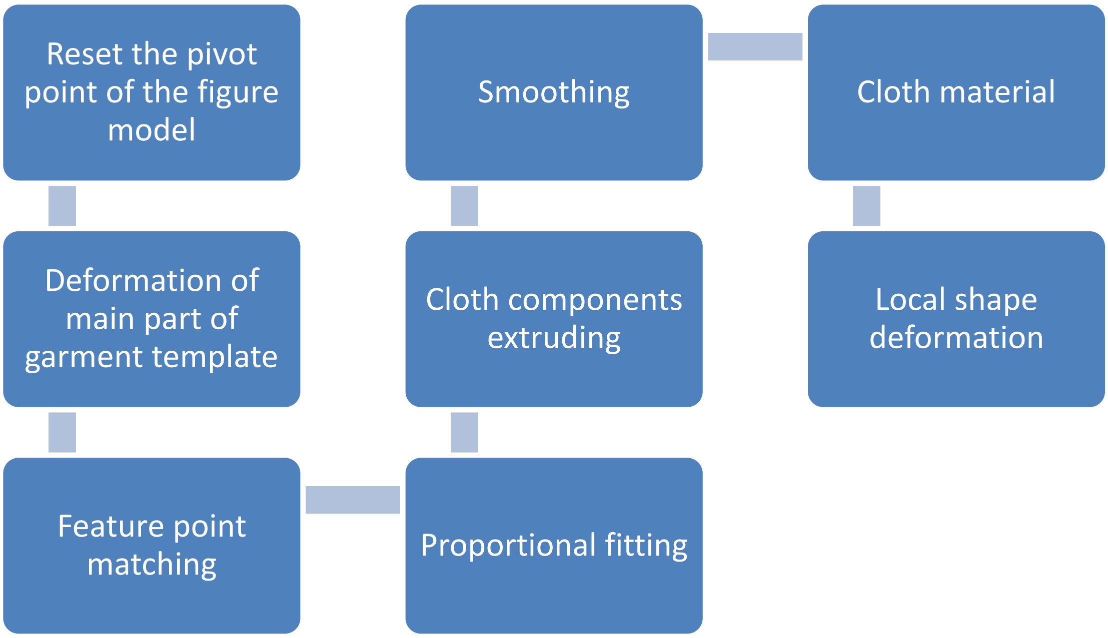


Figure 12: connection of methods in the system

6.1 Reset the pivot point of the figure model

The function of pivot points is to control how objects rotate and scale, and to represent the exact locations of objects in space (Maya user's guide). In other words, they control all transformations related to an object. Pivot points can be changed by using pivot point manipulator manually. The pivot point of a figure model marks the transformation location, which is one of the feature points in my project. For most modelers and animators, the pivot point of figures should be set in the center of the figure's feet, since it is a standard initialization position for them to rig the body and to set key frame for character in animation. In this project, in order to achieve standardization of garment modeling and fitting process, all pivot points of figures are made to move to the feet center automatically. The basic theory and method of pivot point reset can be summarized as follows:

The default position of the feet center of the figure model in object space is (0, 0, 0). The local space is the one in which the point first exists and in which it hasn't undergone any transformations. World space is the final space in which the shapes are drawn. The basic steps of resetting the pivot point are:
1} Obtain the original position of the figure's pivot point in world space.
2) Determine the position of the feet center in object space is (0, 0, 0), calculating its

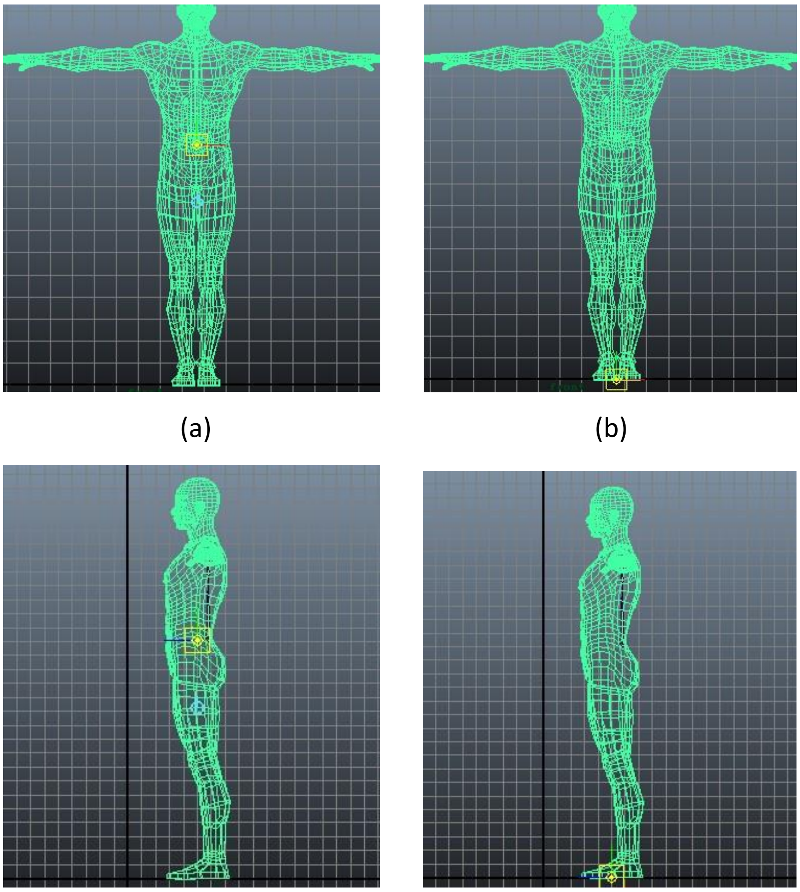
corresponding position in world space.

3) Transform the pivot point from original position to the new position in world space (Transform matrix will be introduced in Chapter 6.2)..

The **algorithm** can be summarized as algorithm 1 shows below.

```
get the selection objects and put value into $objs
 for all $obj in $objs do
    set the pivot position value of the $obj to {0, 0, 0} in object-space
    get the transform matrix to $mtx[16]:
    if ( the size of $pt == 3 && the size of $mtx == 16)
    { $mtx[16] = `xform –query –worldSpace –matrix $transformNode;
    get the position value of the $obj in world space: $pivot multiply $mtx:.
    $res[0] = $pt[0] * $mtx[0] + $pt[1] * $mtx[4] + $pt[2] * $mtx[8] + $mtx[12];
    $res[1] = $pt[0] * $mtx[1] + $pt[1] * $mtx[5] + $pt[2] * $mtx[9] + $mtx[13];
    $res[2] = $pt[0] * $mtx[2] + $pt[1] * $mtx[6] + $pt[2] * $mtx[10] + $mtx[14];
     The result were put into an array $res }
    set the position y value of the $obj to 0
 end for
```

Algorithm 1: The main steps of my algorithm in resetPivot( ).



(a) (b) (c) (d)

Figure 13: Pivot point adjusting automatically. The left two graphs (a), (c) show the original pivot point's position of the figure model in front view and side view; the right ones (b), (d) show the position of pivot point transform to the feet center automatically in front view and side view.

6.2 Geometric Cloth Template Modeling

After resetting the pivot point, the next step is cloth template modeling, which means many different kinds of garment can be built according to one cloth template. In this project, the shirt will be used an example to explain the method. Firstly, the mesh surface for the upper body was set up to be automatic, and was used as the cloth template for the shirt. Based on this cloth template, many different kinds of shirts can be built easily by adding the garment components like collar, pockets, zipper, and sleeves and so on. One main body surface consists of various kinds of cloth components in any combination. Although Arnulph Fuhrmann and Clemens Groß (2003) proposed the idea of bounding surface, there is no need to separate the bounding surfaces to this degree. Compared with the other garment modeling method, this is the first one to propose the idea of a cloth template. It meets customers' urgent needs for the same styles of different clothing to a large extent. In this section, a method was proposed for achieving automatic modeling of bounding surface for upper body. The method of garment components modeling will be described in Chapter 6.5.

The bounding surface for the upper body is based on the automatic deformation of cube mesh. Specifically, most deformations focus on the transformation of feature points p_g. The feature points of the upper body surface are determined by the standard parameters of the shirt. For example, the vertex on the shoulder associates with the shoulder length in the parameter of shirt. A transformation is simply a mapping of a point from one place to another: $p' = p$ * transformMatrix. A single transformation matrix can store more than one transformation:

transformMatrix = translateMatrix * scaleMatrix * rotateMatrix.

In Maya, the xform command is the principal means of querying the transform matrix. The translation is phrased as follows:

Xform –query –worldSpace –translation;

The vertices on the different parts of cube mesh achieve translation on the basis of the corresponding parts on garment parameters table in Chapter 5.5. These include shoulder, waist, breast, back, and neckline. Shoulder was chosen as the starting point in this modeling process due to its location on skeleton. However, just transforming the feature points onto the cube mesh is insufficient to form the main part of cloth template, since the shape around the feature points should form a slope when translating the feature points. The selection function in Mel library can be adopted to construct the shape around the feature points with smooth transitions. I did not use this function since the main part of the garment template of the shirt had no additional vertices except for feature points. That said though, if the modelers intended to expand more complex clothing styles in this system, Soft Selection may be a better choice for reducing the number of feature points.

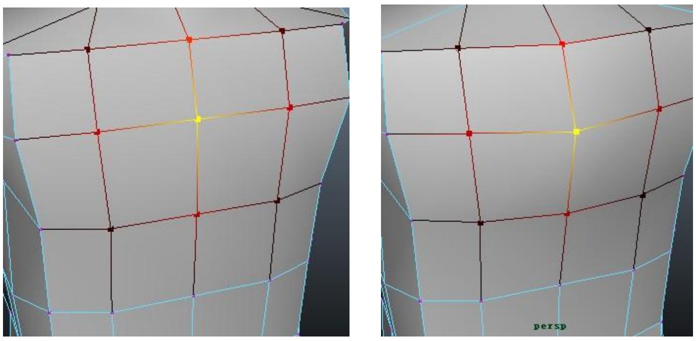


Figure 14: Soft selection. The vertices around the adjusted vertex generate smooth transitions in translation.

The basic steps of automatically building the main surface of a garment template are as follows:

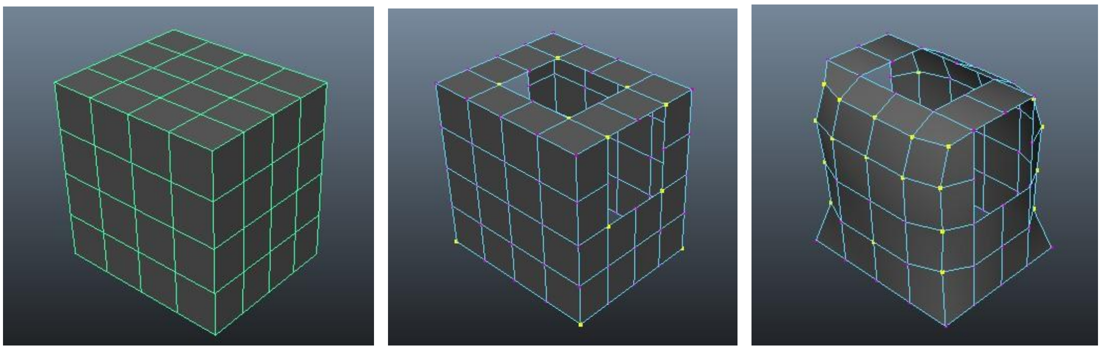
1) Classify the vertices on the mesh cube into various parts according to the parameters of the selected style of cloth (shirt).
2) On the basis of the shoulder parameters, calculate the relative position of corresponding vertices on the mesh cube.
3) Move the vertices to the calculated positions by using the transformation matrix.
4) Based on the other parameters, repeat the calculation process mentioned above.

The **algorithm** can be summarized as algorithm 2 shows below.

---

get the selected objects and put value into $objs
for all $obj in $objs **do**
get the standard parameters of men's shirt
calculating the mean value of width, length, height: $clothWidth $clothHeight $clothDepth
setToolTo CreatePolyCubeCtx
based on $clothWidth $clothHeight $clothDepth, create a polygon cube
set 4 subdivision on x, y, z axis of the cube
delete the surfaces belongs to head, arms and lower body (figure (b))
according to the standard parameters of men's shirt, separate the vertices on meth cube to different part.
Based on the parameters, adjusting the vertices by transformation matrix on shoulder, breast, waistline, back and neckline (figure (c)) .
End for

Algorithm 2: The main steps of my algorithm in createTemplate( )



(a) (b) (c)

Figure 15: Automatic modeling of bounding surface for upper body. The yellow points indicate the adjusted position of vertices after reset the transformation matrix. Finally, the main part of garment template is complete without smoothing.

6.3 Template Fitting Method

Before describing the template fitting method, it is essential to study and analyze the users' needs in garment fitting. The following sections therefore list the basic garment fitting criteria based on a study of users' needs.

6.3.1 Garment Fitting Criteria

1) Proportionality requires the resized cloth to preserve the cloth's features relative position in a way that corresponds to the figure's limbs and body. For example, the waistline of trousers needs to fix on the waist, and the fixed points of windbreaker need to remain on the shoulders. Different styles of cloth own the various relative positions of fixed points. It is also the basic theory of preserving the style of cloth. Therefore, preserving the style of cloth in the proportionality stage means associating a relative location of the fixed point on the body with the points on the cloth and preserving the direction from one to the other during fitting.

2) Fitting requires the resized cloth to keep a proportional distance along the surface normal of the figure model by default. In other words, the resized garment and the surface of the figure model should not intersect apparently in the first step garment fitting.

6.3.2 Feature Points Matching

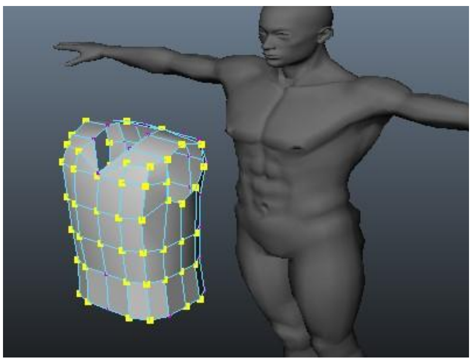
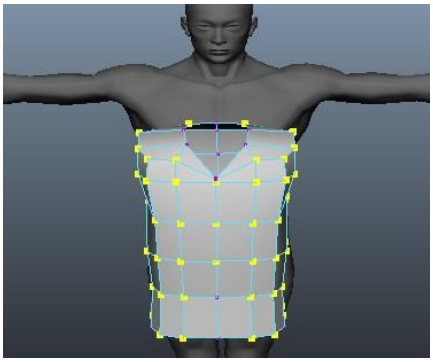
In this project, a garment template fitting involves matching feature points and proportional fitting. The existing feature points extraction methods have been analyzed and discussed in Chapter 3.1. Based on the study of Remi Brouet and Alla Sheffer (2012), I propose an automated feature point matching method. There are three basic steps to this feature matching method:

1) The first step is extracting the feature points of the virtual garment and computing the offset of these feature points. On the basis of the rules defined in Chapter 3.1, the feature points of virtual cloth p_g have been specified in the garment template modeling process; these are almost the same as the standard parameters of garment. Each feature point p_g is associated with offset *o. o* is a preset default value and will be adjusted in next section.
2) The second step is to compute, according to each feature point on the garment, the corresponding points on the target figure by cross-parameterization.
3) The third step is fitting the cloth to the target figure. For each of the feature points on the garment p^c_g, $p^c_g = p^c_t + o * v_{tg}$. Where p^c_t is feature points on the target figure, the v_{tg} is a unit vector in the direction of (p^c_t, p^c_g).

The construction of the transformation matrix is based on the method of Zhong Li and Xianggang Jin (2009). Make F_i be the coordinate vector of the feature points on the figure model and C_i be the corresponding feature points on the garment template model. After obtaining the match relations of key feature points between two models, all the coordinate vector from both models should satisfy the coordinate transformation equation below.

$$\begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix} = \mathrm{R}(\theta_x, \theta_y, \theta_z) \times \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{bmatrix} + T$$

Where n is the number of key feature points, $\mathrm{R}(\theta_x, \theta_y, \theta_z)$ is the rotation matrix, T is the translation vector.

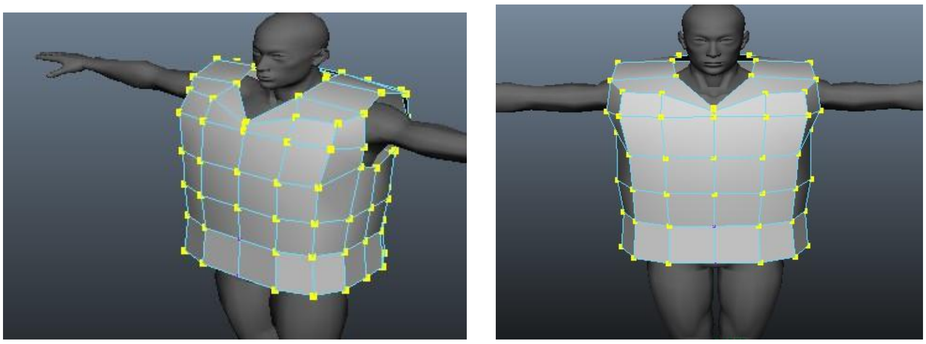


Figure 16: Feature points matching. The top two images present the feature point extracting from garment template in side view and front view. The bottom two images demonstrate the feature points matching effect in side view and front view.

6.3.3 Proportional Fitting

During the feature point matching process, each feature point p^c_g on the target garment keeps a certain distance, the offset *o,* from the target manikin. In order to make the garment proper fit the manikin automatically, I propose the proportional fitting method. The basic steps of proportional fitting are as follows:

1) Assign each feature point p^c_g orthogonally to its nearest bone on p^c_b.
2) Calculate the intersection point p^c_i of the vector (p^c_g - p^c_b) intersect with its nearest skin surface. Then compute the distance *d* between p^c_g and p^c_i.
3) Adjust the offset *o* of each feature point p^c_g on the basis of proportion of $d/|| p^c_g- p^c_b||$

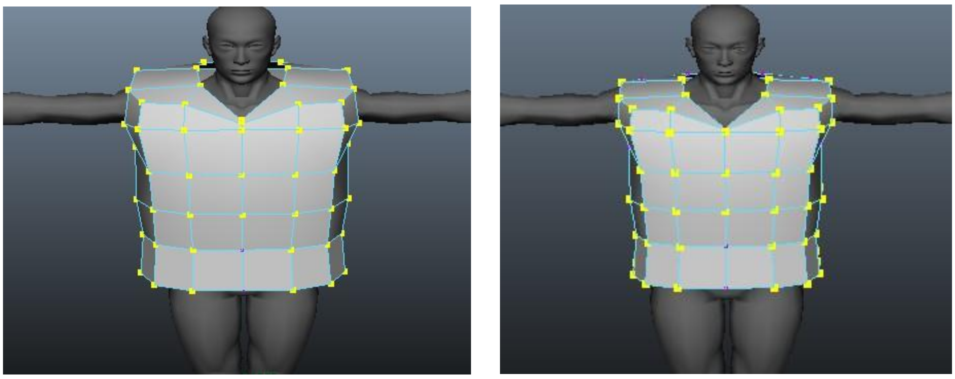


Figure 17: Proportional fitting effect. The left image presents the effect before proportional fitting. The right one presents the proportional fitting effect on target figure.

6.4 Smoothing

There is a built-in function-*polySmooth* in Maya for smoothing the polygon mesh. It adds division to the polygon on mesh. This function is adopted after the garment fitting process.

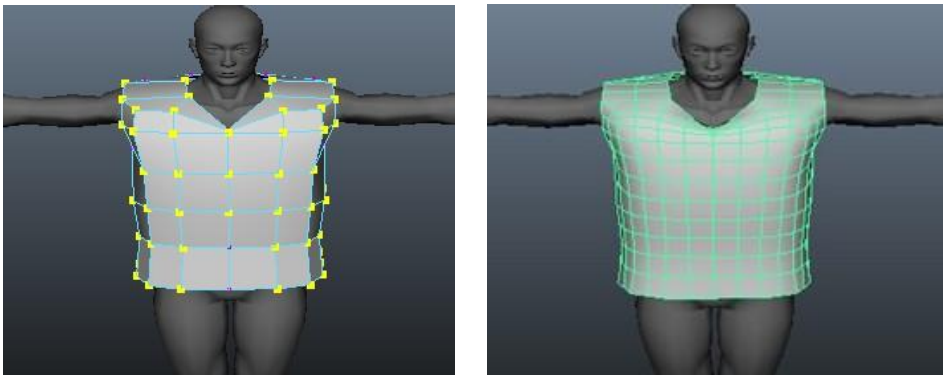


Figure 18: Mesh smooth

6.5 Cloth Components Extruding Method

The aim of this section is to demonstrate the modeling of cloth components automatically according to the user's selection. The categories for cloth components include sleeve, pocket, zipper, button, collar, and belt. All these components are formed by pulling out new polygons from existing vertices, edges or faces on main garment template surfaces. Specifically, each of these components is associated with a set of extruding information on the main mesh surface. The extruding information includes three parts: the name of the extruded vertices, edges or faces, the extruding direction, and the extruding distance. The following sub-sections will outline the main components that are necessary elements to the style of the shirt.

6.5.1 Sleeves Extruding

The extruding of sleeves is separated into two parts: extruding direction and extruding distance.

Extruding direction

The direction of the sleeve from shoulder part to elbow part is the first extruding direction of the sleeve. The second extruding direction of the sleeve is from elbow part to twist part. In order to make the garment fit as many standing postures as possible, the direction of the sleeve cannot be designed according to the fixed one. Accurate detection of the direction of the arm surface of the figure model relies on a series of complex computations that can not be practically explored in this project. Generally, rigging is a necessary step in the process of character production. It is still the essential step before preparing the character for animation. Therefore, the direction of the sleeve can be obtained by calculating the unit vector of bone of arm.

The calculation of the unit vector of the bone of an arm can be separated into two steps. The first step is detecting the bone. It is not as easy for the computer to know which is the bone of the upper arm as it is for us to see and recognize it. However, the positions of each vertex on sewing circle line can be obtained easily, and then the position of scale pivot point can be worked out by calculating the mean value of these vertices. The position of each joint point must then be compared to the scale pivot point so the closest joint point can be recognized. This joint must be the part of bone of upper arm.

The second step is computing the unit vector. The structure of the skeleton must be the top-down hierarchies, so the bone of lower arm is the next of the bone of upper arm. The unit vector can be attained by subtracting the 3D coordinate value of two joints and then dividing by the distance of two joints.

The **algorithm** can be summarized as algorithm 3 shows below

---

Input: $SFlag, $obj
  If the flag of sleeves $SFlag == 1
    Get the positions value of vertices on the circle line of left shoulder
    Calculate the center position value $shoulderC of these vertices
    For all $joint **do**
      Compare $shoulderC to $joint, $shoulderC- $joint
      Return the $joint related to minimum value of ($shoulderC- $joint)
    End for
    Get the joint $leftelbow next to the $leftshoulder
    Calculate the value $leftshoulderOrent of ($leftelbow-$leftshoulder)
    Calculate the unit vector of $leftshoulderOrent
  End if

Algorithm 3: The main steps of my algorithm in sleeves extruding.

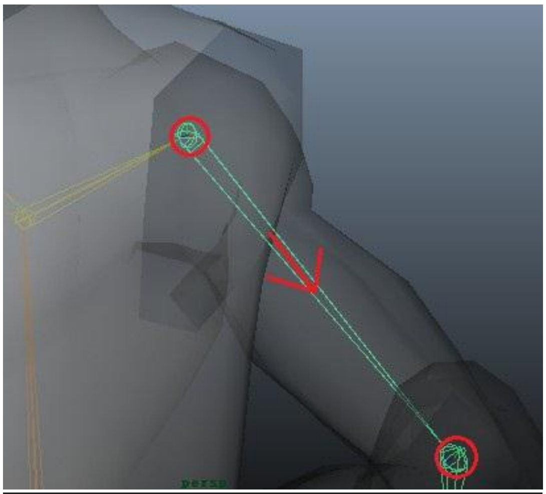
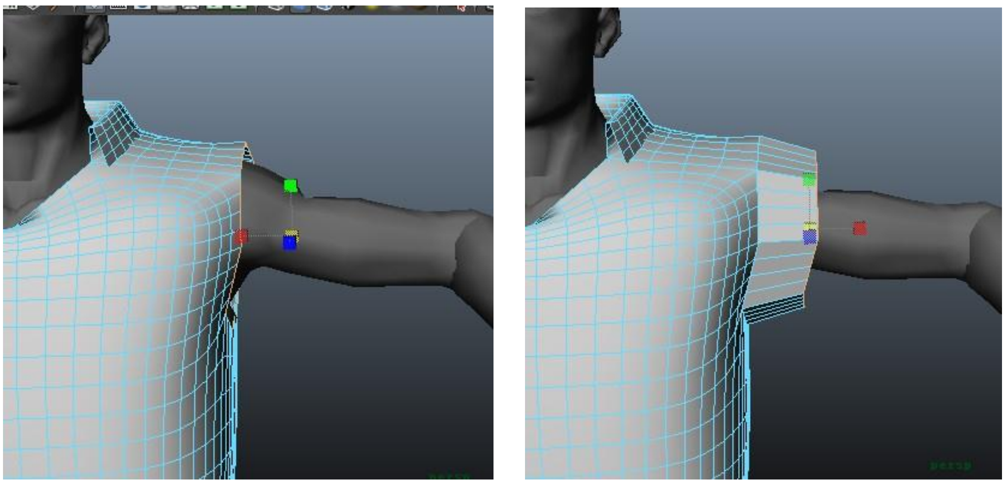


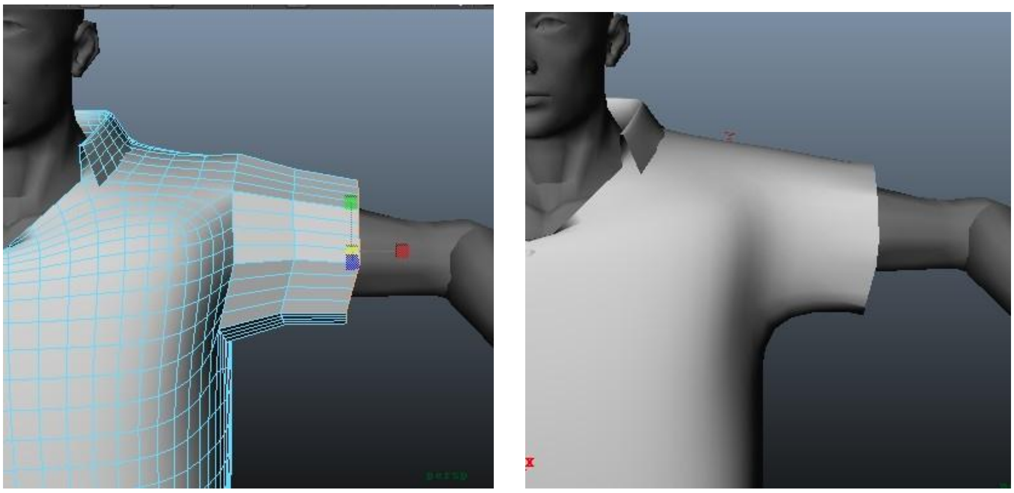
Figure 19: Extruding direction of sleeves. The top left red circle indicates the left shoulder joint. The bottom right red circle indicates the left elbow joint. After obtaining the positions of these joints, the extruding direction (indicated with red arrow) can be obtained.

Extruding Distance

The extruding distance depends on the length of sleeves. Generally, it can be divided into two types: short sleeve and long sleeve. The distance of short sleeves is based on the length of figure's upper arm. The length of the upper arm is equal to the distance of the joint of the shoulder to the joint of the elbow. Since the feature points on sleeves line have been adjusted to fit the arm in the proportional fitting step, the extruded sleeves can fit the arm on the basis of extruding direction and extruding distance.



(a) **(b)**



(c) (d)

Figure 20: Sleeves extruding. The sleeves extruding process can be separated into two stages. The first stage can be seen in images (a) and (b), which extrude a part of distance of sleeves' length and then scale the cuff based on the proportional fitting method. The second stage can be seen in images (c) and (d), which extrude the final distance of sleeves' length.

6.5.2 Pockets

Generally, pocket texture is always been used in cloth modeling of game production, which decreases the polygon number along with the rendering time. A more complex method is to stitch the pocket geometry on the garment. This requires modeling the geometry first then matching the sewing information and stitching the pockets on the cloth.

I propose a simple method to achieve automatic pocket modeling. The automatic process can be separated into the following steps: First of all, on the basis of the number of the pockets, calculate the position of pockets on the cloth. Secondly, according to the size of the cloth template, calculate the proportional size of pocket. Thirdly, extrude the surface in the corresponding place. Next, delete the seal surface on the top of pockets. Finally, smooth the edge of pockets.

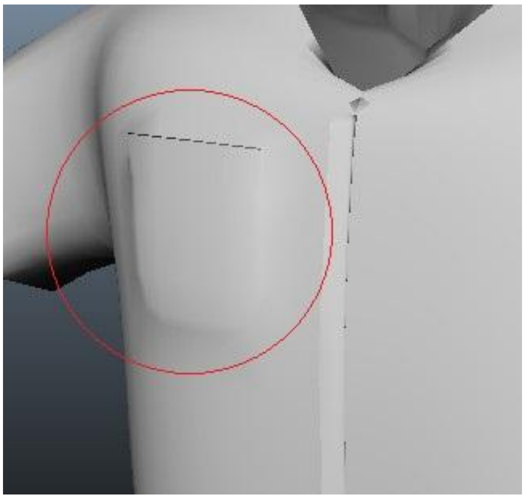
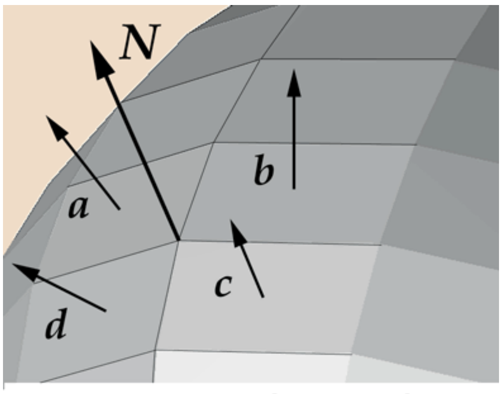


Figure 21: Extruded pocket (pocket was indicated in red circle).

6.6 Local Shape Deformation

By adjusting the local shape of the cloth, the needs of various kinds of non-standard human body shapes can be satisfied. This local shape deformation method can be considered as an extension of smooth skinning, which preserves the pattern design during deformation process. Specifically, in order to preserve the pattern design, vertices to be adjusted should be move along the normal of clothes. Firstly, choose the proper vertices that need to be adjusted. Secondly, determine the three direction (x, y, z) normal values of the selected vertices on the neighbor garment surface. Then average the three vertices normal into one direction. Finally, unitize the normal to one before using the multiplier. The example of averaging four direction surfaces normal around a vertex can be seen on the figure below.



$$N = \frac{a + b + c + d}{4}$$

Figure 22: Vertex normal

The **algorithm** can be summarized as algorithm 4 shows below

```
Input: $amountToMoveFromSlider
   Get the selected vertices and put them into $sel[]
     For all $sel[] do
       If ( judge whether the selected objects are mesh)
          Print a piece of warning "Works only on polygon vertices"
       End if
      Get the normal(x, y, z) value $arrayOfVectors[9] of the selected vertices on
      the surface. And then average the normal:
       Average the value of $arrayOfVectors [0] [3] [6] to the $vN[0];
       Average the value of $arrayOfVectors [1] [4] [7] to the $vN[1];
       Average the value of $arrayOfVectors [2] [5] [8] to the $vN[2];
       Unit the value of <<$vN[0], $vN[1], $vN[2]>> to vector $unitizedNormal
      move -r ($amountToMoveFromSlider * $unitizedNormal.x)
           ($amountToMoveFromSlider * $unitizedNormal.y)
            ($amountToMoveFromSlider * $unitizedNormal.z) $vert;
     end for
        restore original vert selection: select $sel
```

Algorithm 4: The main steps of my algorithm in moveVertAlongNormal( )

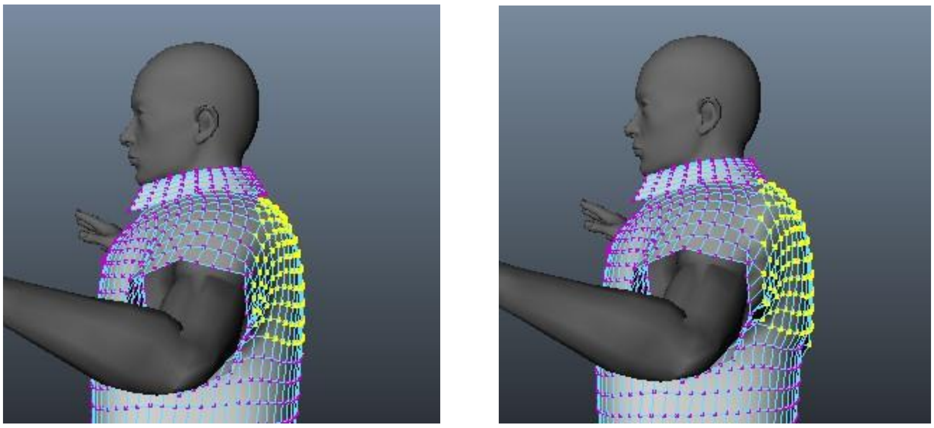


Figure 23: Local shape deformation. The left one shows the shape of cloth before adjusting. The right one presents the result by adjusting the vertices along the normal in 0.21cm

6.7 Garment Material

In my project, the material of the man's shirt is set to Lambert automatically. Lambert is one of the most basic object materials in Maya; it possesses matte surfaces and has no spectacular highlights. The default color of the shirt is set to white. The specific steps of automatically setting the garment material are shown below:

1) Judge whether the garment has material or not. If it doesn't have a material, jump to Step 2.
2) Using the *shadingNode* command, classify the garment *node* as a Lambert *shader* named clothMaterial.
3) Define an association of material properties to the garment as clothSG.
4) Connect the clothMaterial's attribute *outColor* to clothSG's attribute *surfaceShader.*
5) Set the clothMaterial's attribute color to 1 1 1.

This is the fundamental process of automatically setting the material of a garment that does not require the 3D texture mapping. Lambert is used as the *shader* of the cloth; though it cannot reflect all the characteristics of the material of the clothes, it can be replaced by other more suitable cloth *shader* in the future.

6.8 Install Plug-in

The aim of this project is not only propose an automatic garment modeling and fitting method but also to apply this method to the development of practical production. Due to the variety of computers and operating systems, installation is a necessary part in the development of any software or plug-in, as this ensures the software is ready for execution. The process of the installation is as follows:

1) Determine whether the tab Layout shelf exists.
2) If it exists, jump to step 3. If not, print out an error message.
3) Set the parent layout of this control.
4) Set the icon by the image in the icons folder.
5) Connect the response function of the icon to the entry function of the plug-in.

The install instructions are shown below:

1) Copy the mel script (clothfit.mel) to your local user/scripts folder
2) Copy the icon file (fitIcon.bmp) to your user/prefs/icons folder
3) Type: source clothfit.mel; into the Maya command line.
4) Make your shelf is visible and type: installclothFit; into the Maya command line.

After following the above instructions, a new shelf button will appear on the shelf. Click the button and a user interface will appear on the scene.

6.9 User Interface

The structure of the system is clearly presented on the user interface. There are three basic operation steps that can be seen in Figure 24: reset pivot/position, create template and fitting, and adjust vertex. For this project, a menu bar including an *import* function that can read data from .mb or .ma file was created as well. The layout of the user interface is designed to be similar to the layout of the window in Maya. Some exception handlers were added for error operation. The detail of the user interface design is shown below.

```
showMyWindow( )
          if the window of application exist
                delete the window
       else
                create a new one
          set the name and attribute of window myWindow
          set menuBarLayout
                    set the menu "file" and "help"
                          set the menuitems
            setParent ..
          set tabLayout for different styles of clothes
            set columnLayout and its attributes
                    add the button "reset pivot"
                setParent ..
            set columnLayout and its attributes
                print text " Choose cloth template options.."
                add separator line
                add checkbox for the details of clothes
                add button "create template and fitting"
                setParent ..
            set frameLayout and its attributes
            set rowColumnLayout and its attributes
                add separator line
                add floatSliderGrp and its attributes
                add button "Adjust vertex"
            setParent ..
showWindow myWindow
```

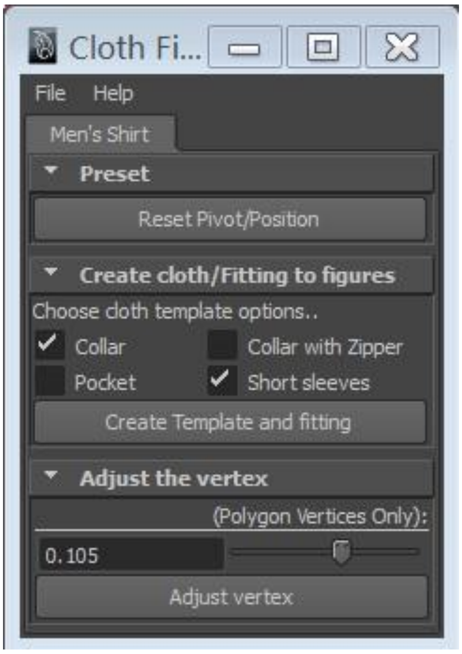


Figure 24: User Interface of the plug-in

6.10 Design of Automation Algorithm

The specific methods involved in this system have been explained separately in the chapter above. The aim of this section is to make the data flow from output of one method to input of another method automatically. The whole process is outlined in Figure 12.

The **algorithm** can be summarized as algorithm 5 shows below

---

createTemplate (int $CFlag, int $ZFlag, int $PFlag, int $SFlag)
1. Insert the selected objects (figure models) into **objs**
2. for ($**obj** in $**objs**)
3. for ($i=0; $i< (size of vertices of **obj**) ; $i++)
4. Acquire the translation attribute of each vertex
5 sort the x, y, z value of these vertices
6. Max value – min value: calculate the preliminary height, width and thickness of figure model
7. Remove disturbances of head, limbs from the figure, calculate the height, width and thickness of upper body.
8. Scale the basic cube mesh according to the size of the upper body
9. According to the value of **Cstyle,** choose the related function, for example, jump into shirt ( )
10. Acquire the garment details options in shirt tab on user interface, which

include collars, pockets, and sleeves and so on.

11. according to the garment details option, set up the corresponding parts of cloth.
12. acquire the central pivot point of upper body then scale and translate the cloth template to it
13. set the garment material: lambert material.
14. smooth the cloth mesh: polySmooth command.

Algorithm 5: The main steps of my algorithm in createTemplate ( )

Figure 25 further presents the flow chart for the automation algorithm.

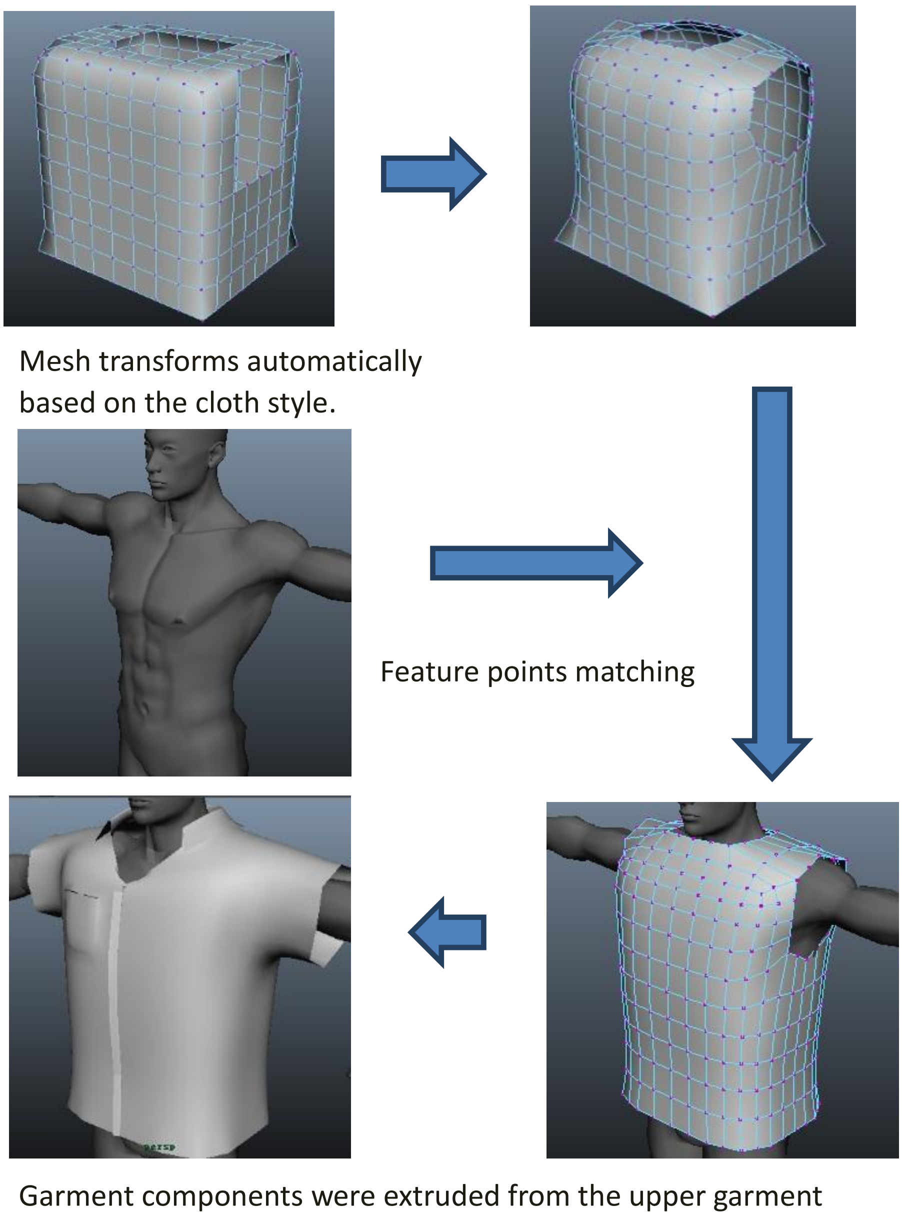


Figure 25: Flow chart for automation algorithm

7 Results and Analysis

In addition to the results of my automatic virtual garment modeling and fitting demonstrated in Chapter 6, I have also performed experiments in testing and comparing result of my method in the following areas: modeling of any combination of garment components, fitting different arms posture, fitting to different shape of bodies, efficiency between different methods, and effect of mass fitting.

7.1 Modeling of Combination of Garment Components

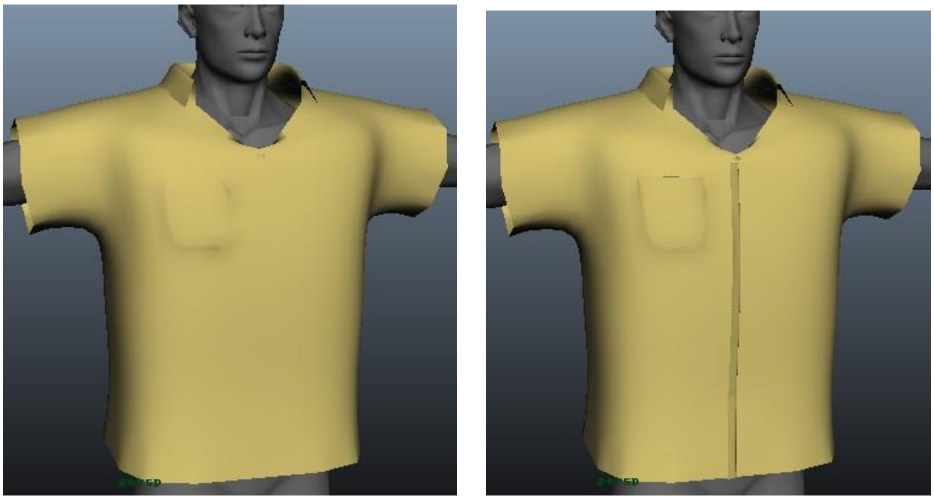
The result of different combinations of cloth components is shown below:

(a)

(b)

(c)

(d)



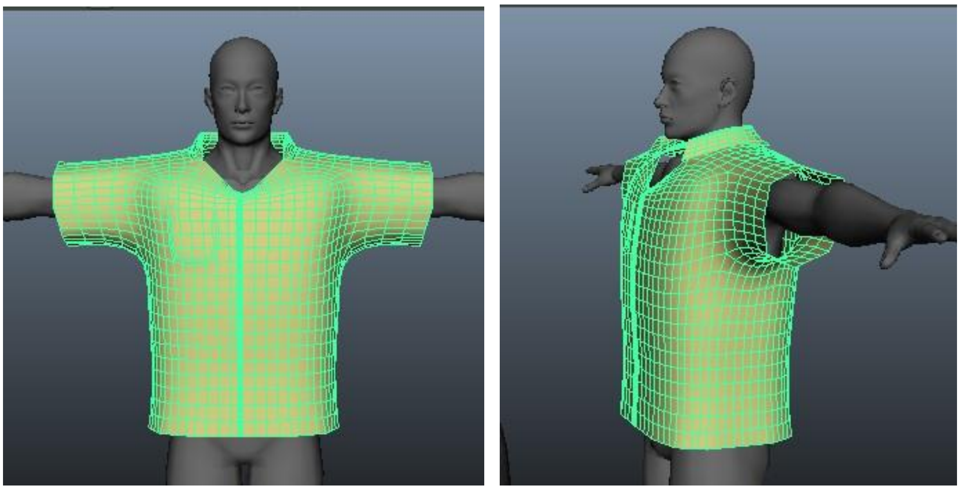
(e) (f)

Figure 26: Modeling of any combination of garment components

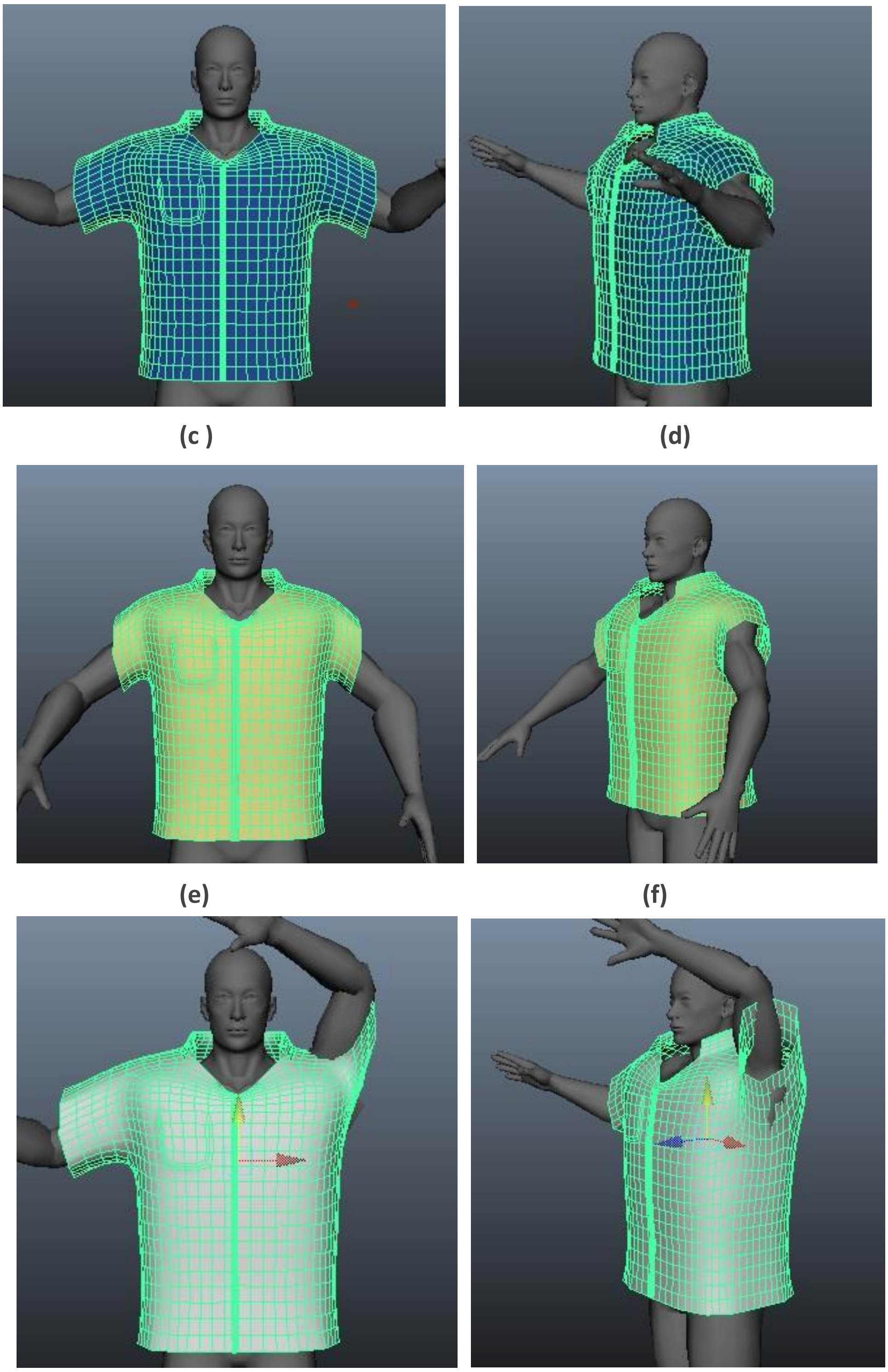
Focus on the modeling effect of the shirt. In Figure 26, images (a) and (b) show the main part of the shirt template, and the garment template with collar respectively, while images (c) and (d) present a cloth template with short sleeves, and a cloth template with collars and sleeves respectively. Image (e) demonstrates a garment template with collars, sleeves and pocket, and image (f) demonstrates a garment template with collars, sleeves, pocket and zipper. All of these shirts are successfully modeled.

7.2 Fitting Different Arms Postures

The results of garment fitting for different arm postures of a figure model are shown in Figure 27.



(a) **(b)**



(c) (d)

(e) (f)

(g) (h)

Figure 27: Garment fitting posture. It can be seen that the figure poses various arm postures in the front view (a) (c) (e) and correspondingly in (b) (d) (f). Garments fit the figure well in these images. However, in images (g) and (h), the garment fails in fitting the figure's arm. The results of this experiment point out that if the arms were raised too high or too low, sleeves will intersect with the arm. This is due to the feature points on the original cuff being preset on the main surface of the garment template, which cannot be affected by the direction and position of arm. The

positions of the feature points on the original cuff determined the range of extruding direction. Therefore, if the arm exceeds the range of the extruding direction of the sleeves, the extruding direction cannot be calculated.

7.3 Fitting to Various Body Shapes

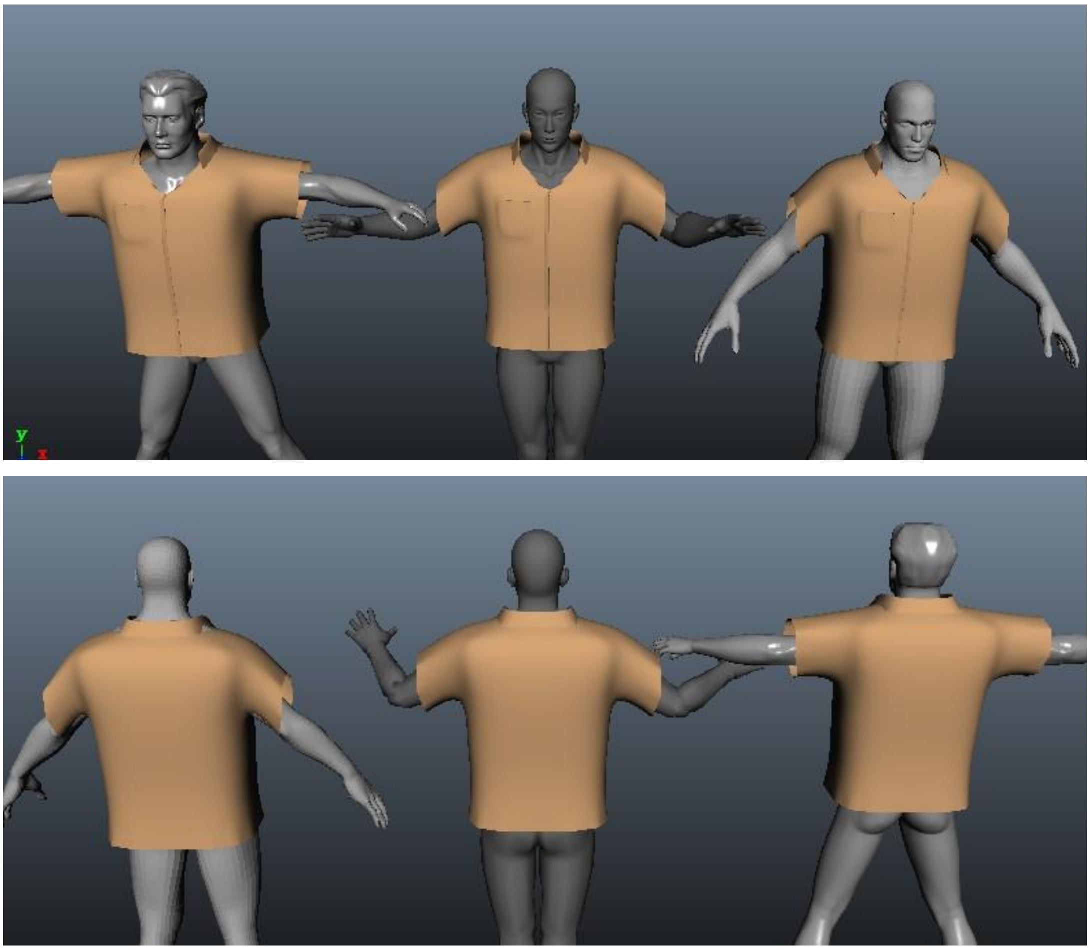


Figure 28: Fitting effect of shirt on various manikins in front view and back view. It can be seen in these two images that feature points matching and proportional fitting method make the shirt fit three different manikins successfully, although there is a small degree of surface intersection on the shoulder of the left manikin in the back view. This is due to the feature points not being sufficient enough to cover all area of vertices on the manikin. This means that there are no feature points in the area of intersection.

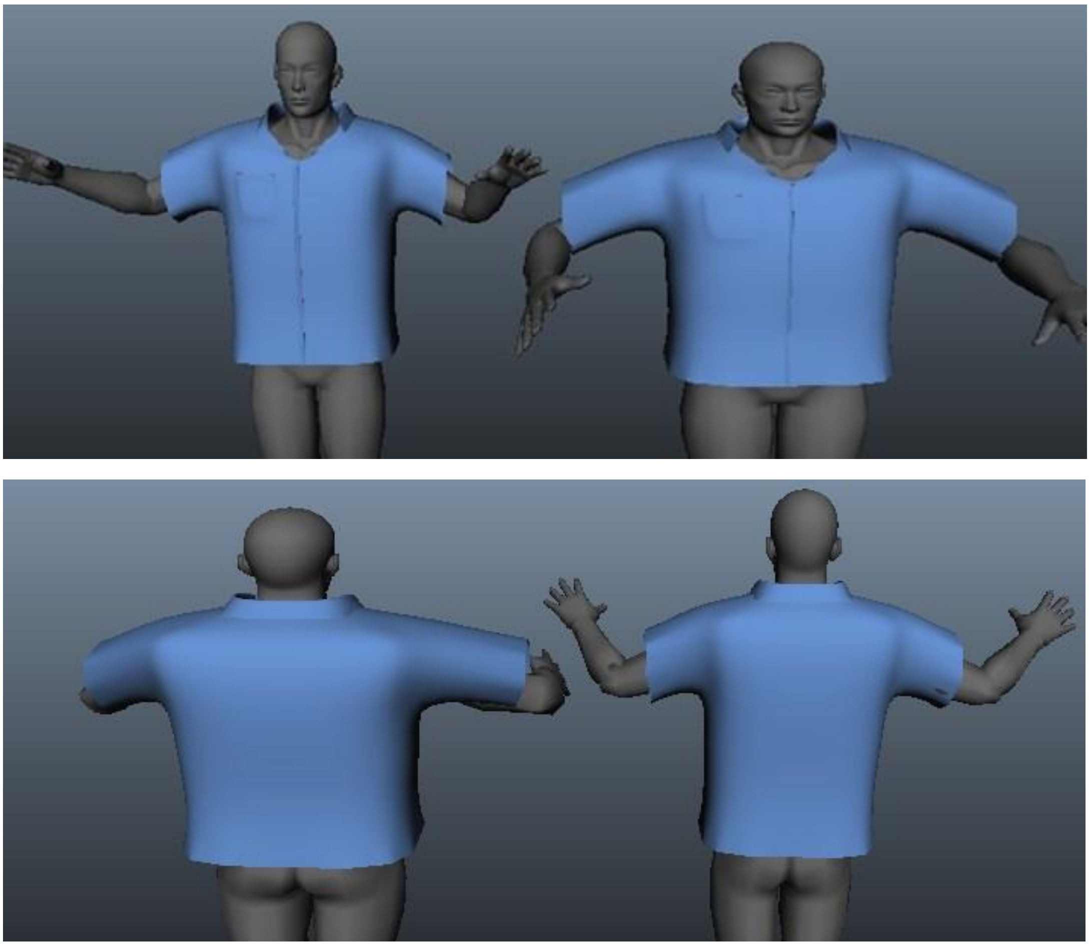
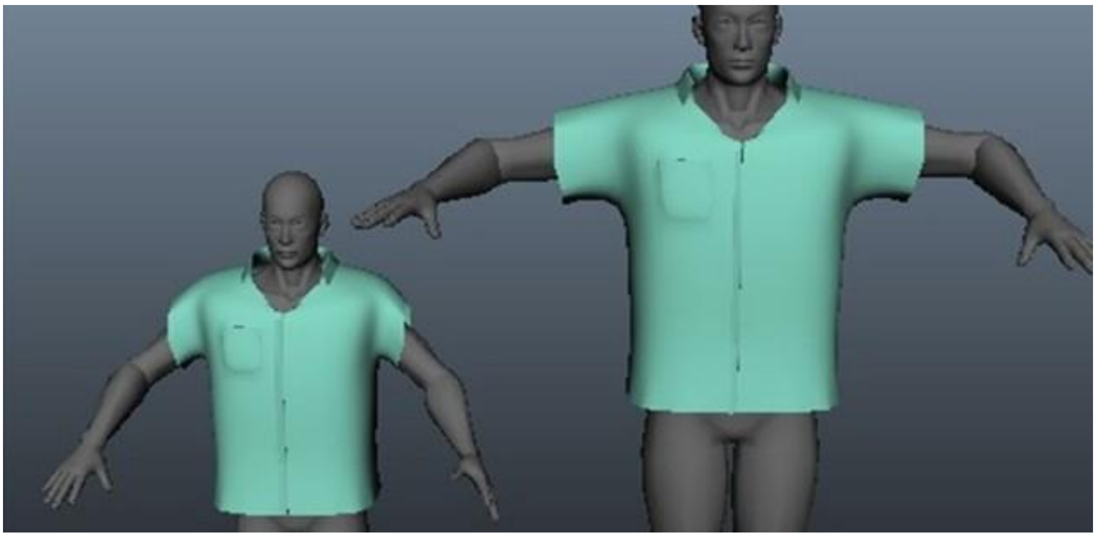


Figure 29: Garment match more different manikins (front and back views): a thin person and a fat person. Garment successfully match these two models



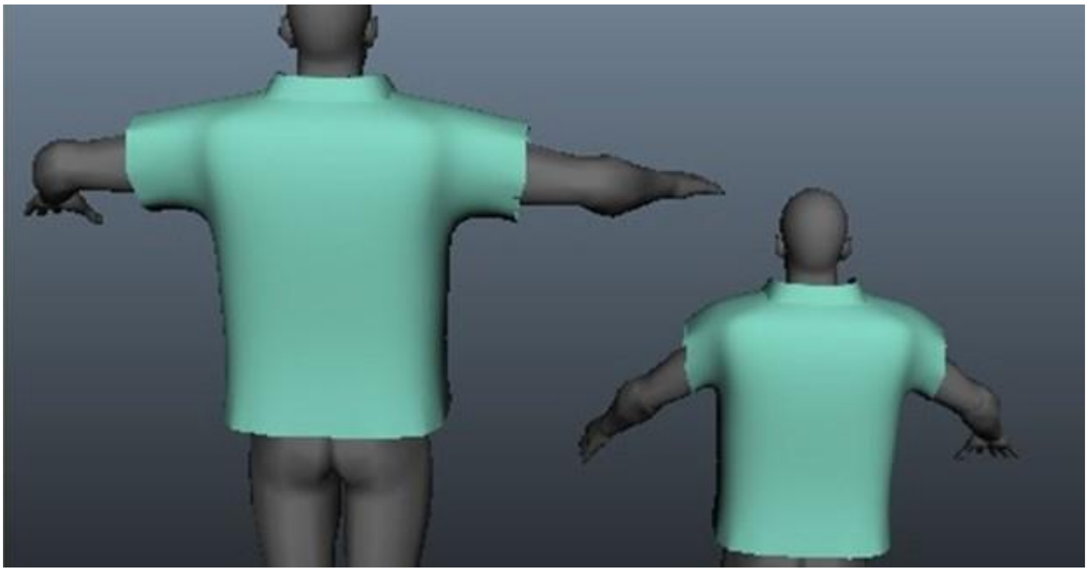


Figure 29: Fitting effect of shirt and two manikins with various body proportions in front view and back view. The two manikins in the above images are formed by proportional deformation of one manikin in different body parts. It can be seen that the garment fits the two manikins successfully.

7.4 Efficiency of this System

The tables below demonstrate the timing of a shirt fit on various shapes of manikins. The timing of others styles of upper garment modeling and fitting will be calculated by using the shirt as a sample with similar number of vertices. The computing time influenced by the number of vertices on the manikins and number of vertices on the garment before smoothing (for style of men's shirt, the total vertices number is almost same as number of feature points). The algorithm of garment template modeling method, components extruding method and template fitting method are test in Maya 2012 on Intel Core 2 Duo, T6500 2.1GHZ.

Vertices on manikin (before smoothing)	8000	12000	15000
Vertices on upper garment	100	100	100
Time in seconds	0.5	0.8	1.1

Table 1: Time testing for one garment on different manikins.

Vertices on manikin (before smoothing)	12000	12000	12000
Vertices on upper garment	100	200	300
Time in seconds	0.8	0.83	0.9

Table 2: Time testing for different garment on one manikin.

It can be seen from the time testing result that the number of vertices on the upper garment has more influence than the complexity of manikins. The timing of the mass fitting, or the accumulation of individual time, was also tested.

As the aim of this system is to reduce labor and save time in the traditional garment modeling process, it is necessary to compare the traditional garment modeling method with this automatic garment modeling and fitting method in terms of both efficiency and accuracy. The compared results are shown in Table 3 below. The experiment object is a same cloth.

	Traditional garment modeling method	Automatic garment modeling and fitting method
Modeling time	2-3 hours	1 second
Accuracy of initial modeling	Low accuracy after pattern sewing	High accuracy with a little intersection
Accuracy after adjustment	High accuracy after adjust vertices and faces for half an hour	Higher accuracy after adjust vertices by local shape deformation method

Table 3: Comparison of traditional garment modeling method and automatic garment modeling and fitting method.

7.5 Effect of Mass Modeling and Fitting

The system that has been designed in this project allows the user to set up multiple manikins at one time by selecting multiple manikins in any order in the *scene* or in the *Outliner*. The function of *Outliner* is to present a hierarchical list of all objects in the scene in outline form (Maya 2012 help file). The figures below demonstrate the garment mass modeling and fitting effect.

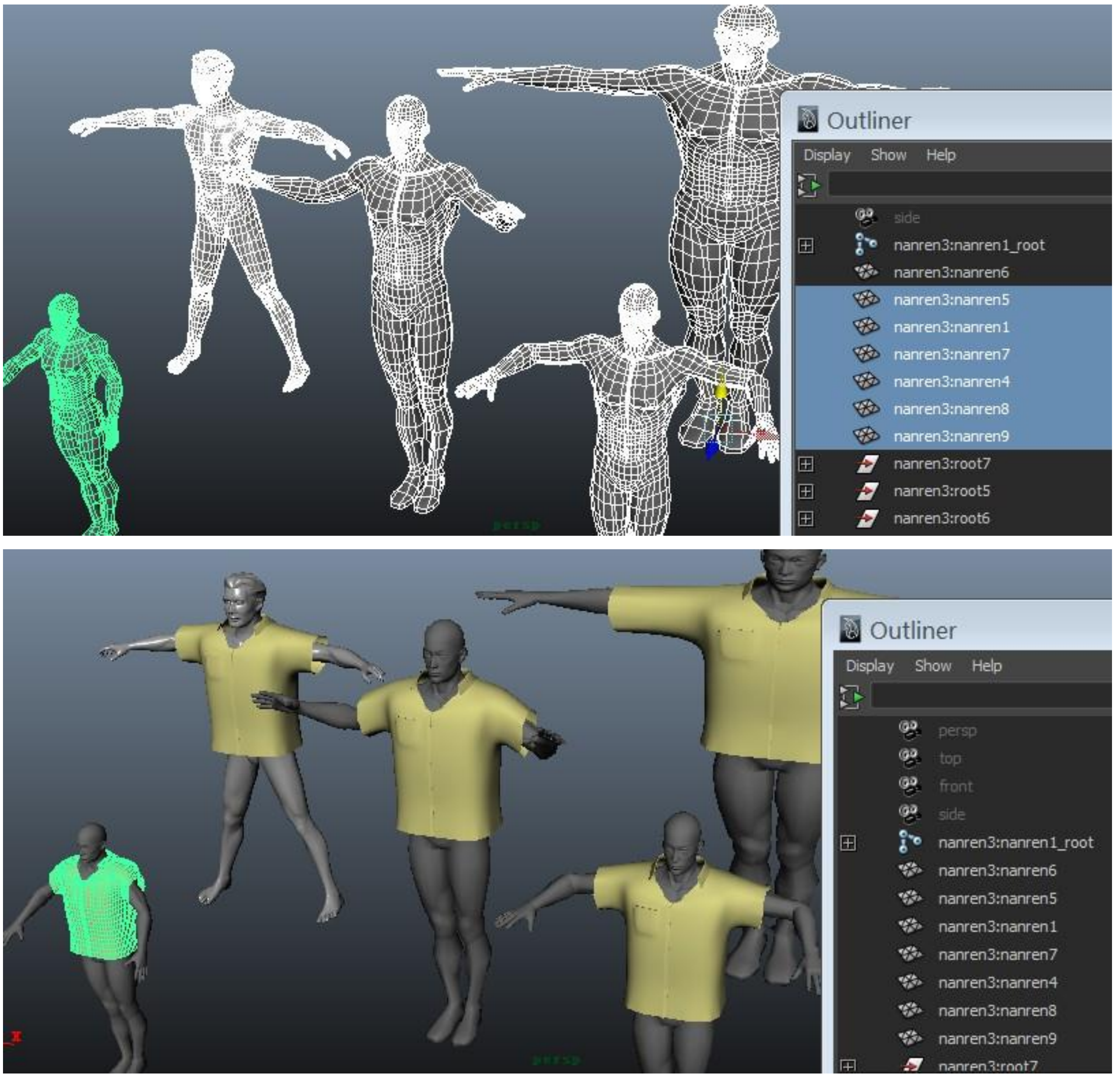


Figure 30: Garment mass modeling and fitting effect

7.6 Summary

The garment template automatic modeling and fitting method performed well in the modeling of any combination of garment components, fitting different arms posture, and fitting different body shapes and mass modeling. However, for the complex manikin, a little intersection must take place in the fitting process.

8 Evaluation

The first objective of this project was to set up the initialization process of the plug-in and to embed it into the Maya interface. This objective can be regarded as successful for the following reasons. First of all, this system is designed with an installer, which means that the system was encapsulated into a packet with one outward interface for the user to install it. The system has been tested on two different operating systems: Windows 7 and Linux Ubuntu. Apart from the script file and icon file directory, there are a variety of different operating systems. The installer can be executed easily by just typing the "installFit" command in command window in Maya. This system was developed in Maya 2012 student version. It can also be installed in the latest version: Maya 2013. Therefore, this system has a good compatibility. Secondly, the user interface is designed with a user-friendly structure and layout. The operation steps are listed clearly on the user interface by using corresponding components to make it simple and understandable. Each button is even associated with a pop-up annotation to stress its function when the mouse moves over it. The layout of the user interface is designed similarly to the default window layout in Maya.

The function of resetting the pivot point of the figure model has also been successfully completed. This function has been tested on five different manikins that consisted of different number of polygons. No matter where the original positions of pivot points are, they will be translated to the center of manikins' feet automatically. The next objective of garment template modeling is also successfully achieved. In order to ensure users could combine any element in design of cloth, the system built the main surface of the garment template first and then extrudes the garment components according to the user's preferences. Based on the parameters of the garment, the feature points can be calculated precisely corresponding to the vertices on the mesh. This system takes shirts as an example to prove that the modeling method is practical and efficient. This system does not provide a complete array of garment styles for two reasons. The first is that, due to the development of society, the styles of clothing are endless. Therefore an efficient, practical modeling method is more important than a complete system. The database of garment can be updated by other users or modelers as they design in the future. The second reason is that it is time-consuming to gather another series of feature point information from garment parameters and test the modeling effect over many trials. Overall, the objective of automatic garment modeling can be considered to have been successfully achieved.

The fourth objective of this project was to achieve automatic garment fitting. From the result in Chapter 7, we can see that the modeled garment could fit manikins successfully in three aspects: completely different manikins, the same manikin with different proportion in body parts, and the same manikin with different proportions in height, width and depth. For the more complex manikins, some intersection may occur due to the limited number of feature points on the garment and manikin. The feature points are selected based on garment parameters. The number of feature points cannot be too high, since this will take much more time in extracting and calculation. Therefore, for more complex manikins, modelers can add relevant number of feature points. If the orientation of the arm is beyond the range of the extruding direction, the intersection will appear during the fitting process. The reason for this phenomenon has been addressed in Chapter 7. However, despite the flexibility of the program, the default posture of manikins must be in a standard standing posture; any other kind of posture will mean the modeler will fail to rig the body (Maya 2012 help file). Therefore, this kind of situation will not be considered in this project. Overall, the objective of automatic garment fitting can be considered as having been successfully achieved as well.

The local shape deformation is the fifth objective in this project. The result can be seen in Chapter 6, in which it was shown that the user can adjust the local shape of the cloth by selecting the vertices first then choosing to expand or shrink distance in the user interface. This local shape deformation method successfully preserves cloth design during the deformation process. One condition of this method is that the operation should be on vertices only.

9 Future Works

It can be seen from the evaluation result that all objectives proposed in the beginning of dissertation have been successfully achieved. Although there are some limitations in the garment automatic modeling and fitting method, this project met these challenges either with respectable success or by outlining a potential solution.

The first limitation is the intersection problem. One significant cause is the lower number of feature points, which could be improved by setting the right amount of feature points according to the complexity of garment and manikin. Another limitation is that the system cannot read figures with a strange posture. The main reason is the feature points extracting on the manikin are sensitive to the quality of the cross-parameterizations; this can be improved in future work. More styles of garment should also be added into the system, like dresses, coats, pants, and so on; this is a necessary step to transform the system into a real product. Lastly, though the local deformation method performs well in this system, it still needs a few operation steps. This part can be designed more intelligently in terms of detecting the local shapes that need to be adjusted in future work.

Reference

Arnulph Fuhrmann, Clemens Gross, Volker Luckas, and Andreas Weber. (2003). Interaction-free dressing of virtual humans. *Computer & Graphics, 27(1):71–82.*

Brouet, R., Sheffer, A., Boissieux, L., Cani, M. (2012). Design Preserving Garment Transfer. *published in "ACM Transactions on Graphics. hal-00695903, version 1 – 16.*

C. Robson, R. Maharik, A. Sheffer, and N. Carr. (2011). Context-aware garment modeling from sketches. *Presented at Computers & Graphics, pp.604-613.*

Chen Hong, Xu Zi Jian, Liu Zi Qiang. (2006). Composite Templates for Cloth Modeling and Sketching. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*

David A. D. Gould. (2003). Complete Maya Programming. *Morgan Kaufmann Publishers.*

Decaudin, P., Julius, D., Wither, J., Boissieux, L., Sheffer, A., Cani, M-P. (2006). Virtual Garments:A Fully Geometric Approach for Clothing Design. *EUROGRAPHICS. Volume 25, Number 3.*

Donald Hearn. (1997). Computer Graphics C Version. *published by Prentice Hall, Inc.*
Emmanuel Turquin, Marie-Paule Cani and John F. Hughes. 2004. Sketching garments for virtual characters. *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling Article No. 28.*

Gillies, M., Ballin, D., Csáji, B., C. (2004). Efficient Clothing Fitting from Data. *Journal of WSCG, Vol.12, No.1-3, ISSN 1213-6972.*

Groß, C., Fuhrmann, A., Luckas, V. (2003). Automatic Pre-Positioning Of Virtual Clothing. *SCCG 2003 ACM proceedings.*

Hing N.Ng and Richard L.Grimsdale. (1996). Computer Graphics Techniques for Modeling Cloth. *IEEE Computer Graphics and Applications.*

Hu, X., Bai, Y. (2009). Review of Cloth Modeling. *IEEE ISECS International Colloquium on Computing, Communication, Control, and Management.*

Li, Z., Jin, X., Barsky, B., Liu, J. (2009). 3D Clothing Fitting Based on the Geometric Feature Matching. *International Conference on Computer-Aided Design and Computer Graphics, Nov 2009*

Meng, Y., Wang, C.L., Jin, X. (2005). Flexible Shape Control for Automatic Resizing of Apparel Products. *Computer-Aided Design*

Nadia Magnenat-Thalmann, Seo Hyewon, and Cordier Frederic. (2004). Automatic Modeling of Virtual Humans and Body Clothing. *d. Comput. Sci. & Technol. VoI.19, No.5, pp.575 584.*

Olaf Etzmu,Michael Keckeisen. (2003). A Fast Finite Element Solution for Cloth Modelling. *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications.*

Turquin, E., J. Wither, L. Boissieux, M. Cani and J. Hughes. (2007). A Sketch-Based Interface for Clothing Virtual Characters. *IEEE Computer graphics and applications. 27(1) , pp. 72-81.*

Vladislav Kraevoy, Alla Sheffer. (2004). Cross-parameterization and compatible remeshing of 3D models. *SIGGRAPH '04 ACM SIGGRAPH 2004 Papers, Pages 861-869.*

Wang, C., Wang, W., Yuen, W.(2005). Design Automation for Customized Apparel Products. *Journal, Computer-Aided Design archive, Volume 37 Issue 7, June, 2005, Pages 675-691.*

Xiao Hu LIU and Yu Wen WU. (2009). A 3D Display System for Cloth Online Virtual Fitting Room. *2009 World Congress on Computer Science and Information Engineering.*

Young, Julia Ditto. (1902). The Rise of the Shirt Waist. *Good Housekeeping, pp.354-357*

Yue L, XiaoGang W. (2009). An automatic clothes designing system based on different databases. *2009 Third International Symposium on Intelligent Information Technology Application Workshops.*

Yue L, XiaoGang W. (2009). Modeling method of clothes for automatic design clothes. *2009 Third International Symposium on Intelligent Information Technology Application Workshops.*

Appendix: Source code

Main part of function of garment modeling and fitting.

```
global proc createTemplate(int $CFlag, int $ZFlag, int $PFlag, int $SFlag)
{
int $i;
float $pointYvalue[],$maxYvalue[],$minYvalue[];
float $bodywScale,$bodybScale;
$objs = `ls -selection`;
for( $obj in $objs)
{
    float $pb[3] = `xform -q -ws -t $obj`;
      //for ($each[] in $vtxAttr)
    for( $i=0; $i< 15000; $i++)
    {
//float $py[3] = $each;
  float $py[3] = `xform -q -ws -t ($obj +".vtx["+$i+"]")`;
  $pointYvalue[$i]= $py[1];
  if ( $pointYvalue[$i] > $maxYvalue[0])
        $maxYvalue[0]= $pointYvalue[$i];
  if ( $pointYvalue[$i] < $minYvalue[0])
        $minYvalue[0]= $pointYvalue[$i];
    }
print ("\n" + $maxYvalue[0] + "\n" + $minYvalue[0]);
$bodyLength = $maxYvalue[0] - $minYvalue[0];
print ("\n" + $bodyLength);
$clothYPosition = $minYvalue[0] + $bodyLength*0.74;

$maxYvalue[0] = 0;
$minYvalue[0] = 0;

$clothWidth = $clothHeight*7/7.2;
$clothDepth = $clothHeight*5.8/7.2;

float $waist1[3] = `xform -q -ws -t ($obj +".vtx[3984]")`;
float $waist2[3] = `xform -q -ws -t ($obj +".vtx[2249]")`;
float $clothWaist =    abs($waist1[0]-$waist2[0]);
print ("\n" + $clothWaist);

float $back1[3] = `xform -q -ws -t ($obj +".vtx[2755]")`;
float $back2[3] = `xform -q -ws -t ($obj +".vtx[2255]")`;
```

```
float $clothBack =    abs($back1[2]-$back2[2]);
print ("\n" + $clothBack);

setToolTo CreatePolyCubeCtx;
polyCube -ch on -o on -w $clothWidth -h $clothHeight -d $clothDepth -sw 4 -sh 4 -sd 4 -cuv 4 ;
move -r $pb[0] $clothYPosition $pb[2];

$clothCube = `ls -selection`;
for( $clothC in $clothCube)
select -r ($clothC + ".f[73:74]") ($clothC + ".f[77:78]") ($clothC + ".f[89:90]") ($clothC + ".f[93:94]")
($clothC + ".f[21:22]") ($clothC + ".f[25:26]") ($clothC + ".f[48:63]");
doDelete;

if ($SFlag == 1)
{
/////////////left sleeve
float $leftshoulder[3] = `xform -q -ws -t ($obj + "_LeftShoulder")`;
float $leftelbow[3] = `xform -q -ws -t ($obj + "_LeftElbow")`;
float                                     $leftshoulderOrent[3]                                     =
{($leftelbow[0]-$leftshoulder[0]),($leftelbow[1]-$leftshoulder[1]),($leftelbow[2]-$leftshoulder[2])};
print ("\n" + $leftshoulderOrent[0]);
print ("\n" + $leftshoulderOrent[1]);
print ("\n" + $leftshoulderOrent[2]);

float $leftshoulderpiv[3] = `xform -q -ws -sp ($clothC + ".e[78]") ($clothC + ".e[82]") ($clothC
+".e[121:122]") ($clothC +".e[132:135]")`;

if ($PFlag == 1)
{
select -r ($clothC + ".f[32:33]") ($clothC + ".f[38:39]");
float $pocket1[3] = `xform -q -ws -t ($clothC +".vtx[200]")`;
polyExtrudeFacet -constructionHistory 1 -keepFacesTogether 1 -pvx $pocket1[0] -pvy $pocket1[1] -pvz
$pocket1[2] -divisions 1 -twist 0 -taper 1 -off 0 -thickness 0 -smoothingAngle 30 ($clothC + ".f[32:33]")
($clothC + ".f[38:39]");
move -r 0 0 0.06;

select -r ($clothC + ".f[524]") ($clothC + ".f[522]");
doDelete;

select -r $clothC;
polySmooth    -mth 0 -dv 1 -bnr 1 -c 1 -kb 1 -ksb 1 -khe 0 -kt 1 -kmb 1 -suv 1 -peh 0 -sl 1 -dpe 1 -ps 0.1
-ro 1 -ch 1 $clothC;

}
```

```
////////////////////
select -r $clothC;
}
}
global proc resetPivot()
{
   $objs = `ls -selection`;
        for( $obj in $objs )
             xform -os -piv 0 0 0 $obj;
             float $fB[3] = `xform -q -ws -t $obj`;
             xform -ws -t $fB[0] 0 $fB[2] $obj;
}
```

Function of local shape deformation

```
global proc moveVertAlongNormal(float $amountToMoveFromSlider)
{
        string $sel[] = `ls -sl -fl`;
        for ($vert in $sel)
        {
                select -r $vert;
                if (`nodeType $vert` != "mesh"){
                warning "Works only on polygon vertices, skipping this selection.";
                continue;
                }
                float $arrayOfVectors[9] = `polyNormalPerVertex -q -xyz`;
                //
                // average the 3 vertex normals into one direction:
                float $vN[3];
                $vN[0]=(($arrayOfVectors[0]+$arrayOfVectors[3]+$arrayOfVectors[6])/3);
                $vN[1]=(($arrayOfVectors[1]+$arrayOfVectors[4]+$arrayOfVectors[7])/3);
                $vN[2]=(($arrayOfVectors[2]+$arrayOfVectors[5]+$arrayOfVectors[8])/3);
                //
                // unitize the normal to one before using the multiplier:
                vector $unitizedNormal = `unit << $vN[0], $vN[1], $vN[2] >>`;
                move -r ($amountToMoveFromSlider * $unitizedNormal.x)
                        ($amountToMoveFromSlider * $unitizedNormal.y)
                        ($amountToMoveFromSlider * $unitizedNormal.z) $vert;
        }
        //
        // restore original vert selection
        select $sel;
}
```

Function of Installer

```
global proc installFit()
{
	global string $gShelfTopLevel;
	if (`tabLayout -exists $gShelfTopLevel`)
	{
		shelfButton
		-parent ($gShelfTopLevel + "|" + `tabLayout -q -st $gShelfTopLevel`)
		-command "showMyWindow"
		-image1 "out_cpClothSolver.png"
		-annotation "Maya cloth fitting!";
	}
	else{
		error "You need a shelf for this Install to complete!";
	}
}
```