A 3D GARMENT DESIGN AND SIMULATION SYSTEM

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Funda Durupınar

July, 2004

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Uğur Güdükbay (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Bülent Özgüç

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Cevdet Aykanat

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

A 3D GARMENT DESIGN AND SIMULATION SYSTEM

Funda Durupınar
M.S. in Computer Engineering
Supervisor: Asst. Prof. Dr. Uğur Güdükbay
July, 2004

In this thesis study, a 3D graphics environment for virtual garment design and simulation is presented. The proposed system enables the three dimensional construction of a garment from its two dimensional cloth panels, for which the underlying structure is a mass-spring model. Construction of the garment is performed through cutting, boundary smoothing , seaming and scaling. Afterwards, it is possible to do fitting on virtual mannequins like in the real life as if in a tailor's workshop. The behavior of cloth under different environmental conditions is implemented applying a physically-based approach. As well as the simulation of the draping of garments, efficient and realistic visualization of garments is an important issue in cloth modelling. There are various material types and reflectance properties for fabrics. We have implemented a number of material and rendering options such as knitwear, woven cloth and standard shading methods such as Gouraud shading. Performance results of the system are presented at the end.

Keywords: garment design, garment simulation, fabric rendering, physically-based modeling.

ÖZET

ÜÇ BOYUTLU GİYSİ TASARIM VE SİMÜLASYON SİSTEMİ

Funda Durupınar
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Asst. Prof. Dr. Uğur Güdükbay
Temmuz, 2004

Bu tez çalışmasında, üç boyutlu bir sanal giysi tasarım ve simülasyon sistemi tanıtılmıştır. Önerilen sistem, üç boyutlu bir giysinin, giysiyi oluşturan panellerden yapılandırılmasına olanak sağlamaktadır. Panellerin temelini bir yay-parçacık modeli oluşturmaktadır. Giysinin oluşturulması, kesme, dikme, çevresini düzeltme ve büyütme/küçültme aşamalarından geçerek gerçekleştirilmektedir. Daha sonra gerçek hayatta bir terzinin atölyesindeki olduğu gibi bir manken üzerinde prova yapmak mümkündür. Kumaşın değişik çevresel koşullardaki davranışı, fiziksel bir yaklaşım izlenerek gerçekleştirilmiştir. Giysilerin salınımının yanısıra, verimli ve gerçekçi görüntülenmesi de kumaş modellemede önemli bir konudur. Kumaşların değişik materyal tipleri ve yansıma özellikleri vardır. Sistemimizde örgü, dokuma ve standart boyama yöntemleri gibi çeşitli materyal özellikleri ve boyama seçenekleri gerçekleştirilmiştir. Programın performans sonuçları tezde sunulmuştur.

Anahtar sözcükler: giysi tasarımı, giysi simülasyonu, kumaş boyama, fiziksel modelleme.

Acknowledgement

I gratefully thank my supervisor Asst. Prof. Dr. Uğur Güdükbay for his supervision, guidance, and suggestions throughout the development of this thesis.

I am gratefully thankful to Prof. Dr. Bülent Özgüç for his support, guidance and great help throughout my master's study.

I would also like to give special thanks to my thesis committee member Prof. Dr. Cevdet Aykanat for his valuable comments.

Besides, I would also like to thank İlknur Kaynar for her invaluable support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cloth simulation has been a challenging issue in computer graphics for a long time. The interest of the computer graphics society in this area has emerged in the 1980's. Since then, extensive research has been done on cloth simulation. Today, there is an increasing demand for the involvement of computer graphics in textile industry and entertainment industry. Especially, computer aided design systems, fashion design programs, new generation movies and computer games require new tools that perform realistic simulations. However, the demands vary for different areas. For instance, a textile engineering program requires accuracy and high precision whereas a computer game is contented with realistic appearance and efficiency. Therefore, a compromise between accuracy and efficiency is essential.

The basic issues to consider when modelling the cloth are as follows ([53]):

- Geometrical representation
- Behavior
- Interaction with the environment
- Rendering

A full cloth simulation program should take all of these issues into consideration. The first issue, geometrical representation, is about the accurate representation of the shape of the cloth. The second one, behavior of the cloth, shows the various properties of the cloth's material such as stress-strain curves. The third issue, which is the interaction with the environment deals with how the cloth behaves under environmental conditions such as gravity, wind or collisions with rigid objects. Finally, rendering the cloth is an open-research area, which involves the visualization of fabrics.

There are basically two methods for the simulation of cloth: geometrical models and physically-based models. Physically-based models are more realistic and easy to implement compared to the geometrical models. Among all the methods, the simplest and the mostly preferred method is the mass-spring system. This is a kind of particle system, in which the set of particles are interconnected by springs.

One step further in cloth simulation is the simulation of garments, which can be considered an assembly of different cloth pieces. There are various computer aided tools for designing garments in the clothing industry. However, these are mostly in 2D and do not allow the fitting of a garment on a mannequin . In recent years, there is a great deal of research for 3D garment design. In most of the developed software tools, the user draws the 2D pattern of the garment parts and the system triangulates these patterns and converts them into physically-based models [12, 13, 40, 53]. In contrast, some systems prefer to dress the virtual mannequins in 3D [7, 15].

1.1 Our Approach

The purpose of our system is to design various garment models with different textile materials and then simulate the garment worn by a virtual mannequin. Our system is built on top of a human modelling and animation tool implemented by Aydemir Memişoğlu and Şahin Yeşil.

We create garments from 2D patterns (actually, 2D panels of 3D geometric points) and then sew them together as in the real-life process of creating a garment. In the existing systems, it is difficult to implement different textile types such as woven or knitted fabrics, since the creation of weave and knit patterns require regular mesh structures. Thus, we first create the garment panels by performing cutting on the rectangular mesh. Cutting is done by selecting the particles comprising the boundary of the cloth panel. The mass-spring system that has been used for the 3D simulation of cloth enables the parametrization of internal forces such as bending, stretching or damping of the fabric. This provides us with the ability to implement different types of fabric by just modifying these parameters. In addition, external forces such as wind, gravity or air resistance can be applied to the cloth and the behavior of the garment under different environmental conditions can be observed.

1.2 The System Architecture

Our system is composed of three modules: motion design, garment design and garment and human simulation. Motion design module allows the user to define various human motion behaviors by adjusting the position, distance and rotation curves of the joint points. The details of this module are out of the scope of this study and they are explained in [35, 61]. Garment design module is an effective tool for creating 3D garments from their 2D panels through cutting and seaming. It enables the designer to position the 3D garment around the virtual character and sew the garment panels. In addition, the system provides different options for rendering garments. Simulation module provides the functionality for human motion, cloth deformation and collision handling. The system architecture is given in Figure 1.1

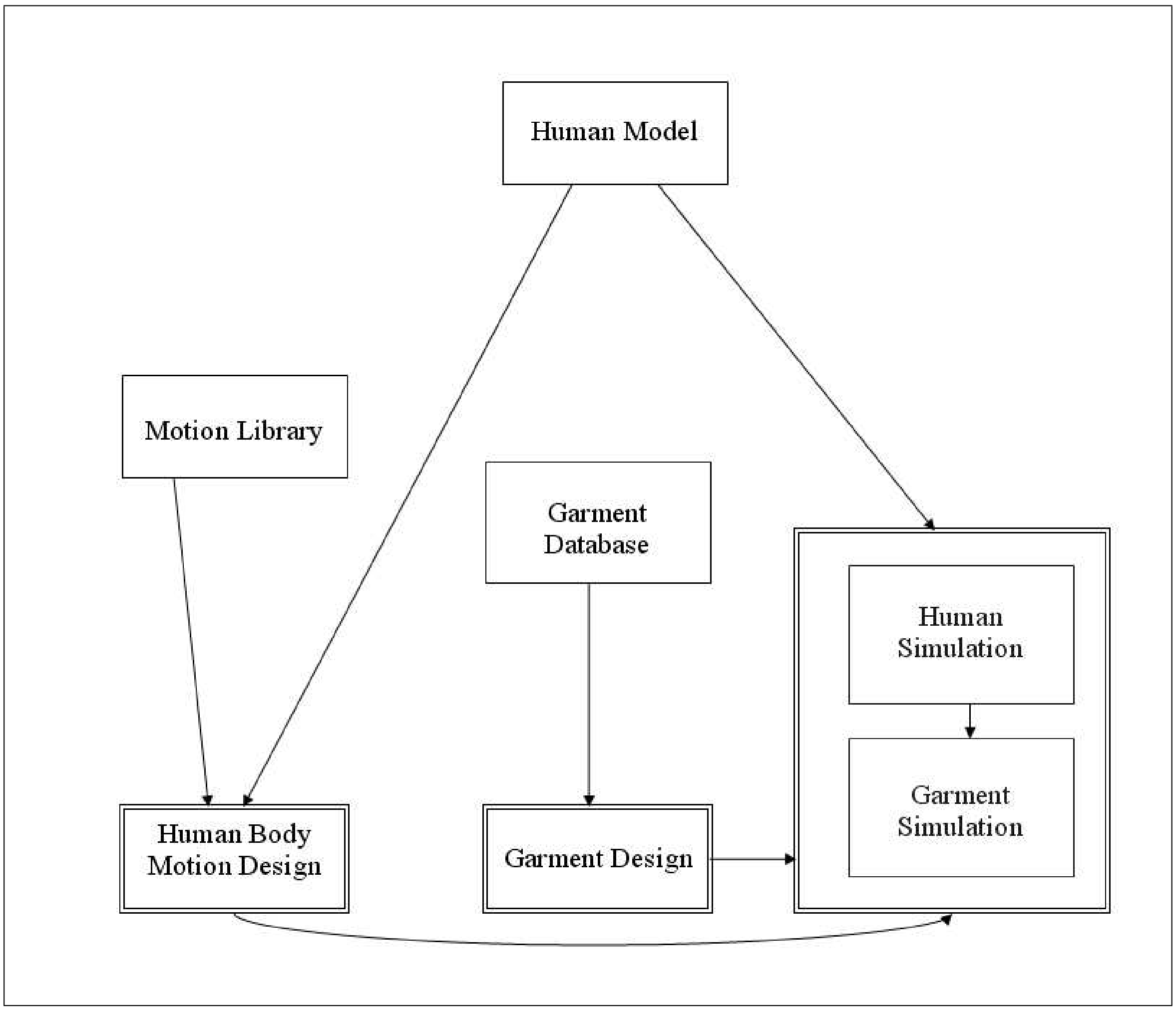


Figure 1.1: The system architecture

1.3 Organization of the Thesis

The organization of the thesis is as follows: Chapter 2 reviews the state of art in garment design, simulation and rendering fields. Chapter 3 explains the garment design process and Chapter 4 describes the garment simulation process in our system. Chapter 5 overviews the collision handling methods adopted in our system. Chapter 6 gives some experimental results of the study. Chapter 7 presents the conclusions and future work about the subject. Finally, the system at work and the user interface are explained in the Appendix.

Chapter 2

Background

Garment simulation has many potential application areas such as the entertainment industry, CAD/CAM systems, e-commerce or textile industry. The evolution of virtual characters in movies, virtual worlds or computer games has created a demand for the dressing of the virtual characters. Hence, the entertainment industry requires fast, convenient and efficient tools for designing garments that look realistic enough. In addition, fashion designers wish to see the appearance of a garment on a virtual mannequin before the garment is manufactured. Similarly, customers of an interactive e-commerce site may opt for trying their garments on and make preferences on the design, texture or color of the garments before they buy them. Thus, CAD/ CAM tools and interactive online stores make use of garment design and simulation systems. Again, these systems require efficiency and realism. On the other hand, textile engineers require accuracy and precision about the behavior of the cloth material. Thus, they deal with the properties of cloth such as Young's modulus, bending modulus, stress-strain curves, etc. [53].

In the following sections, the state of the art in garment simulation and design will be explained in detail. Garment design consists of the construction of three dimensional garments from their components, which are cloth panels in 2D. On the other hand, garment simulation deals with the mechanical and geometrical modelling, as well as the simulation scheme with numerical integration, environmental interaction and rendering issues.

2.1 State of the Art in Garment Design

Currently, there are various software tools for designing apparel in 2D; however, there is a strong need for 3D tools for designers. Therefore, in recent years, such software has been developed extensively. Garment design using computer graphics has two main approaches:

- designing 2D panels, seaming them and converting them into 3D, and
- designing 3D garments around a virtual mannequin and extracting its 2D components.

The most well-known example for the first type of approach is the MIRA-Cloth Software developed in the Miralab. They find this method more intuitive and similar to real life. In [53], it is stated that the MIRACloth System consists of the design of garment patterns, putting patterns on bodies, seaming and constructing garments, animation of garments, defining garment materials and textures, cutting and modifications. In MIRACloth, the design of the patterns consists of editing the garment as a set of 2D polygons linked by seaming lines. Then, these patterns are digitized and triangulated to construct a particle-based system. After that, these patterns are placed around the human body and sewn together. Before the animation, the mechanical parameters of the clothes are set up to reflect the correct behavior of the fabric. Animation is performed by moving the body from a recorded sequence using animation playback. Garments move with the body by means of the reactions to collision response and friction. Similar studies conducted in Miralab include [60, 13, 40].

For instance, the system presented in [60] is one of the initial garment design tools. This is a tool for the interactive design of garments in three dimensions. The system presented for the tool consists of five main parts, which are: the interactive graphic interface for the 2D design of panels, deformable cloth model, pattern library of garment templates, movable human body model and the output interface. The designer draws patterns for 2D garment panels and then these templates are digitized. It is also possible in the system to animate the garments with human movements.

Furthermore, Protopsaltou et al. present a system for producing 3D clothes with realistic behavior [40]. The main purpose of the system is to build an interactive, compelling and realistic virtual shop, where customers can select the garment designs that they prefer and can see these garments on virtually animated bodies. Using the system, the user can create the standard female or male virtual bodies and create and seem together the 2D garment patterns around the virtual human body. This provides the initial shape of the garment.A simulation is made using the seemed garment by applying physical parameters based on real fabric properties. As a result, the whole system enables the customer to visualize clothes, to animate them, and to add interactivity in addition to view garments fitted onto their own virtual bodies. The main problem in that system is such that the simulation of garment movement is performed using a physics based model, making it impossible to achieve real-time performance. Therefore, the simulation results are prerecorded in order to display the animated garment at an interactive rate.

Later, Cordier et al. present a Web application that provides more powerful manipulation of garment-related items such as sizing and pattern derivation [13]. The customer can select various garments and fit these garments onto the 3D mannequin that is adjusted according to his/her measurements. Moreover, the movement of the garment is simulated to give the customer an idea about how the garment will look on his/her body. The main difference of this system from the one presented in [40] is that the garment simulation is calculated on the fly while keeping the response time interactive. The system sends body/garment sizing as well as cloth/skin animation to the client side. Then, the major part of the content to be manipulated is generated on the client side rather than on the server, which provides a minimal response time to the user. Furthermore, a significant step in this system is the application of 3D graphics technology to create and simulate the virtual store.

The Virtual Try-On project [24] leads the development of new VR technologies that forms the basis of a realistic, three dimensional simulation and visualization of garments put on virtual counterparts of real customers. The complete process chain starts with the 3D scanning of the human body up to a photo-realistic 3D presentation of the virtual customer dressed in the selected pieces of garments.

In addition, the work presented in [11] is a simulator that complements traditional CAD systems that are based on 2D graphics by defining a cross-application data exchange format among the different CAD systems and applications in textile industry. Then, the VRML-based 3D previews of the garment are produced so that they could be published on the Web.

The second type of approach is illustrated in [7]. Bonte et al. outline a 3D graphic environment for industrial applications in the clothing industry in their study. Their system enables the design of a garment in 3D, testing different fabric types on the final garment and the automatic generation of 2D patterns from the 3D representation. 3D modelling of the garment is performed by browsing a library to select the closest garment to the final one and creating the 3D shape by stating values or selecting points or lines on the mannequin. In addition, darts, seams, holes and style lines are defined interactively in the 3D shape. 2D patterns are generated only after the completion of the 3D garment. Flattening is performed by flattening the triangles of the 3D meshes one by one with a non-isometric method and then applying a relaxation procedure to refine the 2D points until error is minimized. In order to simulate the garment in 3D, a particle-based approach is adopted.

The works described in [14] and [15] are also similar to [7] and developed within the framework of the Brite Project MASCOT. In addition, the project "CADwalk" for the 3D display of garments in common PCs is also an example of the second approach [43]. In that study, the figurine can be scaled to the customer's measurements in order to show the customer how she/he is going to look like in the new clothes. Roediger et al. follow the geometric approach proposed by Hinds and McCartney in [26].

As a final example, Wang et al. present a new approach for intuitively modelling a 3D garment around a 3D human model by two-dimensional sketches [55]. First, a feature template for creating a customized 3D garment is defined according to the features on a human model; second, the profiles of the 3D garment are specified through 2D sketches; finally, a smooth mesh surface interpolating the specified profiles is constructed by a modified variational subdivision scheme. The system has several advantages compared to the earlier approaches. First, the system provides direct design of garment patterns through 2D strokes in the 3D space. In addition, the system enables the regeneration of the patterns when creating the same style of garment for other human models. Furthermore, the 2D sketched input method provided in the system is beneficial for the designers since most designers still prefer to express their creative design ideas through 2D sketches.

2.2 State of the Art in Garment Simulation

Garment simulation includes the geometrical and mechanical behavior of the clothes, interaction of them with the environment and rendering them. In this section, the approaches for the behavior of the cloth model are examined. Then, collision detection and response techniques are reviewed. Finally, rendering techniques are presented.

2.2.1 Cloth Model

Before the mid 1980s, cloth was modelled as a texture mapped on rigid surfaces. Today, there are two basic approaches in simulating the cloth using computer graphics: geometrical models and physically-based models. In [8], Breen gives a survey about cloth modelling methods.

2.2.1.1 Geometrical Models

Weil [56] defines a geometric approach that approximates the folds in a constrained piece of square cloth. He models a cloth hanging from several constraint points. The cloth structure is modelled topologically as a 2D grid of 3D geometric points. The first step recursively connects constraint points,with catenary curves as an initial approximation. The grid points lying between the constraint points are placed on the 3D catenary curves. When two curves cross, but do not intersect at the same point, the lower curve is eliminated. New constraint points and catenary curves are then added until all points on the catenary curves lie within the convex hull of the constraint points. The second pass uses a relaxation technique to enforce distance constraints between all grid points in order to create smooth cloth-like folds in the rectangular grid. Weil does not include any mechanical information in his model. Cloth stiffness is modelled with a second-order distance constraint. He also includes a rendering technique where the cloth surface is subsequently modelled as a collection of cylinders.

Hinds and McCartney [26] present a system for the interactive design of garments by allowing the user to create a geometric model of a garment by specifying the outlines of the garment panels on a mannequin, and then generating offsets from the mannequin surface. They do not physically model the cloth surfaces, instead, folds are sinusoidal offsets added into the geometrical model.

2.2.1.2 Physically-based Models

Physically-based models are more accurate and easier to implement compared to the geometrical methods.

Spring-Mass Models

Haumann and Parent [25] present the spring mass models. Their models included a point mass, environmental forces, a spring connecting two point masses, a hinge that connects the two triangles formed by two point masses and aerodynamic drag and wind actors. In this model, each vertex is converted into a point mass, each edge into a spring and each set of adjacent faces into a hinge. Given initial conditions, the motion of the model is calculated by applying standard Newton's laws of motion. Their model could not accurately simulate the behavior of woven cloth, only some kind of deformable surface.

Provot [41] describes a similar spring-mass system. The mass particles, arranged in a rectilinear grid are connected with the three types of springs, which were structural, shear and flexion springs. In addition, distance constraints, which eliminate the unacceptable elongations of the cloth are introduced in this method.

Models Based on Elasticity Theory

Feynman [20] simulates some mechanical properties of cloth by defining a set of energy functions over a 2D grid of 3D points. The total energy of this cloth model contains tensile strain, bending and gravity terms. He minimizes the energy of the grid with a stochastic technique and a multigrid method. He assumes that cloth is a continuous flexible material and derived his energy functions from the theory of elastic shells. His energy functions are only based on the distance between points and a simple measure of curvature. His approach does not take into account the shearing behavior of the cloth and the self-intersection of the cloth or interaction of arbitrary solid geometric models.

Taking continuum mechanics and differential geometry as their starting point, Terzopoulos et al. [48] develop a wide range of deformable physically based models. They present a simplified set of equations based on elasticity theory that describe elastic and inelastic deformations, interactions with solid geometry and fracture for flexible curves, surfaces and solids. Their technique for modelling the dynamics of elastic objects uses Lagrange's equations of motion. The same group also presents works that deal with the practical issues involved in stitching flat panels into three-dimensional clothes and reducing the number and complexity of the model parameters presented to the user.

Baraff and Witkin [5] describe a simple cloth continuum model that is motivated more by numerical computing issues than a desire for mechanical accuracy. They present a computational framework for producing clothing simulations based on an implicit numerical integration method. Then, Desbrun et al. [17] extend Baraff and Witkin's approach to produce real-time simulations of cloth-like sheets which may be interactively manipulated by a user while colliding with or sliding over numerous rigid objects.

Particle Models

Breen and House [9] develop a non-continuum particle model for cloth drape that explicitly represents the micro-mechanical structure of cloth via an interacting particle system. Their model is based-on the observation that cloth is best described as a mechanism of interacting mechanical parts rather than a continuous substance, and derives its macro-scale dynamics properties from the micro-mechanical interaction between threads. Crossing points of warp and weft threads are represented by particles. These particles interact with adjacent particles and the environment through mechanical connections represented by energy functions. A stochastic gradient descent technique is used to relax the cloth particles toward a final equilibrium position, producing fabric drape. Afterwards, they showed how this model can be used to produce the drape of specific materials accurately. However, the model produces only draped configurations without motion and its original implementation was slow and inefficient [10].

Finite Element Models

Etzmuss et al. [19] present a model based on finite elements that is particularly designed for numerically stiff materials such as textiles. Their system allows fast time stepping by using an implicit integration method. The nonlinear elasticity problem is reduced to a linear, planar one in each step. In contrast to mass-spring models, which deal with regular quadrilateral meshes, this method works independent of the mesh topology.

2.2.2 Integration Methods

During the computation of the evolution of the cloth model, a differential equation system must be solved. Since the cloth system evolves in discrete time steps, the system should be integrated numerically. There are various integration schemes in the literature. The performance of the integration method depends on the following factors ([54]):

- the computation time for one iteration of the algorithm,
- the time step for one iteration,
- the desired accuracy, and
- the numerical stability.

Among all the methods, the integration schemes can be classified into two: explicit and implicit methods.

- Explicit integration methods compute the state of the next time step out of a direct extrapolation of the previous states. Among the explicit integrators, the most important ones are Euler, Midpoint and Runge-Kutta methods. These are relatively easy to implement. However, they need very small integration time steps to guarantee system stability and accuracy.
- Implicit methods deduce the state of the next time step indirectly from an extrapolation of the next state. These methods are able to use larger steps without loss of stability, but they are more complex to implement because they need to solve large linear systems at every integration step. The use of implicit methods in cloth simulation was first proposed by [5].

The integration methods are explained in Chapter 3 in more detail.

2.2.3 Cloth Rendering

As well as the draping and dynamic deformation of cloth, the visual appearance of cloth is crucial when realism is required. In particular, textiles are mainly divided into knitwear and woven cloth. Therefore, rendering of the cloth can be examined in these two basic types.

2.2.3.1 Knitwear

Meissner and Eberhardt [33] present a system that simulates the real, physically correct appearance of knitted fabrics. They use the produced machine code for knitting machines in their system. For this purpose, they take knitting machine data, generate a data structure representing it and apply a particle system for the physical behavior of the knit pattern. Since their application is almost real-time, they do not integrate time consuming visualization techniques such as ray casting. In order to give the yarn a three dimensional appearance, they use line primitives of different size and brightness. They separate the knitting process into three steps: topology generation, topology reduction and topology refinement. The first step includes the generation of a data structure containing the necessary information of the fabric such as material information, thread course in 3D space and thread interaction points. In addition, this step simulates the operation of a knitting machine. The second step reduces the topology by filtering out all the instable interaction points. Finally, the third step equalizes the distortions by storing the length of the initially distributed thread. To simulate the correct physical behavior, a particle system is adopted. Thus, the dynamics of stretching, repelling and bonding of the yarn are calculated. Spring forces are used to pull or repel the neighboring particles and to simulate the bending of a yarn loop.

Zhong et al. present a method for rendering knitwear on free-form surfaces [63]. Free-form surfaces give the system the potential to consider the interaction of the neighboring yarn loops. In addition, the user can determine the fluffiness of the yarn and the irregularity in the yarn positions. In contrast to [33] in which the pattern is based on industrial knitting machines, Zhong et al. model the knitted fabrics made by hand. There are two basic stitches in knitting, plain stitch and the reverse stitch, namely the right loop and the left loop. The knitwear skeleton is composed of an equal partition of a rectangle Q into quadrilateral tiles based on the stitch pattern, considering the irregularities in the yarn positions. Key points in the parameter domain Q are specified for every loop. Then, these key points are connected by using curve segments to get the loop in 3D. The system is capable of rendering the yarn microstructure efficiently by drawing free-form knitwear as Gouraud-shaded polygons. The fluffiness of the yarn can be controlled by applying some random perturbations in the positions of the Gouraud-shaded polygons. Although this method is efficient, the results for close-up views are not very realistic.

Another system that uses free-form surfaces is the one presented by Xu et al [59]. In this system, photorealistic rendering of knitwear is handled by introducing an element called the lumislice. The lumislice represents the radiance from a yarn cross section. They make use of the fact that the structure of yarn is repetitive and they represent knitwear as a collection of identical lumislices in various positions and orientations. In addition, varying levels of detail for different viewing distances is handled by multiresolution lumislices. The framework makes use of hardware aided transparency blending, thus can be implemented easily with standard graphics APIs such as OpenGL. However, this method does not have interactivity.

2.2.3.2 Woven Cloth

When woven cloth is of concern, we cannot simply do texture mapping or rendering because occlusion and self-shadowing terms come into scene due to the structure of the individual threads and their interweaving pattern. Similarly, we cannot exactly simulate the real structure as a result of the computational cost. For instance, methods like ray-tracing are costly [16]. In general, methods in this area make use of the periodicity of the weaving or knitting patterns.

Woven cloth is examined in milliscale and microscale. Milliscale geometry refers to how threads are interwoven, whereas microscale geometry is about the structure of fibers making up the yarns.

In [16], the lighting is computed using a geometrical model of a stitch. Then, by sampling the stitch regularly within a plane, a view-dependent texture with per-pixel normals and material properties is generated. The proposed method is interactive since it uses precomputing by keeping a lookup table of the color and alpha values for each viewing direction. Also, the method is similar to Bidirectional Texture Functions and virtual ray-tracing. The base geometry of the knits and weaves are modeled by using implicit surfaces. For rendering, first, a normal is estimated for each visible point on the object. Then, the Bidirectional Reflectance Distribution Function (BRDF) that would be suitable to obtain the color of each pixel is found. After that, the light and the viewing directions are mapped into the geometry's local coordinate system using the normals. The software evaluation of the BRDF model returns the three colors and an alpha value from the lookup table, which is then written into the framebuffer.

In [18], the weave of the texture is simulated by procedural displacements of the geometry and the loop is represented as a 2D curve. The 3D appearance is given by displacing the loop. The macroscopic structure of the materials is simulated by using displacement shaders. To model the weave of the material variations, the following function is used.

$$Weave(x) = cos(2 * \pi * x * T_{freq} + Phase) * Height,$$

where, T_{freq} is the number of the threads that is requested, $Height$ is the total amplitude of the weave and $Phase$ is the constant that changes due to the evenness or oddness of the rows.

In order to visualize the close-up appearance of threads, Adabala and Thalmann present a technique for procedurally creating the texture of the twists of thread in [2]. The tightness of the twist of fibers and the roughness of the fiber are the parameters that have been introduced. The shape of the twist is presented by a trigonometric function. These resulting textures are then combined with color parameters and used to create weave patterns with a realistic look.

Adabala et al. [1, 3] extend their work to generate texture from Weaving Information Files (WIFs), generate micro and macro detail separation and weave based BRDF. The BRDF is generated by the WIF information.

2.2.4 Collision Handling

Garments interact with the body or with other garment panels. The shape and the movement of the body affect the garments movement. Thus, in order to obtain realistic simulation of garment panels, collision handling must be achieved in an accurate and efficient way.

Collision handling consists of two phases:

- *Collision detection:* Checking the geometrical contacts and proximities between the objects.
- *Collision response:* Correcting the velocities and positions of the colliding objects by applying constraints or forces.

2.2.4.1 Collision Detection

Collision detection is the most time consuming part of the cloth simulation, since the number of geometrical entities, such as nodes, faces, and edges that the collision detection algorithm has to handle is large. The complexity problem leads the development of algorithms that decrease the number of collision tests. Most of these algorithms make specific assumptions about the objects and design solutions based on the geometry of the objects. Lin and Gottschalk [31] have presented an overview of collision detection methods.

The general case of collision detection is the one involving cloth and an object. A particular case is the self-collision detection or the collision between a garment panel and the human body.

Collision Detection Between Cloth and Human Body

Collision detection between cloth and the human body has been widely studied in the last decade. Most of these methods use geometrical collision detection methods with the optimization techniques in order to reduce the number of checks.

Mezger et al. [37] propose a method by combining and improving the techniques, such as hierarchical bounding volumes [51], collision prediction (proximities) through k-DOPs bounding volumes [30] and heuristics [51]. They improve the efficiency of bounding volume hierarchies by adapted techniques. To prune the hierarchy, extended sets of heuristics, such as curvature and coherence criteria, are used. In addition, for preserving large time steps and stability, oriented inflation of bounding volumes is used to detect not only collisions but also proximities. The advantage of this approach is to detect collisions before they occur.

A voxel-based collision detection method for clothed human animation is presented in [62]. They speed up the performance of the voxel-based method by choosing an appropriate voxel size and using a fast voxelization approach. Based on the voxel method, they propose a self-collision detection method and a simplified collision detection method. Firstly, the efficiency of self-collision detection is improved by taking advantages of the curvature and multi-layer methods. Secondly, the efficiency of cloth/human collision detection is improved by introducing auxiliary line segments. Experimental results demonstrate that their method is efficient for clothed human animation. It has, however, a limit in that it can't be applied to interactive garment simulation in the aspect of speed, and it is hard to apply to fast movement since it can only handle collisions occurred in the current frame.

Vassilev and Spanlang [49] propose a cloth-body hardware-assisted method that uses z-buffer not only to compute the depth maps of the body but also to interpolate the body normal vectors and velocities of each vertex. Although the method is fast and does not depend on the complexity of the human body, the maps need to be precomputed for each simulation, which makes it difficult to use in the applications

In addition to the above methods, approaches for animating dressed humans are proposed based on the property that most parts of a garment do not change position relative to the human body [12, 44]. The system described in [12] performs real-time animation of complex garments by classifying the particles of the garments into layers and applying different animation techniques for each of them. The deformation and collision handling of the layers are done depending on how they lie on the body surface and whether they stick or flow on it. The cloth layers are categorized as: stretch, loose and floating clothes. The results in [12] show that the method achieves real time performance compared to the other cloth simulation approaches. In addition, this method can be used only on top of deformable objects such as virtual humans since it uses the deformation of the underlying object.

Self Collision Detection

Self-collision detection is a special case of collision detection where both of the intersecting geometrical primitives belong to the same deformable model. It is a complex task compared to the collision between cloth and an object because some special cases, such as multiple collisions, collision consistency and adjacency in bounding boxes, should be examined carefully. Several methods have been proposed for handling self-collisions accurately and efficiently.

Particular advances in accelerating the self-collision detection are achieved by Volino et al. [51]. They propose a method that is based on geometrical shape regularities. They used a region-merge algorithm to build hierarchies on top of a polygonal mesh, storing adjacency information for the regions. In order to avoid unnecessary self-collision tests in whole branches of the tree, they use the surface curvature optimization method. This method is based on the property that, when a given zone has sufficiently "low curvature", it cannot self-intersect, and all the zones it includes do not intersect with each other. This means, if the branches of the tree correspond to a zone with a sufficiently low curvature, no self-intersection test is done. Furthermore, they also introduce a technique that observes the history of close regions to guarantee a consistent collision response [50]. In their recent publications, they use the k-DOPs as bounding volumes [52].

Provot [42] uses the bounding box hierarchy and surface curvature method [51] for optimizing collision detection. He creates the bounding box hierarchy of the tree by recursively dividing the cloth piece into zones considering the triangles positions. Then, the curvature information of each node is updated bottom-up by using the two angles of its descendant nodes, α_1 and α_2. The leaf nodes have angle $\alpha = 0$, since they have single normals. In self-collision detection, while parsing the tree, the algorithm eliminates the nodes whose bounding boxes do not intersect. In addition, the surface curvature is also considered. If the angle $\alpha > \Pi$, then self collision detection is applied. Otherwise, it is not.

Huh et al. propose one of the best solutions to self-collision problem [27]. They consider the cloth-cloth collision resolution as a special case of deformable N-body collision resolution. They group the particles into parts and using the law of momentum conservation they handle the collisions between these parts. They create a system of linear equations for resolving collisions using a scheme adapted from the simultaneous resolution method for rigid N-body animation. The linear equations are built from the collision relations for the cyclic relatioships in collisions.

Despite some speedup techniques, such as curvature of the cloth surface [52, 42] and bounding volume hierarchies, self-collision handling often fails to get real-time performance.

Optimizing Collision Detection

Optimization techniques are used for efficient collision detection by preventing $O(n^2)$ comparisons. There exist many algorithms depending on the complexity reduction mechanism behind them [53]. The main groups are:

- *Bounding Volumes:* Complex objects or object groups are enclosed within simpler volumes, such as box, sphere, that can be easily tested for collisions.
- *Projection Methods:* Possible collisions are estimated by using separately the projections of the scene along several axes or surfaces.
- *Subdivision Methods:* The problem is decomposed into smaller space volumes or object regions either on the scene space or on the object space. They are usually evaluated through bounding volume techniques. Hierarchical subdivision schemes add efficiency.

- *Proximity Methods:* The scene objects are arranged according to their geometrical neighborhood. The collisions between these objects are detected based on the neighborhood structure.

2.2.4.2 Collision Response

Once the collisions have been detected for a given frame, their effects, such as preventing objects from interpenetrating, and producing contact reaction, have to be taken into account in the mechanical simulation. Collisions between deformable objects are much more difficult to treat than collisions between rigid objects, because a response for each face or particle has to be computed and care must be taken not to introduce additional stiffness.

Several collision response schemes for cloth animation have been presented. In general, there exist four options for the collision response:

- *Constraint-based*: This approach assumes totally inelastic collision. The particle that has collided with an object is constrained to lie on the surface of the object. The collision response is integrated as a direct correction on the state of the system. This approach is presented by Baraff and Witkin [5]. Later, Volino et al. [52] have used the approach that is based on correction of positions, velocities, and accelerations of colliding particles.

- *Penalty-forces*: In this method, a spring force that keeps particles away from each other is applied.

- *Impulse-based*: This method applies an "instantaneous" change in momentum. The main advantage of this method is it correctly stops all collisions. However, it can have poor numerical performance and it handles persistent contact poorly.

- *Rigid body dynamics*: The basic idea behind this option is if a group of particles start time step collision-free, and move as a rigid body throughout the time step, then they will end time step collision free. For applying this idea to the cloth, we can group particles involved in a collision together and move them as a rigid body. This method is totally failsafe. Until the impact zone includes all colliding particles, we will need to iterate, and merge impact zones. This method is proposed in [42] and it is used as a last resort. Although it is easy to implement, when the particles collide into each other it cannot get dynamic interactions between particles.

Chapter 3

The Garment Design

3.1 The Cloth Model

The cloth model used in our system is a mass-spring model [41], which behaves according to the Newton's second law of motion $F = ma$. This is a specific case of a particle system, in which the particles are connected by spring forces. Particles are objects that have mass, position, and velocity, and respond to forces, but that have no spatial extent. The type and behavior of the cloth is determined by the strength of the spring forces and the topology of the cloth, which is determined by how the springs connect the particles. The mass-spring model is adopted due to its simplicity, efficiency and the capability to simulate the physical behavior of cloth.

The cloth system is simulated by calculating the positions of each particle, which depend on the forces acting on each particle. These forces can be divided into two as the internal and external forces. Internal forces are the spring forces, whereas external forces are the forces such as gravity or wind.

The initial grid structure is a rectangular mesh of particles at the vertices and the springs connecting these particles (Figure 3.1).

3.1.1 Internal Forces

Internal forces are the spring forces between the particles and they determine the mechanical properties of the cloth. These mechanical properties can be categorized into four as follows ([53]):

- *Elasticity*: Characterizes the internal forces resulting from a given geometrical deformation.
- *Viscosity*: The internal forces resulting from a given deformation speed.
- *Plasticity*: Describes how the properties evolve according to the deformation history.
- *Resilience*: Defines the limit at which the structure will break.

3.1.1.1 Springs

A spring, whose behavior depends on the Hooke's law, is defined by the tension between the endpoints of that spring. The forces on the endpoints of the spring (f_i and f_j) depend on the rest length (r_{ij}), the spring coefficient (k_s), the damping constant (k_d), the velocities of the endpoints (v_i and v_j) and the current length of the spring (d_{ij}).

$$\begin{aligned}
f_i &= -[k_s(|d_{ij}| - r_{ij}) + k_d(v_i - v_j)(\frac{d_{ij}}{|d_{ij}|})](\frac{d_{ij}}{|d_{ij}|}) \\
d_{ij} &= pos_i - pos_j \\
f_j &= -f_i
\end{aligned}$$

The spring force magnitude is proportional to the difference between the actual length and the rest length, while the damping force magnitude is proportional to P_i and P_j's speed of approach. Equal and opposite forces act on each particle, along the line that joins them.

Three types of springs are used in order to reproduce the stretching, shearing and bending behavior of cloth.

1. *Structural springs:* These springs connect the vertices adjacent along the row or column of the grid. Namely, they connect the particle [i, j] with particles [i+1, j] and [i, j+1]. The cloth is made up of warp and weft fibers and these springs model this fact. Structural springs serve to keep the cloth in its natural rectangular state.
2. *Shear springs:* They connect the two vertices along the diagonals of each rectangle in the mesh. Namely, they connect the particle [i, j] with particle [i+1, j+1] and the particle [i+1, j] with particle [i, j+1]. The shear springs prevent the cloth from collapsing. They handle the shear stress of cloth.
3. *Bend springs:* These springs connect every other particle along the two directions of the rectangular grid. Namely, they connect the particle [i, j] with particles [i+2, j] and [i, j+2]. These springs prevent the cloth from bending excessively. Keeping these springs stiff restricts the cloth from bending too much out of the original plane of the grid.

Instead of adding shear springs, the realistic behavior of cloth can also be simulated by increasing the resolution of the grid. However, this brings a high computational cost to the system; thus, the realism is achieved by adding shear springs.

3.1.2 External Forces

The external forces that have been implemented in our system are gravity, viscous drag, wind and collision forces. The first three are presented in this section, whereas collision forces are described in Chapter 5.

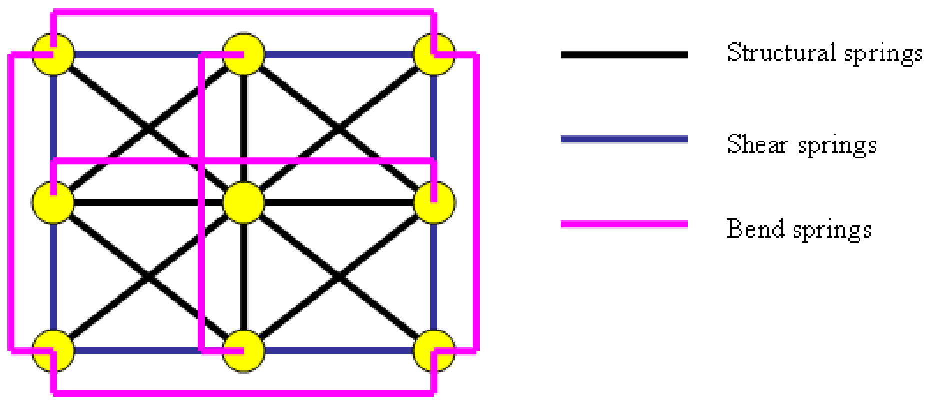


Figure 3.1: Cloth mesh with springs and masses

3.1.2.1 Gravity

The global earth gravity acting on each particle is $f = mg$, where g is a constant vector whose magnitude is the gravitational constant. The gravitational force is applied simply by traversing the system's particles, adding the appropriate force into each particle's total forces.

3.1.2.2 Viscous Drag

The effect of drag is to resist motion, making a particle gradually come to rest in the absence of other influences. In order to enhance numerical stability, at least a small amount of drag should be applied to each particle. Ideal viscous drag has the form $f = k_d v$, where the constant k_d is called the "drag coefficient" [58].

3.1.2.3 Wind

A nonlinear wind formulation in accordance with fluid laws in physics is used in our system.The force acting on a triangle due to wind is always in the direction of the normal vector of that triangle. The force is proportional to the surface area of the triangle, the angle at which the wind hits the triangle, and the speed of the wind.

Each mass point having approximately the same small surface area, we assign a drag force due to wind to a mass point P_i as:

$$F_i^{drag} = k_d \| v_i^{wind} \| (n_i \cdot v_i^{wind}) n_i,$$

where v_i^{wind} is the velocity of the wind, n_i is the surface normal and k_d is a dragging constant. First, the drag forces are calculated for each triangle. Then, the total drag force for each point is calculated as the sum of the drag forces of the triangles surrounding that point [36].

3.1.3 Evolving the Cloth in Time

In order to compute the progression of the system in time, the system must be integrated numerically. The progression is calculated as a sequence of successive positions of the particles making up the cloth, over specific time intervals. The integration methods can be basically classified into two: *explicit* and *implicit* methods [58].

We have implemented both implicit and explicit integrators in our system so that the best one could be selected depending on the circumstances.

3.1.3.1 Explicit Integrators

Euler Integrator: This method is the simplest method for numerical integration. The formula for this method is

$$\begin{aligned} x(t_0 + h) &= x_0 + h x'(t_0) \\ x'(t) &= f(x, t) \end{aligned}$$

In this way, instead of calculating the true integral of the function f in x, we find the approximate solution by taking a step size of h in the direction of the derivative. However, the derivative information is used only at the beginning of the interval.

We can adapt this formula into our system as:

$$x(t_0 + h) \quad = \quad x_0 + hv(t_0)$$

Where, v is the initial velocity of the particle, x_0 is the initial position of the particle and x is the position of the particle at time $t_0 + h$.

By applying a Taylor series expansion, we can see that the error in Euler integrator is $O(h^2)$ [39].

Midpoint Integrator: Euler Integrator may go unstable if large step sizes are taken. From the Taylor series, if we knew the second derivative of x, we could get an error of $O(h^3)$. Thus, we take a trial step to the middle of the interval and then use the value of the x and x' at the middle of the interval to calculate the real step across the interval. We obtain:

$$x(t_0 + h) \quad = \quad x_0 + h(f(x_0 + \frac{h}{2}f(x_0))$$

Runge-Kutta Integrator: The same idea can be further improved for less error rates. We take several Euler steps and in each step, the derivative is evaluated four times: once at the initial point, twice at the midpoints, once at a trial end point. Then, the final function value is calculated as:

$$\begin{aligned}
k_1 &= hf(x_0, t_0)\\
k_2 &= hf(x_0 + \frac{k_1}{2}, t_0 + \frac{h}{2})\\
k_3 &= hf(x_0 + \frac{k_2}{2}, t_0 + \frac{h}{2})\\
k_4 &= hf(x_0 + k_3, t_0 + h)\\
x(t_0 + h) &= x_0 + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4
\end{aligned}$$

In the fourth order Runge-Kutta, the error term is $O(h^5)$.

Adaptive Runge-Kutta: Runge-Kutta integrator can be improved by applying adaptive step size. The purpose is to achieve accuracy in the solution with minimum computational effort [58]. In implementing adaptive integrators, we get two stepsizes, calculate the error and increase or decrease the stepsize depending on the error threshold.

3.1.3.2 Implicit Integrators

Even the higher order explicit methods require small time steps in order to preserve stability. However, since implicit integration methods consider information of the next time-step, large stepsizes can be taken in order to deduce the next state. In [5], Baraff and Witkin proposed to use implicit integration so that the cloth simulation with stiff equations could be possible. The results presented by Baraff and Witkin showed that, implicit integration methods allowed to take large time steps so that the efficiency of the system would not be reduced.

Implicit Euler Integrator: The method in [5] requires the solution of a linear system with $O(n^2)$ computation time, where n is the number of mass points in the cloth. In our study, we have followed the approach presented in [28] in order to implement implicit integration. This method enables the simulation of the system interactively.

The value of $x(t + h)$ is computed as:

$$x(t+h) \quad = \quad x(t) + hv(t+h) \tag{3.1}$$

which is different from explicit integrators since they consider the value of the velocity at time t. However, implicit Euler requires that we know the value of $v(t+h)$. This value is:

$$v(t+h) \quad = \quad v(t) + hM^{-1}f(t+h) \tag{3.2}$$

where,

$$f(t+h) \quad = \quad f(t) + \frac{\partial f}{\partial x}\Delta x(t+h) = f(t) + Jx(t+h) \tag{3.3}$$

So, from (3.1) and (3.3) we get:

$$x(t+h) \quad = \quad f(t) + Jx(t+h) = f(t) + J(v(t) + v(t+h))h \tag{3.4}$$

where,

$$\begin{aligned} \Delta v(t+h) &= hM^{-1}f(t+h) && (3.5) \\ &= hM^{-1}f(t) + h^2M^{-1}Jv(t) + h^2M^{-1}J\Delta v(t+h) && (3.6) \end{aligned}$$

By rearranging Equation (3.6), we finally obtain:

$$(M - h^2J)v(t+h) \quad = \quad hf(t) + h^2Jv(t) \tag{3.7}$$

The solution of (3.7) returns us the value of v at time $t+h$.

The algorithm exploits the fact that the matrix $(M - h^2J)$ is sparse and symmetric. In this way, applying iterative methods solves the problem in real time.

3.2 The Garment Design Process

The garment design process consists of the following phases: First, cloth meshes, each a rectangular grid of vertices, are created. Then, the garment panel is given its desired shape via cutting or selecting individual vertices and moving them or automatically smoothing the boundary. After creating the flat panels, seaming points can be defined on them. This is done by selecting the vertices to attach and adding seams between them. In addition, the obtained garment panels can be scaled to the desired size.

3.2.1 Creating Garment Panels

A garment panel is a rectangular grid showing the positions of particles and springs. While creating these panels, the user can specify the number of particles in the mesh, spring constants for bend, shear and structural springs (which determine the type of the fabric material) and the size of the cloth mesh.

3.2.2 Cutting

There are two options for cutting fabric in order to design garments: either to draw the 2D shape of the garment and then discretize it, or to select the cloth boundary on an already discretized cloth mesh. The latter is more similar to the cutting process in real life and it also preserves the regular structure of cloth. In our system, we prefer the second option since we need regularity for rendering knitted and woven fabric. We select the particles that make up the cloth boundary and then cut it out of the rectangular mesh (Figure 3.2).

When applying cutting, we perform the 8-connected, recursive boundary fill algorithm on the particles making up the cloth mesh, thus by only selecting the boundary of the cloth it is possible to extract the desired pattern. There are also other options such as extracting out pieces and creating holes on the cloth piece as in Figure 3.3.

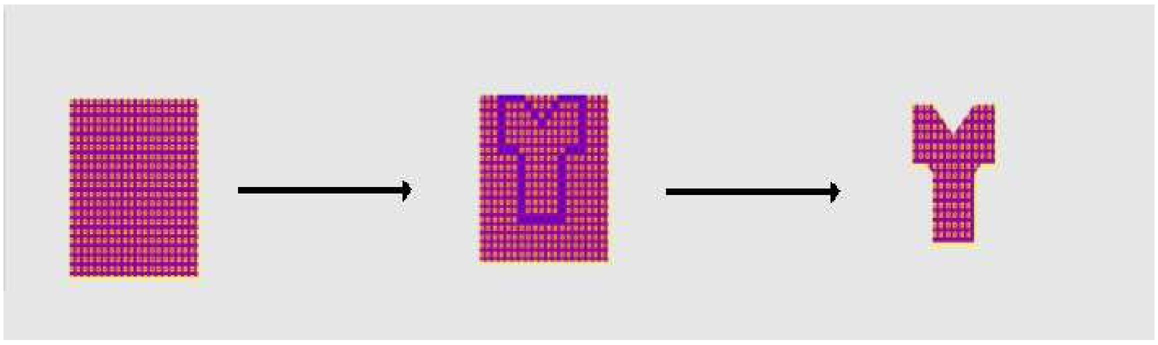


Figure 3.2: The 2D garment panel cutting process

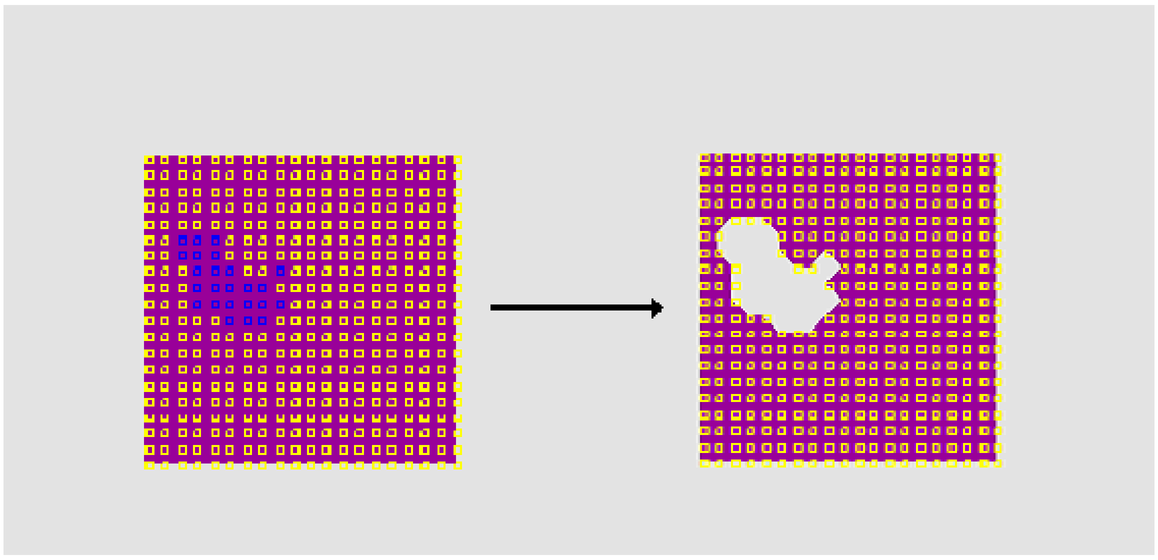


Figure 3.3: Cutting holes in 2D garment panels

3.2.3 Smoothing

After the cutting, we can smooth the boundary of the cloth panels with the intention of preventing the jagged borders. In order to apply smoothing, we select the particles on the boundary that require smoothing as control points, and then calculate the Bézier spline for these control points and move the particles onto the resulting curve. Bézier curves are preferred over other spline approximation methods since they are easy to implement and powerful for designing curves [6]. Figure 3.4 shows the smoothing process on a skirt and shirt.

The Bézier spline approximation is performed on a set of $n+1$ control point positions: $p_k = (x_k, y_k, z_k)$, where $k \in [0, 1, 2, ..n]$. These control points can be blended to vector $P(u)$, to describe the path of an approximating Bézier polynomial function between p_0 and p_n.

$$P(u) = \sum_{k=0}^{n} p_k BEZ_{k,n}(u), \qquad 0 \le u \le 1$$

where $BEZ_{k,n}(u)$ are the "Bernstein polynomials":

$$BEZ_{k,n}(u) = C(n,k)u^k(1-u)^{n-k}$$

and C(n,k) are the binomial coefficients:

$$C(n,k) = \frac{n!}{k!(n-k)!}$$

3.2.4 Seaming

Seaming is performed by defining the seam points between the particles of cloth panels. The process can be seen in Figure 3.5.

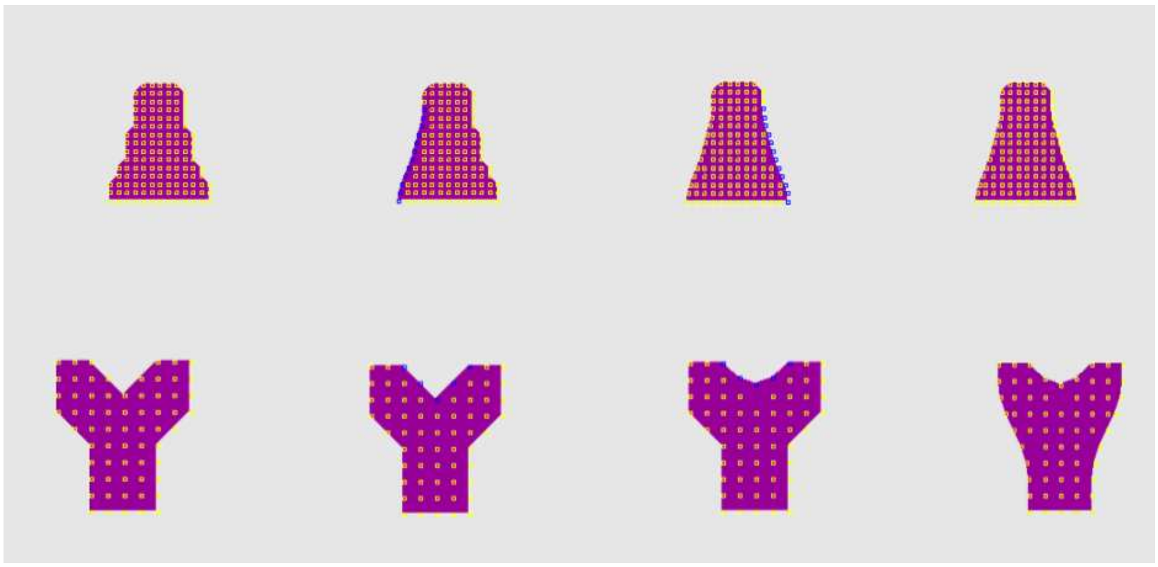


Figure 3.4: Smoothing the boundary of 2D garment panels

Another option is to perform "batch seaming", that is, instead of defining the seam points one by one, selecting the line of seaming on one cloth piece and then bringing the two cloth pieces close enough and seaming the points all at once. This method works by finding the closest pair of particles between the two clothes and adding seams between them. Figure 3.6 shows this process.

3.2.5 Resizing

In order to fit the garment on the mannequin, it should be resized taking the sizes of the mannequin into consideration. This process is performed through scaling the panel as a whole or selecting certain particles and translating them individually.

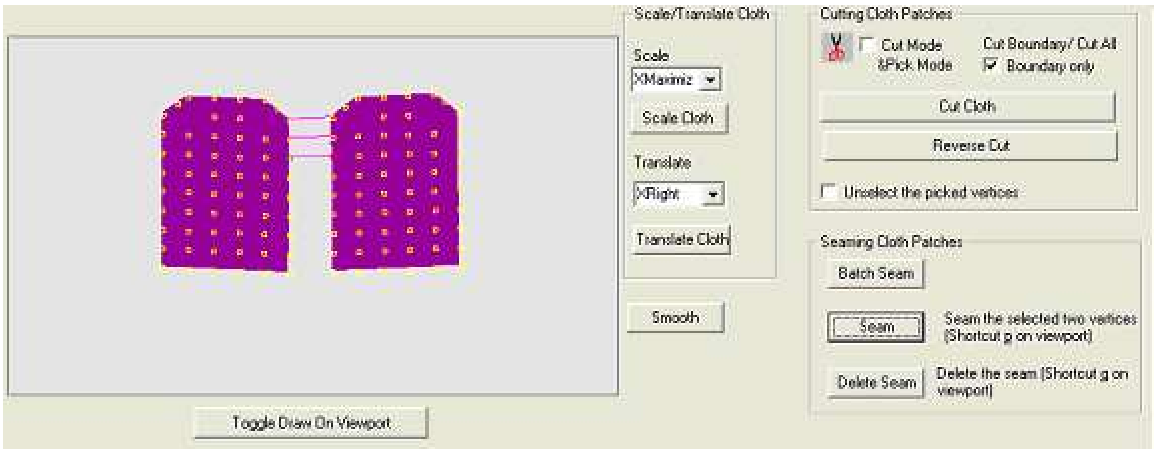


Figure 3.5: Seaming 2D garment panels

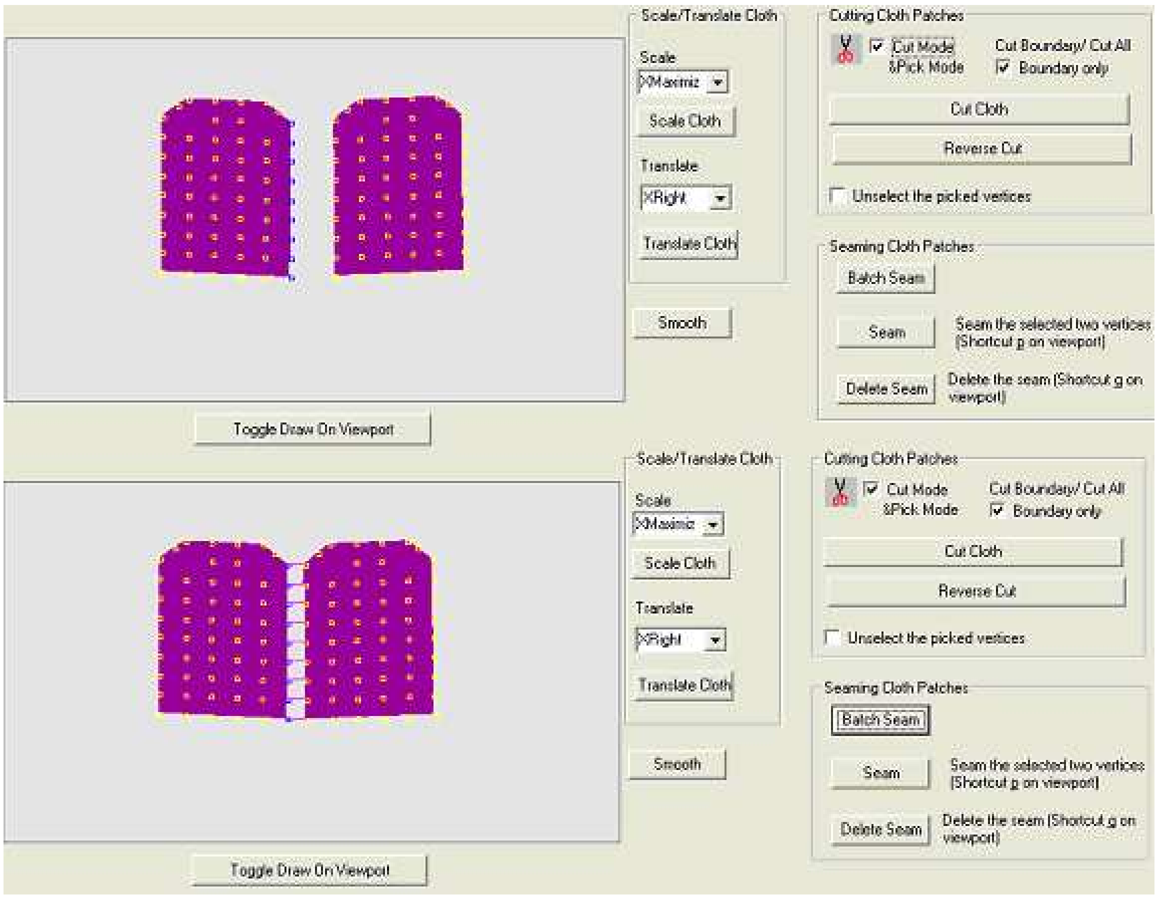


Figure 3.6: Batch seaming 2D garment panels

Chapter 4

The Garment Simulation

The garment panels are transformed from 2D to 3D in order to simulate the three dimensional behavior of garments. Then, the panels are placed around the virtual human and sewn together. The user can modify the rendering options and determine how the garment is going to look like at the end so that he/she can see the final appearance of the garment.

This chapter examines the garment simulation process implemented in the scope of this thesis study.

4.1 Garment Placement

Garment panels are placed around the body by keyboard interaction. It is possible to translate, scale or rotate each garment panel. In addition, local parts of a garment panel can be translated individually. This is achieved by changing the position of the selected particles. In this way, the garment panel can obtain the desired shape.

4.2 Sewing

After garment panels are in their accurate positions around the body, sewing is invoked by applying forces between the seams in garment parts. Seams can be regarded as forces that attract two particles to each other (Figure 4.1), in that sense, they can be regarded as elastic forces. However, simulating the exact behavior of elastic forces is expensive [53] and a much simpler heuristic can solve the problem more efficiently. The heuristic approach is applying symmetrical forces on the two particles so that they pull each other as in the following equations:

$$
\begin{aligned}
p1_{vel} &= c_{attraction} \frac{|p1_{pos} - p2_{pos}|}{\|p1_{pos} - p2_{pos}\|} \\
p2_{vel} &= -p1_{vel}
\end{aligned}
$$

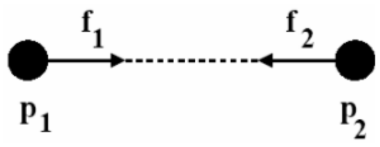


Figure 4.1: Forces acting on particles during the sewing process

The two particles attract each other until they are constrained by collision forces. During the sewing process, no other forces such as gravity are applied on the clothes. After two particles p_1 and p_2 are closer than a threshold, the sewing process is finalized and these particles are combined into one. This is performed by adding spring forces between p_2 and neighbors of p_1 and between p_2 and neighbors of p_1 (Figure 4.2). Neighbor of a particle p means the particle q such that there exists a spring between p and q. In Figure 4.8, the procedure for placing parts of a shirt and sewing them can be seen. By this approach, the garment can be a complex assembly of different textile materials.

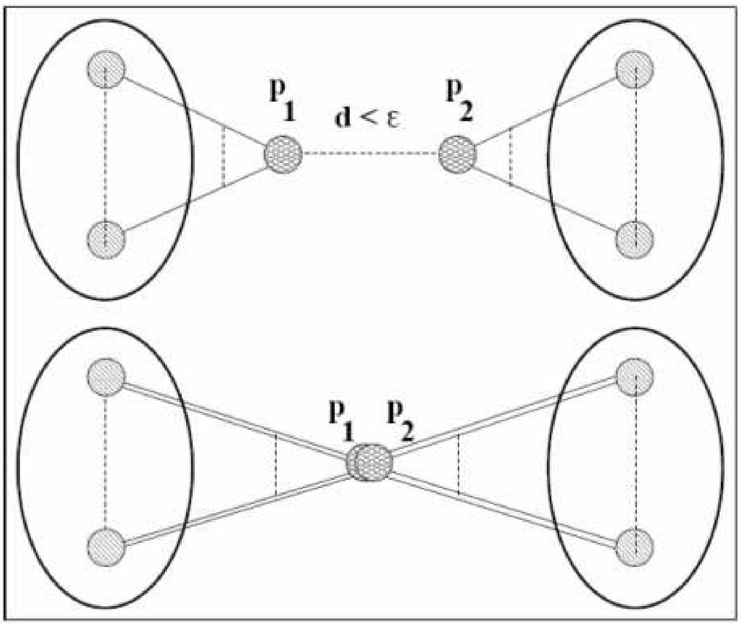


Figure 4.2: Combining two particles into one after sewing

4.3 Attachment Constraints

In order to keep the garment on the virtual model without losing efficiency, some parts of the clothes can be attached to the human body. This approach is followed depending on the type of the garment. For instance, tight clothes can be bound to the human body with attachment constraints. For this purpose, after the virtual human is dressed with the garment, the selected particles are attached to the closest polygon on the virtual human. In this way, those parts of the garment move with the human.

4.4 Rendering Garments

Realistic rendering of clothes is as important as the simulation of their draping since important information about the material the fabric is made of can be obtained via its visual appearance. Besides general rendering techniques such as

Gouraud shading, some shading techniques specifically related to textiles exist. There are various methods to produce garments from yarn, such as knitting, weaving, braiding or knotting. The most important ones among these are knitting and weaving. Thus, we have simulated these two methods in our system. Moreover, standard methods and material-specific BRDFs are also implemented. This section explains the techniques that have been implemented for rendering garments.

4.4.1 Smoothing

Before explaining the methods for weaving and knitting, we should make sure that the surface is smooth enough. In order to have a smooth surface, we rediscretize the surface by taking into account surface geometry. Thus, flat mesh triangles are converted into curved surfaces, which give the cloth a smoother appearance. This method is adopted from [53]. The procedure is very simple since it considers only vector positions and vector normals for each triangle. The discretization scheme is as follows:

Given P_a, P_b, P_c vertex positions and N_a, N_b, N_c vertex normals, let $P = (r_aP_a, r_bP_b, r_cP_c)$ be an intermediate point on the triangle where r_a, r_b and r_c are barycentric coordinates and $N = (r_aN_a, r_bN_b, r_cN_c)$ is the normal of point P. Our aim is to calculate an interpolated point Q. For this purpose, we calculate the contributing points Q_a, Q_b and Q_c that make up Q (Figures 4.3 and 4.4).

$$\begin{aligned} K_a &= P + ((P_a - P) \cdot N)N \\ Q_a &= K_a + \frac{(P_a - K_a) \cdot N_a}{2 + \mu((N \cdot N_a) - 1)} \end{aligned}$$

Then, we blend the three surfaces as:

$$Q = \frac{f(r_a)Q_a + f(r_b)Q_b + f(r_c)Q_c}{f(r_a) + f(r_b) + f(r_c)}$$

where, $f(x) = x^2$

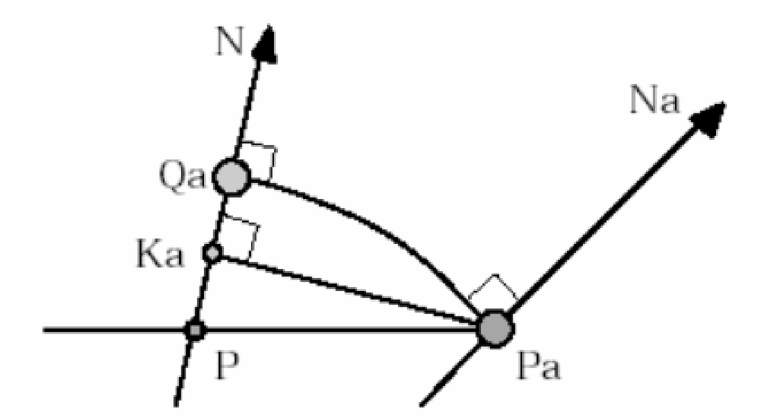


Figure 4.3: Contribution of the vertex Pa to the smoothed vertex P during the smoothing process (reprinted from [53])

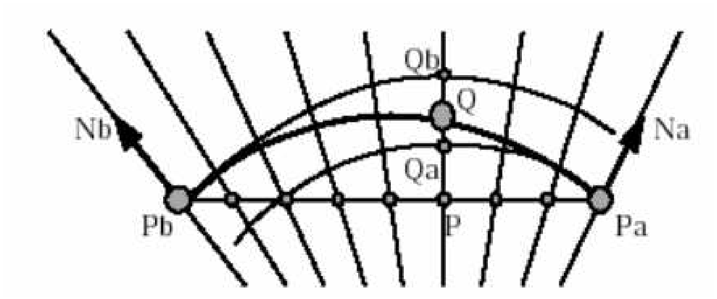


Figure 4.4: Interpolation between vertex contributions during the smoothing process (reprinted from [53])

4.4.2 Knitwear

The structure of knitwear is complicated compared to other techniques like weaving. This is due to the three dimensional geometry of a knit loop. In our system, we make use of the particle system and the mass-spring model of our cloth mesh in order to consider the interaction of neighboring loops. For this purpose, the cloth mesh must consist of quadrilaterals and must be regular. There are two types of basic stitches when knitting: left and right loops. The knitwear pattern, which shows the order of the right and left loops is read from an input file and can be changed interactively in the program.

Each quadrilateral contains one type of loop. The structure of the loop in a quadrilateral is defined by the bonding points (BPs). The position of the bonding points can be determined parametrically by the vertices of the enclosing quadrilateral. The equations obtained from the parametrization of the surface are as follows:

$$\begin{aligned}
BP_1 &= \frac{6}{7}p_{3_{pos}} + \frac{1}{7}p_{4_{pos}} \\
BP_2 &= \frac{5.5}{21}p_{1_{pos}} + \frac{1.5}{21}p_{2_{pos}} + \frac{11}{21}p_{3_{pos}} + \frac{3}{21}p_{4_{pos}} \\
BP_3 &= \frac{1.5}{21}p_{1_{pos}} + \frac{5.5}{21}p_{2_{pos}} + \frac{3}{21}p_{3_{pos}} + \frac{11}{21}p_{4_{pos}} \\
BP_4 &= \frac{1}{7}p_{3_{pos}} + \frac{6}{7}p_{4_{pos}} \\
BP_5 &= p_{3_{pos}} \\
BP_6 &= \frac{5}{7}p_{3_{pos}} + \frac{2}{7}p_{4_{pos}} \\
BP_7 &= \frac{4.5}{28}p_{1_{pos}} + \frac{2.5}{28}p_{2_{pos}} + \frac{13.5}{28}p_{3_{pos}} + \frac{7.5}{28}p_{4_{pos}} \\
BP_8 &= \frac{6}{7}p_{1_{pos}} + \frac{1}{7}p_{2_{pos}} \\
BP_9 &= p_{4_{pos}} \\
BP_{10} &= \frac{2}{7}p_{3_{pos}} + \frac{5}{7}p_{4_{pos}} \\
BP_{11} &= \frac{2.5}{28}p_{1_{pos}} + \frac{4.5}{28}p_{2_{pos}} + \frac{7.5}{28}p_{3_{pos}} + \frac{13.5}{28}p_{4_{pos}} \\
BP_{12} &= \frac{1}{7}p_{1_{pos}} + \frac{6}{7}p_{2_{pos}}
\end{aligned}$$

where, Bp_i is the i'th bonding point and $p_{j_{pos}}$ are the three dimensional positions of vertices.

Due to the thickness of the yarn, these bonding points are moved slightly taking the normal of that quadrilateral into consideration. Then, each bonding point Bp_i is assigned the value $Bp_i + N$, where N is the surface normal. Thus, the knitted fabric looks different when front and back views are considered. Figure 4.5 shows the construction of the loops.

In order to maintain interactivity, complex volumetric models or time consuming methods like the ones in [59, 34] for rendering the yarns in a loop cannot be utilized. 3D effect for each yarn is achieved by drawing cylinders around each yarn and applying texture mapping on these cylinders. However, since this operation also slows down the system, except for close-up views, we draw line primitives of different sizes and colors so as to give a three dimensional look, similar to Meissner et al. [33]. Figure 4.9 shows the close-up view of a knitwear.

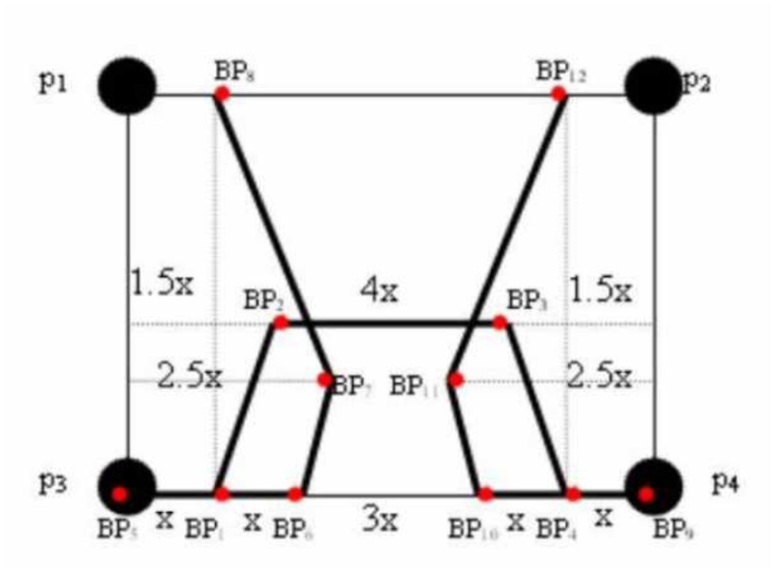


Figure 4.5: Bonding points of a knit loop

4.4.3 Weaving

The woven cloth effect cannot be captured realistically by texture mapping; a cloth simulation program should enable the user to create complex patterns and determine the material of the fabric interactively. At this point, procedural simulation techniques of the visualization of cloth can be adopted. For representing the behavior of woven fabric, the interweaving of threads and the interaction of the light with threads should be carefully integrated. Weaving should be examined in two scales: milliscale and microscale. Milliscale examination refers to the order of the interleaving of threads, whereas microscale examination deals with the fibers constituting the threads [57, 1].

4.4.3.1 Representation of Interwoven Threads

In actual life, weaving can be performed by means of a loom. The idea of the loom is to interleave two sets of perpendicular threads [21, 22, 23]. These threads are the warp and the weft threads as can be seen in Figure 4.6. Warp threads are the vertical ones, whereas the weft threads are the horizontal ones. Weaving patterns can be obtained from two types of interleaving of threads, i.e. warp on weft and weft on warp.

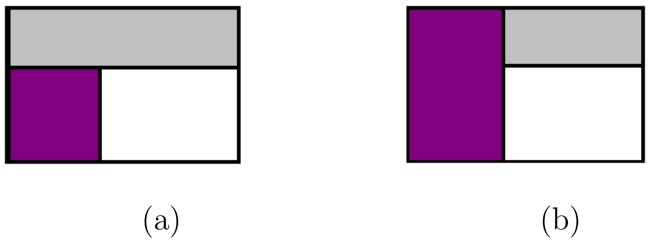


Figure 4.6: Obtaining weave patterns by interleaving threads. (a) weft on warp; (b) warp on weft.

Various weaving patterns can be created by ordering these two thread interleaving types. Our system works as follows: we read the patterns from a pattern description file, and then draw them in three dimensions as warp and weft threads by calculating their positions and finally do texture-mapping on them in order to capture a detailed appearance with the comprising fibers. Texture coordinates are calculated as soon as the weave pattern is read from the file.

We render the threads one by one at each iteration. In this way, our method for rendering woven cloth is similar to the rendering of knitwear since the three-dimensional structure is not ignored. The three dimensional effect can be given by calculating the indices of each of the fibers. Weaving is thus simulated physically. Since the weaving structure is defined procedurally, there is no need to use alpha values for the gaps between the warp and weft threads. Although this reduces the efficiency, the results are more realistic.

When the weave pattern is read from the file, each quadrilateral in the cloth mesh is subdivided into smaller quadrilaterals in order to construct the pattern. The input file consists of a matrix of binary numbers, where 0 denotes weft on warp and 1 denotes warp on weft. For instance, the plain weave matrix M can be identified as:

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

There are three basic patterns of weaving: plain weave, twill weave and satin weave [46]. The thread is woven over and under single threads in a plain weave. In a twill weave, the thread is woven under one thread and over two. Also, the weft courses are organized in a diagonal manner, like in denim. Finally, in a satin weave, the thread is woven under one thread and over more than two threads. The satin weave is a variation on the twill weave, without the diagonal appearance. Weaves other than these three are the variations over them. For instance, the basket weave is a variation over plain weave, in which two threads are used instead of one. These types can be seen in Figure 4.10.

4.4.3.2 Light Interaction with Threads

The shading methods explained in the previous section do not consider an illumination model. However, the material type for the fabric and the anisotropic behavior of cloth can be revealed by an explicit sophisticated illumination model. Anisotropic BRDF (Bidirectional Reflectance Distribution Function) is used to model the scattering of light from individual fibers in the threads in our system. Milliscale illumination is obtained from microscale BRDF that is calculated for each of the warp and weft threads.

Bidirectional Reflectance Distribution Function (BRDF)

When elaborating the surface illumination model, we should consider a reflectance function of the angles of the incident and reflected angles only. The BRDF is defined on the cross product of two hemispheres and intuitively it represents, for each incoming angle, the amount of light that is scattered in each outgoing angle. BRDFs are four dimensional functions, which describe the reflection distribution at a surface point depending on incoming and outgoing light directions. There are various strategies for modeling BRDF [4]. These are:

- *Direct measurement*: This is through gonioreflectometers, which are mechanical devices that vary the direction to a small light source and a spectral sensor and collect a large number of point samples for BRDF.
- *Empirical methods*: Some of these empirical methods are Gouraud and Phong shading.
- *Height correlation methods*: This is the most complete surface representation among the other methods. This can be realized by applying a correlation function directly related to the surface height correlations.
- *Microfacet methods*: These methods assume that the surface consists of a large number of small and flat microfacets each of which reflect light in the specular direction.These methods are somewhere in between the height correlation methods and empirical methods. Although height correlation

methods are more accurate, microfacet methods are comparatively more intuitive and simplistic.

In our system, we have implemented both Schlick's BRDF Model [45] and Ashikmin's BRDF model [4], which are both consistent with the microfacet models.

Schlick's BRDF Model

One of the models that we have adopted in our system is the Schlick's BRDF model [45]. This reflectance model is applied for realistic illumination of the cloth and it is preferred because of the following features:

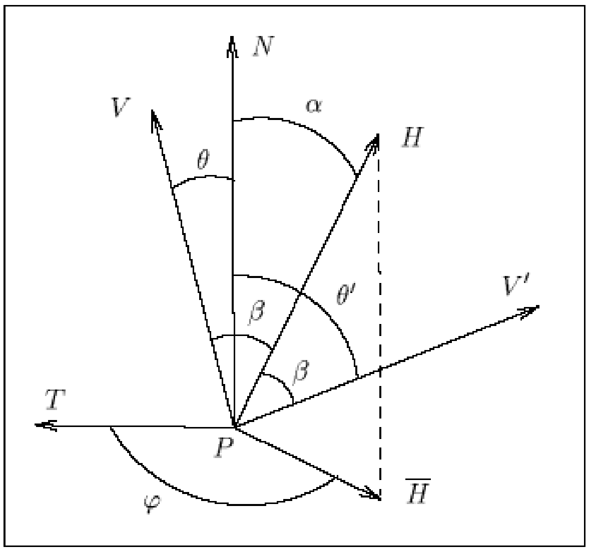
1. It is defined by a small number of parameters.
2. It is expressed by a formulation by varying complexity that can be customized according to the number of physical phenomena the user wants to include (isotropic or anisotropic reflection, homogeneous or heterogeneous materials, spectral modifications, surface self-shadowing).
3. It is fast and simple though it obeys the main lows of physics (Energy conservation law, Helmholtz reciprocity rule, Microfacet theory, Fresnel equation).

The BRDF can be separated into the spectral factor S_λ and the directional factor D as:

$$R_\lambda(t,u,v,v',w) \quad = \quad S_\lambda(u)D(t,v,v',w)$$

Spectral factor should obey to the Fresnel law and it can be approximated by the following equation as explained in [45]:

$$S_\lambda(u) \quad = \quad C_\lambda + (1-C_\lambda)(1-u)^5$$



V	$=$	Outcoming direction of light
V'	$=$	Incoming direction of light
N	$=$	Surface normal vector
T	$=$	Surface tangent vector
H	$=$	Bisector vector of V and V'
$\overline{H}$	$=$	Projection of $H \perp N$

α	$=$	$< H, N >$	t	$=$	$\cos\alpha$
β	$=$	$< V, H >$	u	$=$	$\cos\beta$
θ	$=$	$< V, N >$	v	$=$	$\cos\theta$
θ'	$=$	$< V', N >$	v'	$=$	$\cos\theta'$
φ	$=$	$< \overline{H}, N >$	w	$=$	$\cos\varphi$

Figure 4.7: Angles and vectors for the definition of BRDF (reprinted from [45])

Directional factor is formulated as:

$$D(t, v, v', w) \quad = \quad \frac{1}{4\Pi v v'} Z(t) A(w),$$

where

$$\begin{aligned} Z(t) &= \frac{1}{(1 + rt^2 - t^2)^2} \\ A(w) &= \sqrt{\frac{p}{p^2 - p^2 w^2 + w^2}} \end{aligned}$$

In $Z(t)$ and $A(w)$, r is the roughness factor and p is the isotropy factor.The surface tangent T is taken as direction of each thread. Thus, the surface tangent is calculated along the warps and wefts separately. The definitions of angles and the vectors for BRDF are given in Figure 4.7.

Ashikmin's BRDF Model

In order to simulate different fabric types, Ashikmin's BRDF Model [4] is utilized. According to [4], microfacet models assume that:

- The microfacet normals have an underlying probability density function $p(h)$.

- A microfacet contributes to BRDF for a given pair of directions if and only if it is visible (not shadowed) relative to the lighting direction and the viewing direction.

As a result of these assumptions, the BRDF for the surface is defined as:

$$\Psi(w_{in}, w_{out}) = \frac{p(h)\langle h \cdot n\rangle S_\lambda(h \cdot w_{in})}{4g(w_{in})g(w_{out})},$$

where w_{in} is the incoming light direction, w_{out} is the outgoing light direction, h is the halfway vector, n is the surface normal, S_λ is the fresnel function, and g's are the shadowing terms. In general, $p(h)$ is formulated as:

$$p(h) = c * exp(-tan^2\theta(cos^2\phi/\sigma_x^2 + sin^2\phi/\sigma_y^2)),$$

where σ_x and σ_y are the Gaussian distributions of threads in x and y directions, θ is the angle between the half vector h and the surface normal, ϕ is the azimuth angle of h and c is a normalization constant. The calculation of $p(h)$ determines the material type. For instance, satin and velvet are formulated as follows:

Satin: Ashikmin et al. [4] explain satin as fibers running in one direction with about 70% of the fiber length lying in the relatively flat part of the fiber while the other 30% to the bent parts at the ends. So, the distribution of microfacets is modelled as a linear combination of these two terms of the cylindrical fiber:

$$p(h) = 0.7 * p_{warp}(h) + 0.3 * p_{weft}(h)$$

Velvet: A simple intuitive form of p(h) function written as an "inverse Gaussian" heightfield is enough to capture the main character of the distribution of microfacets in velvet [4]:

$$p(h) = c * exp(-cot^2\theta/\sigma^2).$$

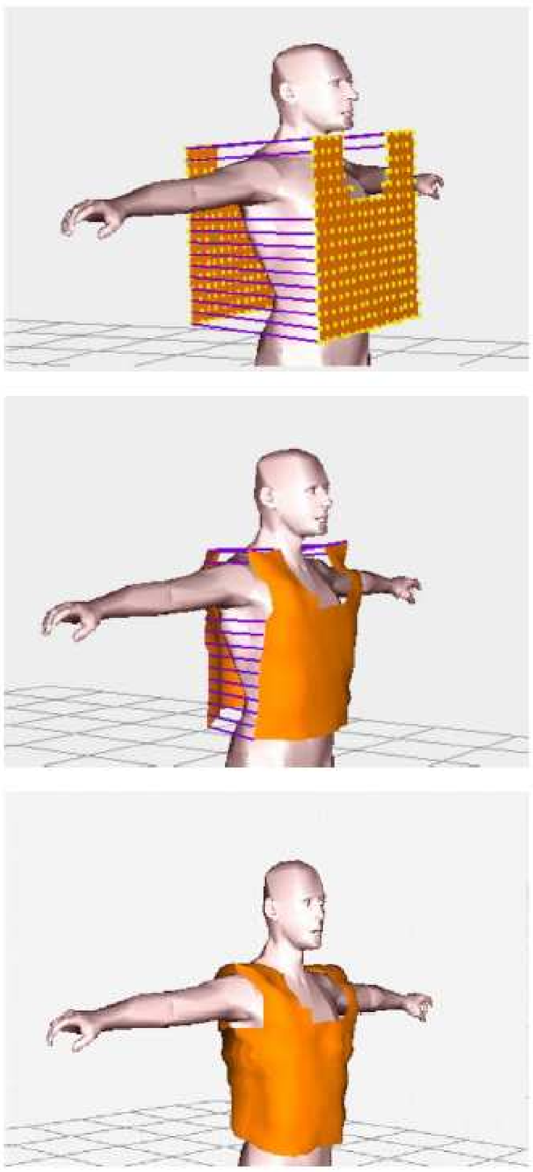


Figure 4.8: The sewing process

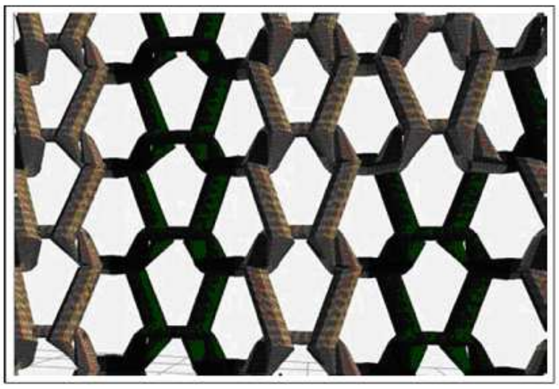
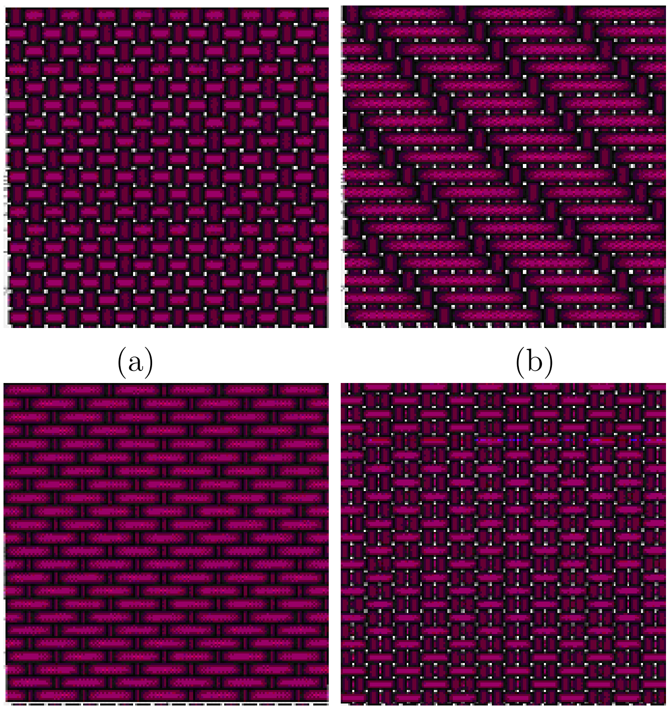


Figure 4.9: Close-up view of knitwear



(a) (b)

(c) (d)

Figure 4.10: Weave pattern examples. (a) plain weave; (b) twill weave; (c) satin weave; (d) basket weave

.

Chapter 5

Collision Handling

Garments interact with the body that wears them or with other garment panels. The shape and the movement of the body affect the garments movement. Thus, in order to obtain realistic simulation of garment panels, collision handling must be achieved in an accurate and efficient way. This chapter overviews the methods that have been adopted for handling collisions. The details of the algorithms and their comparisons can be found in [29].

5.1 Collision Detection Between Garment and Human Body

We need to perform collision detection between garments and skin layer of the human body in order to move the garment in accordance with the human. In general, the intersection test between every particle of garment and every vertex of skin layer can be implemented. However, because of the time constraints, this approach is impractical in the real time applications. Thus, algorithms that decrease the number of collision tests must be applied for getting efficient and accurate results.

In the literature, there exist many approaches that are proposed for decreasing the number of collision tests. One of them is the bounding volume hierarchy, which provides a fast way to perform exact collision detection between complex models. In this study, we use bounding volumes for both human and cloth models. We construct axis-aligned bounding box (AABB) hierarchy for the cloth model, and for comparing their efficiency we utilize oriented bounding box (OBB), AABB and bounding sphere for each part of the human model.

The collision detection algorithm in this study firstly tests all the intersections between the bounding volumes of the human model against the AABB hierarchy of the cloth model. The AABB hierarchy is traversed until the leaf nodes are reached. If an intersection between the two bounding volumes is found (bounding volume of human model and bounding volume in the leaf node of cloth model's hierarchy), then geometrical collision detection methods are applied for testing collisions between triangles.

After detecting bounding volume interferences, the geometrical collision detection tests are applied between the triangles in the intersecting bounding volumes. Collision of two triangles may be in vertex-triangle collision or edge-edge collision. Liu et al. propose a method that consists of constructing a very thin bounding volume around the rigid object for avoiding edge-edge collision detection [32]. The advantage of this method is that it constrains the colliding point over the surface of the bounding volume. We compute the bounding volume by enlarging the surface in the normal directions at each iteration.

The vertex-triangle collision test is done between the vertices of the triangles in the bounding volume of the leaf node (in cloth's hierarchy) and the triangles in the human body part's bounding volume. To allow large time steps and preserve stability, our approach for collision detection and response aims at avoiding collisions before they occur. Therefore, not only interferences are detected but also proximities. The vertex-triangle pairs that are found as colliding or close are stored in a data structure and they are used in the collision response.

The interference detection is achieved by finding whether the particle is in the positive side (above), negative side (below) or on the plane of the triangle. The position of particle with respect to the triangle is found by using the equation of the triangle plane, which divides the space in three subspaces: (1) positive half-space, (2) negative half-space, (3) space that contains all the points lying on the plane. If the particle is in the positive side at time t and in the negative side at time $t + \Delta t$, it can be concluded that interference has occurred during the time step Δt. Thus, the particle-triangle pair must be stored as a collision pair.

The proximity detection is applied if the particle is in the positive side of the collision plane in both t and $t + \Delta t$. The algorithm first computes the distance between the point $P_0 = (x_0, y_0, z_0)$ and the triangle plane $\Pi : ax + by + cz + d = 0$. If the distance is less than a threshold, then the projection of point on the triangle is found and it is determined whether the projected point is inside the triangle. If so, we conclude that the vertex is inside the triangle and we store the particle-triangle pair as a candidate collision.

In order to check the proximity between two edges, S_1 and S_2, we find the pair of points, one on each edge, that are closest to each other and check the distance between them. We use the algorithm described in [47].

5.2 Self-Collision Detection

In our study, we use the AABB hierarchy for the cloth model in order to improve the self-collision detection test. We traverse the hierarchy and we test the intersection of two polygons only if their bounding volumes intersect.

Collision detection between two polygons is done by checking the point-triangle intersection and proximity, similar to the cloth-human body collision detection. In order to avoid edge-edge collision detection, we enlarge the bounding volumes of the cloth model. In addition, adjacent polygons are tested for self-collision detection by using the polygonal adjacency information.

5.3 Collision Response

Collision response must be handled accurately for performing realistic garment simulation. After the collisions have been detected between garments and human body, their effects have to be considered in the mechanical simulation to avoid intersections and to reproduce contact reaction and friction accurately.

In the system, we use penalty forces and constraints for preventing collisions. The type of collision response is chosen depending on the distance between the particle and the triangle's plane, and the particle's position with respect to the triangle. Two distance thresholds $T_{outside}$ and $T_{surface}$ are used for determining the response type. There are three collision types in the system, one for interference and two for proximities. The classification of collisions is similar to the one in [38]. However, the velocity correction and position correction parts are modified for the dynamic case. We employ different collision response approaches for each collision type.

Case 0 (No intersection): If the distance between the point and the triangle is greater than $T_{outside}$, then no response is applied.

Case 1 (Point-Triangle Proximity): If the distance is between $T_{outside}$ and $T_{surface}$, a penalty force for decreasing the probability of collision in the following frames is applied The penalty force is added to the total force applied to the vertex due to gravity, wind, etc.. It avoids intersection with the human body by modifying the direction of the movement for the following frames.

Case 2 (Point-Triangle Intersection): For the second case, where the distance is less than $T_{surface}$, both penalty forces and velocity correction are applied for the colliding particle on the cloth to make the collision smoother. The velocity correction scheme is similar to the method in [49]. However, we have to do some adjustments for the computation of the object velocity and response direction.

Case 3 (Point-Triangle Interference): In this case, the point is in the back of the triangle. This means an intersection has occurred. So, we apply position correction and velocity modification.

Chapter 6

Results

In this chapter, first we present various garment examples with different rendering options. Then we present performance analysis results.

6.1 Visual Results

Some visual results obtained using our system are presented in this section. Figure 6.1 shows a woven cloth emphasizing the difference between the front and back sides. Figure 6.2 shows the deformation of the loops of a knitted cloth. Figures 6.3 and 6.4 successively illustrate dresses made of velvet and satin. Figure 6.5 demonstrates three satin skirts with different weave types and colors. Figure 6.6 shows the close-up view of a woven shirt and trousers with complex weaving patterns. Figure 6.7 shows a knitted shirt with woven trousers. Figure 6.8 demonstrates a multilayered garment with a Gouraud-shaded shirt, knitwear jacket and velvet trousers. Finally, close-up view of a knitted shirt with a complex pattern is illustrated in Figure 6.8.

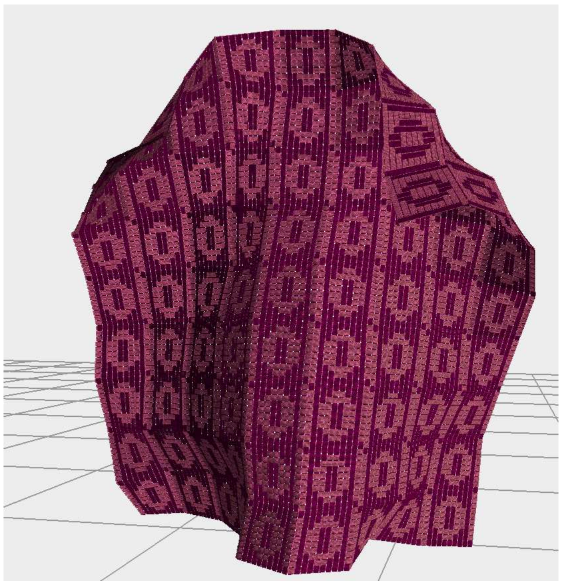


Figure 6.1: Woven cloth

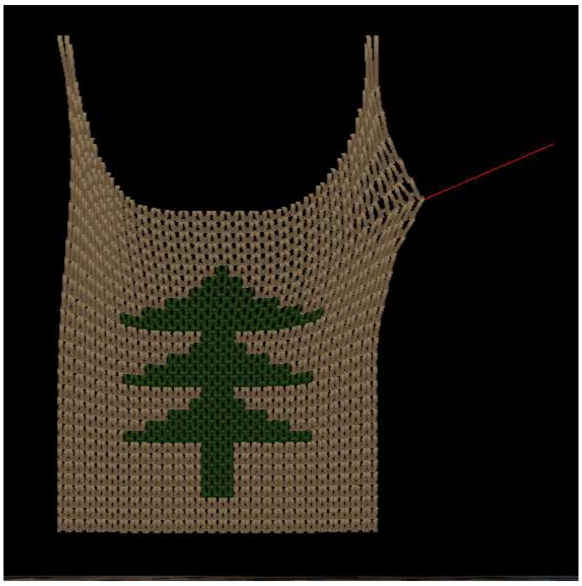


Figure 6.2: Knitted cloth

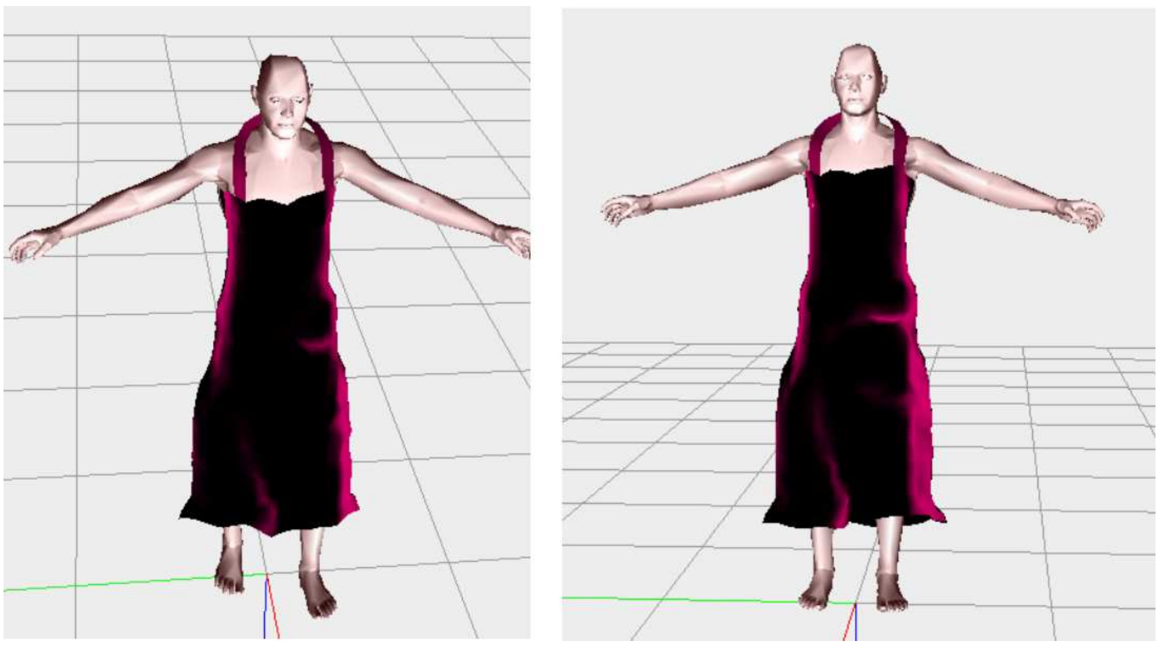


Figure 6.3: Velvet dress

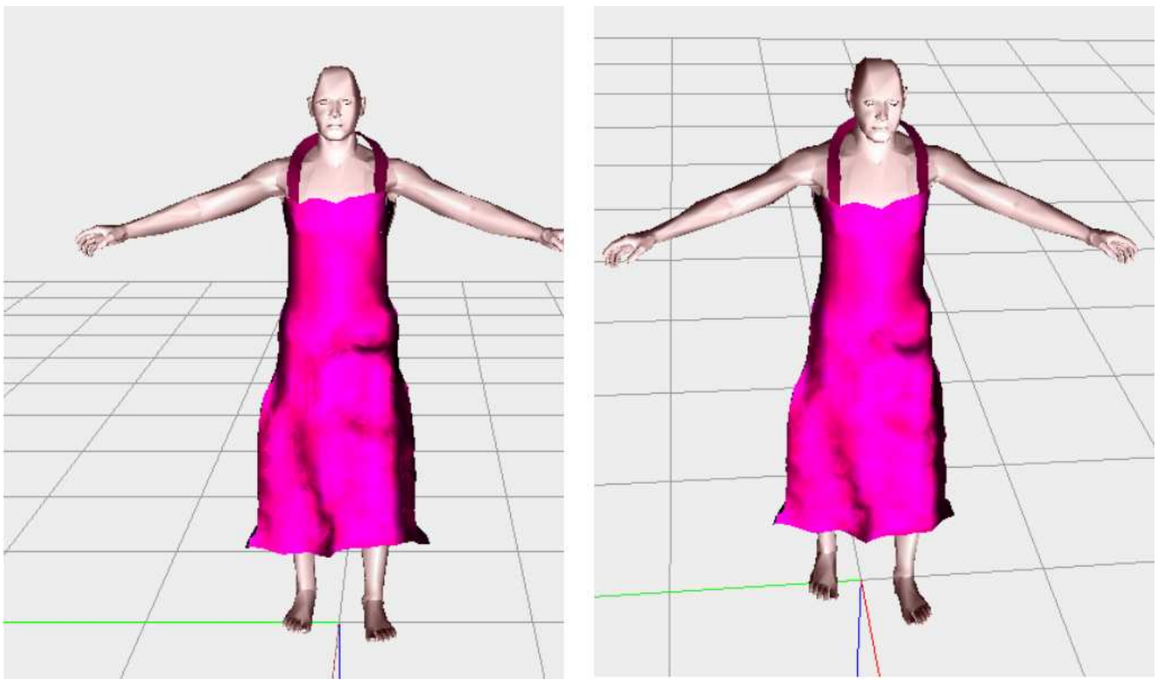


Figure 6.4: Satin dress

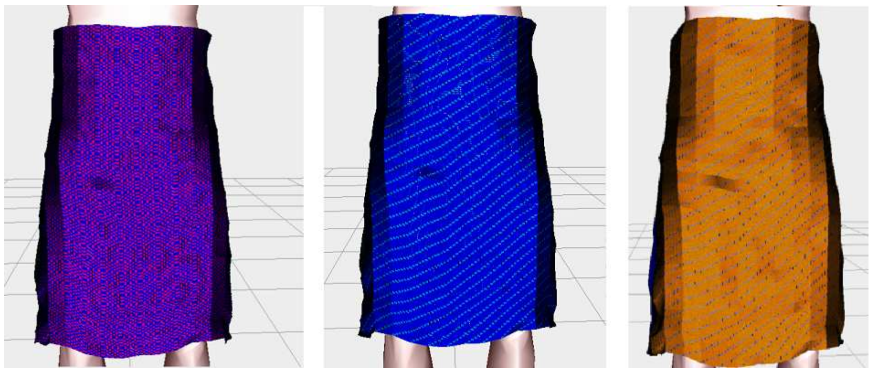


Figure 6.5: Woven skirts with satin BRDF

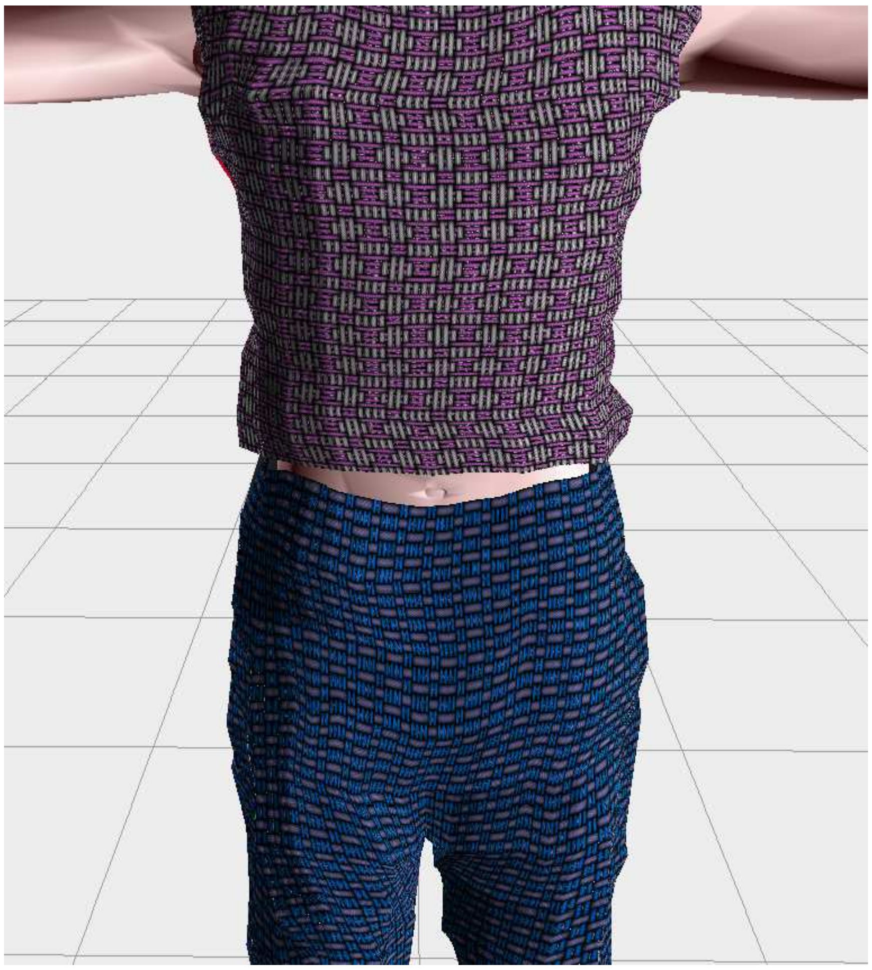


Figure 6.6: Woven shirt and trousers

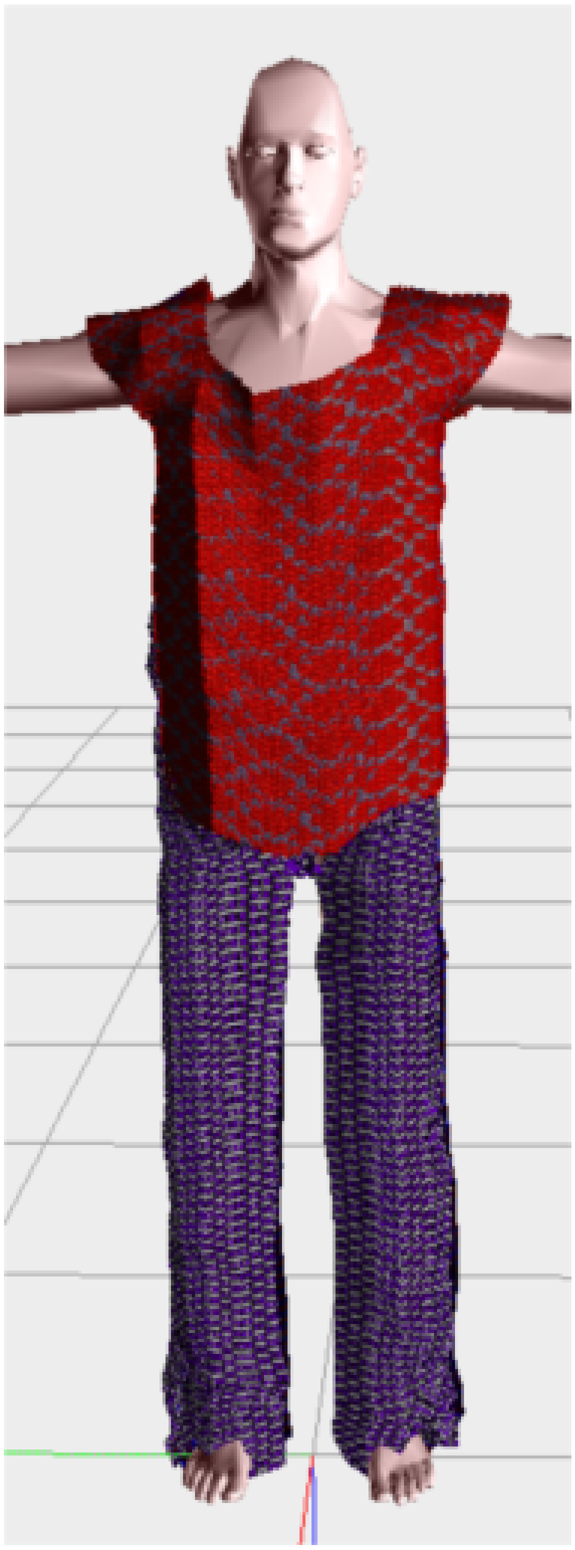


Figure 6.7: Knitted shirt and woven trousers

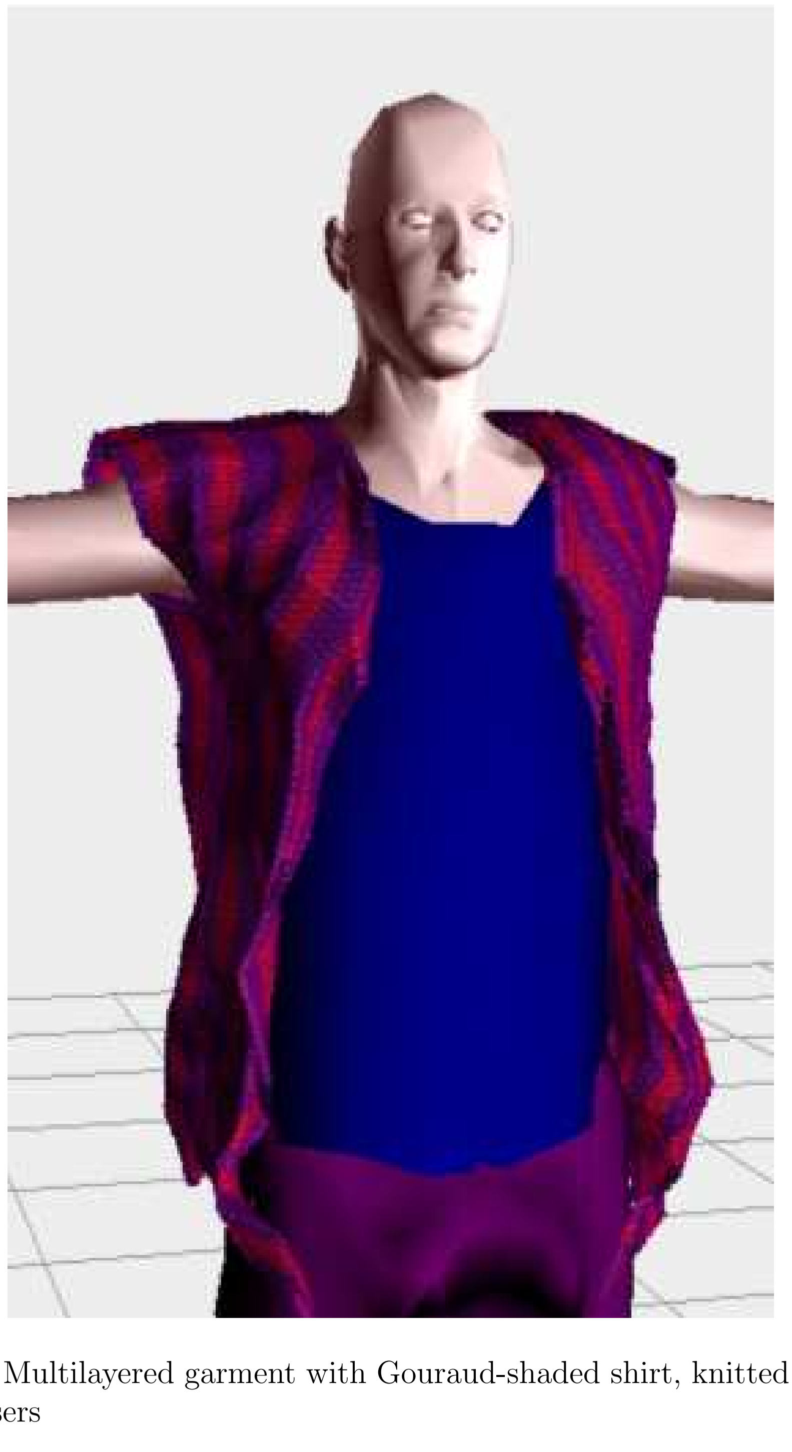


Figure 6.8: Multilayered garment with Gouraud-shaded shirt, knitted jacket and velvet trousers

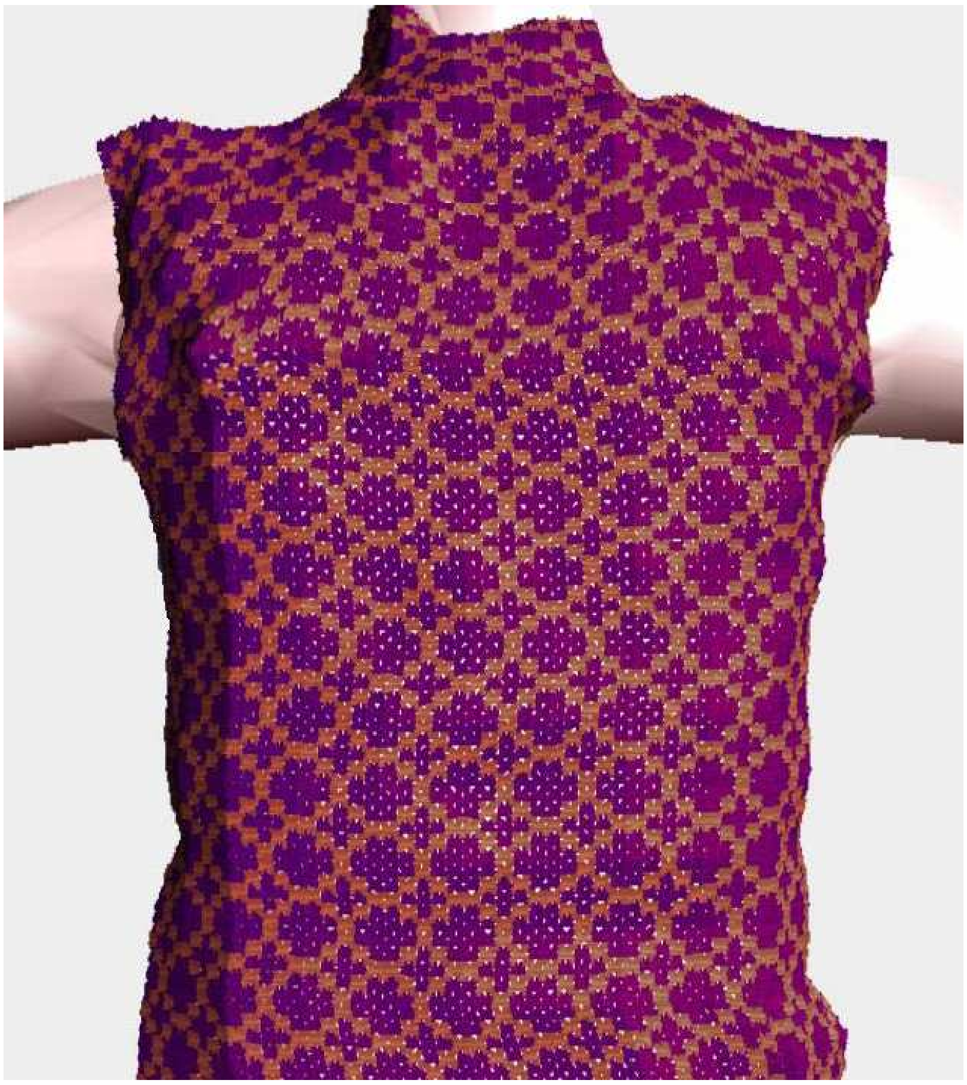


Figure 6.9: Knitted pullover

6.2 Performance Analysis

The system is implemented on a PC (Pentium III, 868 MHz) with 512 MB RAM. The graphics card is NVIDIA GeForce FX 5200 with 128 MB memory. The software platform was Microsoft Visual C++ 6.0 with OpenGL libraries.

The first time performance analysis is done on the rendering times. For this purpose, a cloth mesh with 900 vertices is rendered. A comparison of rendering types is illustrated in Table 6.1. The performance results are highly reduced when complex rendering techniques are applied. For instance, it can be observed that BRDF methods slow down the system as in velvet shading, satin shading and woven cloth with satin BRDF.

Next, Table 6.2 shows the duration at each iteration for the sewing process on garments of two different resolutions. The sewing times increase as two panels approach each other. This is due to the fact that cloth panels also approach the human body and as a result, collision detection/response calculations should be handled.

Table 6.3 shows the duration for the simulation times of garments of two different resolutions. The second column gives the results of a human model dressed with the garments. The third column gives the results of a free-fall test, where collision forces are excluded and the only external force acting on the cloth is the gravitational force. It can be observed that collision handling and high resolution of the cloth are the two factors that reduce frame rates.

Finally, Table 6.4 shows the comparison of different integrator types for a cloth in free-fall state (without collisions). The results are as expected; namely, computation times for Runge-Kutta and implicit integrators are high, whereas explicit Euler and midpoint integrators work faster.

Rendering type	Frame rate
Gouraud shading	50.00
Gouraud shading with smoothing	2.72
Texture mapping	100.00
Velvet shading	2.22
Satin shading	2.00
Woven cloth	2.77
Woven cloth with satin BRDF	0.20
Knitted cloth	1.81
Knitted cloth with detailed rendering of yarns	0.25

Table 6.1: Frame rates for rendering a garment with 900 vertices (in frames per second)

Iteration number	Sewing Time (25 seams)	Sewing Time (66 Seams)
1	0.70	0.58
2	0.82	0.66
3	0.70	0.69
4	0.71	0.76
5	0.84	0.87
6	1.01	1.08
7	1.15	1.32
8	1.28	1.57
9	1.44	1.79
10	1.50	1.98

Table 6.2: The time it takes to sew a garment of 1200 vertices (in seconds)

Number of vertices	Frame rates (with collisions)	Frame rates (free-fall test)
250	0.99	25.00
1200	0.50	12.50

Table 6.3: Frame rates of simulations (in frames per second)

Integrator type	Frame rates
Euler	12.50
Midpoint	7.14
Runge-Kutta of 4'th order	4.50
Runge-Kutta of 5'th order	3.70
Adaptive Runge-Kutta of 5th order	3.84
Implicit Euler	4.16

Table 6.4: Integration times for a cloth of 900 vertices with a stepsize of 0.01 seconds (in frames per second)

Chapter 7

Conclusion

In this thesis study, a 3D graphics tool for virtual garment design and simulation is introduced. The presented system enables the design of various garment models with different fabric types. In addition, the resulting garment is dressed on a virtual mannequin and animated in accordance with the mannequin through collision and friction forces and attachment constraints.

Our system consists of three modules:

- Motion design module
- 2D garment design module
- Human and 3D garment simulation module

Our system is built on top of the human animation system implemented by [35, 61]. The modules related to garment design and simulation are in the scope of this thesis study.

The 2D garment design module enables the user to construct garment panels through cutting, smoothing, scaling and defining seaming points. Although this module is called 2D, in fact it makes use of the two dimensional, flat grid representations of three dimensional cloth pieces. Cutting is done by selecting the particles comprising the boundary of the cloth panel. After the cutting, we smooth the boundary of the cloth panel by using Bézier curves and the boundary particles as the control points.

In our system, we have adopted a particle-based approach, or more specifically a spring-mass system in order to define the topology of the cloth meshes. Our cloth pieces are rectangular grids of particles connected by springs. The mass-spring system that has been used for the 3D simulation of cloth enables the parametrization of internal forces such as bending, stretching or damping of the fabric. This provides us with the ability to implement different types of fabric by just modifying these parameters (parameters of the springs). In addition, external forces such as wind, gravity or air resistance can be applied to the cloth and the behavior of the garment under different environmental conditions can be observed.

After the design part, the garment is fitted on the virtual human by placing the garment panels around the human model and sewing these panels. Then, the human and the garment can be animated. Integration is done on the system and differential equations are solved here in order to simulate the draping of garments. We have implemented explicit and implicit numerical integrators for this purpose.

As well as the simulation of the draping of garments, efficient and realistic visualization of garments is an important issue in cloth modelling. There are various material types and reflectance properties for fabrics. We have implemented a number of material and rendering options such as knitwear, woven cloth or standard shading methods in our system. Here, we can define our own weaving or knitting patterns and can simulate the physical behavior of individual yarns of woven and knitted cloth. Also, by applying various shading techniques, we can render different material types such as satin or velvet.

7.1 Future Work

Currently, one of the problems of the system is that it does not work in real time. Especially, when the cloth resolution increases, frame rates decrease drastically. One of the reasons for this problem is the physical rendering of cloth at milliscale. Since every thread in the cloth is simulated individually, efficiency is highly reduced. In addition, this method also causes a filtering problem when the user zooms out. One solution would be to apply varying levels of detail and mipmapping. However, weave and knit textures should be preprocessed and prepared for mipmapping for this purpose. Moreover, pixel shaders should be incorporated to improve the efficiency and realism of the rendering component. Currently, we are using standard OpenGL functions.

Moreover, fitting the garment on mannequins with different sizes and taking the measurements of the mannequins and creating the garment accordingly is considered to be a future work.

Bibliography

[1] N. Adabala, G. Fei, and N. Magnenat-Thalmann. Visualization of woven cloth. In *Proceedings of the 14th Eurographics Workshop on Rendering*, pages 178–185, 2003.

[2] N. Adabala and N. Magnenat-Thalmann. A procedural thread texture model. *Journal of Graphics Tools*, 8(3):33–40, 2003.

[3] N. Adabala, N. Magnenat-Thalmann, and G. Fei. Real-time rendering of woven clothes. In *Proceedings of ACM Virtual Reality Software and Technology*, pages 41–47, 2003.

[4] M. Ashikmin, S. Premoze, and P. Shirley. A microfacet-based BRDF generator. In *ACM Computer Graphics (Proceedings of SIGGRAPH'00)*, pages 65–74, 2000.

[5] D. Baraff and A. Witkin. Large steps in cloth simulation. In *ACM Computer Graphics (Proceedings of SIGGRAPH'98)*, pages 43–54, 1998.

[6] P. Bézier. *Numerical Control: Mathematics and Applications.* John Wiley & Sons, London, 1972.

[7] T. Bonte, A. Galimberti, and C. Rizzi. A 3D graphic environment for garments design. In *Proceedings of the Workshop on Geometric Modeling*, pages 137–150, 2000.

[8] D. Breen. *"A Survey of Cloth Modeling Methods" in Cloth Modeling and Animation*, edited by D. House, and D. Breen, pages 19–53. A.K. Peters, 2000.

[9] D. Breen, D. House, and P. Getto. A physical-based particle model of woven cloth. *The Visual Computer*, 8(5-6):264–277, 1992.

[10] D. Breen, D. House, and M. Wozny. Predicting the drape of woven cloth using interacting particles. In *ACM Computer Graphics (Proceedings of SIGGRAPH'94)*, pages 365–372, 1994.

[11] L. Chittaro and D. Corvaglia. 3D virtual clothing: from garment design to Web3D visualization and simulation. In *ACM Computer Graphics (Proceedings of SIGGRAPH'03)*, pages 73–85, 2003.

[12] F. Cordier and N. Magnenat-Thalmann. Real-time animation of dressed virtual humans. *Computer Graphics Forum (Proceedings of Eurographics'94)*, 21(3):327–336, 2002.

[13] F. Cordier, H. Seo, and N. Magnenat-Thalmann. Made-to-measure technologies for an online clothing store. *IEEE Computer Graphics and Applications*, 23(1):38–48, 2003.

[14] U. Cugini, A. Galimberti, and C. Rizzi. Industrial research in women clothing industry: project and results. Technical report, Dipartimento di Ingegneria Industriale, Università di Parma, Italy.

[15] U. Cugini and C. Rizzi. 3D design and simulation of men garments. *Journal of Winter School of Computer Graphics*, 10(3), 2002.

[16] K. Daubert, H. Lensch, W. Heidrich, and H. Seidel. Efficient cloth modeling and rendering. In *Eurographics Rendering Workshop'01*, pages 63–70, 2000.

[17] M. Desbrun, W. Schroeder, and A. Barr. Interactive animation of structured deformable objects. In *Proceedings of Graphics Interface (GI'99)*, pages 1–8, 1999.

[18] F. Drago and N. Chiba. *"Procedural simulation of interwoven structures" in Advances in Modelling, Animation and Rendering*, pages 123–138. Springer, 2002.

[19] O. Etzmuss, M. Keckeisen, and W. Strasser. A fast finite element solution for cloth modeling. In *Proceedings of the Pacific Conference on Computer Graphics and Applications (PG'03)*, pages 244–252, 2003.

[20] C. Feynman. Modeling the appearance of cloth. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1986.

[21] A. Glassner. Digital weaving, part 1. *IEEE Computer Graphics and Applications*, 22(6):108–118, 2002.

[22] A. Glassner. Digital weaving, part 2. *IEEE Computer Graphics and Applications*, 23(1):77–90, 2003.

[23] A. Glassner. Digital weaving, part 3. *IEEE Computer Graphics and Applications*, 23(2):84–91, 2003.

[24] C. Gross, A. Fuhrmann, V. Luckas, and J. Encarnacao. Virtual try-on: Topics in realistic, individualized dressing in virtual reality. In *Proceedings of the Virtual and Augmented Reality Status Conference*, 2004.

[25] D. Haumann and D. Parent. The behavioral test-bed: obtaining complex behavior from simple rules. *The Visual Computer*, (4):332–347, 1988.

[26] B. K. Hinds and J. McCartney. Interactive garment design. *The Visual Computer*, (6):53–61, 1990.

[27] S. Huh, D. N. Metaxas, and N. I. Badler. Collision resolutions in cloth simulation. In *Proceedings of Computer Animation*, pages 122–127, 2001.

[28] Y. Kang and H. Cho. Complex deformable objects in virtual reality. In *Proceedings of Virtual Reality Software and Technology (VRST'02)*, pages 11–13, 2002.

[29] İ. Kaynar. Animating dressed virtual humans. Master's thesis, Department of Computer Engineering, Bilkent University, Turkey, 2004.

[30] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Transactions on Visualization and Computer Graphics*, 6:21–36, 1998.

[31] M. Lin and S. Gottschalk. Collision detection between geometric models: A survey. In *Proceedings of the Institute of Mathematics and its Applications (IMA) Conference on Mathematics of Surfaces*, pages 25–32, 1998.

[32] J. Liu, M. Ko, and R. Chang. Collision avoidance in cloth animation. *The Visual Computer*, 12:234–243, 1996.

[33] M. Meissner and B. Eberhardt. The art of knitted fabrics, realistic and physically based modelling of knitted patterns. *Computer Graphics Forum (Proceedings of Eurographics'94)*, 17(3), 1998.

[34] M. Meissner, B. Eberhardt, and W. Strasser. *"Knit Fabrics" in Cloth Modeling and Animation*, edited by D. House, and D. Breen, pages 123–143. A.K. Peters, 2000.

[35] A. Memişoğlu. Human motion control using inverse kinematics. Master's thesis, Department of Computer Engineering, Bilkent University, Turkey, 2003.

[36] M. Meyer, G. Debunne, M. Desbrun, and A. Barr. Interactive animation of cloth-like objects in virtual reality. *Journal of Visualization and Computer Animation*, 12(1):1–12, 2000.

[37] J. Mezger, S. Kimmerle, and O. Etzmuss. Improved collision detection and response techniques for cloth animation. Technical Report WSI-2002-5, Universität Tübingen, 2002.

[38] N. Pelechano. Real-time collision detection between cloth and skinned avatars using OBB. Master's thesis, Department of Computer Science, University College London, 2002.

[39] W. H. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1992.

[40] D. Protopsaltou, C. Luible, M. Arevalo, and N. Magnenat-Thalmann. A body and garment creation method for an Internet based virtual fitting room. In *Proceedings of Computer Graphics International (CGI'02)*, pages 105–122, 2002.

[41] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Proceedings of Graphics Interface (GI'95)*, pages 141–155, 1995.

[42] X. Provot. Collision and self-collision detection handling in cloth model dedicated to design garments. In *Proceedings of Graphics Interface (GI'97)*, pages 177–189, 1997.

[43] K. Roediger and Students of the CADwalk Project. 3D-visualization of garments. In *Proceedings of Computer Graphics International (CGI'98)*, pages 396–401, 1998.

[44] I. Rudomin and J. L. Castillo. Realtime clothing: geometry and physics. In *Poster Proceedings of Winter School of Computer Graphics (WSCG'02)*, pages 45–48, 2002.

[45] C. Schlick. An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum (Proceedings of Eurographics'94)*, 13(3):233–246, 1994.

[46] M. Schweppe. *"A Costume Designer and Animator's Perpective" in Cloth Modeling and Animation*, edited by D. House, and D. Breen, pages 287–307. A.K. Peters, 2000.

[47] D. Sunday. *Distance between Lines and Segments with their Closest Point of Approach.* http://softsurfer.com/Archive/algorithm_0106/algorithm_0106.htm, 2001.

[48] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *ACM Computer Graphics (Proceedings of SIGGRAPH'87)*, volume 21, pages 205–214, 1987.

[49] T. Vassilev and B. Spanlang. Fast cloth animation on walking avatars. *Computer Graphics Forum (Proceedings of Eurographics'01)*, 20(3):260–267, 2001.

[50] P. Volino, M. Courchesne, and N. Magnenat-Thalmann. Versatile and efficient techniques for simulating cloth and other deformable objects. In *ACM Computer Graphics (Proceedings of SIGGRAPH'95)*, pages 137–144, 1995.

[51] P. Volino and N. Magnenat-Thalmann. Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity. *Computer Graphics Forum (Proceedings of Eurographics'94)*, 13:155–166, 1994.

[52] P. Volino and N. Magnenat-Thalmann. Implementing fast cloth simulation with collision response. In *Proceedings of Computer Graphics International (CGI'00)*, pages 257–268, 2000.

[53] P. Volino and N. Magnenat-Thalmann. *Virtual Clothing,Theory and Practice.* Springer-Verlag, 2000.

[54] P. Volino and N. Magnenat-Thalmann. Comparing efficiency of integration methods for cloth animation. In *Proceedings of Computer Graphics International (CGI'01)*, pages 265–274, 2001.

[55] C. Wang, Y. Wang, and M. Yuen. Feature based 3D garment design through 2D sketches. *Computer-Aided Design*, 35(7):659–672, 2003.

[56] J. Weil. The synthesis of cloth objects. In *ACM Computer Graphics (Proceedings of SIGGRAPH'86)*, pages 49–53, 1986.

[57] S. H. Westin, J. R. Arvo, and K. E. Torrance. Predicting reflectance functions from complex surfaces. In *ACM Computer Graphics (Proceedings of SIGGRAPH'92)*, pages 255–264, 1992.

[58] A. Witkin and D. Baraff. Physically-based modeling. *ACM SIGGRAPH Course Notes*, (#25), 2001.

[59] Y. Xu, Y. Lin, C. Zhong, E. Wu, and H. Shum. Photorealistic rendering of knitwear using the lumislice. In *ACM Computer Graphics (Proceedings of SIGGRAPH'01)*, pages 391–398, 2001.

[60] Y. Yang, N. Magnenat-Thalmann, and D. Thalmann. 3D garment design and animation – a new design tool for the garment industry. *Computers in Industry*, 19:185–191, 1992.

[61] M. Ş. Yeşil. Realistic rendering of a multi-layered human body model. Master's thesis, Department of Computer Engineering, Bilkent University, Turkey, 2003.

[62] D. Zhang and M. Yuen. Collision detection for clothed human animation. In *Proceedings of the Pacific Conference on Computer Graphics and Applications (PG'00)*, pages 328–337, 2000.

[63] H. Zhong, Y. Xu, B. Guo, and H. Shum. Realistic and efficient rendering of free-form knitwear. *Journal of Visualization and Computer Animation*, 12(1):13–22, 2001.

Appendix A

The System At Work

Motion control part of the system is implemented by Aydemir Memişoğlu using inverse kinematics based on analytical methods. Realistic rendering using a multi-layered human model that consists of skeleton, muscles and skin is implemented by Mehmet Şahin Yeşil. The garment design and rendering part is implemented by Funda Durupınar. Finally, garment simulation for animating dressed virtual humans including cloth deformation and collision handling is implemented by İlknur Kaynar.

Our application is Single Document Interface (SDI) application implemented using Visual C++ 6.0 and Microsoft Foundation Classes (MFC). The graphics display API OpenGL is used. The top level user interface of the system is seen in Figure A. The elements on the interface can be mainly divided into five parts:

1. *The Main Menu*: This consists of menu bar and toolbar. It basically allows the user to control the application.
2. *The Motion Control, Skinning, Garment Design and Simulation Toolbox*: This toolbox allows user to control the skeleton, generate new motions, and model the skin mesh. In addition, it provides the user to create garments and to simulate them on the human body. It consists of six panels: *Skeleton, Motion, Skinning, Garment Design, Cloth Simulation Parameters* and

Collision Handling.

3. *The Keyframe Editor*: This editor allows the user to generate curves for distance, joint angles in order to characterize the motion of the articulated figure.

4. *The Garment Design Editor*: This editor permits the user to create garment patterns, to cut them and to select particles for seaming.

5. *The Viewing Area*: The viewing area has quad view layout with the front, top, side, and perspective views of the 3D environment.

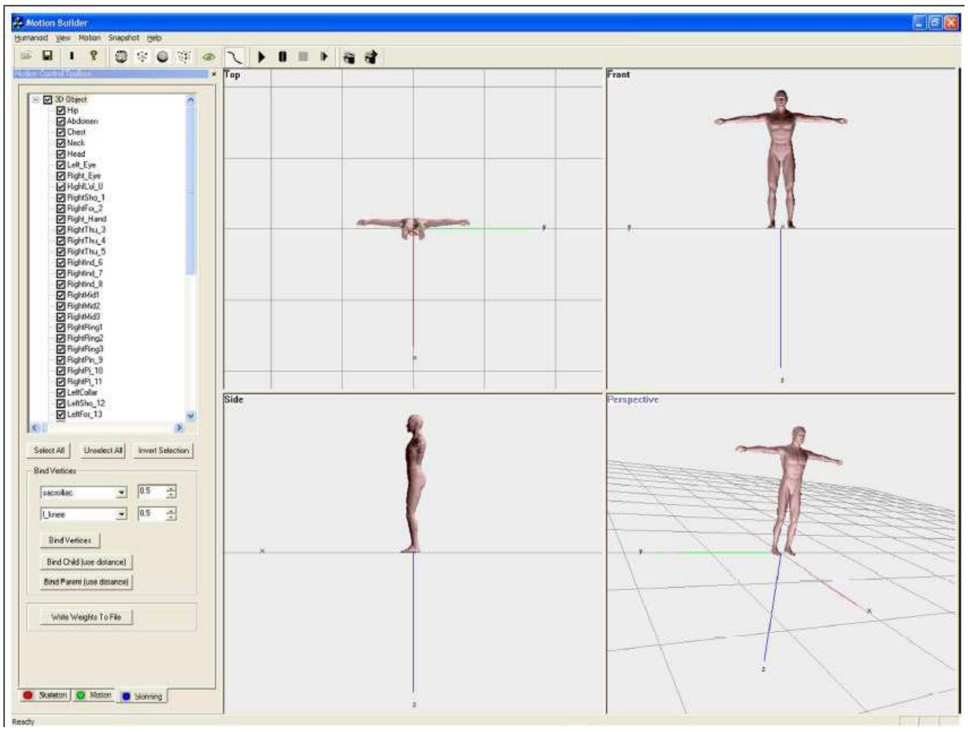
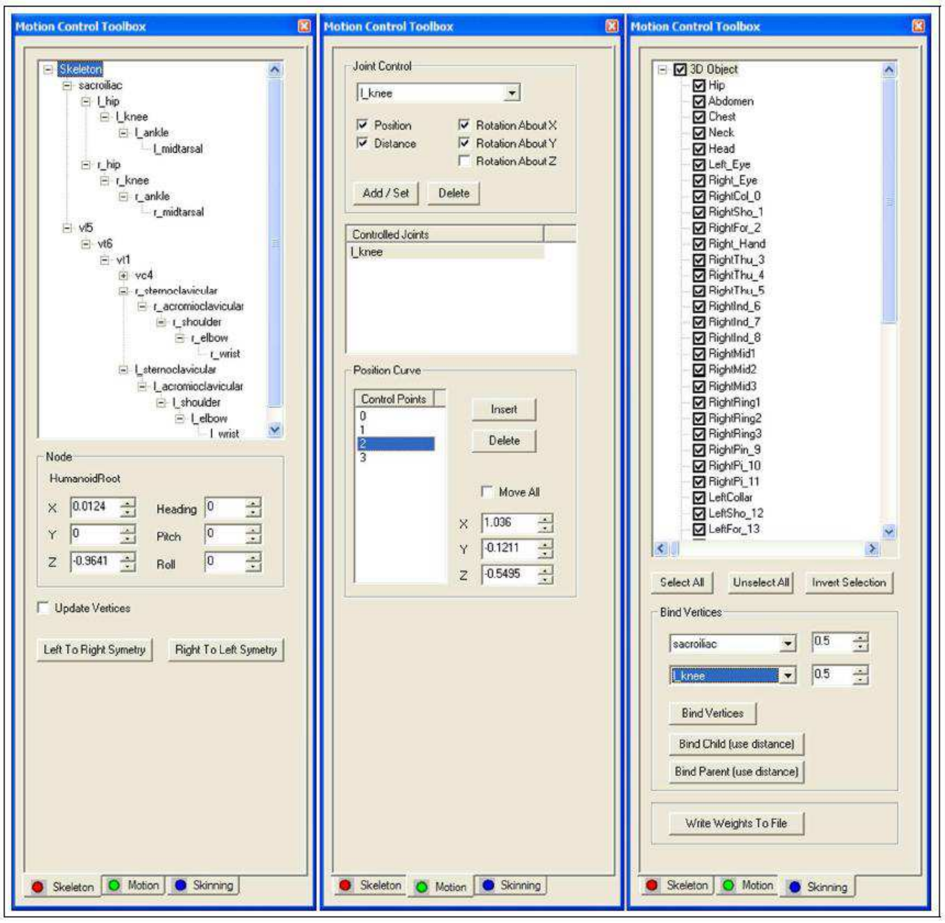


Figure A.1: Top level user interface of the system

The main menu part of the program consists of the menu bar and the other tool- bars. The menu bar includes "File", "View", "Motion", "Snapshot" and "Help" subitems and provides the general functionalities like creating and opening an object model, changing the user interface options, and creating a new motion or loading an existing one. The user also can start, stop, pause and step by step play the loaded animation by using the toolbar. The “Snapshot” menu item and the toolbar gives user the opportunity to take the snapshots of the animation.

The motion control, skinning, garment design and simulation toolbox includes six panels. The first panel, called the skeleton panel, contains skeletal information of the articulated figure. The motion panel of the toolbox allows the user to make modifications on the position curves of the end-effector (ankles and wrists) and root joints of the structure. The skinning panel provides the required functionality for modelling the skin mesh (Figure A.2).



Skeleton
sacroiliac
l_hip
l_knee
l_ankle
l_midtarsal
r_hip
r_knee
r_ankle
r_midtarsal
vl5
vt6
vt1
vc4
r_sternoclavicular
r_acromioclavicular
r_shoulder
r_elbow
r_wrist
l_sternoclavicular
l_acromioclavicular
l_shoulder
l_elbow
l_wrist

Node
HumanoidRoot
X 0.0124 Heading 0
Y 0 Pitch 0
Z -0.9641 Roll 0

Joint Control
l_knee
☑ Position ☑ Rotation About X
☑ Distance ☑ Rotation About Y
☐ Rotation About Z

Controlled Joints
l_knee

Position Curve
Control Points
0
1
2
3
X 1.036
Y -0.1211
Z -0.5495

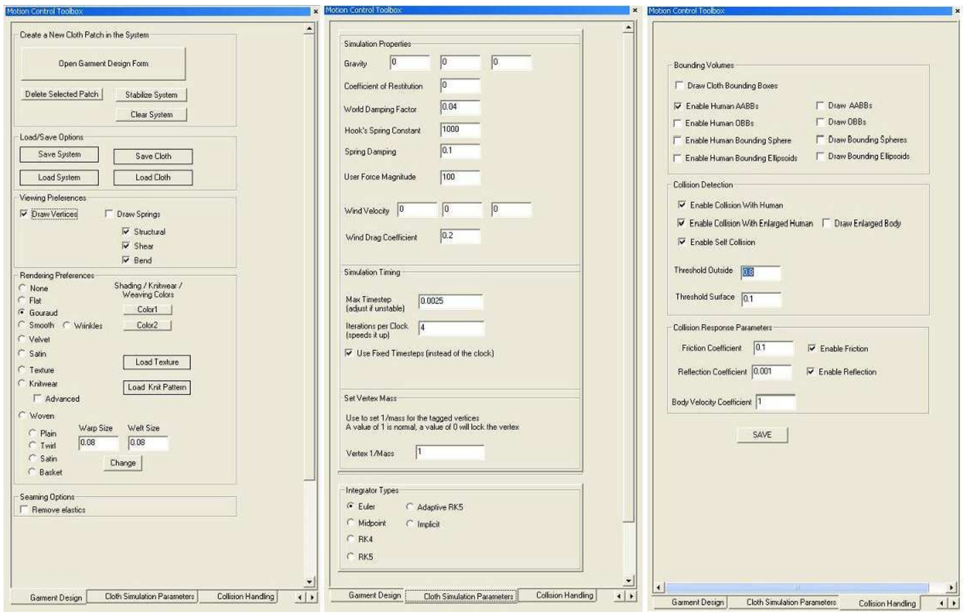
3D Object
☑ Hip
☑ Abdomen
☑ Chest
☑ Neck
☑ Head
☑ Left_Eye
☑ Right_Eye
☑ RightCol_0
☑ RightSho_1
☑ RightFor_2
☑ Right_Hand
☑ RightThu_3
☑ RightThu_4
☑ RightThu_5
☑ RightInd_6
☑ RightInd_7
☑ RightInd_8
☑ RightMid1
☑ RightMid2
☑ RightMid3
☑ RightRing1
☑ RightRing2
☑ RightRing3
☑ RightPin_9
☑ RightPi_10
☑ RightPi_11
☑ LeftCollar
☑ LeftSho_12
☑ LeftFor_13

Bind Vertices
sacroiliac 0.5
l_knee 0.5

Figure A.2: The motion control and the skinning toolbox of the human motion and animation tool

The last three panels are for the garment design and simulation (Figure A.3).

The garment design panel enables the user to load and save garment patterns, to select different rendering options and to open the garment design editor. The cloth simulation parameters panel contains options for setting parameters, such as damping, gravity, wind and time intervals. Lastly, collision handling panel provides selection of bounding volumes for cloth and human body, and setting of collision detection and response parameters.


Motion Control Toolbox

Create a New Cloth Patch in the System
Open Garment Design Form
Delete Selected Patch | Stabilize System | Clear System

Load/Save Options
Save System | Save Cloth | Load System | Load Cloth

Viewing Preferences
☑ Draw Vertices
☐ Draw Springs
☑ Structural
☑ Shear
☑ Bend

Rendering Preferences
○ None
○ Flat
● Gouraud
○ Smooth ○ Wrinkles
○ Velvet
○ Satin
○ Texture
○ Knitwear
☐ Advanced
○ Woven
○ Plain ○ Twill ○ Satin ○ Basket
Shading / Knitwear / Weaving Colors: Color1 | Color2
Load Texture | Load Knit Pattern
Warp Size: 0.08
Weft Size: 0.08
Change

Seaming Options
☐ Remove elastics

Garment Design | Cloth Simulation Parameters | Collision Handling

Motion Control Toolbox

Simulation Properties
Gravity: 0 0 0
Coefficient of Restitution: 0
World Damping Factor: 0.04
Hook's Spring Constant: 1000
Spring Damping: 0.1
User Force Magnitude: 100
Wind Velocity: 0 0 0
Wind Drag Coefficient: 0.2

Simulation Timing
Max Timestep (adjust if unstable): 0.0025
Iterations per Clock (speeds it up): 4
☑ Use Fixed Timesteps (instead of the clock)

Set Vertex Mass
Use to set 1/mass for the tagged vertices
A value of 1 is normal, a value of 0 will lock the vertex
Vertex 1/Mass: 1

Integrator Types
● Euler ○ Adaptive RK5
○ Midpoint ○ Implicit
○ RK4
○ RK5

Garment Design | Cloth Simulation Parameters | Collision Handling

Motion Control Toolbox

Bounding Volumes
☐ Draw Cloth Bounding Boxes
☑ Enable Human AABBs ☐ Draw AABBs
☐ Enable Human OBBs ☐ Draw OBBs
☐ Enable Human Bounding Sphere ☐ Draw Bounding Spheres
☐ Enable Human Bounding Ellipsoids ☐ Draw Bounding Ellipsoids

Collision Detection
☑ Enable Collision With Human
☑ Enable Collision With Enlarged Human ☐ Draw Enlarged Body
☑ Enable Self Collision
Threshold Outside: 0.8
Threshold Surface: 0.1

Collision Response Parameters
Friction Coefficient: 0.1 ☑ Enable Friction
Reflection Coefficient: 0.001 ☑ Enable Reflection
Body Velocity Coefficient: 1

SAVE

Garment Design | Cloth Simulation Parameters | Collision Handling

Figure A.3: The garment design and simulation toolbox

The keyframe editor toolbox will be displayed for a selected joint when one of the controlled joints is double-clicked. While the motion panel includes the controls for manipulating the position curve of a joint, the keyframe editor provides the functionality for editing the distance and orientation curves of the joint (Figure A.4).

Garment design editor contains tools for manipulating garment patterns (Figure A.5). It has options for creating garments, cutting them by selecting the boundary and determining the particles for seaming.

The viewing area of the system includes four view layout; top, front, side, and perspective (Figure A.6). The user can either select and view one of the layouts

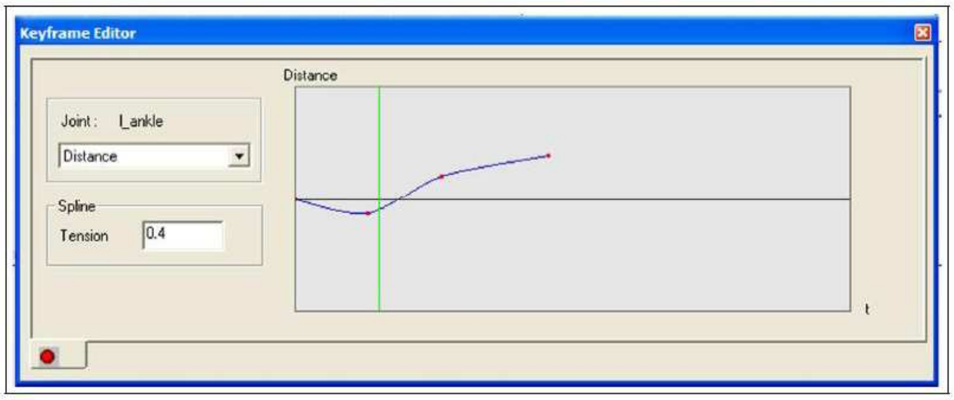
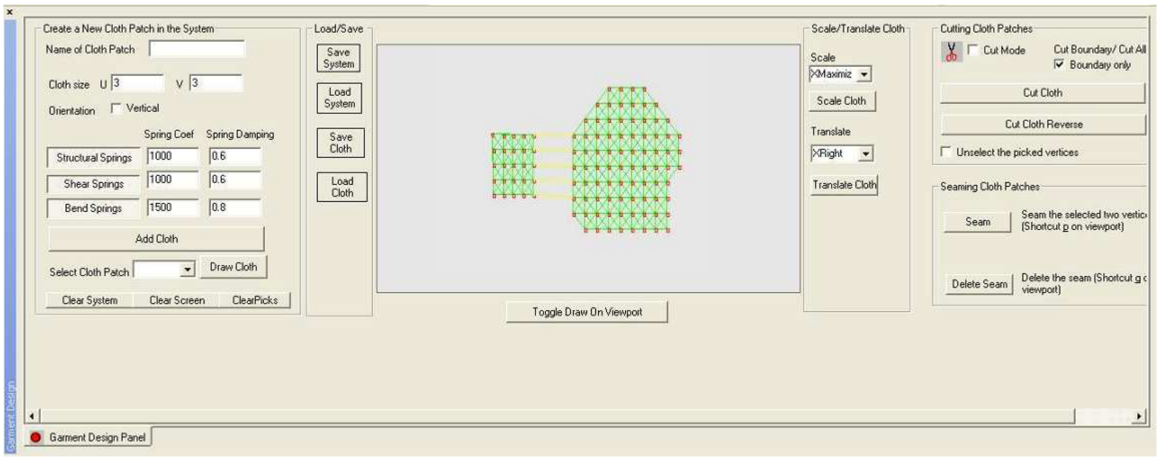


Figure A.4: The keyframe editor of the human motion and animation tool

or use all of them. Zooming and view point change operations can be performed in each frame by using mouse and keyboard keys. All the keyboard and mouse controls are presented in Table A.1.



Cloth size U 3 V 3

	Spring Coef	Spring Damping
Structural Springs	1000	0.6
Shear Springs	1000	0.6
Bend Springs	1500	0.8

Scale XMaximiz

Translate XRight

Cut Boundary/ Cut All [✓ Boundary only]

Figure A.5: The garment design editor

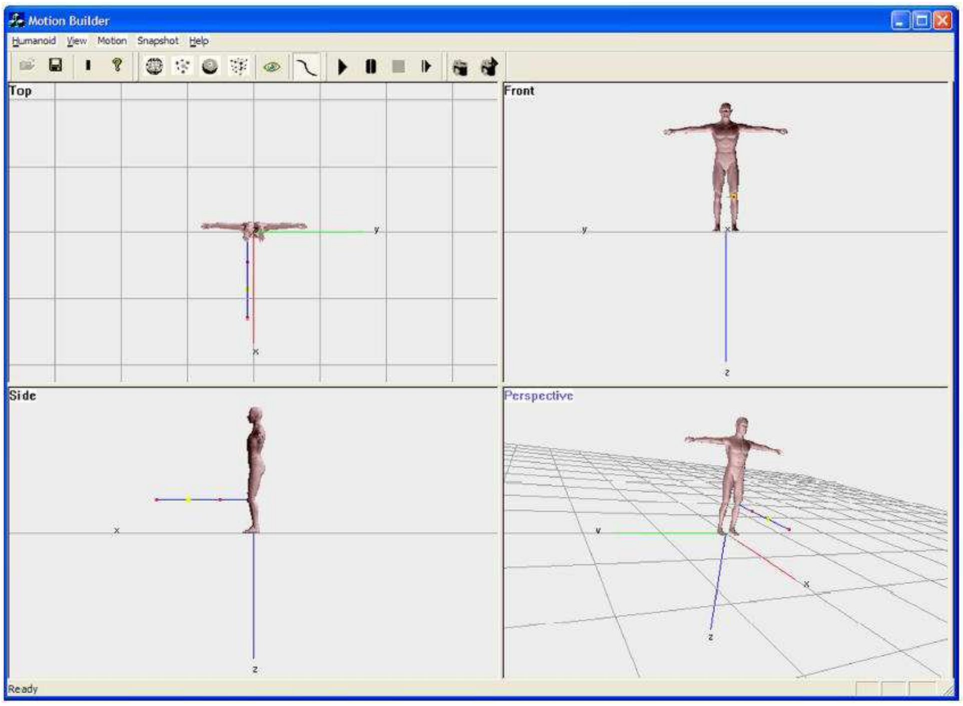


Figure A.6: The viewing area for the Human Motion and Animation Tool

Buttons	Controls
Up	Translates the selected cloth in -z direction
Down	Translates the selected cloth in z direction
Left	Translates the selected cloth in y direction
Right	Translates the selected cloth in -y direction
Home	Translates the selected cloth in x direction
End	Translates the selected cloth in -x direction
1	Scales down the cloth in x axis
2	Scales up the cloth in x axis
3	Scales down the cloth in y axis
4	Scales up the cloth in y axis
5	Scales down the cloth in z axis
6	Scales up the cloth in z axis
X	Rotates clockwise over x axis.
C	Rotates counterclockwise over x axis
Y	Rotates clockwise over y axis
U	Rotates counterclockwise over y axis
Z	Rotates counterclockwise over z axis
A	Rotates counterclockwise over z axis
P	Adds seams between selected particles
Q	Removes seams between selected particles
B	Adds attachment constraint to selected particle
R	Removes the attachment constraint on selected particle
Left Mouse	Selects particles
Left Mouse	Applies user force
Right Mouse	Zooms in/out
CTRL + Left Mouse	Rotates the figure in 3D space
CTRL + Right Mouse	Translates the figure
Shift + Left Mouse	Translates selected cloth in x direction in Garment Design Editor

Table A.1: Keyboard and mouse controls in the system