# Rule-Free Sewing Pattern Adjustment with Precision and Efficiency

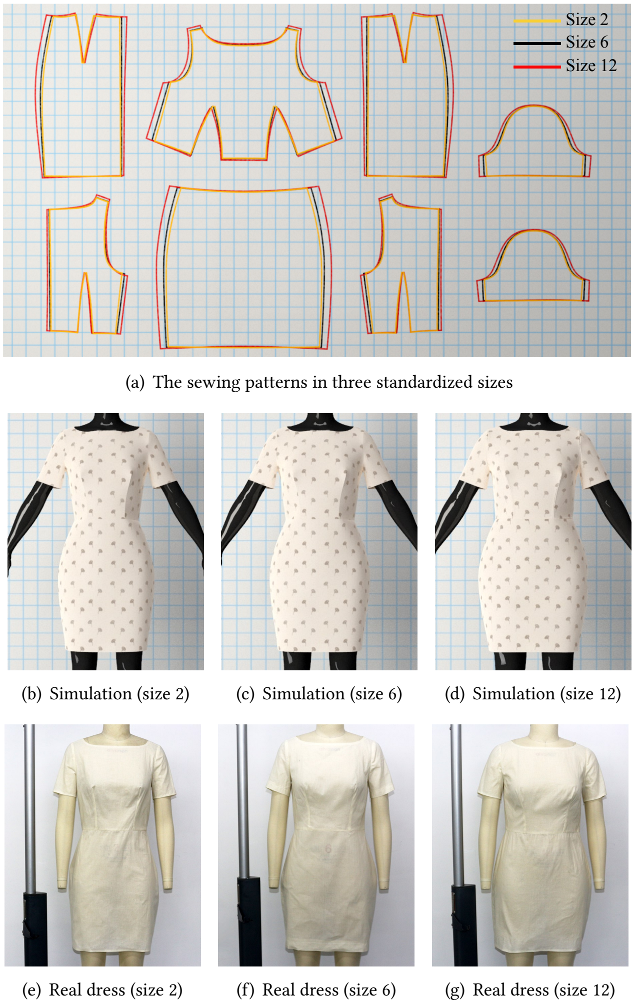
HUAMIN WANG, The Ohio State University and Frilly Inc.

Being able to customize sewing patterns for different human bodies without using any pre-defined adjustment rule will not only improve the realism of virtual humans in the entertainment industry, but also deeply affect the fashion industry by making fast fashion and made-to-measure garments more accessible. To meet the requirement set by the fashion industry, a sewing pattern adjustment system must be both efficient and precise, which unfortunately cannot be achieved by existing techniques. In this paper, we propose to solve sewing pattern adjustment as a nonlinear optimization problem immediately, rather than in two phases: a garment shape optimization phase and an inverse pattern design phase as in previous systems. This allows us to directly minimize the objective function that evaluates the fitting quality of the garment sewn from a pattern, without any compromise caused by the nonexistence of the solution to inverse pattern design. To improve the efficiency of our system, we carry out systematic research on a variety of optimization topics, including pattern parametrization, initialization, an inexact strategy, acceleration, and CPU-GPU implementation. We verify the usability of our system through automatic grading tests and made-to-measure tests. Designers and pattern makers confirm that our pattern results are able to preserve design details and their fitting qualities are acceptable. In our computational experiment, the system further demonstrates its efficiency, reliability, and flexibility of handling various pattern designs. While our current system still needs to overcome certain limitations, we believe it is a crucial step toward fully automatic pattern design and adjustment in the future.

CCS Concepts: • **Computing methodologies → Physical simulation**;

Additional Key Words and Phrases: sewing pattern, made-to-measure, nonlinear optimization, quasistatic simulation, inexact methods

**ACM Reference Format:**
Huamin Wang. 2018. Rule-Free Sewing Pattern Adjustment with Precision and Efficiency. *ACM Trans. Graph.* 37, 4, Article 53 (August 2018), 13 pages. https://doi.org/10.1145/3197517.3201320



(a) The sewing patterns in three standardized sizes

(b) Simulation (size 2) (c) Simulation (size 6) (d) Simulation (size 12)

(e) Real dress (size 2) (f) Real dress (size 6) (g) Real dress (size 12)

Fig. 1. The grading results of a secretary dress example. Given a base sewing pattern (black) in (a) and its simulated garment in (c), our pattern optimization system automatically converts the base pattern into new patterns for making garments that fit new bodies, as shown in (b) and (d). The experiment demonstrates that the real garments in (e) and (g) are able to preserve design details, and the whole optimization process lasts less than two minutes for a mesh with 66K elements.

## 1 INTRODUCTION

Today the multi-billion fashion industry is dominated by *ready-to-wear* garments, made and sold in standardized sizes, from Size 00 to Size 12 for example. To make a ready-to-wear garment, a pattern maker typically develops a sewing pattern in a base size first, such as Size 6. A pattern grader then transforms the pattern from the base size to additional sizes, using a series of *garment grading rules* established over the last two centuries. Compared with a ready-to-wear garment, a *made-to-measure* garment reduces the inventory cost and provides superior fit, by customizing an existing pattern for each specific individual as needed. However, a made-to-measure garment is much more difficult and time consuming to develop, since the individual's body can differ significantly and the *pattern adjustment rules* for disproportionate body parts are rather sparse,

Author's address: Huamin Wang, Department of Computer Science and Engineering, The Ohio State University and Frilly Inc. whmin@cse.ohio-state.edu.

0730-0301/2018/8-ART53 $15.00
https://doi.org/10.1145/3197517.3201320

especially for women's garments. Even if skilled pattern makers can formulate these rules for a specific garment, they can hardly apply the same rules to other garments. Given the vast diversity of fashionable garment designs, it is impractical to create all of the rules ahead of time. This explains why the availability of made-to-measure garments is limited in the fashion industry and why their prices are typically higher than those of ready-to-wear ones.

In this research, we are interested in developing a rule-free automatic sewing pattern adjustment system for making garments that fit to different body shapes. This system will not only improve the realism of virtual humans in the human-centered entertainment industry, but also trigger a deep impact on the future of the fashion industry, by speeding up the pace of fast ready-to-wear fashion and enabling mass production of made-to-measure garments. The success of such a system relies on its precision. The measurement difference between two consecutive standardized sizes is typically two inches or less. This implies that the precision of a usable pattern adjustment system must be within one inch, or 2.54cm, for each body measurement. The requirement on the precision of a single pattern patch can be much stricter than this. For example, pants are made of four panel patches and the precision of each panel patch must be within 1/4 inch, or even stricter for skinny and slim-fit pants. No previous graphics research has studied sewing pattern adjustment at this level of precision, as far as we know.

Another issue associated with sewing pattern adjustment, fundamentally as an inverse elastic design problem, is its computational cost. One way of solving inverse elastic design is to use sequential quadratic programming (SQP) [Skouras et al. 2014]. SQP is restricted from using many efficient linear solvers, since its matrix is indefinite. Finding a suitable merit function and a backtracking line search strategy for SQP [Nocedal and Wright 2006] is also challenging in practice. Therefore, researchers in computer graphics, mechanical engineering, and biomechanics often prefer to solve inverse elastic design by iteratively running two steps: a forward step that reaches quasistatic equilibrium; and an optimization step that adjusts the parameters. This technique has demonstrated its flexibility in solving a variety of inverse problems [Bickel et al. 2009; Lund et al. 2003; Miguel et al. 2016], but it still needs a large computational cost. For instance, the optimization process of a mesh with less than 10K vertices can easily last hours, if not days [Casati et al. 2016]. An immediate implementation of this technique would eliminate the usability of our system, as pattern markers would rather perform manual adjustment.

To develop a precise, efficient, and practically useful sewing pattern adjustment system, we made the following contributions.

- *Problem formulation.* We propose to directly minimize the objective function that evaluates the fitting quality of a garment dressed on a target body, rather than in two phases as did in previous systems. The objective function evaluates the looseness, smoothness, deformation, and relative location of the garment, by comparing it with a base garment dressed on the base body. The function is suitable for a variety of garments, such as bodysuits under constant stretching and skirts with shirring.
- *Optimization method.* Based on the aforementioned two-step idea, we develop a novel implementation that finishes the optimization process of a sewing pattern with 160K triangles in less than five minutes. The success of this implementation is stemmed from our systematic research and analysis on pattern parametrization, initialization, an inexact optimization strategy, acceleration, and CPU-GPU implementation.
- *Usability test.* To evaluate the usability of our system, we apply it to adjust commercial sewing patterns for two dress forms and eight human bodies, and test their fits using real garments, as shown in Fig. 1 and 16. While our system solves the optimization problem mathematically, we do not test a body if it is too different from the base one at this time. It is because the industry considers petite-sized, regular-sized and plus-sized garments as different categories, and often designs them differently.

Our experiment shows that our system is fast, reliable, easy-to-use, GPU-friendly, and memory-efficient. Our pattern makers confirm that the patterns adjusted by our system are balanced and the resulting garments fit target bodies well within an acceptable error threshold. Both the accuracy and the efficiency of our system have space for improvement, by using better human models, physics-based simulators, and optimization methods in the future.

## 2 OTHER RELATED WORK

*Cloth simulation.* Fast and accurate cloth simulation is highly useful in computer animation and computer-aided fashion modeling. While cloth dynamics can be integrated over time explicitly [Bridson et al. 2003], most cloth simulators in computer graphics [Narain et al. 2012; Volino et al. 2009] today use implicit integration to avoid the numerical instability issue. Depending on how the elasticity of cloth is modeled, a cloth simulator can be spring-based [Choi and Ko 2002; Liu et al. 2013], continuum-based [Baraff and Witkin 1998; Narain et al. 2012], or yarn-based [Cirio et al. 2014; Kaldor et al. 2010]. Researchers also investigated the use of position-based constraints [Macklin et al. 2014; Müller 2008] to replace elastic forces. Recently, Liu and colleagues [2013] and Bouaziz and collaborators [2014] developed a projective dynamics method for fast simulation of cloth, which can be interpreted as preconditioned gradient descent [Wang 2015; Wang and Yang 2016] or the alternating direction method of multipliers (ADMM) [Narain et al. 2016]. Besides the elasticity of cloth, researchers have also studied a wide range of other research topics, including inextensibility [Goldenthal et al. 2007], elasticity measurement [Miguel et al. 2012; Wang et al. 2011], hysteresis [Miguel et al. 2013], and collision handling [Bridson et al. 2003; Tang et al. 2016]. While this work relies heavily on the cloth simulation technology, it is relatively independent of any specific simulator and it will benefit from the advance of cloth simulation techniques in the future.

*Garment and pattern design.* Graphics researchers have investigated garment design for decades, but most of their works were focused on dressed characters [Guan et al. 2012] and did not involve the use of a sewing pattern. On the other hand, existing fashion CAD softwares, such as Optitex and Marvelous designer, allows users to design 3D garments by interactively modifying 2D sewing patterns only. To achieve a specific design or fitting goal in 3D, users must perform trial and error in 2D over time. Umetani and colleagues [2011] proposed to predict the needed pattern adjustment for a specific design goal, by precomputing a sensitivity matrix that linearizes the relationship between pattern parameters and garment features. A more common practice adopted by existing co-design systems [Brouet et al. 2012; Decaudin et al. 2006; Meng et al. 2012; Wang et al. 2005] is to design 3D garments first, and then apply off-the-shelf surface parametrization methods to flatten garments into sewing patterns. Since such a geometric strategy fails to consider the draping effect, the resulting sewing pattern often cannot recover the originally designed garment precisely, as Bartle and colleagues [2016] demonstrated. Their solution is a fixed point optimization method that tries to calculate the sewing pattern inversely from the garment, in which quasistatic simulation is treated as a black box. As discussed next in Section 3, this inverse pattern design problem may not have an exact solution. This fact was also pointed out by Casati and collaborators [2016]. Different from our work, their research was focused on estimating the 3D reference shape of a garment with frictional contacts and their problem is difficult in its own way.

## 3 PROBLEM OVERVIEW

Let $\boldsymbol{\theta} \in \mathbb{R}^B$ be a set of parameters controlling the shape of a sewing pattern, and $\texttt{Sim}_\phi(\boldsymbol{\theta}) : \mathbb{R}^B \rightarrow \mathbb{R}^{3N}$ be a simulation function generating the $N$ vertex positions of a 3D quasistatic garment, dressed on a target body represented by a signed distance field $\phi$. Pattern adjustment can be defined as the minimization of an objective function:

$$\boldsymbol{\theta} = \arg\min f\left(\texttt{Sim}_\phi(\boldsymbol{\theta}), \boldsymbol{\theta}, \phi, \texttt{Sim}_{\bar{\phi}}(\bar{\boldsymbol{\theta}}), \bar{\boldsymbol{\theta}}, \bar{\phi}\right), \tag{1}$$

in which $\bar{\boldsymbol{\theta}}$ controls the base pattern and $\bar{\phi}$ represents the base body. Intuitively, the objective function evaluates the fitting quality of $\boldsymbol{\theta}$, by treating the base size as a reference. This optimization problem is complicated, since the objective function depends on not only $\boldsymbol{\theta}$, but also the simulated garment in the quasistatic state: $\texttt{Sim}_\phi(\boldsymbol{\theta})$. To simplify this problem, existing techniques often choose to introduce an intermediate variable $\mathbf{x} = \texttt{Sim}_\phi(\boldsymbol{\theta}) \in \mathbb{R}^{3N}$ and solve the problem in two phases. In the first phase, the 3D garment is transferred from body $\bar{\phi}$ to body $\phi$:

$$\mathbf{x} = \arg\min f\left(\mathbf{x}, \boldsymbol{\theta}, \phi, \texttt{Sim}_{\bar{\phi}}(\bar{\boldsymbol{\theta}}), \bar{\boldsymbol{\theta}}, \bar{\phi}\right), \tag{2}$$

and in the second phase, $\boldsymbol{\theta}$ is calculated to satisfy:

$$\texttt{Sim}_\phi(\boldsymbol{\theta}) = \mathbf{x}, \tag{3}$$

which is known as an *inverse pattern design* problem. A common practice adopted in the past is to ignore fabric elasticity and solve the second phase by flattening each 3D garment patch into a 2D pattern piece. To address elastic deformation of fabrics during inverse pattern design, Bartle and colleagues [2016] invented a fixed point method for direct 3D garment editing. While their method produces more plausible sewing patterns, a fundamental issue related to the two-phase approach remains: Equation 3 may not have an exact solution. An intuitive example is a sagging flag, which cannot stand up in its quasistatic state no matter how we optimize its reference shape. Therefore, the solution to Phase 2 fails to meet the same minimization goal achieved by Phase 1 and it is only sub-optimal, as Fig. 2 shows.

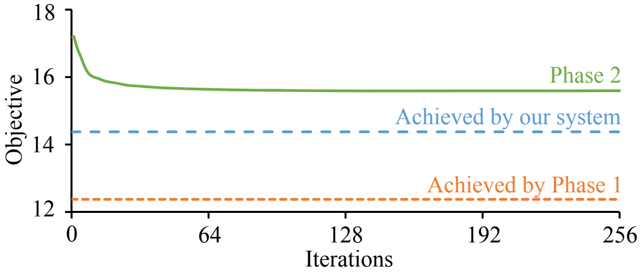


Fig. 2. The outcome of a two-phase approach in the Carmen skirt example. The plot shows handling pattern adjustment in two phases is not optimal, since inverse pattern design may not have a solution and it ignores the objective function outlined in Phase 1. Here we use our system to solve inverse pattern design, by minimizing the positional cost only.

In this paper, we choose to solve the constrained optimization problem in Equation 1 immediately, without any compromise caused by the two phases. To begin with, we will introduce the formulations of the simulator $\texttt{Sim}_\phi(\boldsymbol{\theta})$, the parameters $\boldsymbol{\theta}$, and the objective function $f$ in Section 4, 5, and 6. After that, we will present our optimization system in Section 7, built upon the state-of-the-art research on GPU-based simulation and optimization. Results and evaluations will be provided in Section 8.

## 4 QUASISTATIC SIMULATION

Quasistatic simulation of a 3D garment, i.e., $\texttt{Sim}_\phi(\boldsymbol{\theta})$, is a major component in our system. While quasistatic simulation has been extensively studied in computer graphics for decades, our simulator must meet two unique requirements. First, the tangent stiffness matrix of the total force exerted on the garment must be evaluable and invertible. Second, the simulator must be fast and stable.

### 4.1 Simulation Model

To meet the first requirement, we propose to ignore cloth-body friction and self collision of cloth. We believe it is a reasonable assumption for sewing pattern adjustment, since we prefer garments to drape naturally and relaxedly into a unique configuration. We note that ignoring self collision does not prevent the system from handling multi-layered garments, such as the pants example in Fig. 14, as long as self collision does not greatly affect the draped garment shape.

The key idea behind our simulation model is to handle cloth-body collision by a collision potential energy, not projection. This naturally avoids the discontinuity problem of sewing pattern optimization, occurred in [Casati et al. 2016]. Let $\mathbf{x} \in \mathbb{R}^{3N}$ be the vertices of a 3D garment. Its quasistatic state satisfies:

$$\mathbf{x} = \arg\min \left\{ E(\mathbf{x}) + F(\mathbf{x}) + G(\mathbf{x}) - \sum_i \mu \log(\phi(\mathbf{x}_i)) \right\}, \tag{4}$$

in which $E(\mathbf{x})$ is the elastic potential energy, $F(\mathbf{x})$ is the sticking energy, $G(\mathbf{x})$ is the gravitational energy, and $\mu$ is a log barrier parameter. The purpose of the sticking energy is to fix certain vertices

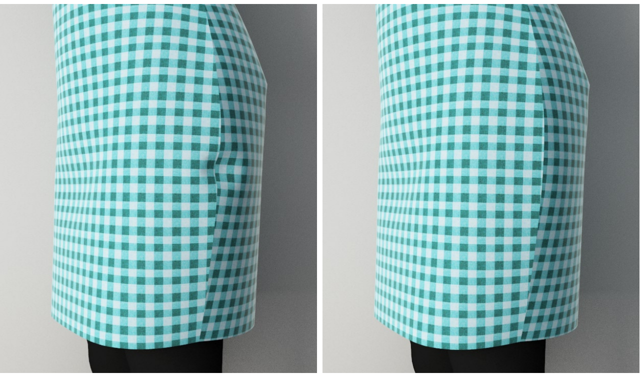

(a) Before re-initialization (b) After re-initialization

Fig. 3. A dent caused by a local minimum of the secretary dress example. While we can remove it by re-initializing quasistatic simulation, we found it to be often unnecessary as its impact on the overall shape is small.

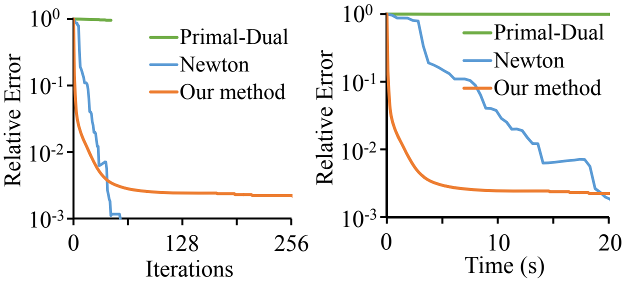


Fig. 4. The convergence of different simulation methods. Quasistatic simulation of a virtual garment can be computationally expensive, due to high nonlinearity and stiffness of cloth-body collision and cloth planar deformation. This experiment shows that our GPU-based simulator performs the best, within a short period of computational time.

at given locations, accounting for the existence of belts and pins. The elastic energy contains the elastic planar energy, modeled by a mass-spring system and the elastic bending energy, modeled by the quadratic model [Bergou et al. 2006]. Let $\{s_i, s_j\}$ be the vertices of spring $s$, either belonging to two patches if $s$ is a sewing spring. or the same patch if $s$ is a patch edge. We define the elastic spring energy as:

$$\begin{cases} E_s^{\mathsf{S}}(\mathbf{x}) = \frac{Y^{\mathtt{sewing}}}{2} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2, & \text{if } s \text{ is sewing,} \\ E_s^{\mathsf{S}}(\mathbf{x}) = \frac{Y^{\mathtt{patch}}}{2\|\mathbf{p}_i - \mathbf{p}_j\|} \left(\left\|\mathbf{x}_i - \mathbf{x}_j\right\| - r_s \left\|\mathbf{p}_i - \mathbf{p}_j\right\|\right)^2, & \text{otherwise,} \end{cases} \tag{5}$$

where $Y^{\mathtt{sewing}}$ and $Y^{\mathtt{patch}}$ are constant Young's moduli, $\mathbf{p}_i$ and $\mathbf{p}_j$ are vertex positions in the pattern, and $r_s$ is a scaling factor for shirring effects, such as those in the Carmen skirt example. According to Equation 4, the collision force at vertex $i$ is: $\mu\nabla\phi(\mathbf{x}_i)/\phi(\mathbf{x}_i)$, which behaves like a barrier at $\phi(\mathbf{x}_i) = 0$ as $\mu$ converges to zero. In our system, we set $\mu = 0.01$ as constant for simplicity. The collision energy model fails if $\phi(\mathbf{x}_i) < 0$. In that case, we project $\mathbf{x}_i$ onto the body surface instead.

While we prefer the quasistatic state to be unique, it is often not, due to the local minima. We typically observe this issue as dents being stuck during the parameter optimization process, as shown in Fig. 3a. This issue can be resolved by re-initializing quasistatic simulation, or using larger bending forces. Since most of these dents are small and we apply strong regularization as discussed in Subsection 5.1, we do not specifically address the issue in our current system.

### 4.2 Simulation Method

We explore several ways to solve Equation 4. First, we test Newton's method and we find that it cannot converge as fast as we expected, as shown in Fig. 4. The main reason is because the collision is highly nonlinear, which makes quadratic approximation ineffective. Next we test the primal-dual idea, which has proven to be efficient in dynamic simulation of a stiff system [Tournier et al. 2015]. To do so, we introduce two dual variable vectors: $\boldsymbol{\lambda}^{\mathsf{C}} \in \mathbb{R}^N$ for modeling stiff collision and $\boldsymbol{\lambda}^{\mathsf{S}} \in \mathbb{R}^S$ for modeling stiff springs, in which $S$ the number of springs. The resulting nonlinear system becomes:

$$\begin{cases} -\nabla(E^{\mathsf{B}} + F + G) - \sum_s \lambda_s^{\mathsf{S}} \nabla \left\|\mathbf{x}_{si} - \mathbf{x}_{sj}\right\| + \sum_i \lambda_i^{\mathsf{C}} \nabla\phi(\mathbf{x}_i) = \mathbf{0}, \\ \forall s, \quad -\left\|\mathbf{x}_{si} - \mathbf{x}_{sj}\right\| + (r_s + \lambda_s^{\mathsf{S}}/Y) \left\|\mathbf{p}_{si} - \mathbf{p}_{sj}\right\| = 0, \\ \forall i, \quad \phi(\mathbf{x}_i)\lambda_i^{\mathsf{C}} - \mu = 0, \end{cases} \tag{6}$$

where $E^{\mathsf{B}}$ is the elastic bending energy. By applying Newton's method to Equation 6, we obtain the primal-dual search direction for $\mathbf{x}$, $\boldsymbol{\lambda}^{\mathsf{S}}$, and $\boldsymbol{\lambda}^{\mathsf{C}}$ in every iteration. Unfortunately, Fig. 4 shows the primal-dual method performs worse, since it needs a small step length to preserve $\lambda_i^{\mathsf{C}} > 0$. In the end, we choose to implement our quasistatic simulator based on the gradient descent method, with Jacobi preconditioning and Chebyshev acceleration [Wang and Yang 2016]. After GPU parallelization, our simulator takes 78ms to finish one iteration in the Carmen skirt example, nearly half of which is used for cloth-body collision. In comparison, Newton's method and the primal-dual method finish one iteration in 0.47s and 2.09s, respectively.

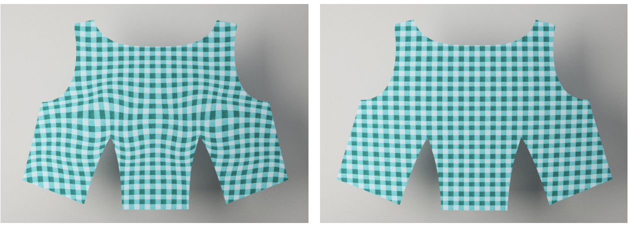
## 5 PARAMETRIZATION AND REGULARIZATION

The next design choice in our system is the parameters controlling the pattern shape: $\boldsymbol{\theta} \in \mathbb{R}^B$. Naturally, we can set the parameters as all of its 2D vertices: $\mathbf{p} \in \mathbb{R}^{2N}$. However, only boundary vertices in this parametrization controls the shape, while interior vertices are merely spatial samples. If the parametrization includes all of the vertices, the pattern interior can contain self-intersection or sample distribution change caused by overfitting, as shown in Fig. 5a. Pattern optimization also suffers more from the local minimum issue, due to a large parameter space. For these reasons, we propose to parameterize the sewing pattern by its boundary vertices.

Suppose that the boundary vertices $\mathbf{p}^{\mathsf{B}}$ of a sewing pattern has been given. We determine the interior vertices $\mathbf{p}^{\mathsf{I}}$ by minimizing the difference of vertex displacement:

$$\mathbf{p}^{\mathsf{I}} = \arg\min \frac{1}{2} \left\| \begin{bmatrix} \mathbf{L}^{\mathsf{I}} & \mathbf{L}^{\mathsf{B}} \end{bmatrix} \begin{bmatrix} \mathbf{p}^{\mathsf{I}} - \bar{\mathbf{p}}^{\mathsf{I}} \\ \mathbf{p}^{\mathsf{B}} - \bar{\mathbf{p}}^{\mathsf{B}} \end{bmatrix} \right\|^2, \tag{7}$$

where $\mathbf{L}^{\mathsf{I}} \in \mathbb{R}^{(2N-B)\times(2N-B)}$ and $\mathbf{L}^{\mathsf{B}} \in \mathbb{R}^{(2N-B)\times B}$ are the components of the hinge-edge-based Laplacian operator matrix [Bergou et al. 2006] evaluated from the base pattern $\bar{\mathbf{p}}$. The analytic solution



(a) Parametrization by all vertices (b) Parametrization by boundary vertices

Fig. 5. The optimization results of using two parametrizations. When the system selects all of the vertices to parametrize a pattern, it tends to distort the pattern interior, unless it uses strong smooth regularization. In contrast, the parametrization by boundary vertices is free of this issue.

to this problem is:

$$\mathbf{p}^{\mathrm{I}} = \bar{\mathbf{p}}^{\mathrm{I}} - \left(\mathbf{L}^{\mathrm{I}}\right)^{-1} \mathbf{L}^{\mathrm{B}} \left(\mathbf{p}^{\mathrm{B}} - \bar{\mathbf{p}}^{\mathrm{B}}\right), \tag{8}$$

using the fact that $\mathbf{L}^{\mathrm{I}}$ is symmetric positive definite. Since $\mathbf{L}^{\mathrm{I}}$ and $\mathbf{L}^{\mathrm{B}}$ are constant, we pre-factorize $\mathbf{L}^{\mathrm{I}}$ and precompute $\mathbf{L}^{\mathrm{B}}$ for fast runtime computation. We note that this parametrization is fundamentally equivalent to the positive mean value coordinates method [Joshi et al. 2007]. But instead of calculating the coordinates explicitly, we define the interpolation process procedurally, as the parameter space is still relatively large. Fig. 5b shows that the parametrization avoids the vertex distribution change and the overfitting issue.

### 5.1 Regularization

Once we parameterize the sewing pattern by its boundary vertices, we formulate regularization terms on boundary vertices as well. Here we implement two regularization terms. The fixing regularization term penalizes the changes of boundary vertices:

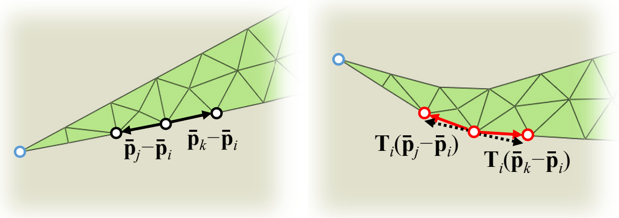
$$R^{\mathtt{fix}} = \frac{\delta^{\mathtt{fix}}}{2} \sum_i \|\mathbf{p}_i - \bar{\mathbf{p}}_i\|^2, \tag{9}$$

in which $i$ is a boundary vertex and $\delta^{\mathtt{fix}}$ is a regularization strength coefficient. The actual use of $R^{\mathtt{fix}}$ is to ensure that the Hessian of the total regularization is positive definite, a property desired by preconditioning as discussed in Section 7. Meanwhile, the bending regularization term tries to preserve the shape of the boundary curve:

$$R^{\mathtt{cur}} = \frac{\delta^{\mathtt{cur}}}{2} \sum_i w_i \left\|\mathbf{p}_j + \mathbf{p}_k - 2\mathbf{p}_i - \mathbf{T}_i(\bar{\mathbf{p}}_j + \bar{\mathbf{p}}_k - 2\bar{\mathbf{p}}_i)\right\|^2, \tag{10}$$

in which $\delta^{\mathtt{cur}}$ determines the regularization strength, $j$ and $k$ are the two vertices adjacent to boundary vertex $i$, $w_i$ controls the importance of vertex $i$, and $\mathbf{T}_i$ is an affine transformation matrix used for removing rotation and uniform scaling, as shown in Fig. 6. We typically use a smaller $w_i$ if $i$ is a corner vertex (in blue), so that the angle between two curves can be more adjustable.

The two aforementioned regularization terms do not need any user intervention. Some other regularization terms, such the preservation of patch symmetry and right corner angles, may become necessary in the future, especially if the target body differs significantly from the base body. We will allow the system to accept these regularization terms as part of user interaction.


$\bar{\mathbf{p}}_j - \bar{\mathbf{p}}_i$
$\bar{\mathbf{p}}_k - \bar{\mathbf{p}}_i$
$\mathbf{T}_i(\bar{\mathbf{p}}_j - \bar{\mathbf{p}}_i)$
$\mathbf{T}_i(\bar{\mathbf{p}}_k - \bar{\mathbf{p}}_i)$

(a) The original pattern (b) The adjusted pattern

Fig. 6. Bending regularization. This regularization term tries to minimize the difference between the current edge vectors (in red) and the original edge vectors (in black), after being transformed by $\mathbf{T}_i$.

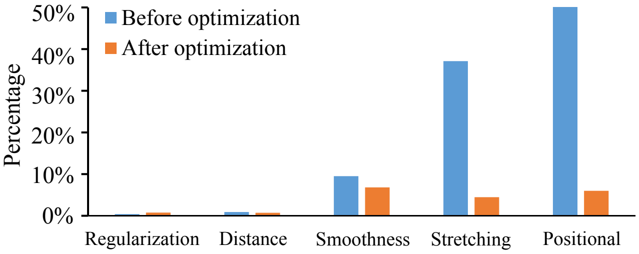


Fig. 7. The percentage of each cost term with respect to the initial objective, before and after optimization. This result is reported from the Carmen skirt example, which uses a strong smoothness cost coefficient to preserve designed skirt wrinkles.

## 6 COST FUNCTIONS

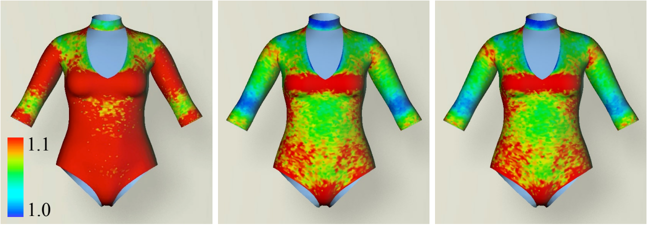
We define the objective $f$ as the sum of regularization functions and cost functions. The cost functions evaluate the fitting quality of $\mathbf{x}$ worn by $\phi$, given $\bar{\mathbf{x}} = \mathrm{Sim}_{\bar{\phi}}(\bar{\boldsymbol{\theta}})$ worn by $\bar{\phi}$ as a reference. Our system includes four cost functions and their contributions in the Carmen skirt example are displayed in Fig. 7. The user controls the strength of each cost function by a coefficient, which can be intuitively decided by the nature of a garment. For instance, the bodysuit uses little distance or smoothness cost, since it is supposed to fit tightly.

*Distance cost.* We use a distance cost function to compare the looseness of garment $\mathbf{x}$ with the looseness of $\bar{\mathbf{x}}$ on a per-vertex basis. Let $\mathbf{x}_i$ and $\bar{\mathbf{x}}_i$ be the positions of the same vertex $i$, and $\phi$ and $\bar{\phi}$ be the signed distance functions of the two bodies. We define this cost as:

$$C^{\mathtt{dis}} = \frac{\delta^{\mathtt{dis}}}{2} \sum_i \left|\phi(\mathbf{x}_i) - \bar{\phi}(\bar{\mathbf{x}}_i)\right|^2, \tag{11}$$

where $\delta^{\mathtt{dis}}$ controls the cost magnitude. Since the distance is subject to many factors other than the looseness, we typically choose a small $\delta^{\mathtt{dis}}$ to lessen its influence in the total objective.

*Smoothness cost.* The smoothness cost function is used to test whether the surface of $\mathbf{x}$ has the same smoothness as the surface of $\bar{\mathbf{x}}$. Our implementation of this function is based on the discrete hinge-edge bending model under an isometric deformation assumption [Garg et al. 2007]. Let $\{ei, ej, ek, el\}$ be the four vertices in the local neighborhood of a hinge edge $e \in \bar{\mathbf{p}}$. We formulate this cost



(a) Before optimization (b) After optimization (c) Reference

Fig. 8. Stretching ratios of the bodysuit example. By optimization, our system reduces the difference between the deformation of the bodysuit sewn from the adjusted pattern and worn by the target body, as shown in (b), and that of a bodysuit sewn from the base pattern and worn by the base body, as shown in (c).

function as:

$$C^{\mathrm{smo}} = \frac{\delta^{\mathrm{smo}}}{2} \sum_e \left( \|\mathbf{L}_e \mathbf{x}\| - s_e \, \|\mathbf{L}_e \bar{\mathbf{x}}\| \right)^2 , \qquad \mathbf{L}_e \mathbf{x} = L_{ei} \mathbf{x}_{ei} + L_{ej} \mathbf{x}_{ej} + L_{ek} \mathbf{x}_{ek} + L_{el} \mathbf{x}_{el}, \tag{12}$$

in which $\delta^{\mathrm{smo}}$ is a magnitude coefficient, and $\mathbf{L}_e$ is an edge smoothing kernel matrix defined by four constants $L_{ei}$, $L_{ej}$, $L_{ek}$, and $L_{el}$. Here we use $s_e$ to indicate whether both surfaces are convex/concave, or not. By using the dot product between the kernel edge product and the surface normal to determine if a surface is convex or concave, we calculate $s_e$ as:

$$s_e = \frac{(\mathbf{L}_e \mathbf{x} \cdot \mathbf{n}_e)(\mathbf{L}_e \bar{\mathbf{x}} \cdot \bar{\mathbf{n}}_e)}{\left|(\mathbf{L}_e \mathbf{x} \cdot \mathbf{n}_e)(\mathbf{L}_e \bar{\mathbf{x}} \cdot \bar{\mathbf{n}}_e)\right|}, \tag{13}$$

where $\mathbf{n}_e$ and $\bar{\mathbf{n}}_e$ are the surface normals of $e$, defined as the average of the two triangle normals.

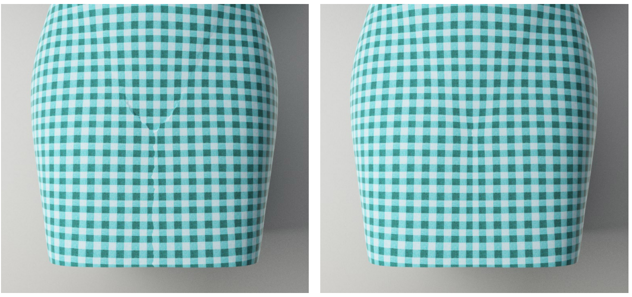
*Stretching cost.* When worn by the human body, garments made of elastic fabrics are often under constant stretching. The bodysuit in Fig. 8 is one such example. We want garment $\mathbf{x}$ to keep the same stretching ratio as $\bar{\mathbf{x}}$ on a per-edge basis. Let $\{\mathbf{x}_{ei}, \mathbf{x}_{ej}\}$ and $\{\mathbf{p}_{ei}, \mathbf{p}_{ej}\}$ be the two vertex positions of the same edge $e$ in the garment and the pattern. The stretching cost function is:

$$C^{\mathrm{str}} = \frac{\delta^{\mathrm{str}}}{2} \sum_e \left| \frac{\|\mathbf{x}_{ei} - \mathbf{x}_{ej}\|}{\|\mathbf{p}_{ei} - \mathbf{p}_{ej}\|} - \frac{\|\bar{\mathbf{x}}_{ei} - \bar{\mathbf{x}}_{ej}\|}{\|\bar{\mathbf{p}}_{ei} - \bar{\mathbf{p}}_{ej}\|} \right|^2 , \tag{14}$$

in which $\delta^{\mathrm{str}}$ is a coefficient controlling the cost magnitude. Fig. 8 demonstrates the effect of this cost on a bodysuit example. Here the color indicates the stretching ratio magnitude of local edges.

*Positional cost.* Finally, we would like to maintain the relative position of every garment vertex with respect to the human body. To do so, we first perform a deformation transfer of $\bar{\mathbf{x}}$ from being worn by $\bar{\phi}$ to being worn by $\phi$. Let $t_{i0}$ be the triangle of the base body mesh closest to $\bar{\mathbf{x}}_i$. We can use its vertex-triangle distance and its barycentric coordinates to determine its transferred vertex position $\tilde{\mathbf{x}}_{i0}$. The problem with this transfer though is the discontinuity, when $t_{i0}$ jumps from one local minimum to another, as shown in Fig. 9a. To solve this problem, we use the second local minimum triangle $t_{i1}$ and we calculate the final transferred vertex position as:

$$\tilde{\mathbf{x}}_i = \left(0.5 + \frac{\min(d_{i0} - d_{i1}, 2B)}{4B}\right) \tilde{\mathbf{x}}_{i0} + \left(0.5 - \frac{\min(d_{i0} - d_{i1}, 2B)}{4B}\right) \tilde{\mathbf{x}}_{i1}, \tag{15}$$



(a) Simple deformation transfer (b) Our deformation transfer

Fig. 9. The deformation transfer results. Simply transferring each cloth vertex by the deformation of its closest body triangle will cause discontinuity, due to the local minima on the legs and the torso as the bottom of this secretary dress shows. Our method lessens the problem by providing a smoothing transition between two local minima, as shown in (b).

in which $d_{i0}$ and $d_{i1}$ are the distances to the two triangles, and $B$ is a transition bandwidth. Fig. 9b shows the transferred garment using Equation 15 lessens the discontinuity problem. Once we obtain $\tilde{\mathbf{x}}_i$, we formulate this cost function as:

$$C^{\mathrm{pos}} = \frac{\delta^{\mathrm{pos}}}{2} \sum_i u_i \, \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 , \tag{16}$$

in which $\delta^{\mathrm{pos}}$ controls the overall magnitude and $u_i$ controls the contribution of vertex $i$. We typically make $u_i$ greater if vertex $i$ is on the boundary of a pattern patch, to emphasize the importance of seams and boundaries on the garment appearance.

## 7 OPTIMIZATION METHOD

Given the simulator, the parametrization, and the objective function in Section 4, 5 and 6, we now need to solve the minimization problem:

$$\boldsymbol{\theta} = \arg\min f \left( \mathtt{Sim}_{\phi}(\boldsymbol{\theta}), \boldsymbol{\theta}, \phi, \mathtt{Sim}_{\bar{\phi}}(\bar{\boldsymbol{\theta}}), \bar{\boldsymbol{\theta}}, \bar{\phi} \right), \tag{17}$$

in which $\boldsymbol{\theta}$ is the only variable and $f = R^{\mathrm{fix}} + R^{\mathrm{cur}} + C^{\mathtt{dis}} + C^{\mathrm{smo}} + C^{\mathrm{str}} + C^{\mathrm{pos}}$. Similar to [Casati et al. 2016], we construct our optimization method as a gradient-based line search method using two iterative steps. In the first step, the method performs quasistatic simulation to compute $\mathbf{x} = \mathtt{Sim}_{\phi}(\boldsymbol{\theta})$, the garment sewn from pattern $\boldsymbol{\theta}$ and worn by body $\phi$. In the second step, it calculates the gradient of the objective function and uses that to guide the update of the pattern parameters $\boldsymbol{\theta}$. The method repeats the two steps until the objective cannot be further reduced. Our focus here is on the second step.

By definition, we consider $f$ as a function of $\mathbf{x}$ and $\boldsymbol{\theta}$, in which $\mathbf{x}$ itself is a function of $\boldsymbol{\theta}$. The gradient of the objective $f$ is:

$$\left(\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\theta}}\right)^{\mathsf{T}} = \left(\frac{\partial \mathbf{p}}{\partial \boldsymbol{\theta}}\right)^{\mathsf{T}} \left(\frac{\partial f}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial f}{\partial \mathbf{p}}\right)^{\mathsf{T}}, \tag{18}$$

where $\partial f / \partial \mathbf{x}$ and $\partial f / \partial \mathbf{p}$ are directly obtainable from the definition of $f$, and $\partial \mathbf{p} / \partial \boldsymbol{\theta}$ is given by Equation 8 for $\boldsymbol{\theta} = \mathbf{p}^{\mathsf{B}}$. To further establish the relationship between $\mathbf{x}$ and $\mathbf{p}$, we use the condition that $\mathbf{x}$ has reached quasistatic equilibrium: $\mathbf{f}(\mathbf{p}, \mathbf{x}) = \mathbf{0}$, in which

**Algorithm 1:** Sewing Pattern Adjustment

```
Input: The base body $\bar{\phi}$, the base pattern $\bar{\boldsymbol{\theta}}$, the new body $\phi$
Output: The pattern parameters $\boldsymbol{\theta}$
$\bar{\mathbf{x}} \leftarrow \texttt{Sim}_{\bar{\phi}}(\bar{\boldsymbol{\theta}})$;
$\tilde{\mathbf{x}} \leftarrow \texttt{Transfer}(\bar{\mathbf{x}}, \bar{\phi}, \phi)$;
$\boldsymbol{\theta}^{(0)} \leftarrow \texttt{Initialize}(\bar{\boldsymbol{\theta}}, \bar{\mathbf{x}}, \tilde{\mathbf{x}})$;
$\mathbf{x}^{(0)} \leftarrow \texttt{Sim}_{\phi}(\boldsymbol{\theta}^{(0)})$;
$\mathbf{P} \leftarrow \texttt{Preconditioner}$;
Initialize $S$ and $s$;
for $k = 1 \ldots \infty$ do
    $\boldsymbol{\theta}^{(k)} \leftarrow \boldsymbol{\theta}^{(k-1)} - s\mathbf{P}^{-1}\texttt{Grad}\left(\mathbf{x}^{(k-1)}, \boldsymbol{\theta}^{(k-1)}\right)$;
    if $k \bmod M = 0$ then
        $\mathbf{x}^{(k)} \leftarrow \texttt{Sim}_{\phi}(\boldsymbol{\theta}^{(k)})$;
        if $f(\mathbf{x}^{(k)}, \boldsymbol{\theta}^{(k)}) > f(\mathbf{x}^{(k-M)}, \boldsymbol{\theta}^{(k-M)})$ then
            $k \leftarrow k - M$;
            Increase $S$ and decrease $s$;
            Continue;
        end
        else
            Decrease $S$ and increase $s$;
        end
    end
    else
        $\mathbf{x}^{(k)} \leftarrow \texttt{Sim}_{\phi}(\boldsymbol{\theta}^{(k)}, S)$;
    end
end
return $\boldsymbol{\theta}^{(k+1)}$;
```

$\mathbf{f} \in \mathbb{R}^{3N}$ is the stacked force. According to the implicit function theorem, we have:

$$\frac{\partial \mathbf{x}}{\partial \mathbf{p}} = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right), \tag{19}$$

under the assumption that the tangent stiffness matrix $\partial \mathbf{f}/\partial \mathbf{x}$ is invertible. We note that we do not need to explicitly evaluate $\partial \mathbf{x}/\partial \mathbf{p}$ or $\partial \mathbf{p}/\partial \mathbf{p}^{\mathsf{B}}$, since doing so requires matrix inversion. Instead we solve two linear systems in every gradient calculation. This technique is commonly known as the *adjoint method*.

While the basic methodology of our optimization system is not new and it has been explored in many areas, the real challenge is how to implement it with sufficient efficiency. The efficiency is of particular importance to the usefulness of our system, since users, i.e., pattern makers, may still need to tune strength coefficients for optimal pattern outcomes. In this section, we study the performance improvement of our system from four perspectives: initialization, an inexact strategy, acceleration, and CPU-GPU implementation.

*Initialization.* Like other iterative methods, our system requires a good initial guess to reach its solution fast. A naïve idea is to simply treat the base pattern as the initialization, but that often generates a large initial objective. Instead, we propose to treat the transferred garment $\tilde{\mathbf{x}}$ as a fixed quasistatic state $\mathbf{x}$, and run an incomplete optimization process with the stretching cost and the regularization only. Since $\mathbf{x} \equiv \tilde{\mathbf{x}}$ is fixed, this process avoids $\partial \mathbf{x}/\partial \mathbf{p}$ and it can be finished in only a few seconds. The resulting initial pattern reduces the stretching cost and the positional cost reasonably well, which are the two dominant ones in the objective function. In our experiment, the objective generated by this initial pattern is only one sixth to one tenth of the objective given by the base pattern.

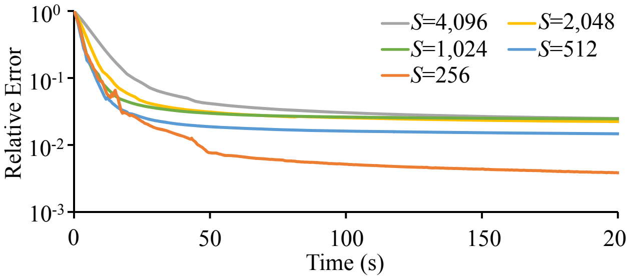
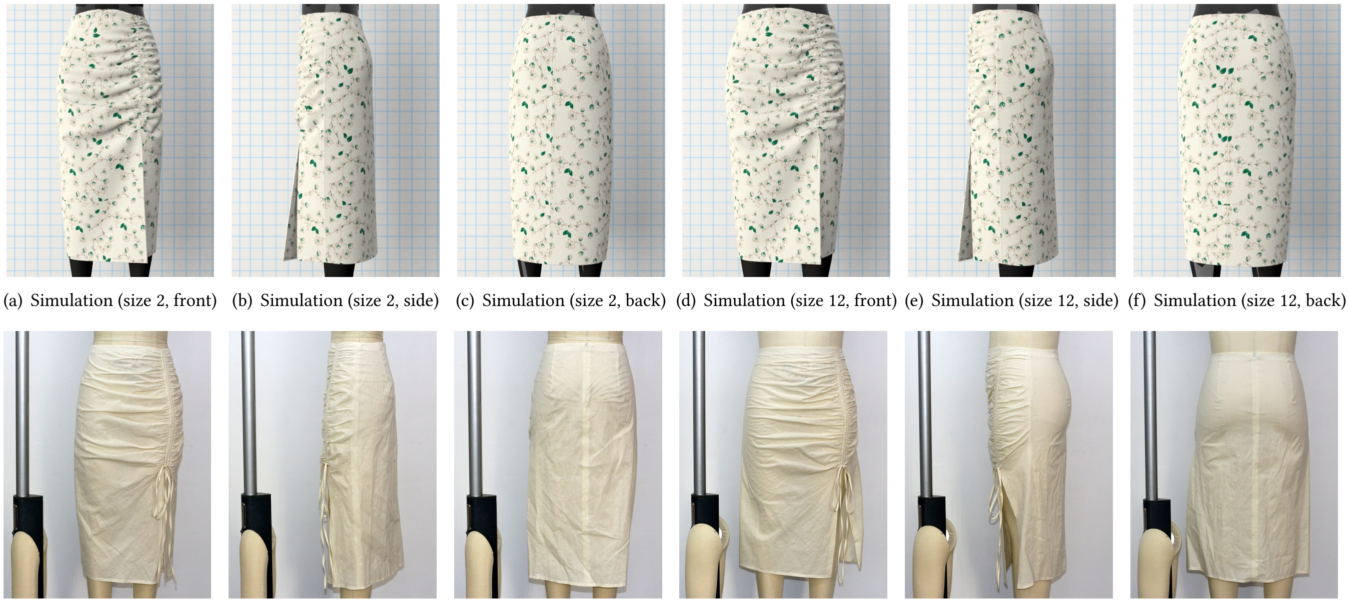


Fig. 10. The convergence of our system when choosing different $S$, the number of simulation iterations per outer iteration. In general, the system performs faster as $S$ decreases, before divergence happens. Here the relative error is defined as $(f - f^*)/(f^0 - f^*)$, where $f$ is the current objective, $f^0$ is the initial objective, and $f^*$ is the final objective after sufficient outer iterations.

*An inexact strategy.* The two-step optimization approach can be computationally expensive, if it reaches quasistatic equilibrium exactly in every outer (optimization) iteration as did in [Casati et al. 2016]. To reduce the computational cost without slowing down the convergence rate much, we propose to approximate the quasistatic state by using a smaller $S$, the number of simulation iterations per outer iteration, as shown in Fig. 10. This inexact idea has been investigated for solving inverse problems using a primal-dual formulation [Haber et al. 2004; Quirynen et al. 2017] before. To achieve better performance, we use a primal formulation and we must specifically address the balance between the efficiency and the divergence risk. Our solution is to adaptively adjust the accuracy of quasistatic simulation by long-range backtracking search [Nocedal and Wright 2006]. Specifically, after $M$ inexact outer iterations, e.g., $M = 32$, we run one exact outer iteration with sufficient simulation accuracy. If the total cost fails to drop since the last exact iteration, we increase the accuracy of quasistatic simulation, decrease the step length, and relaunch the optimization from the last exact outer iteration. In the worst case, our inexact two-step approach is reduced to the exact approach, which is a standard line search method. Therefore, our approach is guaranteed to converge, even if the objective increases temporarily due to inexactness.

*Acceleration.* Although we can use the calculated gradient to update the pattern parameters directly, we prefer not to do so, since gradient descent is known to have a low convergence rate. This is because the gradient often cannot act as a good search direction, which prevents the update from taking larger step lengths.

To speed up the convergence rate, a typical way is to multiply the gradient by the inverse of a positive definite preconditioning matrix $\mathbf{P}$. Given the cost functions defined in a nonlinear least squares fashion, we formulate the objective into: $f(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{v}^{\mathsf{T}}\mathbf{v} + R(\boldsymbol{\theta})$, in



(a) Simulation (size 2, front) (b) Simulation (size 2, side) (c) Simulation (size 2, back) (d) Simulation (size 12, front) (e) Simulation (size 12, side) (f) Simulation (size 12, back)

(g) Real skirt (size 2, front) (h) Real skirt (size 2, side) (i) Real skirt (size 2, back) (j) Real skirt (size 12, front) (k) Real skirt (size 12, side) (l) Real skirt (size 12, back)

Fig. 11. The automatic grading results of a Carmen skirt example. Our system can flexibly handle complex patterns designed to create garments with shirring, as demonstrated on the left side of this Carmen skirt.

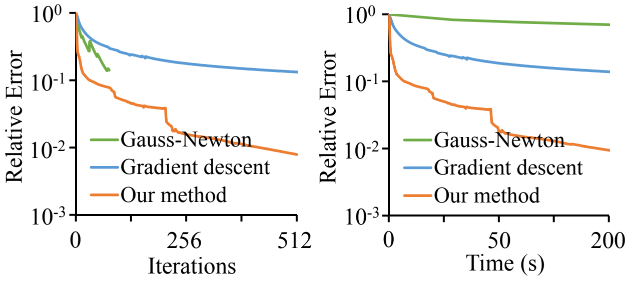


Fig. 12. The convergence of different optimization methods. Our method uses the approximate Hessian of the regularization terms as the preconditioner. Compared with plain gradient descent, our method has a faster convergence rate; and compared with Gauss-Newton, our method uses a small preconditioning cost per iteration.

which $\mathbf{v}$ is a stacked cost vector and $R(\boldsymbol{\theta})$ contains all of the regularization terms. For simplicity, we consider the positional cost only, in which case $\mathbf{v}$ is a function of $\mathbf{x}$. When $\mathbf{P}$ is the Hessian matrix of the objective $f$:

$$\mathbf{P} = \frac{\partial}{\partial \boldsymbol{\theta}} \left( \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}} \right)^{\mathsf{T}} \mathbf{v} + \left( \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}} \right)^{\mathsf{T}} \left( \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}} \right) + \frac{\partial}{\partial \boldsymbol{\theta}} \left( \frac{\partial R}{\partial \boldsymbol{\theta}} \right)^{\mathsf{T}}, \quad (20)$$

the method becomes Newton's method. The main issue with Newton's method is that it is extremely complicated to evaluate the first term in Equation 20, especially due to the bending force. In practice, researchers often ignore the first term and choose to use the Gauss-Newton method or its variations, such as Levenberg-Marquardt. To avoid the calculation of $(\partial \mathbf{f}/\partial \mathbf{x})^{-1}$ in the second term, Bickel and colleagues [2009] evaluated $(\partial \mathbf{v}/\partial \mathbf{x})(\partial \mathbf{x}/\partial \theta_i)$ separately for every parameter $\theta_i$, each of which requires solving a linear system with matrix $\partial \mathbf{f}/\partial \mathbf{x}$. While this matrix can be factorized only once per iteration, the evaluation of the second term still takes approximately 58 seconds for the Carmen skirt example in Fig. 11, as demonstrated in our experiment. Unfortunately, the use of the second term fails to significantly improve the convergence rate and worsens the overall performance, as shown in Fig. 12. This is not surprising, given the fact that strong regularization used in our system makes the norm of the third term at least 1,000 times greater than that of the second term. In the end, we choose to ignore the first two terms and use the third term only, which is the Hessian of $R$. To further reduce the computational cost, we treat the rightmost term in Equation 10 as constant and derive the approximate Hessian as the preconditioner $\mathbf{P}$, similar to the local/global geometric approach [Brouet et al. 2012; Liu et al. 2008] and projective dynamics [Bouaziz et al. 2014; Liu et al. 2013]. Since the resulting matrix is constant, we pre-factorize it for fast runtime computation. Fig. 12 shows our method outperforms plain gradient descent and Gauss-Newton.

Besides preconditioning, we also implemented L-BFGS acceleration. Although it can accelerate the convergence occasionally, it introduces instability into the search direction, which can trigger more frequent long-range backtracking steps. Overall, we think its benefit is limited and we suggest not to use it. We note that L-BFGS acceleration has an unpleasant side effect: pattern patches often become rotated after optimization, as exhibited in our experiment. These rotations must be removed for orientation-specific patterns.

*CPU-GPU Implementation.* We implement our system on a hybrid CPU-GPU architecture. Specifically, the system performs quasistatic simulation, iterative linear solver, and Jacobian matrix

Table 1. Statistics and timings of our examples. Here the timings do not include precomputation or initialization time, which is typically a few seconds. The cost strength coefficients are provided for $\delta^{\mathtt{fix}}$, $\delta^{\mathtt{cur}}$, $\delta^{\mathtt{dis}}$, $\delta^{\mathtt{smo}}$, $\delta^{\mathtt{str}}$, $\delta^{\mathtt{pos}}$, respectively.

| Example Name<br>(#Vert., #Tri., #Patches ) | Target<br>Body | Strength<br>Coefficients (1/m$^2$) | Objective<br>Reduction | Maximum<br>Displacement | Opt. Time<br>per Iteration | Time<br>per Iteration | # of<br>Iterations | Total<br>Time |
|---|---|---|---|---|---|---|---|---|
| Secretary Dress<br>(34K, 66K, 8) | Size 2<br>Size 12 | 100, $10^6$<br>1, 0.05, 1, 100 | 37.7%<br>16.6% | 2.4mm<br>5.4mm | 0.267s | 0.451s | 256 | 115s |
| Bodysuit<br>(22K, 42K, 9) | Size 2<br>Size 12 | 100, $10^6$<br>0.01, $5 \times 10^{-5}$, 5, 100 | 72.6%<br>43.0% | 1.2mm<br>6.3mm | 0.149s | 0.284s | 256 | 72.7s |
| Carmen Skirt<br>(30K, 59K, 4) | Size 2<br>Size 12 | 100, $10^6$<br>0.01, 0.01, 1, 100 | 70.0%<br>18.8% | 9.9mm<br>20.8mm | 0.290s | 0.446s | 512 | 228s |
| Alana Jumpsuit<br>(58K, 110K, 19) | Alex<br>Jenny | 100, $10^6$<br>1, 0.05, 1, 100 | 32.8%<br>31.2% | 17.4mm<br>33.5mm | 0.502s | 0.800s | 256 | 201s |
| Pants<br>(85K, 163K, 29) | Jiang<br>Zhou | 100, $10^6$<br>1, 0.05, 1, 100 | 43.7%<br>22.8% | 14.3mm<br>14.6mm | 0.663s | 1.147s | 256 | 294s |

evaluation on the GPU, while the rest on the CPU. Our simulator and our iterative linear solver are based on the Jacobi method with Chebyshev acceleration, which are highly parallelizable on the GPU.

A special feature of our system implementation is about the Jacobian matrix of the force with respective to the sewing pattern: $\partial\mathbf{f}/\partial\mathbf{p}$. It includes the Jacobian matrix of the gravitational force, the Jacobian matrix of the spring force, and the Jacobian matrix of the bending force. Unlike $\partial\mathbf{f}/\partial\mathbf{x}$, this matrix is asymmetric and much more complicated to evaluate. Fortunately, since the use of this matrix is to perform a matrix-vector product with the adjoint state, we can avoid explicit evaluation of this matrix and implement the matrix-vector product procedure even more conveniently on the GPU. In the Carmen skirt example, the Jacobian matrix evaluation takes 150ms and the matrix-free product takes 73ms on the CPU. In comparison, the matrix-free product takes only 3ms on the GPU.

## 8 RESULTS

Our system uses the Intel MKL PARDISO library for CPU computation and the CUDA library for GPU computation. Our experiment was performed on an Intel Core i7-5820K 3.3GHz processor and an NVIDIA GeForce GTX TITAN X Graphics Card. Table 1 summarizes the statistics and the timings of our examples. It shows that the system is able to sufficiently reduce the objective within 256 to 512 iterations. The overall objective reduction ratio is highly example-specific and it depends on the quality of the initialization. In general, the initialization is more effective for slim-fit and second-skin garments, such as the bodysuit example shown in Fig. 17. The initialization is also more effective when the target body is slimmer than the base body.

We treat the Size 6 female form made by alvanon® as our base body by default. Given the measurements of a human body, we use our in-house human body deformation software to generate the corresponding human body mesh. This software solves a nonlinear optimization problem, whose objective is to minimize body deformation and the difference between body landmarks and the measurements at the same time. Given the body mesh, we convert it into a signed distance field for fast cloth-body collision handling, as discussed in Section 4. The resolution of the signed distance field is 2.5mm in our experiment.

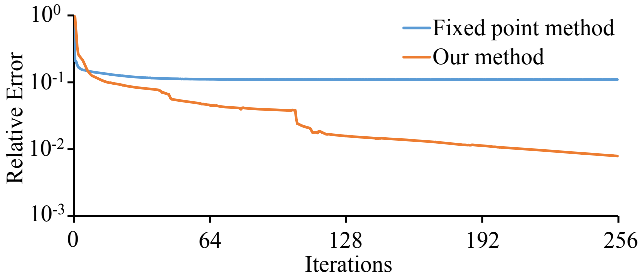
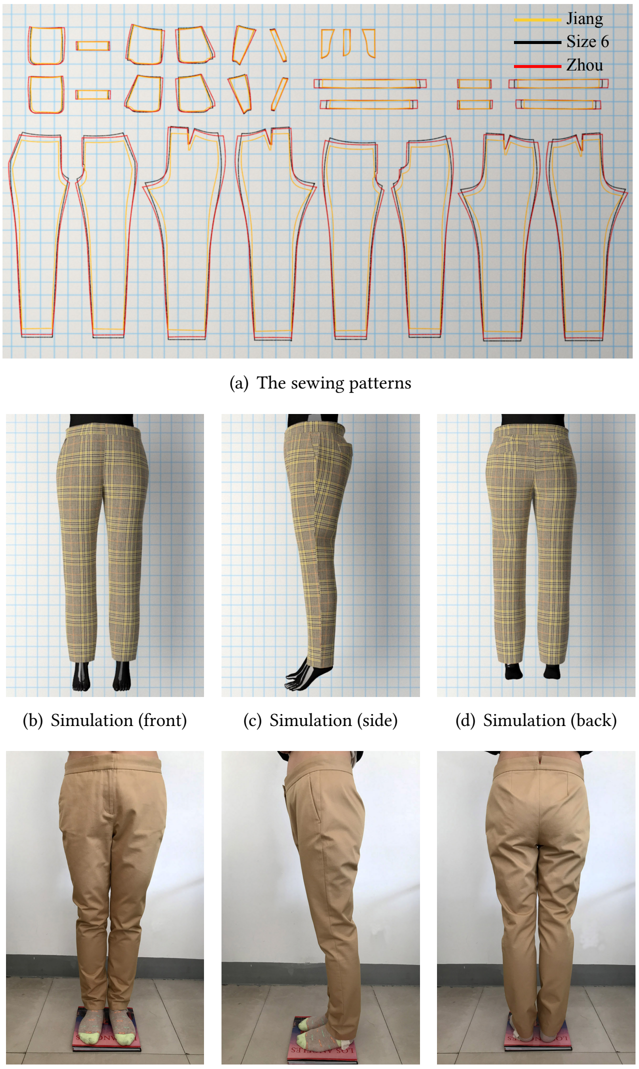


Fig. 13. The outcomes of the fixed point method and our method. Unlike the fixed point method, our method does not stagnate and it is able to converge, given sufficient computational time.

We use several fabrics in our grading and made-to-measure tests, to be discussed later in Subsection 8.3 and 8.4. The white semi-transparent fabric, shown in Fig. 1, is cotton woven fabric used by pattern makers for regular test purposes. The thick fabric used in the pants example in Fig. 14 is the actual woven fabric for production. Finally, the white fabric used in the bodysuit example in Fig. 17 is made of spandex and it can stretch by approximately 20 percent.

### 8.1 Comparison to a Fixed Point Method

The strength of the fixed point method [Bartle et al. 2016] is that it can treat the simulator as a black box. On the other hand, its weaknesses are also obvious: it is developed for solving the inverse pattern design problem (in Equation 3) and it misses the objective, if the problem does not have a solution. To compare the fixed point method with our method, we implement it on a mass-spring system as well and we tune our system to minimize the positional cost only, as did before in Section 3. Fig. 13 shows the fixed point method reduces the objective rapidly in the first few iterations, but then it stagnates. We find stagnation occurs, no matter whether the inverse problem has an exact solution or not. In contrast, as a line search method, our method converges after a sufficient number of iterations. We note that our method should be able to reduce the objective even further, if it treats all of the vertices as the parameters, like the fixed point method does.



(a) The sewing patterns

(b) Simulation (front)

(c) Simulation (side)

(d) Simulation (back)

(e) Real pants (front)

(f) Real pants (side)

(g) Real pants (back)

Fig. 14. The made-to-measure results of a pants example. In this example, the pants are made of multiple fabric layers and the sewing pattern contains 29 patches. By ignoring self collision, our system produces the sewing pattern that matches the body of Zhou.

## 8.2 A Multi-Layered Case

Many garments are made of multiple fabric layers. Even a simple pants example shown in Fig. 14 contains an outer layer and an inner layer, and its pockets are made by two additional layers. As shown in Table 1, this example contains 29 pattern patches. Under the assumption that self intersection among these layers have limited influence on patch shapes and garment seams, we ignore self intersection in simulation and optimization, as mentioned in Subsection 4.1. Fig 14 shows our system is able to optimize the sewing pattern in this multi-layered example, and the resulting fitting quality is plausible, according to pattern makers and the individual.

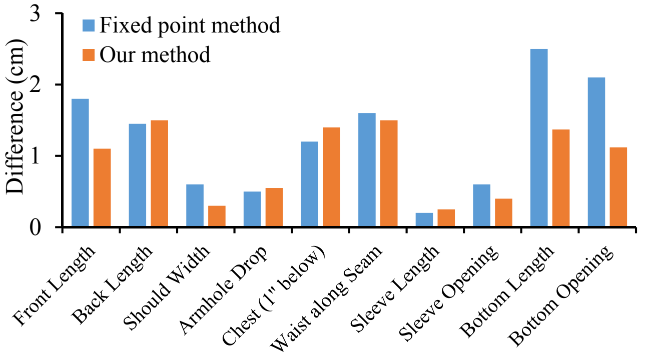


Fig. 15. The maximum differences between automatically and manually graded secretary dress patterns in Size 10, 12, and 14. This plot shows that, in general, our method is more precise than the fixed point method.

An interesting idea we have not explored yet is to optimize pattern patches in batches. The outer layer of a garment is often stiffer and it determines the overall garment shape more. This suggests that we can optimize outer patches first, and then optimize inner patches with the outer layer being fixed. We expect this idea to improve both the result quality and the system performance.

## 8.3 Grading Tests

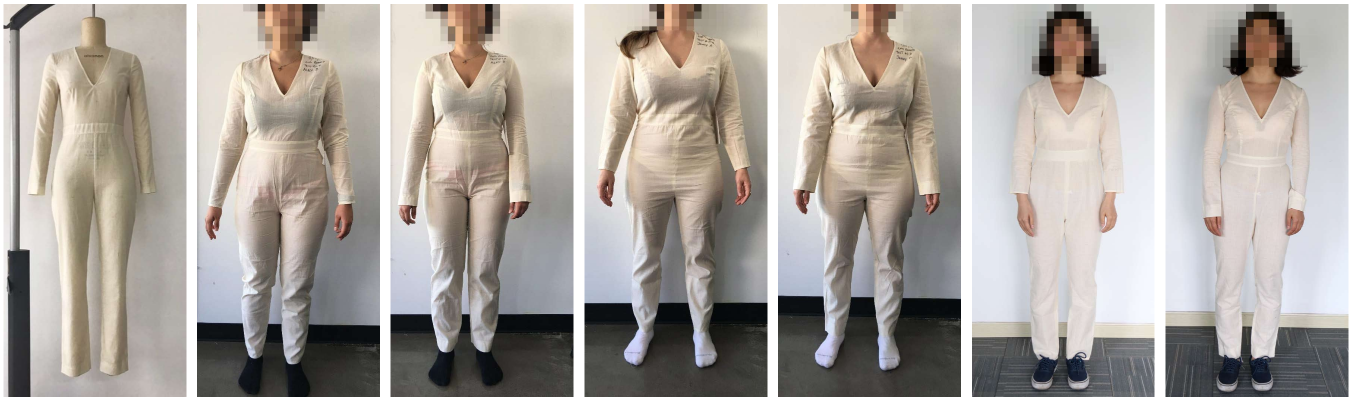
Our first real-world experiment is to test whether the system can perform automatic grading to generate sewing patterns for other standardized bodies. In this experiment, we use the standard forms made by Alvanon®, whose measurements are accessible from the Alvanon website [2018]. Since these bodies are standardized with similar proportions, our grading results look promising and plausible, as shown in Fig. 1, 11, and 17.

To quantitatively evaluate our results, we compare them with manually graded patterns using the distances between pattern landmarks. Manually graded patterns are created by standard grading rules and verified by human body tests, so they are sufficiently precise. Fig. 15 shows that the maximum differences between automatically and manually graded results are less than 1.5cm. Therefore, our results are also precise. In comparison, the results generated by the fixed point method [Bartle et al. 2016] differ more significantly from manual results. We note that manual results contain manmade errors as well and they should not be simply treated as the ground truth. But at least, the experiment indicates the precision of our system is more reliable and acceptable for garment production.

One issue we noticed from the experiment that the system tends to magnify existing artifacts in the base pattern, especially when the target body is larger than the base body. For instance, a curvy seam on the base garment will become more curvy on the adjusted garment. This is because the system cannot identify artifacts and it preserves them as design features instead. To avoid this problem, we must develop the base pattern to meet a higher standard.

## 8.4 Made-to-Measure Tests

In the next experiment, we evaluate the ability of our system to customize patterns for people with various body shapes and proportions. This experiment involves two sewing pattern designs and



(a) Base garment (b) Alex (ours) (c) Alex (rule-based) (d) Jenny (ours) (e) Jenny (rule-based) (f) Liu (ours) (g) Liu (rule-based)

Fig. 16. The made-to-measure results of the jumpsuit example. Compared with a rule-based system, our new system produces more plausible and consistent jumpsuit patterns for people with different body shapes and proportions.

Table 2. Part of the fitting results of the jumpsuit example. This table shows that our system suffers fewer minor and major fitting issues than our previous rule-based system. Major issues are highlighted in bold.

| Name | Our System | Rule-based System |
|---|---|---|
| Alex | Small cuffs | **Long and tight sleeves** |
| | Excessive front rise | **Tight thighs (cannot sit)** |
| | | **Cannot move arms easily** |
| | Curved pants seams | Curved pants seams |
| | Excessive back top | Excessive back top |
| Jenny | Small cuffs | **Long and tight sleeves** |
| | | **Cannot move arms easily** |
| | Curved pants seams | Curved pants seams |
| | Excessive back top | Excessive back top |
| Liu | High arm drops | **Tight waist** |
| | Excessive hip | **Too long sleeves** |
| | | Excessive back top |
| | | Loose chest |
| | | Short front top |
| | | Wrong shoulder slope |

eight individuals, whose bodies are measured by hand with a 1/4 inch precision. Some of the individuals, such as Alex, have disproportionate body shapes and struggle to find tightly fit garments in the market. To perform sewing pattern adjustment, we use both our system and a rule-based system, the latter of which requires pattern makers to manually specify a number of rules, i.e., the relationships between body measurements and pattern landmarks.

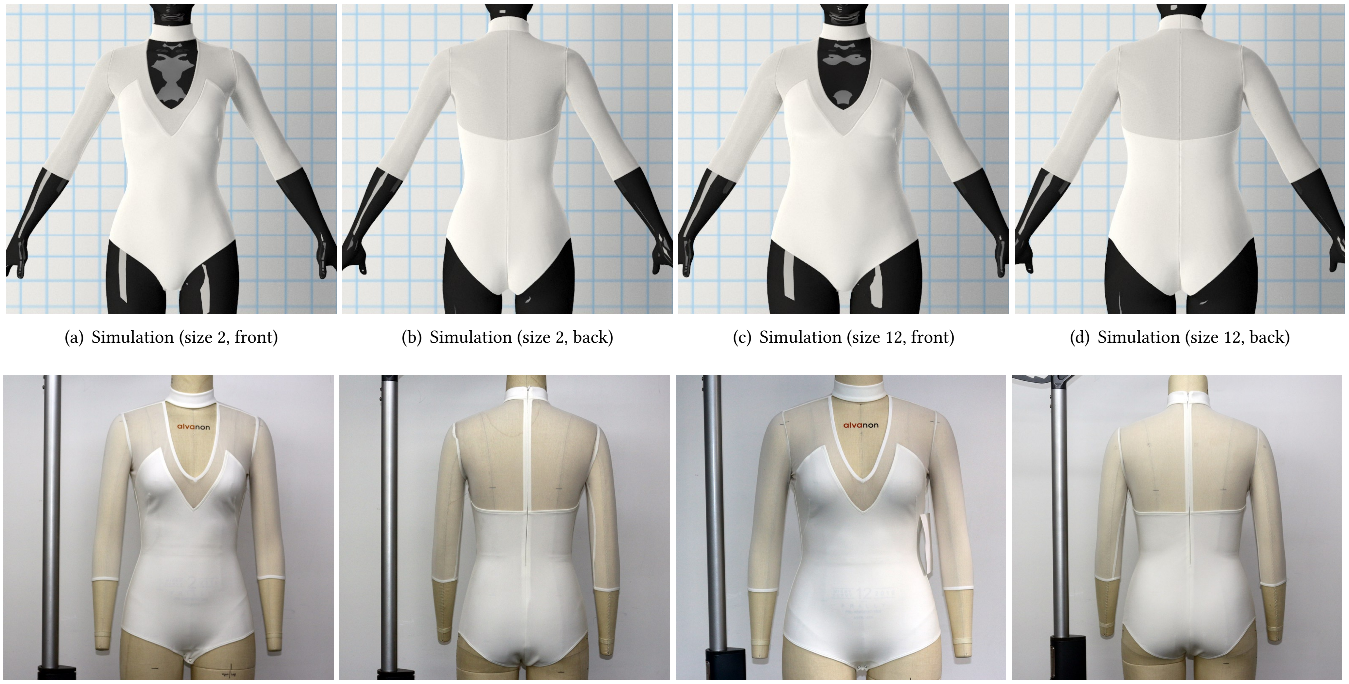
First, our pattern makers confirm that the pattern results produced by both systems are balanced and free of noticeable errors. Next we ask the individuals to wear the garments, as shown in Fig. 14 and 16, and we evaluate the fitting quality together with the pattern makers, the designers, and the individuals. Part of our results are shown in Table 2. In general, the rule-based system suffers from many major issues, such as long and tight sleeves, excessive wrinkles on the back, and short rise that may even prevent the individual from wearing the garment. These issues can be quickly identified by the pattern makers and the designers, without feedbacks from the individuals. Overall, the rule-based system is not satisfactory, as the rules overly simplify the body-pattern relationship and fail to address many important body shape aspects. In comparison, our system provides better fits and our results are free of major issues. Our results do still contain several minor issues, which we believe are largely caused by insufficient body measurements, as listed below.

- **Armholes.** The fitting quality near the shoulders relies on many measurements, including shoulder width, bicep, shoulder slope, armhole circumference, and armhole drop. Currently, our system considers shoulder width and bicep.
- **Crotch.** The fitting quality near the crotch affects the sitting comfort. It can be affected by high hip, low hip, front rise, and back rise. Currently, we use high hip and the total rise only.
- **Cuff, knee, and calf.** The quality near the aforementioned regions requires not only more measurements, but also more details on the body shape. It is also related to an individual's standing, sitting, and walking habits.

We would like to emphasize that our made-to-measurement experiment is much more ambitious than it should be. In practice, we can apply standard grading rules to obtain the standard-sized pattern closest to the body first, and then perform customized pattern adjustment from there. Since standard grading rules have been well justified for standard-sized bodies, doing this should produce better pattern results than our current practice does, which starts the adjustment directly from the base pattern in Size 6.

## 8.5 Limitations

The greatest limitation of our pattern adjustment system is that it depends heavily on the accuracy of the human body model. Hand measurements have errors, and they can be inconsistent when they are measured by different people. A natural solution is to build



(a) Simulation (size 2, front) (b) Simulation (size 2, back) (c) Simulation (size 12, front) (d) Simulation (size 12, back)

(e) Bodysuit (size 2, front) (f) Bodysuit (size 2, back) (g) Bodysuit (size 12, front) (h) Bodysuit (size 12, back)

Fig. 17. The grading results of a bodysuit example. Unlike other garments, a bodysuit is designed to fit the body tightly and its fabric patches are constantly stretched, when it is worn by the body.

the human body model from 3D scan. The problem though is that the scanned waist is often larger than it should be, since the belly tends to be more relaxed during the scanning process. An alternative solution is to use smart bodysuits with stretch sensors, such as the ZOZOSUIT [2018]. While it addresses the waist issue, it can be difficult to measure the chest with accuracy. Having said that, 3D scan and smart bodysuits are still advantageous for their consistency and they can become more useful in the future, once they are strengthened by measurement corrections.

So far, our system is developed for human bodies with limited differences from the base body. If body differences are large, sewing patterns may need to be significantly modified with remeshing. The system does not consider complex fabric material properties. In reality, pattern makers often develop different sewing patterns for the same garment made of different fabrics. We are not sure how significant the material factor is yet. Like many other line search methods, our system suffers from the local minimum issue. This issue is demonstrated when different initializations lead to different results with similar objectives sometimes. Our system relies on its simulator and it suffers from the same issues as the simulator does, such as the locking issue. The simulator and the system are suitable for single-layered garments and simple multi-layered garments. We expect the modeling, simulation, and optimization of complex multi-layered garments, such as multi-layered wedding dresses, to be much more difficult. Our system does not consider the consistency of fabric textures along seams, and it may cause fabric textures to become mismatched after optimization. To address this issue, we must introduce texture-specific regularization terms. Currently, we ask pattern makers to manually create annotations and seam allowances, both of which are required for production.

## 9 CONCLUSIONS AND FUTURE WORK

In this paper, we present a novel and practical system that customizes sewing patterns for a new target human body, without any user-specified rule. This is achieved by using a series of generic cost functions to describe the fitting quality of a garment and formulating pattern adjustment into a parameter optimization problem. Our work reveals that many nonlinear optimization ideas are applicable to the development of this system, and our experiment confirms its effectiveness and efficiency in handling various pattern designs.

We are interested in using more accurate elastic models in our simulator, once elasticity measurement of cloth becomes more precise and accessible. To do so, we can simply replace the elastic energy in Equation 5 by a hyperelastic strain energy. The stretching cost function can be modified to use a continuum-based model as well, if we replace stretching ratios by deformation gradients.

In the near future, we plan to address the imminent issues related to garment production, such as annotation, seam allowance, fabric material, and additional regularization and cost functions. We then would like to study pattern adjustment for more body types, especially plus-sized bodies. To put our system into practical use, we need more system evaluations and fitting tests. In the long term, we plan to investigate automatic sewing pattern development and adjustment from designs, rather than base patterns. If successfully deployed, such a technique will greatly shorten the production process from design to manufacturing.

## ACKNOWLEDGMENTS

I would like to thank Sheng Yang for software testing, Meng Zhang for pattern preparation and production, Justyna Smart and Angelina Aiken for garment testing and evaluation, and the R&D team and the production team at Frilly Inc. for supporting this research. This work is partially funded by collaborative gifts from Adobe and NVIDIA.

## REFERENCES

Alvanon. 2018. Alvanon Form Measurements. https://alvanon.com/measurements-specs/. Online; accessed April 16, 2018.

David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. 43–54.

Aric Bartle, Alla Sheffer, Vladimir G. Kim, Danny M. Kaufman, Nicholas Vining, and Floraine Berthouzoz. 2016. Physics-driven Pattern Adjustment for Direct 3D Garment Editing. *ACM Trans. Graph. (SIGGRAPH)* 35, 4, Article 50 (July 2016), 11 pages.

Miklos Bergou, Max Wardetzky, David Harmon, Denis Zorin, and Eitan Grinspun. 2006. A Quadratic Bending Model for Inextensible Surfaces. In *Proceedings of SGP*. 227–230.

Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Wojciech Matusik, Hanspeter Pfister, and Markus Gross. 2009. Capture and Modeling of Non-linear Heterogeneous Soft Tissue. *ACM Trans. Graph. (SIGGRAPH)* 28, 3, Article 89 (July 2009), 9 pages.

Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph. (SIGGRAPH)* 33, 4, Article 154 (July 2014), 11 pages.

R. Bridson, S. Marino, and R. Fedkiw. 2003. Simulation of Clothing with Folds and Wrinkles. In *Proceedings of SCA*. 28–36.

Remi Brouet, Alla Sheffer, Laurence Boissieux, and Marie-Paule Cani. 2012. Design Preserving Garment Transfer. *ACM Trans. Graph. (SIGGRAPH)* 31, 4, Article 36 (July 2012), 11 pages.

Romain Casati, Gilles Daviet, and Florence Bertails-Descoubes. 2016. *Inverse Elastic Cloth Design with Contact and Friction.* Technical report. Inria Grenoble Rhône-Alpes, Université de Grenoble.

Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but Responsive Cloth. *ACM Trans. Graph. (SIGGRAPH)* 21, 3 (July 2002), 604–611.

Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A. Otaduy. 2014. Yarn-level Simulation of Woven Cloth. *ACM Trans. Graph. (SIGGRAPH Asia)* 33, 6, Article 207 (Nov. 2014), 11 pages.

Philippe Decaudin, Dan Julius, Jamie Wither, Laurence Boissieux, Alla Sheffer, and Marie-Paule Cani. 2006. Virtual Garments: A Fully Geometric Approach for Clothing Design. *Computer Graphics Forum (Eurographics)* 25, 3 (2006), 625–634.

Akash Garg, Eitan Grinspun, Max Wardetzky, and Denis Zorin. 2007. Cubic shells. In *Proceedings of SCA*. 91–98.

Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. 2007. Efficient Simulation of Inextensible Cloth. *ACM Trans. Graph. (SIGGRAPH)* 26, 3, Article 49 (July 2007).

Peng Guan, Loretta Reiss, David A. Hirshberg, Alexander Weiss, and Michael J. Black. 2012. DRAPE: DRessing Any PErson. *ACM Trans. Graph. (SIGGRAPH)* 31, 4, Article 35 (July 2012), 10 pages.

Eldad Haber, Uri M. Ascher, and Douglas W. Oldenburg. 2004. Inversion of 3D Electromagnetic Data in Frequency and Time Domain Using an Inexact All-At-Once Approach. *Geophysics* 69, 5 (2004), 1216–1228.

Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. 2007. Harmonic Coordinates for Character Articulation. *ACM Trans. Graph. (SIGGRAPH)* 26, 3, Article 71 (July 2007).

Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2010. Efficient Yarn-based Cloth with Adaptive Contact Linearization. *ACM Trans. Graph. (SIGGRAPH)* 29, 4, Article 105 (July 2010), 10 pages.

Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J. Gortler. 2008. A Local/Global Approach to Mesh Parameterization. In *Proceedings of SGP*. 1495–1504.

Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan. 2013. Fast Simulation of Mass-spring Systems. *ACM Trans. Graph. (SIGGRAPH Asia)* 32, 6, Article 214 (Nov. 2013), 7 pages.

E. Lund, H. Moller, and L.A. Jakobsen. 2003. Shape Design Optimization of Stationary Fluid-Structure Interaction Problems with Large Displacements and Turbulence. *Structural and Multidisciplinary Optimization* 25, 5–6 (Dec. 2003), 383–392.

Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. 2014. Unified Particle Physics for Real-time Applications. *ACM Trans. Graph. (SIGGRAPH)* 33, 4, Article 153 (July 2014), 12 pages.

Yuwei Meng, Charlie C. L. Wang, and Xiaogang Jin. 2012. Flexible Shape Control for Automatic Resizing of Apparel Products. *Comput. Aided Des.* 44, 1 (Jan. 2012), 68–76.

E. Miguel, D. Bradley, B. Thomaszewski, B. Bickel, W. Matusik, M. A. Otaduy, and S. Marschner. 2012. Data-Driven Estimation of Cloth Simulation Models. *Comput. Graph. Forum* 31, 2pt2 (May 2012), 519–528.

Eder Miguel, David Miraut, and Miguel A. Otaduy. 2016. Modeling and Estimation of Energy-Based Hyperelastic Objects. *Computer Graphics Forum (Eurographics)* 35, 2 (May 2016), 385–396.

Eder Miguel, Rasmus Tamstorf, Derek Bradley, Sara C. Schvartzman, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Steve Marschner, and Miguel A. Otaduy. 2013. Modeling and Estimation of Internal Friction in Cloth. *ACM Trans. Graph. (SIGGRAPH Asia)* 32, 6, Article 212 (Nov. 2013), 10 pages.

Matthias Müller. 2008. Hierarchical Position Based Dynamics. In *Proceedings of VRIPHYS*. 1–10.

Rahul Narain, Matthew Overby, and George E. Brown. 2016. ADMM $\supseteq$ Projective Dynamics: Fast Simulation of General Constitutive Models. In *Proceedings of SCA*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 21–28.

Rahul Narain, Armin Samii, and James F. O'Brien. 2012. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Trans. Graph. (SIGGRAPH)* 31, 6, Article 152 (Nov. 2012), 10 pages.

Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization (2rd Ed.).* Springer.

Rien Quirynen, Sebastien Gros, and Moritz Diehl. 2017. Inexact Newton-Type Optimization with Iterated Sensitivities. *SIAM Journal on Optimization* 28, 1 (2017), 74–95.

Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. 2014. Designing Inflatable Structures. *ACM Trans. Graph. (SIGGRAPH)* 33, 4, Article 63 (July 2014), 10 pages.

Min Tang, Huamin Wang, Le Tang, Ruofeng Tong, and Dinesh Manocha. 2016. CAMA: Contact-Aware Matrix Assembly with Unified Collision Handling for GPU-based Cloth Simulation. *Computer Graphics Forum (Eurographics)* 35, 2 (2016), 511–521.

Maxime Tournier, Matthieu Nesme, Benjamin Gilles, and François Faure. 2015. Stable Constrained Dynamics. *ACM Trans. Graph. (SIGGRAPH)* 34, 4, Article 132 (July 2015), 10 pages.

Nobuyuki Umetani, Danny M. Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive Couture for Interactive Garment Modeling and Editing. *ACM Trans. Graph. (SIGGRAPH)* 30, 4, Article 90 (July 2011), 12 pages.

Pascal Volino, Nadia Magnenat-Thalmann, and Francois Faure. 2009. A Simple Approach to Nonlinear Tensile Stiffness for Accurate Cloth Simulation. *ACM Trans. Graph.* 28, 4, Article 105 (Sept. 2009), 16 pages.

Charlie Wang, Yu Wang, and M. Yuen. 2005. Design Automation for Customized Apparel Products. *Comput. Aided Des.* 37, 7 (June 2005), 675–691.

Huamin Wang. 2015. A Chebyshev Semi-iterative Approach for Accelerating Projective and Position-based Dynamics. *ACM Trans. Graph. (SIGGRAPH Asia)* 34, 6, Article 246 (Oct. 2015), 9 pages.

Huamin Wang, James F. O'Brien, and Ravi Ramamoorthi. 2011. Data-driven Elastic Models for Cloth: Modeling and Measurement. *ACM Trans. Graph. (SIGGRAPH)* 30, 4, Article 71 (July 2011), 12 pages.

Huamin Wang and Yin Yang. 2016. Descent Methods for Elastic Body Simulation on the GPU. *ACM Trans. Graph. (SIGGRAPH Asia)* 35, 6, Article 212 (Nov. 2016), 10 pages.

ZOZO. 2018. ZOZOSUIT. https://zozosuit.com/teaser. Online; accessed April 16, 2018.