공학박사학위논문

Methods for Easier and Trouble-Free Construction, Editing, and Simulation of Virtual Garments

가상 의복의 생성, 수정 및 시뮬레이션을 위한 조작이 간편하고 문제를 발생시키지 않는 방법에 대한 연구

2016년 2월

서울대학교 대학원

협동과정 계산과학전공

정 일 회

Methods for Easier and Trouble-Free Construction, Editing, and Simulation of Virtual Garments

가상 의복의 생성, 수정 및 시뮬레이션을 위한 조작이 간편하고 문제를 발생시키지 않는 방법에 대한 연구

지도교수 : 고 형 석

이 논문을 공학박사 학위논문으로 제출함

2015년 12월

서울대학교 대학원
협동과정 계산과학전공

정 일 회

정일회의 공학박사 학위논문을 인준함

2016년 2월

위 원 장	이 경 무	(인)
부위원장	고 형 석	(인)
위 원	조 남 익	(인)
위 원	강 명 주	(인)
위 원	김 종 준	(인)

Abstract

Methods for Easier and Trouble-Free Construction, Editing, and Simulation of Virtual Garments

Ilhoe Jung

Interdisciplinary Program in Computational Science and Technology

The Graduate School

Seoul National University

This dissertation presents new methods for the construction, editing, and simulation of virtual garments. First, we describe a construction method called TAGCON, which constructs three-dimensional (3D) virtual garments from the given tagged and packed panels. Tagging and packing are performed by the user, and involve simple labeling and two-dimensional (2D) manipulation of the panels; however, it does not involve any 3D manipulation. Then, TAGCON constructs the garment automatically by using algorithms that (1) position the panels at suitable locations around the body, and (2) find the matching seam lines and create the seam. We perform experiments using TAGCON to construct various types of garments. The proposed method significantly reduces the construction time and cumbersomeness.

Secondly, we propose a method to edit virtual garments with synced 2D and 3D modification. The presented methods of linear interpolation, extrapo-

lation, and penetration detection help users to edit the virtual garment interactively without the loss of 2D and 3D synchronization.

After that, we propose a method to model the non-elastic components in the fabric stretch deformation in the context of developing physically based fabric simulator. We find that the above problem can be made tractable if we decompose the stretch deformation into the immediate elastic, viscoelastic, and plastic components. For the purpose of the simulator development, the decomposition must be possible at any stage of deformation and any occurrence of loading and unloading. Based on the observations of various constant force creep measurements, we make an assumption that, within a particular fabric, the viscoelastic and plastic components are proportional to each other and their ratio is invariant over time. Experimental results produced with the proposed method match with general expectations, and show that the method can represent the non-elastic stretch deformation for arbitrary time-varying force.

In addition, we present a method to represent stylish elements of garments such as pleats and lapels. Experimental results show that the proposed method is effective at resolving problems that are not easily resolved using physically based cloth simulators.

Keywords: virtual garment construction, virtual garment editing, cloth simulation, non-elastic deformation

Student Number: 2012-30911

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Digital Clothing

Since humans began to design and make clothes in the paleolithic age, many studies for creating functional and well-designed clothes have been carried out by various industries such as textile science and fashion design. Recent studies into clothing have attempted to develop a new method called digital clothing, which is the application of digital technology to clothing technology. Digital clothing is an interdisciplinary technology whose development has benefited from collaborations between many different engineering and science fields such as computer graphics (CG), computer-aided design (CAD), textile science, and fashion design.

The most common process involved in garment creation can be summarized using the following four steps: *(1) design*, *(2) pattern making*, *(3) sewing*,

Figure 1.1: Virtual garments displayed at www.digitalclothing.org.

and *(4) verification by draping*. The CAD community has focused on developing the *(1) design* and *(2) pattern-making processes*. As a result, most common clothing-design and pattern-making tools include CAD systems. Unlike the CAD community, the goal of the CG community is to represent realistic clothes using 3D animation in movies and games, and this is related to *(4) verification by draping*. Today, virtual garments that are represented using CG are very realistic as the human eye can be tricked. The realistic virtual garments shown in Figure 1.1 are created using digital clothing software. In addition, realistic clothing animations have been used in movies such as *Frozen*, *Tangled*, and *The Lord of The Rings*.

With the advances in digital clothing technology, not only is it used to create clothing, but it is also used for sales and marketing, which have become important topics. Because of the emerging trends to shop for clothes online,

there is a huge demand for the development of virtual try-on systems. An example can be viewed at www.metail.com, where the user can fit the item of clothing on an avatar of himself or herself. As a result, digital clothing is not just tied to the use of CG and CAD, but has become a new independent subject. Below, we list four technologies that are key topics in the evolution of digital clothing technology, as described in Ko [38].

Physical Cloth Simulation Because of the novel works performed by many researchers [65, 63, 4, 63, 21, 14, 71], the realistic motion of virtual clothing sometimes leads persons to believe that there is no room for further enhancements in the simulation of physical clothing. However, to obtain high-quality scenes, virtual clothes in movies are simulated not in real time, but over a very long time. In addition, the types of fabric are limited to avoid using cumbersome physical models. However, virtual fitting systems, which are in demand on the market, must be able to represent various kinds of fabrics in real time. Therefore, we are required to develop a novel physical simulation method to achieve those goals.

Garment Modeling To represent virtual clothes in a virtual fitting system, an obvious condition is that the garments to be represented must first exist in the system. Over the past twenty years, many researchers [72, 66, 67, 25, 64, 62, 40, 22, 26, 68, 53, 54, 32, 10] have tried to develop appropriate ways of creating garments. In addition, many commercial and academic tools have been

developed to assist with the garment-design process. However, none of these methods or tools is entirely suitable for various applications such as movies, games, and virtual fitting systems. In particular, the number of 3D garments that must be entered can exceed one thousand per week to support a virtual fitting system for the online market. In this case, we need a simple and robust system that can create virtual garments in a short time.

Fabric Rendering Fabric rendering is another important technology for the development of realistic virtual garments. Some recent rendered images of fabrics appear realistic, and this is a result of many studies [1, 3, 19, 30, 41, 42, 61, 74, 76, 77, 78]. However, many kinds of fabrics are composed of complex woven structures, and they are difficult to represent on a virtual system. To increase the range of garment types available to a virtual fitting system, there is a need for more studies to render the complex structure of fabrics.

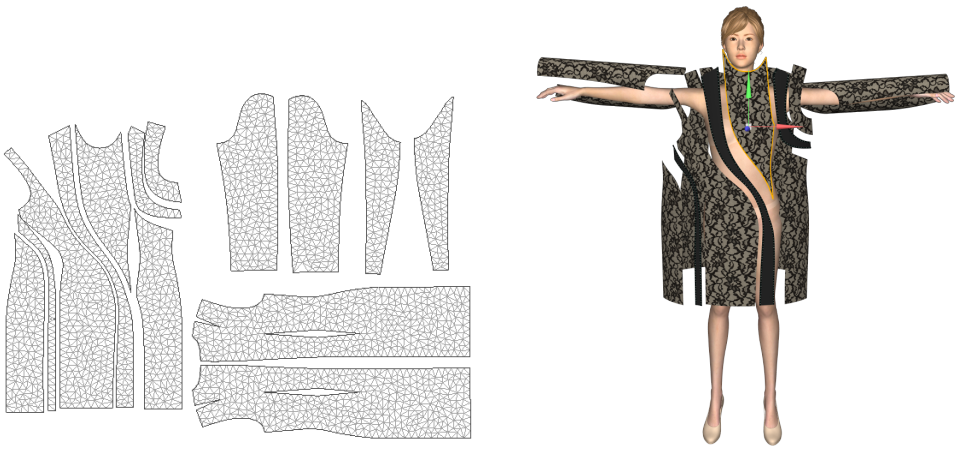
Avatar Generation Users of virtual fitting system desire to see realistic scenes where their avatars appear to duplicate a real body. Because of the ongoing attempts to use CG to create realistic human face, hair, and body [5, 7, 6, 9, 8, 12, 17, 16, 18, 29, 36, 37, 49, 50, 69, 75, 39], its realization is expected to be in the near future. Some issues that remain are the need to increase the similarities between avatars and humans, as well as the reduction of the computation time needed to generate the avatar.

In this thesis, we focus on the development of novel *garment modeling*

and *physical-cloth simulation* methods.

1.2 Garment Modeling

Over the past five years, we have focused on educating persons in the fashion industry regarding the use of digital clothing software to construct various types of garments. Many of these persons are fascinated to see the simulated results, but when they try to *construct* the garments themselves, the fascination rapidly changes to frustration. For example, consider a garment consisting of the panels shown in Figure 1.2 (a). Positioning the panels around the body and creating seams between them requires hundreds of panel-dragging operations and view changes in the 3D window, and minor errors can lead to surprising outcomes. Even for expert users, the task is cumbersome and difficult. The



(a) (b)

Figure 1.2: Manual construction of the garment: (a) the given panels, (b) positioning of each panel with 3D manipulation.

CG field has made significant progress in the accuracy and speed of physical simulations, but it has made relatively less progress in enhancing their usability.

The above inconvenience needs to be resolved to enable such construction to be achieved for thousands of garments per week, which is easily the case when operating a commercial virtual fitting system.

In this work, we propose an automatic garment-construction method to minimize the inconvenience caused. We analyzed various kinds of clothing patterns to find classification rules that exist between them. Based on our analysis and study of previous research, we noted that there are several types of panels and lines of garments, which we refer to as *tags*. All of the described garment-construction algorithms employed in this thesis are based on that classification. Therefore, we call the proposed method tag-based automatic garment construction (TAGCON). The proposed method effectively reduced the need for attention-intensive work in the preparation of virtual garments.

It is not usually the case that a garment created from an original design fully satisfies the expectation of a designer, even when the garment is created using the TAGCON. There is usually a need for post processing to modify the design and fit of aspects of garments. Three types of modification operation may be performed by a designer:

– **Changing Design:** Modify the design elements of a garment.

– **Grading:** Resize a garment pattern based on the grading rules to fit a

specific body.

– **Editing:** Modify local parts of a garment. For example, modify the length of sleeves and pants.

The development of each subject faces specific challenges, and various studies have therefore been performed to resolve these issues. In this thesis, we propose a garment-editing method that can be easily integrated with a digital-clothing system. The proposed editing method helps the user to instantly confirm changes to the 3D garment shape from 2D changes in the pattern. Because this method handles the penetration during the editing stage, it does not cause a penetration problem by abrupt garment editing.

1.3 Physical Cloth Simulation

To represent various kinds of garments in a virtual try-on system, an included physical simulator must can represent various physical properties to simulate various kinds of created garments. In this thesis, we present a simulator that can represent non-elastic physical properties of the fabric [45].

So far, most clothing simulators have been developed under the assumption that cloth is elastic material. With some amount of parameter tuning, those simulators could produce visually realistic results. Nevertheless, deformation of cloth is not elastic. When deformation occurs, some part of it is restored immediately, but some part (viscoelastic component) is restored over a duration, and some other part (plastic component) is never restored. Stress–strain

measurement of various fabrics clearly reveals that deformation of cloth has non-elastic components. For the clothing simulation techniques to be useful in both animation and clothing industries, the above non-elastic components should be accounted for. This work is an attempt to model the non-elastic component of cloth deformation focusing on the stretch, in the context of developing a clothing simulator.[1]

When deformation occurs, cloth has a general tendency to go back to the original condition. Such a tendency is well represented with the previously proposed elastic models. That said, this work develops a new physical model that can be used for augmenting an existing elastic model to account for non-elastic components as well.

We find the proposed framework that augments an existing elastic simulator into a general simulator practically attractive, since the above conversion can be done with a small amount of programming effort and the augmentation introduces only a marginal extra computation to the existing simulator.

Although the proposed simulation can represent various types of garments, it is still difficult to represent stylish elements such as pleats and lapels. Pleats and lapels include folds that cannot be easily represented because the shapes of garments are completely smooth when the simulation is started. Therefore, we propose a method called pre-folding, which preprocesses the panel to obtain

[1]We limit the scope of the non-elasticity study in this thesis to the stretch deformation, leaving the non-elasticity in shear and bending unexplored. One reason for such limited handling is because, in the mechanism, the non-elasticity in stretch is very different from those in shear and bending.

a folded configuration before starting the simulation. The proposed method is shown to effectively represent the fold in stylish elements.

1.4 Dissertation Overview

The remainder of this dissertation is organized as follows. First, we discuss previous studies which are related with our work. In Chapter 3, we describe an automatic garment construction method based on the pattern analysis. In Chapter 4, we propose a method to edit constructed garments with synced 2D pattern and 3D garment shape. In Chapter 5, we introduce a physically based cloth simulator that can represent non-elastic stretch deformation. In Chapter 6, we present a simple method to represent stylish elements of garment such as pleats and lapel. Finally, we summarize proposed methods and conclude this dissertation.

Chapter 2

Previous Work

Digital clothing technology has become an indispensable technology not only in the visual-effects industry, but also in the fashion and textile industry. This is because of significant advances in the areas of physical simulation, fabric rendering, garment modeling, and avatar generation. Research into digital clothing technology has gradually increased based on the demand by industries. In this chapter, we describe studies that are highly relevant to the suggested framework. First, we discuss garment-modeling methods that have been studied over the past twenty years. Then, we discuss garment-simulation methods that are related to the development of non-elastic cloth simulators.

2.1 Garment Modeling

The most common garment-modeling processes can be categorized as design, pattern making, panel positioning, and sewing processes. First, most works fo-

cused on making convenient design tools that assist with actions such as drawing, cutting, and pasting. Actually, those works were applicable not only for garment modeling, but also for modeling a variety of other products. Only after the passage of time did persons begin to show interest in improving specific garment-modeling processes because CG and CAD technologies had reached sufficiently advanced levels.

McCartney et al. [51] proposed a new design cycle including pattern flattening, pattern development systems and draping engines. Igarashi and Hughes [40] proposed techniques for positioning and manipulating clothes using 3D characteristics. Protopsaltou et al. [62] and Cordier et al. [22] represented a virtual garment framework for online shopping. Their works resolved various problems regarding the application of digital clothing in Internet environments. Fuhrmann et al. [28] positioned garment patterns with a 3D human scan using two steps, geometric pre-positioning and physically based end-positioning. Kim and Kang [47] and Kim et al. [46] used body scan data to generate garment patterns.

Volino et al. [70] introduced a virtual fitting system that can modify a garment design interactively. Umentani [68] also presented an advanced interactive garment modeling and editing system based on the sensitivity analysis. Meng et al. [52] employed a hierarchy of ellipsoids to find the optimal position of the panels around the body. To create complex garments, they proposed four types of user interactions to control the panel position, namely, move, rotate, fix, and drag. Another method that can be used for interactive garment

editing was introduced by Meng et al. [53]. They introduced user operations for style editing and pattern alternations based on geometrical and physical methods.

Meng et al. [54], Brouet et al. [15], and Jeong and Ko [44] resized a garment while preserving its shape. Jeong et al. [43] reported a method for creating a garment from a single photograph.

Recently many studies have focused on the fact that most garment designs began with a concept sketch. Works that are based on this fact can be classified as sketch-based design methods.

Sketch-based design Sketch-based garment design methods analyze a user's sketch to make patterns and create 3D garments. Wang et al. [72] used garment feature templates and sketched profiles to create a garment. Turquin et al. [66] created garments from an outline drawn over the silhouette of a body. Turquin et al. [67] provided various user modes such as the contour mode, folding mode, and front/back modes. Therefore, it can represent a wider variety of fashion elements than their previous work [66]. Decaudin et al. [25] obtained information of contour and seam-lines from sketches to create not only complete 3D garment, but also patterns that may be developed. Robson et al. [64] proposed a context-aware sketch interpretation method that is based on the observation of key factors.

Sketch-based design methods have an advantage in that their processes are familiar to designers, but they also have problems such as pattern flattening,

which makes it hard to create a complex garment for manufacturing. In our work, we assume that the patterns of a garment are already given, so we can focus on panel positioning and seam creation, which do not suffer from the pattern-flattening problem.

The work that is most relevant to our garment-modeling method is the study by Berthouzoz et al. [10], which extracts panels from a PDF file and parses sewing patterns to construct the virtual garment automatically. This machine-learning-based method requires several training data sets, but does not require any information from the user. It is a probabilistic model, and success is therefore not guaranteed. They reported that the success ratio was about 68%, and it may vary when the user constructs a garment that is different in type from the ones in the training data set.

In contrast to Berthouzoz et al. [10], we propose a deterministic method[1], that does not call for the training data set. Further, as long as the input garments are within the inherent range of the method, its success ratio is over 96%. The key idea of our method is that if the program positions the panels at *appropriate* locations, the spatial relationship (e.g., adjacency) between lines can be an essential cue for identifying the seam-line pair. We concluded that determining appropriate locations of the panels is not possible without any information, in this thesis, we developed a systematic approach for the user to provide the information, namely panel tagging and packing.

[1]The proposed method also contains some probabilistic ingredient. However, compared to Berthouzoz et al. [10], the probabilistic ingredient is much less.

2.2 Physical Cloth Simulation

The CG field made a remarkable progress in physically-based clothing simulation over the past three decades. Since Terzopoulos et al. [65] introduced the first physically-based cloth model, many researchers (including Baraff and Witkin [4], Volino et al. [71], Provot [63], Bridson et al. [14], and Choi and Ko [21]) have contributed with new improvements in the numerical stability, computational cost, and realism.

The graphics community started to consider the non-linearity, anisotropy, and hysteresis of cloth in the development of simulators. Provot [63], Bridson et al. [14], and Müller [56] used strain-limiting as a means to approximate the non-linearity in the stretch. Volino et al. [71] attempted modeling of the non-linearity in stretch by taking the curve from the actual measurements. Utilization of the data collected from Kawabata evaluation system was attempted by Breen et al. [13] in the calculation of the static draping of various fabrics. Another data-driven elastic model was proposed by Wang et al. [73] which used a piece-wise linear elastic model to represent the non-linearity in various fabrics, along with an invention of a novel measurement system. Chen et al. [20] proposed a method for modeling friction and air effects between cloth and deformable bodies. However, none of the above works explicitly modeled the hysteresis of cloth.

Lahey [48] and Ngo-Ngoc and Boivin [59] are the works that pioneered incorporation of the hysteresis effect to the clothing simulators. Both works

assumed that fabric hysteresis results from the friction between weaves, and resorted to the friction models proposed by Dahl [24] and Bliman and Sorine [11]. Recently, Miguel et al. [55] also resorted to Dahl's friction model and reproduced non-elastic stretch and bending behaviors of cloth impressively.

The proposed work complements the limitations of Miguel et al. [55], in which (1) the dynamic friction model does not explicitly include the plastic deformation, and (2) it is hard to represent the time effect (as in creep/relaxation). Our method resolves the above problems by modifying the rest shape of a cloth in a simulator based on the Kelvin model instead of applying the friction force to vertices.

The plastic deformation of elastic material has been modeled by Müller and Gross. [57] Recently, Narain et al. [58] proposed a method for modeling the bending plastic deformation in thin shells through adaptive re-meshing.

The creep of fabrics has been a well-studied phenomenon in textile science. However, the subject has not been studied much yet in the context of clothing simulation. A book of Hearle and Morton [35] is one of the books that are frequently referenced. The book surveys various findings regarding the physical properties of fabrics, including non-elastic deformations. In particular, the book includes an in-depth description on creep. Gutpa and Kumar [33] developed a creep model for the textile fiber, taking the general approach of the material science. Cui and Wang [23] decomposed creep into four components, namely instantaneous elastic, instantaneous plastic, viscoelastic, and viscoplastic, and modeled each component independently.

Chapter 3

Automatic Garment Construction from Pattern Analysis

The ideal scenario regarding garment construction would involve a program automatically constructing garments without any input from the user. However, this is not possible; in fact, it is an ill-posed problem. From the perspective of usability, the following two principles have been identified:

- **Determinism:** Sure-success is preferred to occasional-failure, even if the method requires some input from the user.
- **Dimensional Overhead:** Simple labeling is preferred to 2D or 3D manipulation. If manipulation needs to be done (to the panels), doing it in 2D is preferred to doing it in 3D.

In this study, we propose a new technique that is based on the above two principles, which enable the computer to perform the construction job (panel

positioning and seam creation) in a deterministic way. The technique requires the user to do some preparation called tagging-and-packing for the panels involved. However, the required preparation is simple labeling or manipulation in the 2D window.

After the user prepares tagged-and-packed panels (which do not need to be positioned with respect to the body), the TAGCON system then produces the completed (positioned, seamed, and draped) garment in 3D, as shown in Figure 3.1.

1. **Cylindrical Positioning:** Locates the panels on the surface of cylinders (Figure 3.2) to enable them to surround the avatar.

2. **Seam Creation:** Identifies the line pairs to be seamed.

3. **Draping Simulation:** Performs a physically based simulation of the constructed garment on the avatar.

The *tag-and-pack* process (the details of which are presented in Section 3.1) is not considered to be significant overhead to the pattern-making experts, but it provides essential tips to TAGCON, significantly reducing the construction efforts.

The remainder of this chapter is organized as follows. The first and second sections describe methods to classify panels and lines. Then, in the third section, we introduce a method to create seams based on those classifications. Finally, we present experiment results that were created using the proposed method, and we conclude this work.

3.1 Panel Classification

In the clothing field, panels are often named according to the position that they occupy during the garment construction. A few examples are the *top front* panel and *right sleeve* panel, which enable us to guess where they should be positioned for the construction. Berthouzoz et al. [10] referenced the panel name for identifying the matching seam line pairs. Unfortunately, each manufacturer may use a different naming scheme, and the name may not be related to the panel's 3D position. Therefore, we devise a convention called *panel tagging*, where every panel in our system should be named according to a rule.

3.1.1 Panel Tagging

To classify the panel efficiently, we assume that the tagging must satisfy the following two conditions:

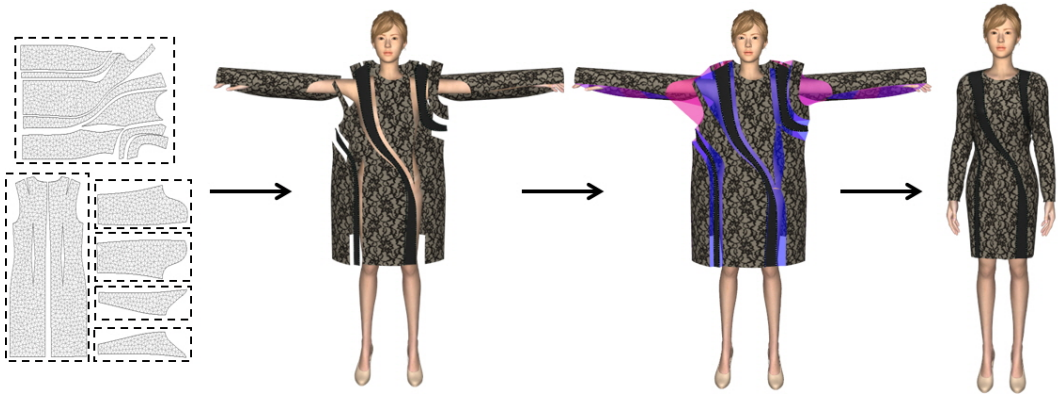


Figure 3.1: The TAGCON process.

– **Coverage with Fixed Key Words:** The tagging should involve only a fixed number of keywords. The number of keywords should not increase as more diverse garments are covered.

– **Position Encoding Capability:** The tagging should be able to uniquely determine the panel's body-relative (sensible) position.

An important finding of this work is that such a tagging convention exists. In fact, we propose one such tagging scheme based on the *cylindrical abstraction* of the situation. Topologically, we note that any garment can be abstracted as a number of cylinders. For example, a T-shirt can be abstracted as consisting of a top, left sleeve, and right sleeve cylinders, and a subset of the full cylinder set is shown in Figure 3.2.

In the context of the cylindrical abstraction, we tag the panel P as (α, β, λ) where α, β, and λ are predicates regarding P. α designates the particular

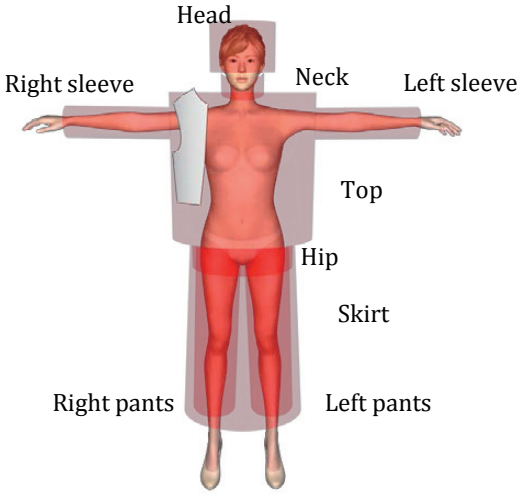
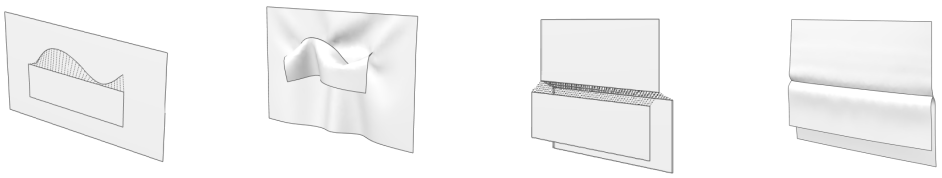


Figure 3.2: The cylindrical abstraction.

cylinder, and β takes either the front or back. For example, the (Top, Front, λ) panel can be positioned on the cylindrical surface as shown in Figure 3.2. λ represents the layer to which P belongs. In most garments, $\lambda = 0$ for all of the panels. However, in the cases shown in Figure 3.3, the panels need to be ordered in-to-out. $\lambda = \pm 1, \pm 2, \ldots$ are used to encode the outer/inner layers. The case shown in Figure 3.3 (a) and (b), where the seam is made with an interior line is called the *T-seam*. The case shown in Figure 3.3 (c) and (d), where multiple panels are seamed with a single contour is called the *Y-seam*. We refer to the other (usual) seams as *normal seams*. Note that the layering of the panels is essential to enable us to construct complex structured garments, and such a situation would not be identifiable without a tip from the user.[1]



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

Figure 3.3: T-seam and Y-seam. (a)-(b): T-seam before and after the simulation, (c)-(d): Y-seam before and after the simulation.

[1]Note that all the algorithms presented in this paper assume that a single garment is being constructed; even if two panels are in the different layers, they will belong to the same garment. If the user wants to create multiple garments, e.g., a blouse and pants, he/she has to run TAGCON twice, once for the blouse, and another time for the pants.

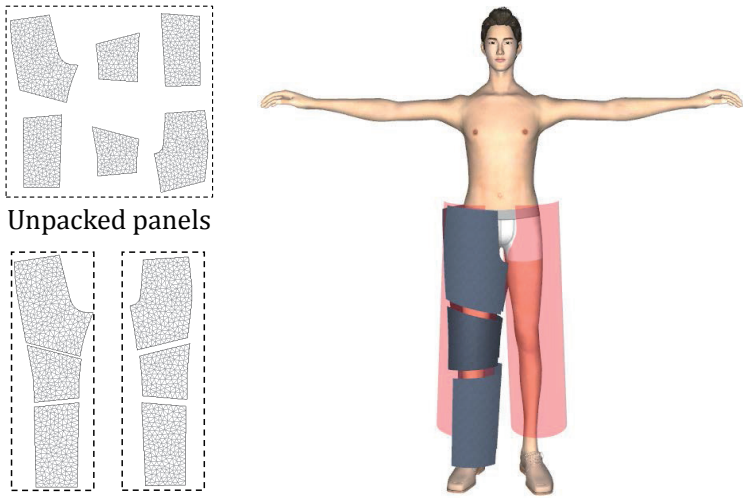


Figure 3.4: The panels before packing, after packing and on the cylinder.

3.1.2 Panel Packing

There is a limitation in the proposed panel tagging. For example, when the right pants consists of multiple panels as shown in Figure 3.4, the tagging cannot resolve the situation completely with (Right pants, Front, *) and (Right pants, Back, *) only.

Considering this limitation, the panel-tagging idea may be considered unsuitable. However, if we dismiss the tagging, garment construction will become more or less a jigsaw puzzle situation. Unfortunately, it is not a well-posed jigsaw puzzle; we can create a case consisting of many small panels for which only the original designer knows the answer.

In this work, we propose a simple but very effective approach to cope with the above limitation of panel tagging, which forms another main finding of this study. Note that when multiple panels should belong to the same (α, β, λ),

clothing experts know how those panels should be arranged. The basic idea of panel packing is to let a clothing expert do the arrangement. Then, in the subsequent panel positioning, the packed panels are moved as a group (with their relative positions being fixed).

When the user has completed the packing, the pack is positioned relative to the cylinder as follows. First, we create a (2D) bounding box that encloses the panels. Then, we wrap the bounding box to the tagged cylinder. Here, the only degree-of-freedom (DOF) that needs to be specified is the vertical offset within the cylinder, because the bounding box should come at the horizontal center of the half cylinder. We obtained the vertical offset for each cylinder from the cloth experts. The proposed method is not very sensitive to the choice of the vertical offset.

The atomic unit in the panel positioning is a pack. Even for unpacked panels, we create a bounding box and regard it as a pack. Then, we locate it on the cylinder as described above.

3.1.3 Tagging-and-Packing Process

Next, we discuss in detail the process regarding the tagging and packing of the garment by the user.

0. Initially, for all of the panels the layer λ is automatically tagged as zero.

1. If there are panels to be packed, the user performs the packing (by positioning them in 2D). The remaining panels are automatically packed as

single panel packs.

2. The user tags each pack, i.e., sets (α, β, λ). In most cases, λ does not need to be changed from its initial value. In most garments, there are only about ten packs for which the user has to set (α, β), by selecting the half-cylinder from the drop-down menu.

3.2 Classification of Seam-Line

There is another kind of tagging to be done, namely *seam line tagging*[2], which tags each line segment that will participate in a seam. Berthouzoz et al. [10] performed this classification probabilistically using machine learning. In contrast with their works, we propose a deterministic method to find seam-lines based on the geometric analysis of lines of panels.[3]

We devised a seam-line classification method based on the fact that a line pair of a seam sweep (a sweep is defined as being between the matching seam-line pair) penetrating the body cannot be connected, and these kinds of line pairs are located on the boundary of the body. The brief algorithm of the seam-line classification is described as below.

1. Find the borderlines (the lines cross the body silhouette).

2. Classify the neighbor lines as candidate lines, which are located on the nearest cylinder of borderlines.

[2]In contrast, we will call the above (α, β, λ) as the *panel tagging*.

[3]We note that the clothing experts prefer the deterministic situation

3. Find the seam-line matches of the borderlines and candidate lines.

4. Classify the non-matched borderlines as non-seam-lines.

5. Classify the lines of the seam sweep that penetrate the body as non-seam-lines.

Some of the seam lines tagged above require further classification. We call this the *secondary seam line tagging*, which should be done to the T-seams and Y-seams. The T-seam lines can be identified by the program without the need for user specification, but the Y-seam lines need to be explicitly tagged as such by the user. This process of seam creation is integrated with the seam-creation process. We further describe the detailed process in the next section.

3.3 Seam Creation

Our main idea in the development of automatic seam creation is *divide-and-conquer*. We classify the seams into the following three categories:

- **Intra-pack Seams:** used to seam lines belonging to the same pack.
- **Inter-pack Seams:** used to seam lines belonging to two different packs. We further categorize the inter-pack seams into *inter-half-cylinder seams* (that join the front and back half cylinders) and *inter-cylinder seams* (that join two cylinders).
- **Inter-layer Seams:** consist of the T-seams and Y-seams (see Section 3.1.1).

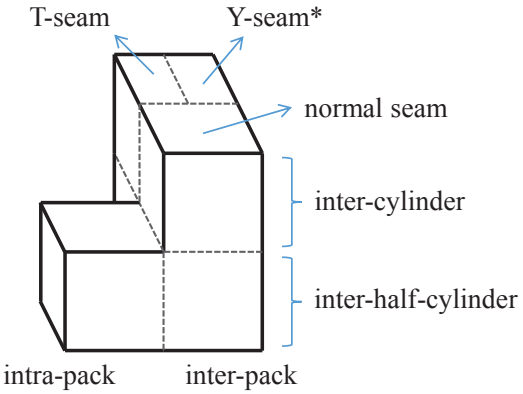


Figure 3.5: Complete classification of the seam lines. * indicates the seam lines that need to be explicitly secondary-tagged by the user.

TAGCON uses different algorithms to seam the above three categories. In summary, Figure 3.5 shows the complete TAGCON classification of the seam lines, where only the layered-seam lines need explicit secondary tagging.

3.3.1 Creating the Intra-pack Seams

TAGCON assumes that the user composes the pack such that every matching seam line pair is the closest neighbor to each other (i.e., no other line comes between them.). Then, the intra-pack seams can be identified by the following simple algorithm:

For all lines l_i in the same pack,

1. Find the closest line among the ones that comes within the field-of-view(FOV) of l_i.[4]

[4]The FOV (field of view) is defined from the center of the line, as shown in Figure 3.6, where the spread angle is a user-controlled parameter.

2. If there is no such line within the FOV, then l_i is categorized as an inter-pack seam line.

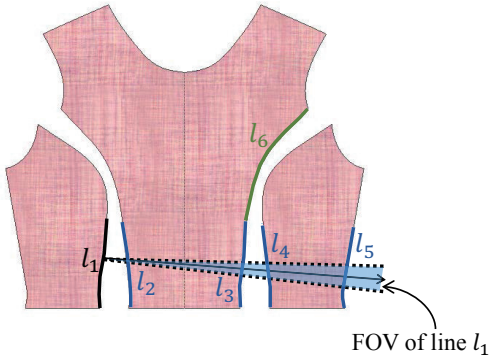


Figure 3.6: The lines l_2, l_3, l_4, and l_5 are in the FOV of line l_1. The closest line l_2 is regarded as the matching seam-line pair for l_1.

3.3.2 Creating the Inter-pack Seams

Identifying the inter-pack seams calls for a more complex algorithm than identifying the intra-pack seams. To estimate the probability of two candidate lines in 3D being inter-pack seamed, TAGCON uses the following criteria:

- M_{ld}: The distance should be short.
- M_{lo}: The angle between the planar normal of one line (i.e., the normal within the common plane) and that of the corresponding line should be small.
- M_{ll}: The length should be similar.
- M_{lc}: The curvature should be similar.

– M_{sa}: The lines should be *seamable*. If the two lines belong to non-adjacent cylinders (e.g., sleeve and pants), those lines are *inherently unseamable*. There are cases in which two lines belonging to two adjacent cylinders should never be seamed. For example the inner lines of the pants should not be seamed with outer lines. Such cases will be called *conventionally unseamable*. Conventional unseamablility is garment specific knowledge, and can differ for each garment type.

Based on the above criteria, we define the objective function $M(i,j)$ for finding the matching inter-pack seam lines as

$$M(i,j) = M_{ld}(i,j)M_{lo}(i,j)M_{ll}(i,j)M_{lc}(i,j)M_{sa}(i,j), \tag{3.1}$$

where

$$\begin{aligned}
M_{ld}(i,j) &= \frac{d_{max} - ||\mathbf{p}_i - \mathbf{p}_j||}{d_{max}}, && (3.2)\\
M_{lo}(i,j) &= \frac{1 - \mathbf{n}_i \cdot \mathbf{n}_j}{2}, && (3.3)\\
M_{ll}(i,j) &= \frac{L_{diff} - |L_i - L_j|}{L_{diff}}, && (3.4)\\
M_{lc}(i,j) &= \frac{\kappa_{diff} - |\kappa_i - \kappa_j|}{\kappa_{diff}}, && (3.5)\\
M_{sa}(i,j) &= 0 \text{ or } 1. && (3.6)
\end{aligned}$$

Equation 3.2 represents the proximity, where d_{max} is the threshold distance (a controllable parameter), and $\mathbf{p}_i$ and $\mathbf{p}_j$ are the positions of lines i and j. Equation 3.3 represents the orientation match, where $\mathbf{n}_i$ and $\mathbf{n}_j$ are the pla-

nar normals of lines i and j, respectively. Equation 3.4 represents the length match, where L_{diff} is the threshold length difference (a controllable parameter), and L_i and L_j are the lengths of lines i and j, respectively. Equation 3.5 represents the curvature match, where κ_{diff} is the threshold curvature difference (a controllable parameter), and κ_i and κ_j are the curvatures of lines i and j, respectively. Equation 3.6 represents the seamability, which takes a value of either zero or one based on the line pair's inherent and conventional unseamability.

When the intra-pack seams are completed, then all the remaining seam lines have to be inter-pack or inter-layer seamed. Because all the inter-layer seams are already tagged or identified (see Section 3.1.1), only the inter-pack seams are not yet known whether it should be inter-half-cylinder or inter-cylinder seamed. TAGCON performs that classification based on the pre-established garment construction theory. Then, TAGCON first creates the inter-cylinder seams, after which it creates the inter-half-cylinder seams according to the following two procedures, respectively.

Creating the Inter-Cylinder Seams The matching inter-cylinder seam lines are identified in the following procedure.

1. For a particular line i, calculate the objective function value $M(i, j)$ for all the adjacent lines j in the other cylinder to find the *best individual match* for i.

2. Find the *best ring match* by rotating the distal cylinder about the axis

and calculating the circumferential summation of $M(i, j)$. From the best individual match, only three clockwise and counterclockwise clicks were sufficient to find the best ring match for the cases shown in this paper.[5]

Creating the Inter-Half-Cylinder Seams The matching inter-half-cylinder seam lines are identified in the following procedure.

1. For a particular line i, calculate the objective function value $M(i, j)$ for all the adjacent lines j in the other half of cylinder to find the *best individual match* for i.

2. Sort the line pairs by $M(i, j)$ in descending order.

3. Create seams of the line pairs in the order of the sorted list. When a line in a line pair is already included in created seams, that line does not participate in further creating processes.

Because the two half cylinders are aligned in 3D, the identification of the matching inter-half-cylinder seam lines does not require an iterative operation, as with the creation of the inter-cylinder seam algorithm.

[5]When two cylinders are joined (e.g., the sleeve to the bodice at the armhole), we call it a ring match. In such a case, with the bodice being fixed, the sleeve needs to be rotated to find the optimal match. Here, a click refers to the clockwise or counter clockwise rotation of the sleeve by one seam-line with respect to the armhole.

3.3.3 Creating the Inter-layer Seams

At this point, the panels are layered, and the lines subject to the T-seams and Y-seams are in their 3D position. Because these lines are conspicuous from other types of lines, the creation of the seams between them can be done by looking at the value of $M(i, j)$.

3.3.4 Seam-creation Process

We summarize the seam-line classification and seam-creation process in sequence as below.

1. Classify all the contour lines as normal seam-lines and all interior lines as inter-layer seam-lines.
2. Find the inter-layer seam-line matches.
3. Find the intra-pack seam-line matches.
4. Classify the remaining seam-lines as inter-cylinder and inter-half-cylinder seam-lines.
5. Classify the inter-cylinder seam lines as non seam-lines if there were no corresponding lines in the adjacent cylinder panels.
6. Find the inter-cylinder seam-line matches.
7. Find the inter-half-cylinder seam-line matches.
8. Create the seams of found matches.

9. Delete the seam if the seam sweep penetrated the body.

3.4 Experiments

To test TAGCON, we obtained from clothing experts 30 sample garments of various types (including a blouse, one-piece, skirt, pants, T-shirt, hooded top, and outdoor clothing).[6] First, we performed the tagging-and-packing process for all the samples, which took about 30 s for the simplest garment and about 3 min for the most complex garment. We averaged about 2 min per garment. Then, we created the 3D garment automatically using the proposed method, which took less than 5 s for each of the sample garments. When the clothing experts created the above samples without TAGCON, the simplest garment took about 5 min and the most complex one took about 1.2 h. Figures 3.10 and 3.11 show that our method works for various kinds of garments. The proposed method successfully created 29 garments out of 30 test samples. TAGCON provides the visual cue about the certainty of the identified matching seams, as shown in Figure 3.1. The seam sweep is shown in blue when it is very certain, and in red when it is very uncertain. If there exists a problem, such a visual cue is helpful to interactively locate the problematic place so that the seam can be processed (delete and re-identified) manually.

We performed draping simulations of the resultant garments on the virtual human model. We used a physically based clothing simulator that was devel-

[6]Some sample garments were created using captured images of real garment panels. The capturing environment is shown in Figure 3.7.

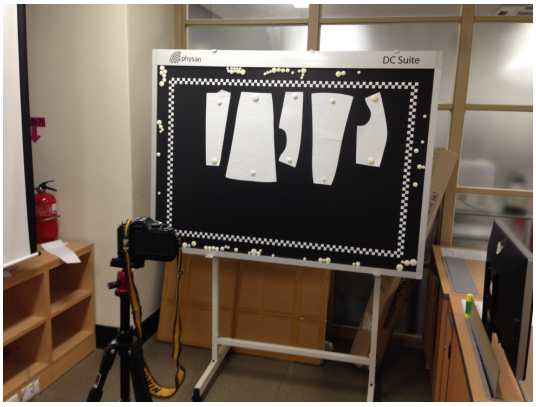


Figure 3.7: The capturing environment that we designed to obtain panel images.

oped to represent various kinds of garments. The detailed methods used in the simulator are described in Chapter 5.

As shown in Figure 3.8, some garments obtained in the produced results are composed of panels having unique shapes. It is difficult to accommodate those unconventional cases using the machine-learning-based method. These examples demonstrate that TAGCON is flexible in covering various types of garments.

TAGCON is based on the cylindrical abstraction, and assumes that each panel can be assigned to a cylinder. If a panel cannot belong to a cylinder, it is difficult to locate the panel. For instance, in the top front panel (which bears the character 'A') of Figure 3.9, the sleeve panels are not separated, so it cannot be accommodated to a cylinder. This kind of unconventional garment (one from among 30 sample garments) cannot be processed using TAGGON, but this inability is deterministic and predictable.

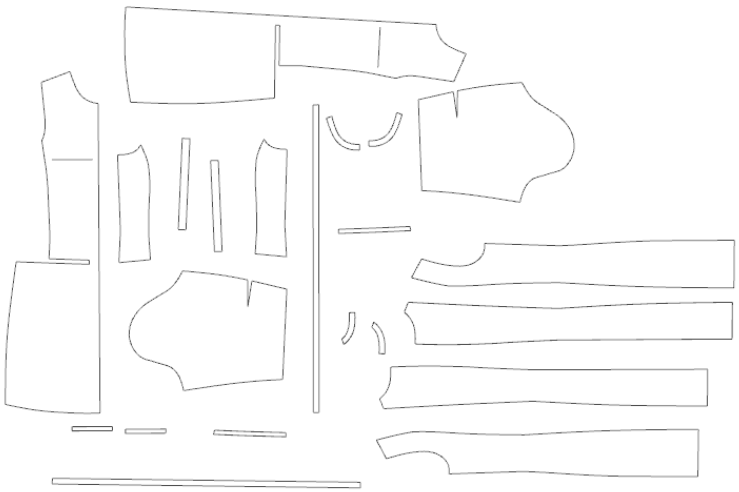


Figure 3.8: Panels of a one-piece that can be created using the proposed method.

Figure 3.9: A garment that cannot be automatically created with TAGCON.

3.5 Conclusion

In this study, we presented a new technique called TAGCON, which constructs the 3D virtual garment from the given tagged and packed panels. Tagging and packing should be done from the user, and involves the simple labeling and 2D

manipulation of the panels, but does not involve any 3D manipulation. Then, TAGCON constructs the garment automatically via the proposed algorithms (1) to position the panels at sensible locations around the body, and (2) to find the matching seam lines and create the seam.

TAGCON takes a deterministic approach and allows inter-layer seams (T-seams and Y-seams), which contributes to increasing the complexity of the garment that the method can accommodate, as demonstrated in Figure 3.10 and Figure 3.11.

Figure 3.10: Sample garments that are automatically constructed using the proposed method.

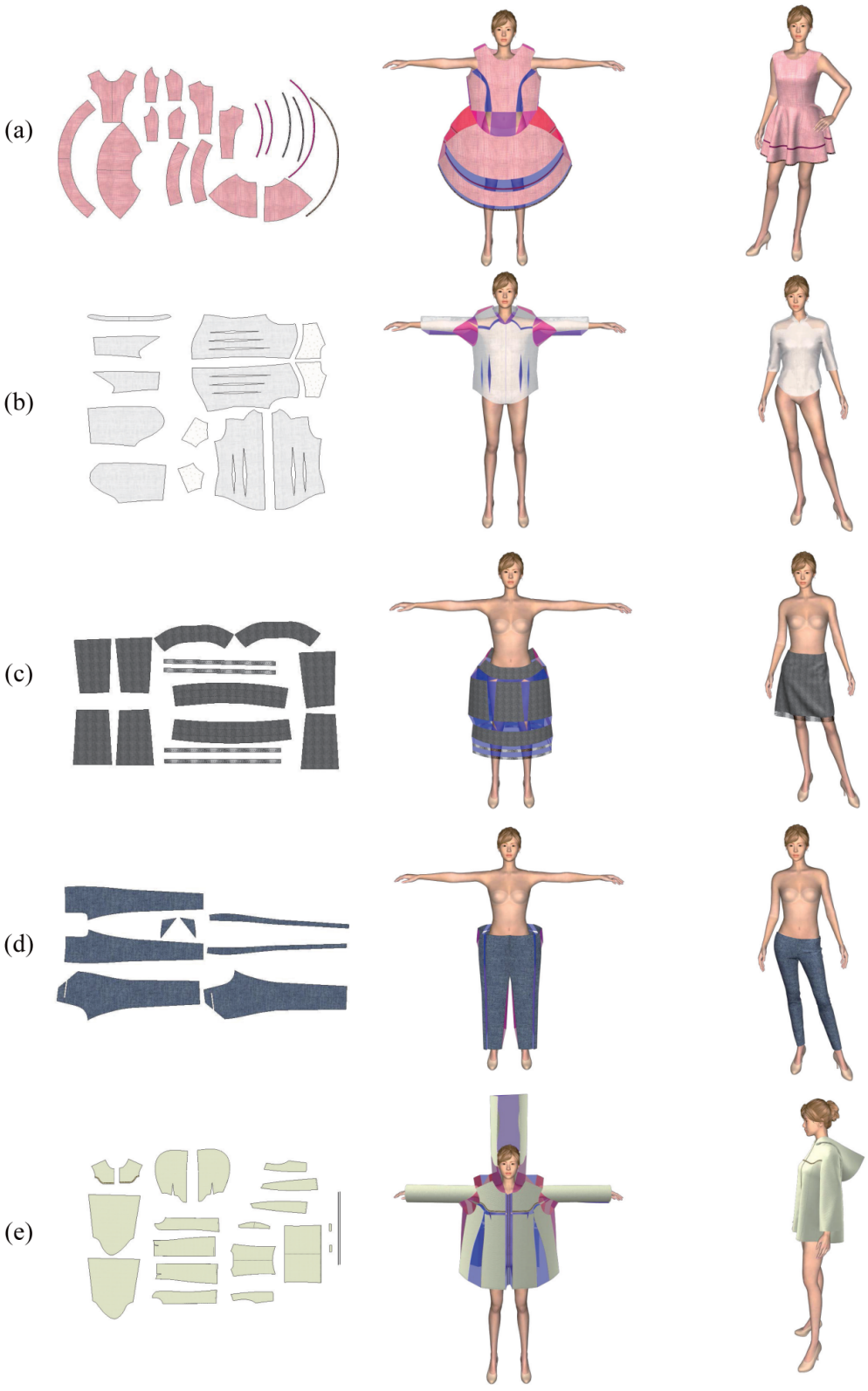


Figure 3.11: Five sample garments: (a) one-piece, (b) blouse, (c) skirt, (d) pants, and (e) hooded top with the input panels, automatically created seams, and draped.

Chapter 4

Synced Garment Editing

4.1 Introduction to Synced Garment Editing

The garment editing process is an iterative trial-and-error process that cannot be easily done. Umetani et al. [68] described this job as solving a brain teaser, which requires many user interactions. If a garment must be re-simulated from the first frame for every single pattern editing, the editing time will increase as long as the user gives up the use of that tool. Therefore, a necessary condition of a practical editing tool is that the user can immediately confirm changes in the 3D garment that result from 2D pattern editing. This means that the 2D garment pattern and 3D simulation result must be synced during the editing process. We call the tool that satisfies the above condition as the *Synced Garment Editing* tool.

To develop a synced garment editing tool, there are three important con-

ditions that must be satisfied.

1. Maintain the topology between the original garment and the edited garment.

2. The edited garment must not penetrate the body.

3. All operations must be processed instantly.

As described in Section 2.1, studies were performed to satisfy the above conditions. We classify those studies into two categories: (1) *geometric approaches* and *(2) sensitivity analysis*.

(a) (b)

Figure 4.1: An example of synced garment editing. The editing of the pattern in (a) is immediately represented at the 3D view in (b).

4.2 Geometric Approaches vs. Sensitivity Analysis

In a synced garment editing tool, if the original 2D pattern P_o is modified to a new pattern P_n, a similar geometric change between P_o and P_n must occur between the original 3D garment G_o and the new garment G_n, with maintaining the above three conditions. Therefore, the main problem confirms the inference of the positions of vertices in G_n from G_o based on the changes in P_o and P_n. Volino et al. [70] and Meng et al. [53] used a geometric method to resolve that problem.

Geometrical Approach Volino et al. [70] did not change the 3D shape of the garment on the editing stage when a simple resizing operation was applied to a pattern. In this case, they only changed the mesh structure in the 2D pattern, and then delegated the simulator to update the 3D garment shape. Their system changed the 3D garment shape directly only after significant modification was applied to a pattern. In this case, they found positions of vertices in G_n by the simple linear interpolation of positions of vertices in G_o using the relation between P_o and P_n. When a vertex in P_n is not in the contour of P_o they used the extrapolation of neighboring vertices.

Meng et al. [53] also used an interpolation method to obtain the positions of vertices, but this method did not use neighboring vertices; instead, it used feature points for interpolation. Their method is beneficial to preserving stylish elements because of feature-based interpolation.

The above two methods did not consider body-garment penetration on the

editing stage, but delegated the physical cloth simulator to handle the penetration problem, as shown in 4.2.

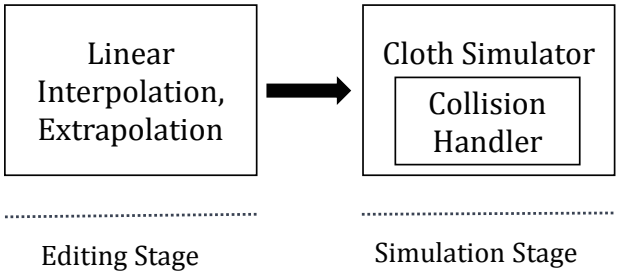


Figure 4.2: The process of geometric approach editing. This methods do not resolve the penetration problem on the editing stage.

Sensitivity Analysis Umetani et al. [68] proposed a novel method for fast synced garment editing. They performed a sensitivity analysis to predict the 3D garment shape from cached data. Further, they employed re-meshing using positive coordinates, progressive refinement, and the modified St. Venant-Kirchhoff cloth model to create a stable and fast system. To prevent collisions between the garment and body, they used a virtual spring method that is based on the distance field in Frisken et al. [27]. They represented the friction using the moving-anchor spring method proposed by Hasegawa et al. [34]. This method showed a remarkable performance once the user can confirm a 3D garment shape interactively during rapid pattern editing such as when using a mouse-drag operation.

4.3 Trouble Free Synced Garment Editing

Our goal of developing a synced garment-editing tool is aimed at developing a useful digital clothing solution that includes various techniques such as convenient garment modeling, realistic fabric rendering, and accurate avatar generation, and stable physics simulation. For this reasons, there are limited resources that can be used for garment editing. We consider that the sensitivity analysis method [68] shows an excellent performance, but is too complex for integration with digital clothing systems that have such a limited resource environment. Therefore, we developed a synced garment-editing tool that is based on the geometric method, which is relatively easy to integrate.

When the user modifies the contour of patterns by moving lines or points, we triangulate the inner contour area of a new 2D pattern P_n using Delaunay triangulations. Then we update the positions of vertices in G_n using linear interpolation and extrapolation , which is similar to the method used by Volino et al. [70]. The proposed method updates the vertex position of G_n using the equations below,

$$\mathbf{v}_i^n = \lambda_1 \mathbf{v}^o{}_1 + \lambda_2 \mathbf{v}^o{}_2 + \lambda_3 \mathbf{v}^o{}_3, \tag{4.1}$$

$$\mathbf{p}_i^n = \lambda_1 \mathbf{p}^o{}_1 + \lambda_2 \mathbf{p}^o{}_2 + \lambda_3 \mathbf{p}^o{}_3, \tag{4.2}$$

where $\mathbf{v}_i^n$ is a vertex position in P_n, $\mathbf{v}_1^o$, $\mathbf{v}_2^o$, $\mathbf{v}_3^o$ are positions of vertices in P_o and λ_1, λ_2, and λ_3 indicate barycentric coordinate. $\mathbf{p}_i^n$ is the vertex position of the G_n which is matched with $\mathbf{v}^i$ and $\mathbf{p}_1^o$, $\mathbf{p}_2^o$, $\mathbf{p}_3^o$ are the positions of vertices

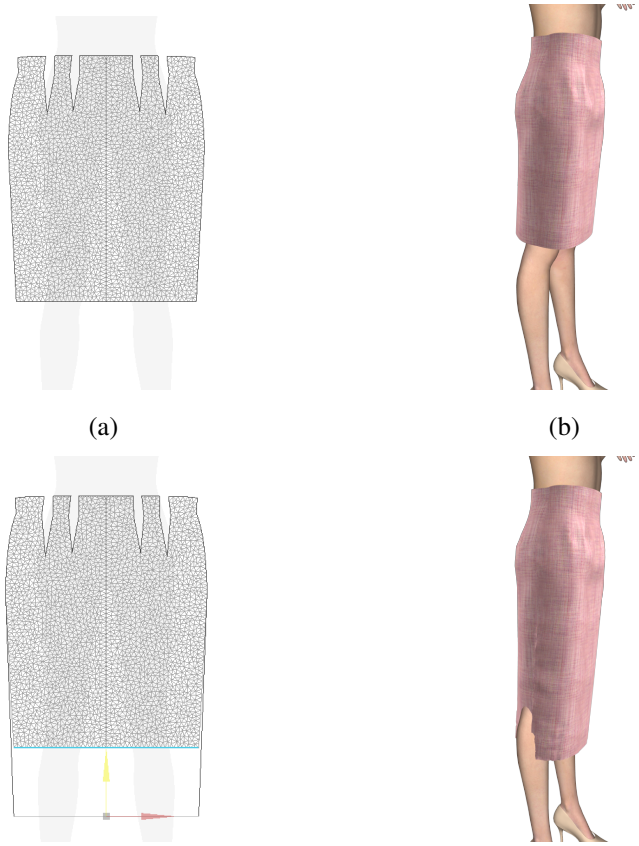
that match with $\mathbf{v}_1^o$, $\mathbf{v}_2^o$, $\mathbf{v}_3^o$ in G_o.

Now, we can obtain the new positions of vertices. The remaining problem is to detect and handle collisions between the garment and the body. General cloth simulators, including collision detectors and handlers, are good at preventing penetration when a pre-simulated result is in a collision-free state. However, when significant penetration occurs, it is not easy to solve that situation using those simulators. At times, we cannot resolve such penetration permanently or spend so much time.

When the user introduced instantaneous significant changes in the pattern, as shown in Figure 4.3(c), the simulator could not resolve that problem, as shown in Figure 4.3(d). Previous synced garment editing methods did not consider resolving body-garment penetration that was caused by such rapid changes. Based on our experiences, this situation can often occur depending on the inclination of a user. Therefore, we conclude that this problem must be solved during the editing stage, as shown in Figure 4.4.

Our penetration-handing method is similar to that of Meng et al. [52]. We detect whether there are vertices of G_n inside the body meshes. If vertices are located inside the body, we repel those vertices to the outside of the body. To reduce the detection time, we propose the spatial partitioning method, as shown in Figure 4.5. First, we divide the body into n-tags, and we then divide each tagged body into several parts based on the height of the cylinder. The detailed penetration detection method is described in Algorithm 1.

The first two for loop statements in Algorithm 1 can be precomputed be-



(a) (b)

(c) (d)

Figure 4.3: When the original pattern (a) was changed to the new pattern (c), the new garment (d) created from the original garment (b) penetrated the body. Immediate changes to patterns can generate a serious penetration problem.

fore editing or simulation. The last for loop statement is only performed during the editing stage. Because we developed a trouble-free situation based on the above process, we simulate G_n using a developed cloth simulator. Figure 4.6 shows that our method resolves the problem caused by rapid pattern editing.

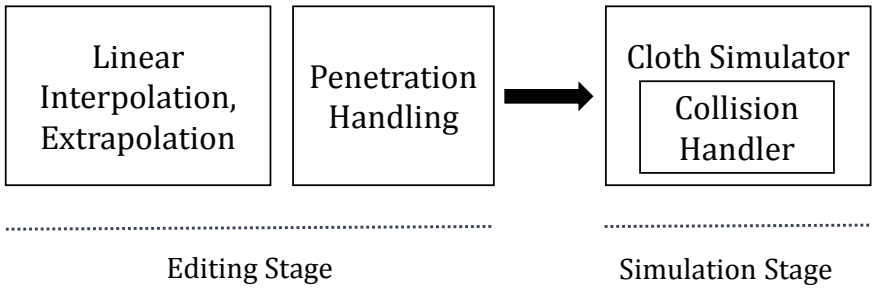


Figure 4.4: The process of proposed synced garment editing. Performing penetration handling is added on the editing stage, that was not performed on the editing stage in previous methods.

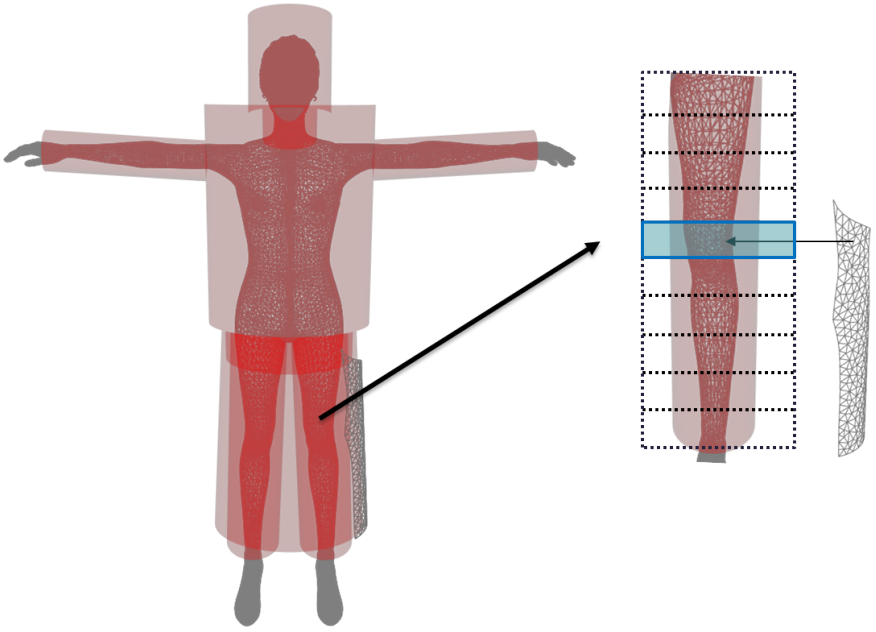


Figure 4.5: We partitioned the body into several parts to reduce the detection time.

Algorithm 1 Penetration detection.

for $n = 0$ to *number of bones* N_B **do**
    Create a line segment representing each bone B_n.
    Divide the segment of B_n as constant intervals I_k^n (k = 1,2,...,K_I^n).
end for
for $i = 0$ to *number of body triangles* **do**
    Find the nearest bone B_n of each body triangle t_i.
    Project the center of t_i to the line segment of B_n.
    Find that projected point belongs to which interval I_k^n.
    Save the indices of vertices in t_i to separate memory space (B_n, I_k^n).
end for
for $i = 0$ to *number of garment vertices* **do**
    Find the nearest bone B_n of each garment vertex v_i.
    Project v_i to the line segment of B_n.
    Find that projected v_i belongs to which interval I_k^n.
    Detect whether v_i is inside of triangles in the memory (B_n, I_k^n).
end for

(a) (b) (c)

Figure 4.6: The proposed methods successfully resolve the problem of garment editing: (a) penetration problem, (b) result after penetration handling, (c) final simulated result.

Chapter 5

Physically Based Non-Elastic Clothing Simulation

The effort to characterize the non-elastic components is not completely new; it has been an important subject in textile science. Many experiments have been performed to analyze the non-elastic behavior of cloth, including creep and relaxation. This work is an attempt to model the non-elastic component by observing the creep phenomenon. Although there has been a lot of study on creep in textile science, we note that the previous findings cannot be directly used for simulator development. The general tendency of cloth deformation does not provide the full information for the specific setup being simulated. In some cases, it calls for non-trivial generalization. This work makes necessary generalization of the previous findings in the textile science so that the results can be used in the context of simulator development. For exam-

ple, from the creep model that works for a constant load, we derive a general force–displacement relationship that works for arbitrary occurrences of loading and unloading.

There have been a number of models to represent the non-elastic components based on friction models, such as Dahl [24] and Bliman and Sorin [11]. The models that have been introduced so far (e.g., Lahey [48], Ngo-Ngoc and Boivin [59], Miguel [55]) can represent the hysteresis of fabric, but do not consider the plastic deformation that occurs over long-term exertion of external forces. This work incorporate the time-varying non-elasticity as well as plasticity to the derivation in such a way that the user can control the non-elastic nature by adjusting a few intuitive parameters. We show that the proposed general model (i.e., the model that combines elastic and non-elastic models) can produce plausible results in response to various user inputs.

5.1 Classification of Deformation

To set up the background for the development of the proposed non-elastic model, this thesis makes an analytical review of creep along with its time-dependent nature. When the stretching load is applied/released, we note that some part of deformation/recovery does not occur instantly, and some part of deformation is never recovered. In modeling the non-elasticity of cloth stretch, this work finds that decomposition of stretch deformation based on the time taken for the stretch/recovery is essential, since modeling the individual com-

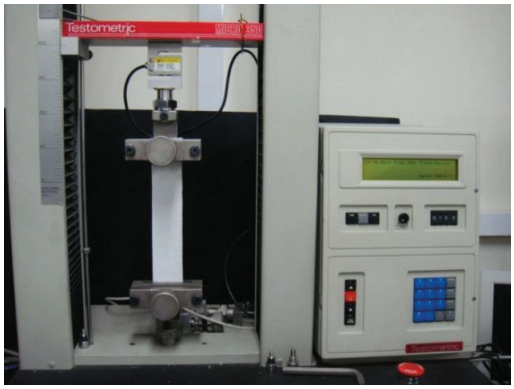


Figure 5.1: Our equipment for measuring stretch deformation.

ponents (immediate elastic, viscoelastic, and plastic) is more tractable to formulate than the combined deformation.

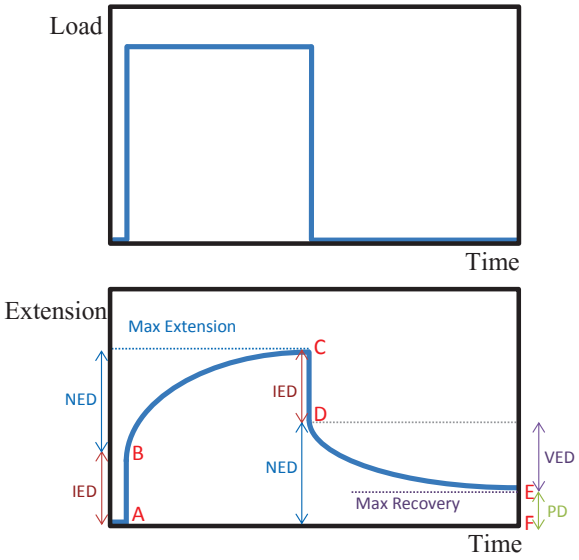


Figure 5.2: Load-time and extension-time plot during a constant force creep.

For the analysis of stretch deformation, we need to measure the load-time and extension-time relations. For the measurements, we used the equipment shown in Figure 5.1, in which the stretching load can be controlled. When

the applied stretching force is kept constant (we will call this setup a *constant force stretch*), the extension of the fabric increases over time as shown in the curve segment B-C of Figure 5.2. Such a phenomenon is called creep.

Making observation of a *constant force creep* (i.e., the creep occurring in a constant force stretch), we can classify cloth deformation into the following three components such that at any moment the total amount of deformation is the sum of those three components.

- **Immediate Elastic Deformation (IED):** This component refers to the deformation and recovery that occurs immediately after the load application and removal. In Figure 5.2, the vertical extension at the beginning (i.e., the curve segment A-B) and the vertical shortening at the force removal (the curve segment C-D) represent the immediate[1] elastic deformation. We assume the lengths of A-B and C-D are the same.

- **Viscoelastic Deformation (VED):** When a constant force stretch starts, after the initial IED of A-B, an additional deformation is in progress over time, as shown in the curve segment B-C of Figure 5.2. For the case of the recovering part, after the initial IED recovery of C-D, further recovery occurs over time as shown in the curve segment D-E of Figure 5.2. Note that, among the total extension occurred during B-C, some is recovered (D-E) but some is not (E-F). The deformation recov-

[1]Rigorously speaking, elastic deformation does not occur immediately but occurs over a duration. So it is more correct to say IED is the deformation that occurs within an epsilon duration after the load application/removal. See Appendix A for more details.

ered (D-E) when sufficient time is given for the recovery is called the VED.

– **Permanent Deformation (PD):** Among the total extension occurred during a stretch deformation, the component that is never recovered (e.g., the amount E-F of Figure 5.2) is called the PD. Note that for the curve segment B-C, VED and PD are mixed.

The creep occurred over B-C, which could not be decomposed into VED and PD during the *extension* phase, can be decomposed into VED and PD by observing the *recovery* phase. For the development of a non-elastic model, the decomposition of the extension into VED and PD must be possible without observing the recovery phase. In fact, the decomposition must be possible when the release of the load started at any point along B-C, or even after C. Development of a non-elastic model having such properties is the goal of the next section.

5.2 Modeling Non-Elastic Deformations

Our modeling of VED and PD starts from reviewing the mass-spring system. In a mass-spring system, the applied force f and the length x of the spring is related by

$$f = -K(x - L) \tag{5.1}$$

where the capital lettered quantities L and K are the constants representing the *rest length* and the stiffness, respectively, of the spring. The lower-case letters f and x are variables representing the applied force and the current spring length, respectively. However we note that this model, whose force versus deformation plot is shown in Figure 5.3 (a), is valid only for an ideal Hookean spring, thus not valid for modeling the stretch of cloth which is non-linear and non-elastic. To represent the non-linearity, people use

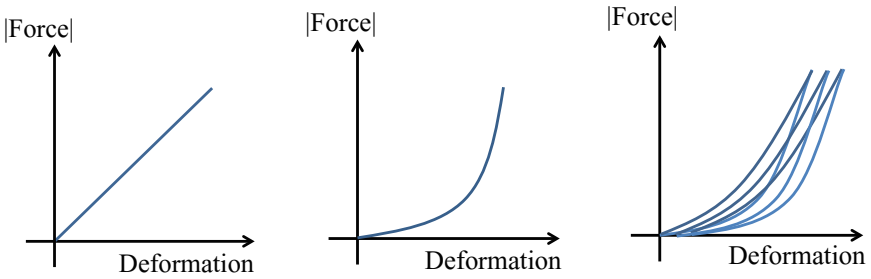
$$f = -k(x - L) \tag{5.2}$$

instead of Equation 5.1, in which k is a variable of x. The above equation can represent the non-linearity of cloth as shown in Figure 5.3 (b), but still cannot represent the non-elasticity. To further represent non-elasticity, the equation we finally use in this work is

$$F = -k(x - l) \tag{5.3}$$

in which l is another variable representing the *unloaded length*.

The difference between the unloaded length and the rest length needs to be highlighted to avoid confusions in the subsequent discussions. The rest length is the length before any deformation is applied to cloth, and is denoted with the constant L. Completely elastic cloth recovers to the rest length if the applied force is removed. On the other hand, non-elastic material does not recover to the rest length, but the length continuously evolves during the load application and removal. That evolving length is called the unloaded length



(a) (b) (c)

Figure 5.3: Force-deformation plots for (a) Equation 5.1, (b) Equation 5.2, (c) Equation 5.3. (c) shows the force-deformation relationship over multiple cycles of loading and unloading.

in this work and is denoted with the variable l. By using variables for both stiffness and unloaded length, the model can represent both non-linearity and non-elasticity as shown in Figure 5.3 (c).

5.2.1 Development of the Non-Elastic Model

We can decompose the unloaded length l as

$$l = L + l_{nonelastic}, \tag{5.4}$$

where L is the rest length of cloth, and $l_{nonelastic}$ is the additional strain occurred due to VED and PD. In the development of the *general model* (i.e., the model that includes IED, VED, and PD altogether), we make the assumption (which we call the *recovery to the unloaded length (RUL) assumption*) that the spring tries to recover to the varying unloaded length $L + l_{nonelastic}$ rather than the constant rest length L. This model, which is apparently elastic (since

the spring recovers to $L + l_{nonelastic}$), can emulate non-elasticity by supplying varying values of $l_{nonelastic}$ over time. For the development of a general deformation model based on the RUL assumption, an essential job is to find out the function that can supply the value of $l_{nonelastic}$ at any moment based on the previous deformation history. This job can be made tractable if we decompose l into three components: IED, VED, and PD.

Decomposing those deformation components from a typical measurement is not trivial. When a controlled setup (such as the constant force creep) is used, IED can be separated from the rest of the deformation by observing the strain that immediately occurs after the force application. However, even in such a controlled setup, it is challenging to decompose the remainder into VED and PD. Our method for decomposing VED and PD will be presented a few paragraphs later.

Creep is quite relevant to our development of the non-elasticity model. Often, creep is approximated with the Kelvin model:

$$\frac{d\varepsilon(t)}{dt} = \lambda(\frac{\sigma(t)}{E} - \varepsilon(t)), \tag{5.5}$$

where $\varepsilon(t)$ is the total creep at time t, $\sigma(t)$ is the stress at time, E is the modulus of elasticity, and λ is the retardation coefficient that is related to the duration taken until the creep is saturated. For the special case when the stress is constant ($\sigma(t) \equiv \sigma_0$), the solution of the above differential equation can be

given with the closed form

$$\varepsilon(t) = \frac{\sigma_0}{E}(1 - e^{-\lambda t}). \tag{5.6}$$

Plotting of this equation tells us that the total creep asymptotically approaches to $\frac{\sigma_0}{E}$, which allows us to predict the *maximum non-elastic deformation*, i.e., the amount of VED + PD after sufficient duration of stretch under the constant stress σ_0.

Although Equation 5.6 is widely used in the textile science, it cannot be used for the simulator development since the simulator should operate for varying stress. In this work, therefore, we use the differential form given in Equation 5.5 in which stress is a variable. However, we can get an insight from Equation 5.6 that $\frac{\sigma_0}{E}$ should represent the maximum non-elastic deformation. In fact, we define a new variable ε_{sat} as $\varepsilon_{sat}(t) = \frac{\sigma(t)}{E}$. According to Equation 5.6, ε_{sat} represents the maximum non-elastic deformation when σ is constantly applied for a sufficient duration. However, we note that its meaning can be extended to the varying stress situation in general. Although the stress varies, its effect to non-elastic deformation is fully accounted for, since it is constantly integrated.

Unfortunately, even Equation 5.5 does not provide satisfactory solution to the problem we attempt to solve in this work, since it gives only the total creep (VED + PD), but it does not give the decomposed values of VED and PD. Such decomposition is essential for modeling the recovering process when

the applied force is removed.

To solve the above problem, we propose a modified Kelvin creep model. In this work, the total creep is expressed as a sum $\varepsilon(t) = \varepsilon_v(t) + \varepsilon_p(t)$, where $\varepsilon_v(t)$ is the VED and $\varepsilon_p(t)$ is the PD.

Observations of various creep measurements tell us that the unloading part of the creep is similar to the loading part except that the sign of gradient is opposite and the saturation value and speed can be different. Based on this observation, we model the VED with the Kelvin model, although the loading and unloading part can have different parameter values.

Another simplification we make is that, in the loading, VED and PD are proportional, that is, their ratio $\varepsilon_v(t) : \varepsilon_p(t) = (1-\alpha) : \alpha$ is invariant throughout the loading period. This is a wild assumption compared to Nikolic et al. [60], who studied the ratio of VED and PD in various fabrics by performing an extensive amount of measurements. According to Nikolic et al. [60], in some fabrics the ratio is maintained almost the same but in general the ratio can vary over time. Although this work currently investigates only a special case of Nikolic et al. [60], we note that the findings in Nikolic et al. [60] can be incorporated into our framework in the future if the ratio-plot (over time) is provided.

With the above assumptions, we can model VED and PD such that the resultant models cover all the considerations made in this section by formulating the time derivatives of the VED ε_v and the PD ε_p as

$$\frac{d\varepsilon_v}{dt} = \begin{cases} \lambda_l\{(1-\alpha)\varepsilon_{sat} - \varepsilon_v\}, & \text{if } \varepsilon_v < (1-\alpha)\varepsilon_{sat} \\ \lambda_u\{(1-\alpha)\varepsilon_{sat} - \varepsilon_v\}, & \text{if } \varepsilon_v \geq (1-\alpha)\varepsilon_{sat} \end{cases} \tag{5.7}$$

and

$$\frac{d\varepsilon_p}{dt} = \begin{cases} \lambda_l(\alpha\varepsilon_{sat} - \varepsilon_p), & \text{if } \varepsilon_p < \alpha\varepsilon_{sat} \\ 0, & \text{if } \varepsilon_p \geq \alpha\varepsilon_{sat}, \end{cases} \tag{5.8}$$

where λ_l and λ_u are the rate of relaxation in loading and unloading, respectively, whose physical meaning is the same with Kelvin model.

When simulating a constant force creep with Equations 5.7 and 5.8, on loading, $\varepsilon_v(t)$ and $\varepsilon_p(t)$ both asymptotically approach to the saturation values $(1-\alpha)\varepsilon_{sat}$ and $\alpha\varepsilon_{sat}$. However, on unloading, $\varepsilon_v(t)$ asymtotically decreases and approaches to zero as shown in Figure 5.4, whereas $\varepsilon_p(t)$ maintains its saturation value $\alpha\varepsilon_{sat}$. When $\sigma(t)$ is changed to zero at the start of unloading, ε_{sat} also changes to zero. So $\frac{d\varepsilon_v}{dt} = \lambda_l\{(1-\alpha)\varepsilon_{sat} - \varepsilon_v\}$ changes to $\frac{d\varepsilon_v}{dt} = -\lambda_u\varepsilon_v$, whereas $\frac{d\varepsilon_p}{dt} = 0$.

We note that (1) the values of VED and PD can now be supplied for arbitrary time-varying force applications, and (2) the results generated for the particular cases well match to our expectation. The above was possible by the ratio between VED and PD, and by introducing a new variable ε_{sat} which has been conventionally a constant but is used as a variable in this work for

controlling the amount of VED and PD.

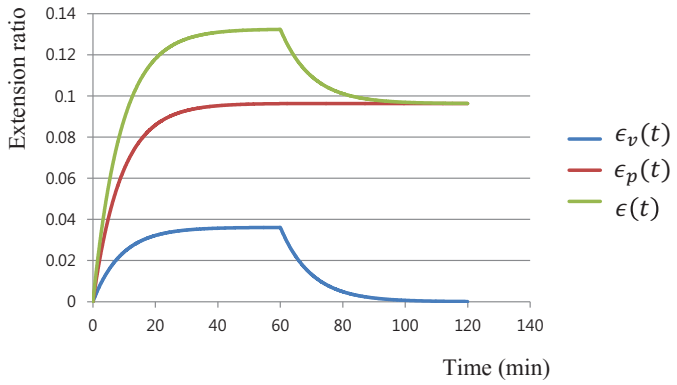


Figure 5.4: The solutions of Equations 5.7 and 5.8.

5.2.2 Parameter Value Determination

With the mathematical model derived in Equations 5.7 and 5.8, the user can control the non-elastic characteristic of the fabric by adjusting the parameter values of λ_l, λ_u, E and α. If there exist measurement data, then, instead of interactively adjusting the parameters, the parameters can be fit to the data. This section describes the method for determining the parameter values in reference to the measurements.

For the data collection, we had each fabric undergo 1-hour long loading followed by 1-hour long unloading, assuming that 1-hour loading/unloading reaches the full saturation/relaxation. (In some kinds of fabrics, the full saturation cannot be reached in one hour. For more details on saturation/relaxation, refer to the Appendix A.)

From the above measurements, α can be calculated by taking the ratio of

the saturated PD (which can be estimated at the end of the unloading) and the saturated VED + PD (at the end of the loading). Then, E can be determined by $E = \frac{\sigma}{\varepsilon_{sat}}$.

With the α and E calculated above, we can now determine λ_l with the least square method referring to the measurements for the loading part. λ_u can be determined similarly referring to the unloading part. Running the least square method for 86,400 frames of loading and unloading took less than 1 s.

5.3 Implementation

This section describes how the proposed non-elastic model can be used to upgrade an existing elastic simulator into a comprehensive one. In this work, as an example, we take the simulator proposed by Baraff and Witkin [4]. However, the upgrading procedure can be similarly applied to other existing elastic simulators.

In an elastic simulator using the implicit time integration based on Equation 5.1 with the constants K and L, for advancing one frame of the simulation, the simulator firstly updates the force, then the velocity, and finally the position. One way (which we took in this work) of incorporating the non-linear and non-elastic component into the elastic simulator is to simply add the following step at the end of the above normal implicit time integration step: namely, updating l of Equation 5.3 to $L + l_{nonelastic}$ and k to the non-constant stiffness value, as summarized in Algorithm 2. In Algorithm 2, the udpate of

Algorithm 2 Non-elastic non-linear cloth simulation.

```
1: for n = 0 to number of frames do
2:     for i = 0 to number of vertices do
3:         update force f_i
4:         for j = 0 to number of vertices do
5:             update stiffness matrix ∂f_i/∂x_j
6:         end for
7:     end for
8:     solve system matrix
9:     for i = 0 to number of vertices do
10:        update velocity v_i
11:        update position x_i
12:    end for
13:    for t = 0 to number of triangles do
14:        update stiffness k_t
15:        update unloaded length l_t
16:    end for
17: end for
```

the stiffness matrix is done by regarding l and k as constants.

For precise simulation, derivative of k and l must be considered for calculating the stiffness matrix. However, it can make the simulator impractically complex. We note that the above approximation did not produce any noticeable artifacts. We attribute this outcome to the fact that the rates of change of k and l are small in normal situations.

Baraff and Witkin [4] represents the stretch energy at vertex $\mathbf{x}$ in a triangular piece consisting of the 3D vertices $\mathbf{x}_i$, $\mathbf{x}_j$, and $\mathbf{x}_k$ (whose 2D material UV coordinates are (u_i, v_i), (u_j, v_j), and (u_k, v_k), respectively as shown in Fig-

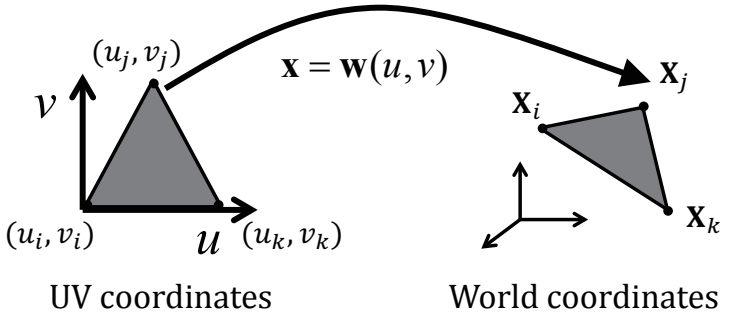


Figure 5.5: **w** is the 2D to 3D mapping function. It maps the UV coordinates (in the material space) to the 3D world coordinates.

ure 5.5) as

$$E_{\mathbf{C}}(\mathbf{x}) = \frac{k}{2}\mathbf{C}(\mathbf{x})^T\mathbf{C}(\mathbf{x}), \tag{5.9}$$

where k is the stiffness constant and

$$\mathbf{C}(\mathbf{x}) = a\begin{pmatrix} ||\mathbf{w}_u(\mathbf{x})|| - b_u \\ ||\mathbf{w}_v(\mathbf{x})|| - b_v \end{pmatrix} \tag{5.10}$$

is a 2D vector that represents how much the cloth is stretched (compared to the rest state) along the u and v directions. Here a is the area of the triangle, and **w** is the mapping from a point in the UV coordinates to its counterpart in the 3D world coordinates. $\mathbf{w_u}$ and $\mathbf{w_v}$ are its derivatives with respect to u and v. As introduced in Baraff and Witkin [4], $\mathbf{w}_u$ and $\mathbf{w}_v$ can be approximated as

$$\begin{pmatrix} \mathbf{w}_u & \mathbf{w}_v \end{pmatrix} = \begin{pmatrix} \Delta\mathbf{x}_1 & \Delta\mathbf{x}_2 \end{pmatrix} \begin{pmatrix} \Delta u_1 & \Delta u_2 \\ \Delta v_1 & \Delta v_2 \end{pmatrix}^{-1}, \tag{5.11}$$

where $\Delta\mathbf{x}_1 = \mathbf{x}_j - \mathbf{x}_i, \Delta\mathbf{x}_2 = \mathbf{x}_k - \mathbf{x}_i,$ and

$$\begin{aligned} \Delta u_1 = u_j - u_i,\ \Delta u_2 = u_k - u_i, \\ \Delta v_1 = v_j - v_i,\ \Delta v_2 = v_k - v_i. \end{aligned} \tag{5.12}$$

In an elastic simulator, $\Delta u_1, \Delta u_2, \Delta v_1, \Delta v_2$ in equation 5.11 are the rest lengths along u and v directions, respectively (with the role of L in Equation 5.1), thus constant over the entire simulation. To apply the proposed non-elastic model, instead of $\Delta u, \Delta v$, for the unloaded length we use $\Delta u^{rest} + \Delta u^{nonelastic}$, $\Delta v^{rest} + \Delta v^{nonelastic}$, respectively. For each frame of the simulation, the non-elastic components $\Delta u^{nonelastic}$ and $\Delta v^{nonelastic}$ are supplied with Equations 5.7 and 5.8.

For the implementing the other parts of simulator, we use the shear deformation model from Baraff and Witkin [4] and the hinge-based bending deformation model from Grinspun et al. [31]. To include non-linearity, we update k using the piece-wise linear model which is similar with Wang et al. [73]. However, the non-elastic components of the shear and bending deformation are not considered in this work.

5.4 Experiments

We implemented the proposed method on an i7 computer using C++ and OpenGL. To experiment the controllability of the proposed framework, we created the setup shown in Figures 5.6 and 5.7.

In Figure 5.6, the four sides of a square-shape fabric were fixed. Then, a cylinder was made to push the fabric for 3 s, as shown in Figure 5.6(a). We performed the push test to three fabrics that have different E values (with all the other parameter values being the same). Small E means the stretch reaches a high saturation value. The simulated results in Figures 5.6(b)–(d) show that effect.

In Figure 5.7, a cylindrical-shape fabric was fitted to the cylinder and the top point of the fabric was fixed to the cylinder. We varied the radius of the cylinder up to 1.5 times of the original radius, as shown in Figure 5.6(a). That radius was maintained for 1 s. Then, the radius was varied back to the original size. We performed this experiment to three fabrics that have different λ_l values (with all the other parameter values being the same). The captured results are shown in Figure 5.6(b)–(d). λ_l represents the rate of non-elastic deformation. With the same duration of stretch, therefore, the fabric with a larger λ_l experiences more non-elastic deformation, and that effect is observable in Figure 5.6(b)–(d).

Another experiment was performed on an one-piece dress to see the effect of changing loading duration. We pinched and stretched the two waist points

Fabric	E	α	λ_l	λ_u
Cotton twill warp	32,600	0.897	0.0886	0.00200
Cotton twill weft	14,100	0.873	0.0275	0.00243
Silk warp	54,000	0.742	0.0572	0.00157
Silk weft	21,600	0.794	0.311	0.00190
Denim warp	25,600	0.667	0.0331	0.0147
Denim weft	42,700	0.728	0.0540	0.00310
Nylon warp	49,600	0.729	0.0594	0.00534
Nylon weft	30,800	0.705	0.0376	0.0598
Polyester warp	80,400	0.674	0.00508	0.00197
Polyester weft	10,700	0.645	0.0227	0.00870

Table 5.1: The summary of the parameter values for the five sample fabrics that were determined from the measurements.

with the clothespins, as shown in Figure 5.8. Figure 5.8(b) and (c) show the results when the stretching was released after 1 s and after 1 hr, respectively. The difference is conspicuous.

The above three experiments show that an user can control the amount of hysteresis effect by adjusting the parameters. In the following experiment, we tested the proposed non-elastic model for reproducing five real world fabrics, namely cotton twill, silk, denim, nylon, and polyester. For the test, we measured the constant force creep for each of those fabrics. The applied constant force was 20 kg, and the loading and unloading duration was 1 hr each. The size of the fabric sample was 8 cm $\times$ 15 cm. Using the method described in the Parameter value determination section, we obtained the parameter values summarized in Table 5.1.

Figure 5.9 compares the measurements with the reproduction according to the proposed non-elastic model. We can observe that it produces a general match. The proposed method is based on the Kelvin model, which is simple to integrate but not necessarily the most accurate. There exists a more accurate (but complex) creep model, namely that of Asayesh and Jeddi [2]. However, that model could not be easily integrated into the cloth simulator. Considering that the measurement data can also introduce inaccuracies, we think the proposed non-elastic model, which works for any occurrences of loading and unloading, is practically viable.

Another experiment was performed in the setup shown in Figure 5.10, in which three coaxial cylinders are stacked vertically. The cloth is attached on the top and bottom (red) cylinders along the circumference. At the rest state, the cloth just fit to the cylinder without any stretch. Firstly, the top cylinder was rotated by $+45^\circ$ (counterclockwise), stayed in that configuration for 1 s, then returned to the home configuration. Then, the bottom cylinder was rotated by $+45^\circ$, stayed in that configuration for 1 s, then returned to the home configuration. Finally, the top and bottom cylinders were simultaneously rotated by $+45^\circ$ and -45°, respectively, stayed in that configuration for 1 s, then were returned to the home configuration. The rate of rotation was $45^\circ/sec$. To dramatize the effect of the non-elastic deformation in a short duration of simulation, we used 10 times the λ_u, λ_l values listed in Table 5.1.

Even though we incorporated the non-elasticity only in the stretch deformation, the simulations exhibit very different deformation characteristics under

the same external force situation. E is smaller in the cotton twill than nylon (Table 5.1), thus we can predict that the cotton twill will have larger non-elastic deformation than the nylon. Figure 5.10 shows that the simulated results agree with the prediction. (The deformation occurring in this experiment is not purely a stretch. Our simulator includes the shear deformation but does not include the non-elastic aspect of shear. Nevertheless, the non-elastic stretch alone produced visually plausible results.)

In all the simulations reported in this work, the amount of extra computation taken in addition to the conventional elastic simulation is negligible, thus we omit the time analysis of it.

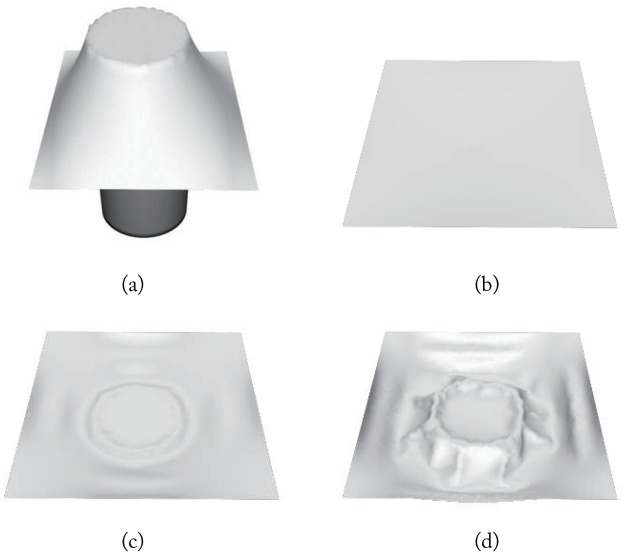


Figure 5.6: The push test. α=0.8, λ_l=0.05, λ_u=0.05 were common to all cases: (a) the setup; (b) E=3000; (c) E=300; (d) E=30.

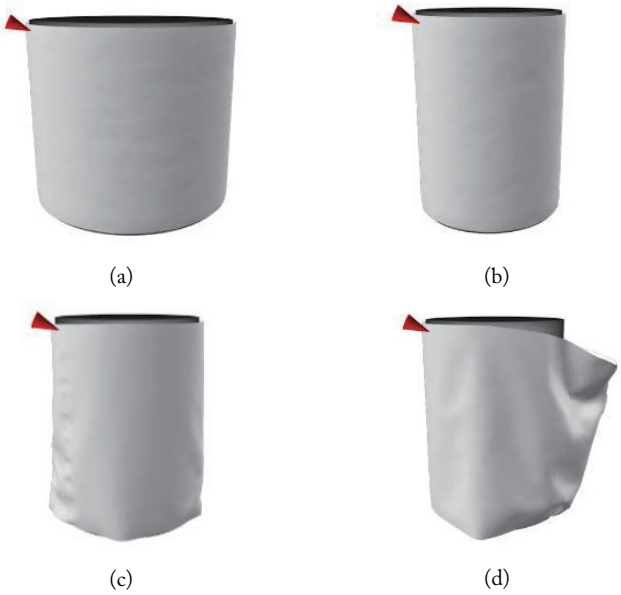


Figure 5.7: The cylinder expansion test. α=0.8, E=300, λ_u=0.05 were common to all cases: (a) the setup at the full expansion; (b) λ_l=0.005; (c) λ_l=0.025; (d) λ_l=0.05.

Figure 5.8: Pinching and stretching the one-piece at the waist of virtual human model. α=0.8, E=300, λ_l=0.05, λ_u=0.05: (a) the setup; (b) when released after 1 s; (c) when released after 1 hr.

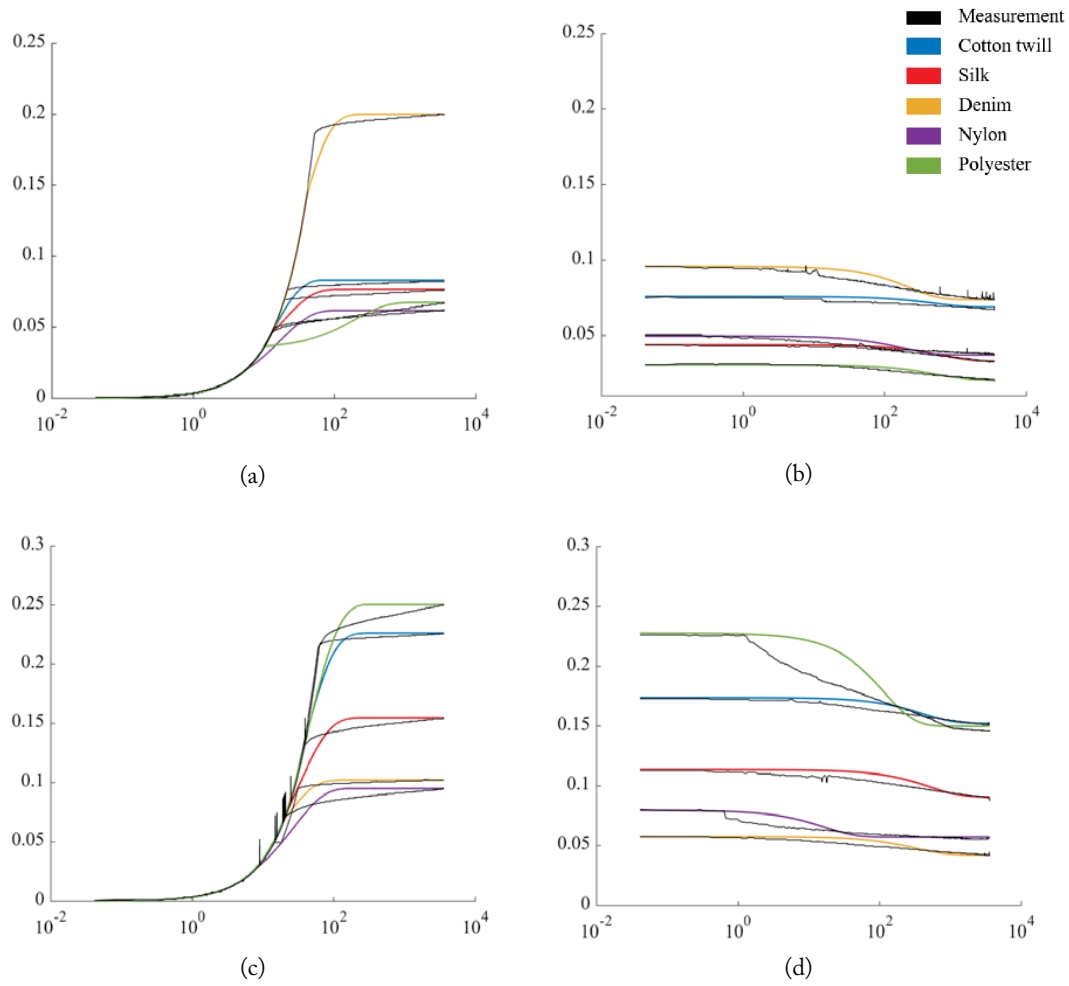


Figure 5.9: Measurements vs. reproduction of the constant force creep of the five sample fabrics. The measurements are shown in black and the reproduced results are shown in colors. The x-axis represents the time (sec) in log-scale and the y-axis represents the ratio of elongation with respect to the rest length. (a) Loading-warp. (b) Unloading-warp. (c) Loading-weft. (d) Unloading-weft.

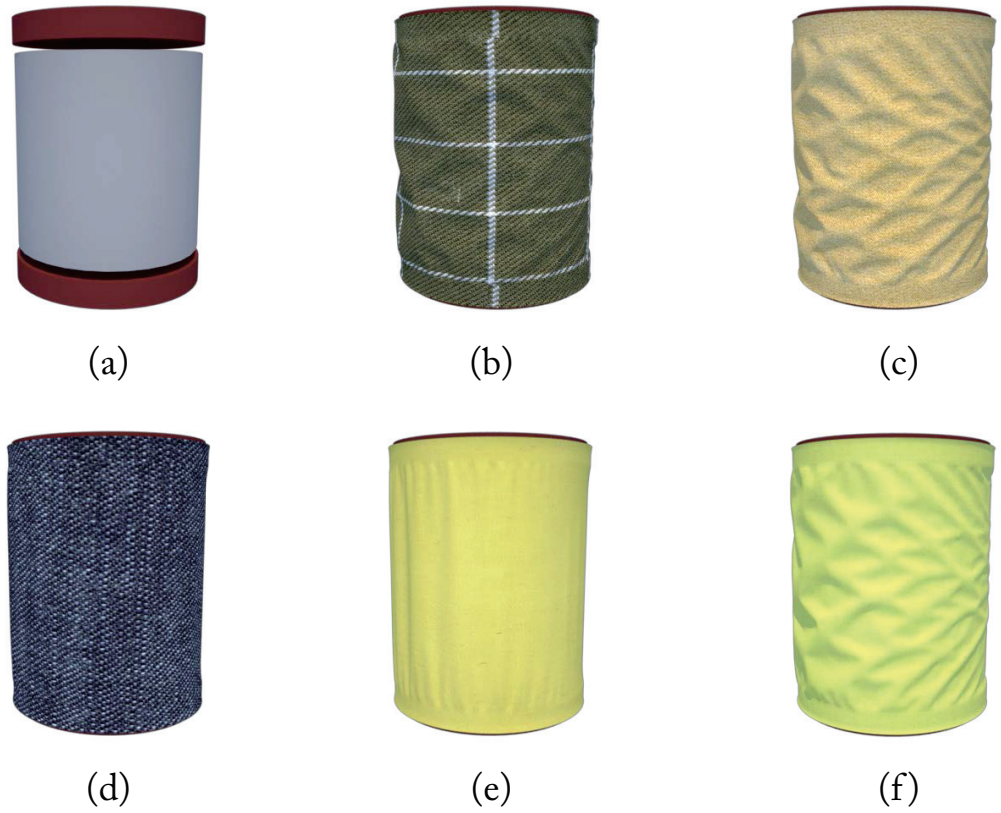


Figure 5.10: The twisting experiment. The top and bottom pieces of the cylinder, which are attached with cloth, are rotated. The images (excluding (a)) show the final results of the simulation for the five sample fabrics. (a) The cylinders for setup. (b) Cotton twill. (c) Silk. (d) Denim. (e) Nylon. (f) Polyester.

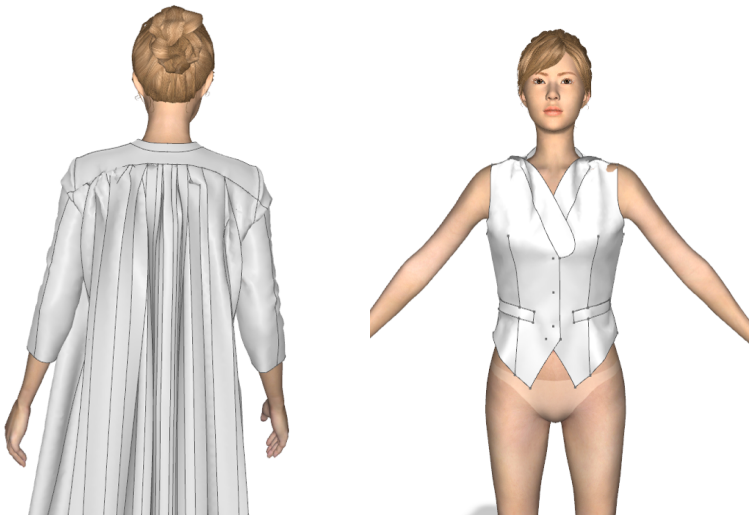
Chapter 6

Tangle Avoidance with Pre-Folding

6.1 Problem of the First Frame Tangle

Even if a cloth simulation was performed using the newest physical cloth simulator and high performing processors, critical problems can be remained, especially for complex structured garments. For example, when creating sharp fold features such as pleats, tuck, lapel on a garment, folds may not formed as intended or first frame tangle (FCT) can occur. Often the problem is persistent; In some cases, the problem does not go away until the end of the simulation.

Shapes of pleats, tuck and lapel are determined by how to fold the panel. The physical meaning of fold in simulation is setting a crease angle to the lines. The problems are (1) sometimes too much simulation time is needed to reach that angle, (2) do not reach that angle even if sufficient simulation



(a) (b)

Figure 6.1: Problems of creating sharp features in (a) pleats, (b) lapel.

time was spent. For example, we cannot resolve the problem to get desired shape of pleats of Figure 6.1(a) by simply increasing time resolution, since the the problem is caused by the vertex moving into the wrong side, the fate of which often within the round-off error. Also, we cannot get the desired shape of lapel of Figure 6.1(b) by setting a crease angle of folding line, even if sufficient simulation time was given.

The above anomalies occur since the simulation is started with completely smooth panels. If we warp initial shape of panel to be similar with desired result, the anomalies can be decreased. The previous studies for representing stylish elements of garments [25, 67, 64] tried to create smooth wrinkle and fold. But non of those works tried to create sharp feature of pleats, tuck, and lapel. So we propose a simple method to create sharp features of garments called pre-folding. Our method preprocesses the panel to gets into a folded

configuration to help successful simulation of those features. The proposed method helps the user to control the shape of the features by a few intuitive parameters.

In the following section, first we describe the proposed method with developed user interface. Then, we demonstrate that this simple method resolves an important problem of simulation.

6.2 Tangle Avoidance with Pre-Folding

We create a simple user interface to set the pre-folding direction and angle. To fold a panel at initial shape, user have to do following tasks.

1. Create an interior line that will be axis of fold.
2. Select the line just created.
3. Select a direction of fold from drop-down menu.
4. Input an angle of the fold.

The created user interface is shown in Figure 6.2

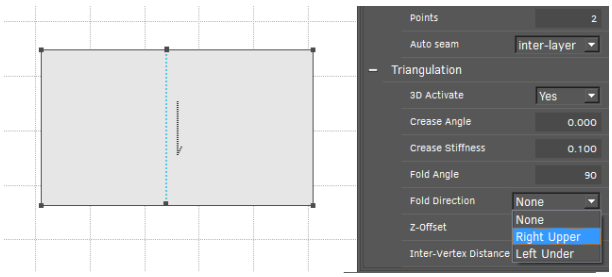


Figure 6.2: The user interface for pre-folding in our system.

Figure 6.3: A flat panel and a folded panel using the proposed system.

(a) (b)

Figure 6.4: Results of simulated garments with lapel: (a) woman jacket without pre-folding, (b) woman jacket with pre-folding.

When the user ended the above tasks, developed system separates the panel into warping region and stationary region from user inputs. Then, it rotates the vertices of warping region respect to rotation axis (selected line) by angle of fold. Using this method, the panel can be folded as shown in Figure 6.3.

We can represent a lapel of garments using the proposed method, as shown in Figure 6.4. This result shows that pre-folding method resolves the problems

that cannot be resolved by setting crease angle parameters of the simulator.

When a number of fold is small such as a lapel, the user can easily make folds of a garment simply using above user interface. However, to create pleats, a number of folds can be more than dozens. In this case, the user can be tiresome to the situation that set all the parameters of each folds manually. To reduce a such cumbersomeness, we integrate the pre-folding process with a pleats creation process. To create pleats, the user have to carry out following tasks.

1. Select side lines for pleating.
2. Select a stationary side.
3. Select a type of pleats.
4. Input a number of slash line.
5. Input a depth of side.
6. Input an angle of in-fold and out-fold.

Using the above inputs, our digital clothing system set the crease angle to slash lines and seamed the segments of side lines to represent pleats. But, an initial shape of garments had been completely smooth until we developed pre-folding method. In this case, often simulation of pleats could not be done as we desired as shown in Figure 6.6(a). To resolve this problem, we pre-fold the panel using slash line as folding rotation axis, and using angles of in-fold

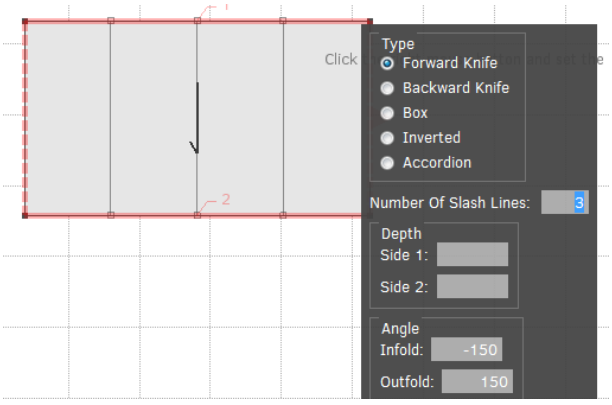
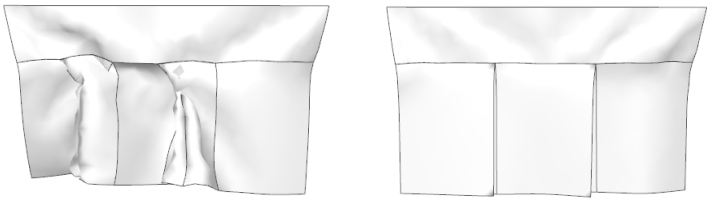


Figure 6.5: The user interface for creating pleats in our system.



(a) (b)

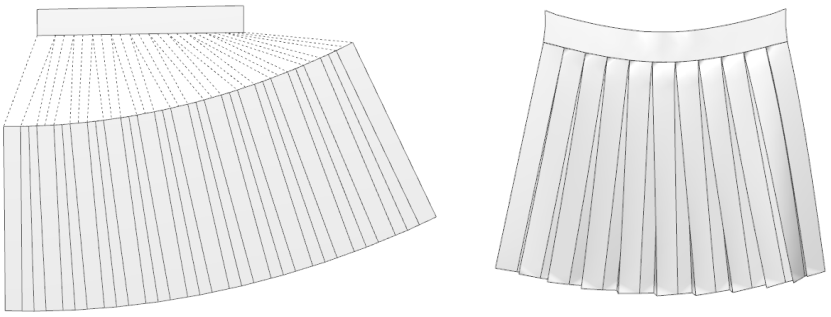
Figure 6.6: Simulated results of pleats with (a) non-folding, (b) pre-folding.

and out-fold.[1] The desired shape of pleats in Figure 6.6(b) is the result of the above method.

Another benefit of representing pleats using pre-folding method is the reduction of simulation converging time. When simulating pleats which have a lot of fold, the duration between starting simulation and converging steady state can be long due to a difference of a initial shape and a final result. The proposed method can reduce that gap of initial and final shape. For exam-

[1]We can adjust this method for creating various types of pleats such as forward knife, backward knife, box, inverted and box pleats.

ple, the simulation of the pleats in Figure 6.7 converged to steady state in a second using pre-folding, but convergence time was about 5 seconds without using pre-folding.



(a) (b)

Figure 6.7: A pattern of pleats and simulated result.

Chapter 7

Conclusion

This dissertation focuses on important problems related to digital clothing, particularly as it pertains to garment modeling and physical cloth simulation. Using the proposed method, an increased number of garment types can be represented on a virtual fitting system, thus enhancing the utilization of digital clothing systems.

For improved designer convenience, in this thesis, we presented a new technique called TAGCON, which constructs a 3D virtual garment from the given tagged and packed panels. Tagging and packing should be done by the user, and it involves simple labeling and 2D manipulation of the panels. However, it does not involve any 3D manipulation. Then, TAGCON constructs the garment automatically via the proposed algorithms (1) to position the panels at suitable locations around the body, and (2) to find the matching seam lines and create the seam. TAGCON takes a deterministic approach and allows inter-

layer seams (T-seams and Y-seams), which contribute to increasing the complexity of the garment that the method can accommodate.

Note that TAGCON has practical significance. First, the proposed method clearly reduces the time taken for the garment construction. In addition to this reduced construction time, the proposed method reduced the variation in the construction time across different garments. Without this technique, the time varies from several minutes to hours, while with the technique, the time varies from 0.5 to 3 min for the examples used in this thesis. TAGCON also significantly reduces the variation in the construction time for novice and expert users. Consequently, TAGCON allows the construction job to be scheduled more predictably. Secondly, the method significantly reduces the cumbersome nature of the job. Considering the trial-and-error that is necessary without the proposed technique, users perceive this reduction to be more critical than the time reduction.

Because designers want to modify garments to resize and redesign constructed garment, we presented a method called trouble-free synced garment editing to modify the constructed garment interactively. The proposed garment-editing method is different from other geometric approaches in that it handles penetration problems during the editing stage. We demonstrated that this method is effective at handling rapid pattern modifications.

In order to represent various kinds of garments in virtual fitting systems, we proposed a method to model the non-elastic components during deformation due to fabric stretching in the context of physically based fabric simula-

tors. When developing the simulator, the controllability is important. We abstracted the non-elastic stretch characteristics of the fabric into four intuitive parameters, namely α, λ_l, λ_u, E.

The job of modeling the non-elastic component of stretch can be made tractable if we decompose the stretch deformation into the immediate elastic, viscoelastic, and plastic components. We found that the decomposition is possible under the assumption that the viscoelastic and plastic components are proportional and their ratio is invariant during the loading. Under this assumption, we developed a non-elastic stretch model based on Kelvin's equation.

Experimental results produced with the proposed method show that it can reproduce the non-elastic nature of stretch deformation in a controllable way. Another contribution of this work is related to the parameter identification. By making decomposed handling of immediate elastic, viscoelastic, and plastic deformations, and by referring to the previous studies on the creep phenomenon, this work developed a method to identify the parameters of the proposed non-elastic model in such a way that does not call for sensitive measurements or a complicated optimization process.

In addition, we introduced a method called pre-folding to represent stylish garment elements. We demonstrated the results produced with this method, to show how the tangle problem can be avoided by making a folded configuration. Even though this method is very simple, it resolves important problems to increase the variety of clothes that can be represented on virtual fitting systems.

Appendix A

Simplification in the Decomposition of Stretch Deformation

When a new string is installed on a guitar, the string needs to be adjusted constantly since the string stretches over time, especially for the case of a nylon string. Here, a question arises. Would the creep halt? Would there be any moment after which the string does not stretch any further while maintaining that constant tension? The correct answer would be no. However, the stretching rate will drop significantly as the stretch progresses. This phenomenon is called work hardening.

For a practical purpose, in this work, we assume that the creep halts after a *finite* duration of time when a constant force is applied. The terminal deformation is called the maximum possible work hardening (MPW). No more plastic deformation can be made if the MPW associated with the applied force

is smaller than the current amount of work hardening.

Let us suppose the above guitar string is now released. There will be an instant shortening. Then, there will be further gradual shortening. Would there be any moment after which the string does not shorten any further? Although the answer is no, we can imagine the amount of shortening will drop to a negligible level after a certain duration. Similarly to MPW, we will assume the shortening will reach the maximum possible work loosening (MPL) in a finite duration of time.

More specifically, we will use the notation $\langle e_1, e_2 \rangle$ to specify the finite duration of hardening and loosening. For example, if we perform a simulation with $\langle 0.001, 3600 \rangle$, it means that, when a stretched string is released, the deformations occurred during [0, 0.001] and [0.001, 3600] (in seconds) are counted as IED, VED, respectively, and the remainder is counted as PD. In a stretching deformation, the deformations occurred during [0, 0.001] and [0.001, 3600] are counted as IED, VED, respectively, and it is assumed further stretch does not to occur after that.

The above is a simplification we make in order to simulate the hysteresis effect with a reasonable amount of computation. Without the above simplifying assumption, we wound not be able to have a clear distinction of IED, VED, and PD.

Bibliography

[1] ADABALA, N., MAGNENAT-THALMANN, N., AND FEI, G. Visualization of woven cloth. In *Proceedings of the 14th Eurographics Workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2003), EGRW '03, Eurographics Association, pp. 178–185.

[2] ASAYESH, A., AND JEDDI, A. A. Modeling the creep behavior of plain woven fabrics constructed from textured polyester yarn. *Textile Research Journal 80*, 7 (2010), 642–650.

[3] ASHIKMIN, M., PREMOŽE, S., AND SHIRLEY, P. A microfacet-based brdf generator. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., pp. 65–74.

[4] BARAFF, D., AND WITKIN, A. Large steps in cloth simulation. In *In Proc. of SIGGRAPH '98* (New York, NY, USA, 1998), ACM, pp. 43–54.

[5] BEELER, T., BICKEL, B., BEARDSLEY, P., SUMNER, B., AND GROSS, M. High-quality single-shot capture of facial geometry. *ACM Trans.*

Graph. 29, 4 (July 2010), 40:1–40:9.

[6] BEELER, T., BICKEL, B., NORIS, G., BEARDSLEY, P., MARSCHNER, S., SUMNER, R. W., AND GROSS, M. Coupled 3d reconstruction of sparse facial hair and skin. *ACM Trans. Graph. 31*, 4 (July 2012), 117:1–117:10.

[7] BEELER, T., HAHN, F., BRADLEY, D., BICKEL, B., BEARDSLEY, P., GOTSMAN, C., SUMNER, R. W., AND GROSS, M. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph. 30*, 4 (July 2011), 75:1–75:10.

[8] BÉRARD, P., BRADLEY, D., NITTI, M., BEELER, T., AND GROSS, M. High-quality capture of eyes. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 223:1–223:12.

[9] BERMANO, A. H., BRADLEY, D., BEELER, T., ZUND, F., NOWROUZEZAHRAI, D., BARAN, I., SORKINE-HORNUNG, O., PFISTER, H., SUMNER, R. W., BICKEL, B., AND GROSS, M. Facial performance enhancement using dynamic shape space analysis. *ACM Trans. Graph. 33*, 2 (Apr. 2014), 13:1–13:12.

[10] BERTHOUZOZ, F., GARG, A., KAUFMAN, D. M., GRINSPUN, E., AND AGRAWALA, M. Parsing sewing patterns into 3d garments. *ACM Transactions on Graphics 32*, 4 (July 2013), 85:1–85:11.

[11] BLIMAN, P., AND SORINE, M. Friction modeling by hysteresis operators. application to dahl, sticktion and stribeck effects. *Pitman Research Notes in Mathematics Series* (1993), 10–10.

[12] BOUAZIZ, S., WANG, Y., AND PAULY, M. Online modeling for realtime facial animation. *ACM Trans. Graph. 32*, 4 (July 2013), 40:1–40:10.

[13] BREEN, D. E., HOUSE, D. H., AND WOZNY, M. J. Predicting the drape of woven cloth using interacting particles. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (1994), ACM, pp. 365–372.

[14] BRIDSON, R., MARINO, S., AND FEDKIW, R. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), Eurographics Association, pp. 28–36.

[15] BROUET, R., SHEFFER, A., BOISSIEUX, L., AND CANI, M.-P. Design preserving garment transfer. *ACM Trans. Graph. 31*, 4 (2012), 36.

[16] CAO, C., HOU, Q., AND ZHOU, K. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph. 33*, 4 (July 2014), 43:1–43:10.

[17] CAO, C., WENG, Y., LIN, S., AND ZHOU, K. 3d shape regression for real-time facial animation. *ACM Trans. Graph. 32*, 4 (July 2013), 41:1–41:10.

[18] CHAI, M., ZHENG, C., AND ZHOU, K. A reduced model for interactive hairs. *ACM Trans. Graph. 33*, 4 (July 2014), 124:1–124:11.

[19] CHEN, Y., LIN, S., ZHONG, H., XU, Y.-Q., GUO, B., AND SHUM, H.-Y. Realistic rendering and animation of knitwear. *IEEE Transactions on Visualization and Computer Graphics 9*, 1 (Jan. 2003), 43–55.

[20] CHEN, Z., FENG, R., AND WANG, H. Modeling friction and air effects between cloth and deformable bodies. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 88.

[21] CHOI, K.-J., AND KO, H.-S. Stable but responsive cloth. *ACM Trans. Graph. (Proc. of SIGGRAPH 2011) 21*, 3 (July 2002), 604–611.

[22] CORDIER, F., SEO, H., AND MAGNENAT-THALMANN, N. Made-to-measure technologies for an online clothing store. *IEEE Computer graphics and applications 23*, 1 (2003), 38–48.

[23] CUI, S.-Z., AND WANG, S.-Y. Nonlinear creep characterization of textile fabrics. *Textile Research Journal 69*, 12 (1999), 931–934.

[24] DAHL, P. A solid friction model. Tech. rep., DTIC Document, 1968.

[25] DECAUDIN, P., JULIUS, D., WITHER, J., BOISSIEUX, L., SHEFFER, A., AND CANI, M.-P. Virtual garments: A fully geometric approach for clothing design. In *Computer Graphics Forum* (2006), vol. 25, Wiley Online Library, pp. 625–634.

[26] FONTANA, M., CARUBELLI, A., RIZZI, C., AND CUGINI, U. Clothassembler: a cad module for feature-based garment pattern assembly. *Computer-Aided Design and Applications 2*, 6 (2005), 795–804.

[27] FRISKEN, S. F., PERRY, R. N., ROCKWOOD, A. P., AND JONES, T. R. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 249–254.

[28] FUHRMANN, A., GROSS, C., LUCKAS, V., AND WEBER, A. Interaction-free dressing of virtual humans. *Computers & Graphics 27*, 1 (2003), 71–82.

[29] GHOSH, A., FYFFE, G., TUNWATTANAPONG, B., BUSCH, J., YU, X., AND DEBEVEC, P. Multiview face capture using polarized spherical gradient illumination. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (New York, NY, USA, 2011), SA '11, ACM, pp. 129:1–129:10.

[30] GLUMAC, R., AND DOEPP, D. Generalized approach to rendering fabric. In *ACM SIGGRAPH 2004 Sketches* (New York, NY, USA, 2004), SIGGRAPH '04, ACM, pp. 120–.

[31] GRINSPUN, E., HIRANI, A. N., DESBRUN, M., AND SCHRÖDER, P. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville,

Switzerland, Switzerland, 2003), SCA '03, Eurographics Association, pp. 62–67.

[32] GUAN, P., REISS, L., HIRSHBERG, D. A., WEISS, A., AND BLACK, M. J. Drape: Dressing any person. *ACM Trans. Graph. 31*, 4 (July 2012), 35:1–35:10.

[33] GUPTA, V., AND KUMAR, S. A model for nonlinear creep of textile fibers. *Textile Research Journal 47*, 10 (1977), 647–649.

[34] HASEGAWA, S., AND FUJII, N. Real-time rigid body simulation based on volumetric penalty method. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on* (2003), IEEE, pp. 326–332.

[35] HEARLE, J. W., AND MORTON, W. E. *Physical properties of textile fibres*. Elsevier, 2008.

[36] HU, L., MA, C., LUO, L., AND LI, H. Robust hair capture using simulated examples. *ACM Trans. Graph. 33*, 4 (July 2014), 126:1–126:10.

[37] HUANG, H., CHAI, J., TONG, X., AND WU, H.-T. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. *ACM Trans. Graph. 30*, 4 (July 2011), 74:1–74:10.

[38] HYEONG-SEOK, K. *Introdunction to Digital Clothing*. Gyomoon, 2015.

[39] ICHIM, A. E., BOUAZIZ, S., AND PAULY, M. Dynamic 3d avatar creation from hand-held video input. *ACM Trans. Graph. 34*, 4 (July 2015), 45:1–45:14.

[40] IGARASHI, T., AND HUGHES, J. F. Clothing manipulation. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2002), UIST '02, ACM, pp. 91–100.

[41] IRAWAN, P., AND MARSCHNER, S. Specular reflection from woven cloth. *ACM Trans. Graph. 31*, 1 (Feb. 2012), 11:1–11:20.

[42] JAKOB, W., ARBREE, A., MOON, J. T., BALA, K., AND MARSCHNER, S. A radiative transfer framework for rendering materials with anisotropic structure. *ACM Trans. Graph. 29*, 4 (July 2010), 53:1–53:13.

[43] JEONG, M.-H., HAN, D.-H., AND KO, H.-S. Garment capture from a photograph. *Computer Animation and Virtual Worlds 26*, 3-4 (2015), 291–300.

[44] JEONG, M.-H., AND KO, H.-S. Draft-space warping: grading of clothes based on parametrized draft. *Computer Animation and Virtual Worlds 24*, 3-4 (2013), 377–386.

[45] JUNG, I.-H., LEE, S.-B., KIM, J.-J., RYU, H.-N., AND KO, H.-S. Modeling the non-elastic stretch deformation of cloth based on creep analysis. *Textile Research Journal 86*, 3 (2016), 245–255.

[46] KIM, S., AND KYU PARK, C. Basic garment pattern generation using geometric modeling method. *International Journal of Clothing Science and Technology 19*, 1 (2007), 7–17.

[47] KIM, S. M., AND KANG, T. J. Garment pattern generation from body scan data. *Computer-Aided Design 35*, 7 (2003), 611–618.

[48] LAHEY, T. Modelling hysteresis in the bending of fabrics.

[49] LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph. 28*, 5 (Dec. 2009), 175:1–175:10.

[50] LI, H., YU, J., YE, Y., AND BREGLER, C. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph. 32*, 4 (July 2013), 42:1–42:10.

[51] MCCARTNEY, J., HINDS, B., SEOW, B., AND GONG, D. Dedicated 3d cad for garment modelling. *Journal of Materials Processing Technology 107*, 1 (2000), 31–36.

[52] MENG, Y., MOK, P. Y., AND JIN, X. Interactive virtual try-on clothing design systems. *Comput. Aided Des. 42*, 4 (Apr. 2010), 310–321.

[53] MENG, Y., MOK, P. Y., AND JIN, X. Computer aided clothing pattern design with 3d editing and pattern alteration. *Comput. Aided Des. 44*, 8 (Aug. 2012), 721–734.

[54] MENG, Y., WANG, C. C. L., AND JIN, X. Flexible shape control for automatic resizing of apparel products. *Comput. Aided Des. 44*, 1 (Jan. 2012), 68–76.

[55] MIGUEL, E., TAMSTORF, R., BRADLEY, D., SCHVARTZMAN, S. C., THOMASZEWSKI, B., BICKEL, B., MATUSIK, W., MARSCHNER, S., AND OTADUY, M. A. Modeling and estimation of internal friction in cloth. *ACM Transactions on Graphics (TOG) 32*, 6 (2013), 212.

[56] MÜLLER, M. Hierarchical position based dynamics. In *In Proc. of Virtual Reality Interactions and Physical Simulations* (2008), The Eurographics Association.

[57] MÜLLER, M., AND GROSS, M. Interactive virtual materials. In *Proceedings of Graphics Interface 2004* (2004), Canadian Human-Computer Communications Society, pp. 239–246.

[58] NARAIN, R., PFAFF, T., AND O'BRIEN, J. F. Folding and crumpling adaptive sheets. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 51.

[59] NGO, C., AND BOIVIN, S. Nonlinear cloth simulation. *Technical report, INRIA Research Report* (2004).

[60] NIKOLIC, M. D., AND MIHAILOVIC, T. V. Investigation of fabric deformations under different loading conditions. *International Journal of Clothing Science and Technology 8*, 4 (1996), 9–16.

[61] POULIN, P., AND FOURNIER, A. A model for anisotropic reflection. *SIGGRAPH Comput. Graph. 24*, 4 (Sept. 1990), 273–282.

[62] PROTOPSALTOU, D., LUIBLE, C., AREVALO, M., AND MAGNENAT-THALMANN, N. A body and garment creation method for an internet based virtual fitting room. In *Advances in Modelling, Animation and Rendering*. Springer, 2002, pp. 105–122.

[63] PROVOT, X. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface* (1995), Canadian Information Processing Society, pp. 147–147.

[64] ROBSON, C., MAHARIK, R., SHEFFER, A., AND CARR, N. Smi 2011: Full paper: Context-aware garment modeling from sketches. *Comput. Graph. 35*, 3 (June 2011), 604–613.

[65] TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. Elastically deformable models. In *ACM Siggraph Computer Graphics* (1987), vol. 21, ACM, pp. 205–214.

[66] TURQUIN, E., CANI, M.-P., AND HUGHES, J. F. Sketching garments for virtual characters. In *In Courses, ACM, J. A. Jorge and J. F. Hughes, Eds., SIGGRAPH '06* (2004), pp. 175–182.

[67] TURQUIN, E., WITHER, J., BOISSIEUX, L., CANI, M.-P., AND HUGHES, J. F. A sketch-based interface for clothing virtual characters. *IEEE Comput. Graph. Appl. 27*, 1 (Jan. 2007), 72–81.

[68] UMETANI, N., KAUFMAN, D. M., IGARASHI, T., AND GRINSPUN, E. Sensitive couture for interactive garment modeling and editing. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2011) 30*, 4 (2011), 90.

[69] VENKATARAMAN, K., LODHA, S., AND RAGHAVAN, R. Technical section: A kinematic-variational model for animating skin with wrinkles. *Comput. Graph. 29*, 5 (Oct. 2005), 756–770.

[70] VOLINO, P., CORDIER, F., AND MAGNENAT-THALMANN, N. From early virtual garment simulation to interactive fashion design. *Computer-Aided Design 37*, 6 (2005), 593–608.

[71] VOLINO, P., MAGNENAT-THALMANN, N., AND FAURE, F. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Transactions on Graphics 28*, 4 (2009), Article–No.

[72] WANG, C. C., WANG, Y., AND YUEN, M. M. Feature based 3d garment design through 2d sketches. *Computer-Aided Design 35*, 7 (2003), 659–672.

[73] WANG, H., O'BRIEN, J. F., AND RAMAMOORTHI, R. Data-driven elastic models for cloth: modeling and measurement. *ACM Transactions on Graphics (TOG) 30*, 4 (2011), 71.

[74] WANG, J., ZHAO, S., TONG, X., SNYDER, J., AND GUO, B. Modeling anisotropic surface reflectance with example-based microfacet synthesis. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 41:1–41:9.

[75] WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. Realtime performance-based facial animation. In *ACM SIGGRAPH 2011 Papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 77:1–77:10.

[76] WESTIN, S. H., ARVO, J. R., AND TORRANCE, K. E. Predicting reflectance functions from complex surfaces. *SIGGRAPH Comput. Graph. 26*, 2 (July 1992), 255–264.

[77] XU, Y.-Q., CHEN, Y., LIN, S., ZHONG, H., WU, E., GUO, B., AND SHUM, H.-Y. Photorealistic rendering of knitwear using the lumislice. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 391–398.

[78] ZHAO, S., JAKOB, W., MARSCHNER, S., AND BALA, K. Building volumetric appearance models of fabric using micro ct imaging. In *ACM SIGGRAPH 2011 Papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 44:1–44:10.

초 록

본 논문은 가상 의복의 생성, 수정 및 시뮬레이션에 관한 새로운 방법들을 제시한다. 첫 번째로, 우리는 분류된 의상 원판에 대한 가상 의복 생성 방법을 기술한다. 사용자는 의상 원판의 분류를 위해서 이 차원 사용자 환경에서 의상 원판에 대한 간단한 작업을 수행 해야 하며, 삼 차원 작업은 필요로 하지 않는다. 사용자가 원판을 분류한 후, 원판의 아바타(avatar) 주변 자동 배치 방법 및 자동 봉제 생성 방법 등의 제안된 방법을 통해서 자동으로 가상 의복을 생성한다. 본 의상 생성 방법은 다양한 종류의 의상을 생성할 수 있으며, 의상 생성에 필요한 시간과 노력을 크게 감소시킨다.

두 번째로, 우리는 동기화 된 이 차원 의복 패턴 및 삼 차원 가상 의복의 수정 방법을 제안한다. 본 논문에서 사용한 선형 보간법, 선형 외삽법, 침투 탐지 방법은 사용자가 이 차원과 삼 차원의 동기 상태를 잃지 않고 즉각적으로 가상 의복을 수정할 수 있도록 도와준다.

다음으로 우리는 물리 기반 직물 시뮬레이터 개발을 위한 직물의 비-탄성 인장 변형을 모델링 할 수 있는 방법을 제안한다. 우리는 본 문제가 직물의 변형을 즉각 탄성 변형, 지연 탄성 변형, 영구 변형으로 나누어 다룬다면 해결하기 용이해짐을 발견하였다. 직물 시뮬레이터를 개발하기 위해서는 이와 같은 직물 변형의 분류가 어떠한 상태의 변형이나, 가해지는 힘 등에 대해서도 가능해야 한다. 우리는 다양한 일정 하중 크립(creep) 실험에 대한 관찰에 기반하여, 특정 종류의 직물에 대해서는 지연 탄성 변

형과 영구 변형이 시간과 관계 없이 일정한 비율을 유지한다고 가정하였다. 제안된 방법을 사용하여 만들어진 실험 결과들은 일반적인 기대치와 거의 일치하며, 임의로 가해지는 힘에 대해서도 정상 동작함을 보여준다.

또한, 우리는 옷깃, 주름 같은 의복의 꾸밈 요소들을 표현할 수 있는 방법을 제시하였다. 실험 된 결과들은 제시된 간단한 방법이 물리 기반 시뮬레이터에서 다루기 쉽지 않은 문제들을 효과적으로 해결함을 보여준다.

주요어: 가상 의복 생성, 가상 의복 수정, 의복 시뮬레이션, 비탄성 변형
Keywords: virtual garment construction, virtual garment editing, cloth simulation, non-elastic deformation

학 번: 2012-30911