Physics-driven Pattern Adjustment for Direct 3D Garment Editing

Aric Bartle[1] Alla Sheffer[2] Vladimir G. Kim[3] Danny M. Kaufman[3] Nicholas Vining[2] Floraine Berthouzoz[3]
[1]Stanford University [2]University of British Columbia [3]Adobe Research

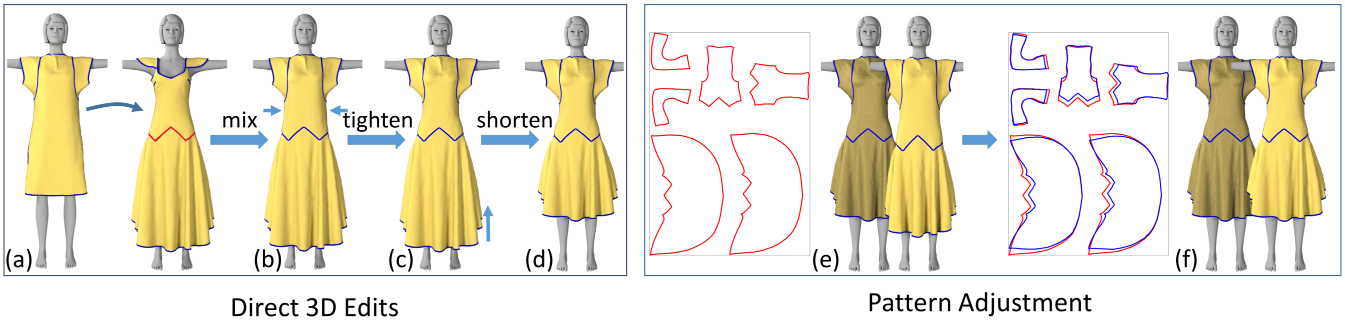


Figure 1: *Direct 3D garment editing. (left) In our UI a designer can mix garments, change fit and length with a handful of mouse click-and-drag operations. Our algorithm then computes a target garment that satisfies user constraints while preserving the source style. (Right) Resimulation of the resulting garment using patterns generated by 2D parameterization (here [Liu et al. 2008]) (red) introduces significant deviation from the 3D target (shown for reference in the back), our automatically adjusted 2D patterns (blue) ensure that the resulting garments retain the desired shape after physical simulation. Computation and interaction combined start-to-end took under 5 minutes.*

Abstract

Designers frequently reuse existing designs as a starting point for creating new garments. In order to apply garment modifications, which the designer envisions in 3D, existing tools require meticulous manual editing of 2D patterns. These 2D edits need to account both for the envisioned geometric changes in the 3D shape, as well as for various physical factors that affect the look of the draped garment. We propose a new framework that allows designers to directly apply the changes they envision in 3D space; and creates the 2D patterns that replicate this envisioned target geometry when lifted into 3D via a physical draping simulation. Our framework removes the need for laborious and knowledge-intensive manual 2D edits and allows users to effortlessly mix existing garment designs as well as adjust for garment length and fit. Following each user specified editing operation we first compute a *target* 3D garment shape, one that maximally preserves the input garment's style–its proportions, fit and shape–subject to the modifications specified by the user. We then automatically compute 2D patterns that recreate the target garment shape when draped around the input mannequin within a user-selected simulation environment. To generate these patterns, we propose a fixed-point optimization scheme that compensates for the deformation due to the physical forces affecting the drape and is independent of the underlying simulation tool used. Our experiments show that this method quickly and reliably converges to patterns that, under simulation, form the desired target look, and works well with different black-box physical simulators. We demonstrate a range of edited and resimulated garments, and further validate our approach via expert and amateur critique, and comparisons to alternative solutions.

Keywords: Garment design

Concepts: **•Computing methodologies → Mesh models;**

1 Introduction

Designing complex garments is a time-consuming and knowledge-intensive task. It takes an expert designer multiple hours to create a real or virtual outfit from scratch [Brouet et al. 2012]. To save time, designers frequently use existing garments as a starting point and create new garment looks by combining or modifying existing designs. The editing operations they commonly use include mixing together existing garment elements, changing hem or sleeve lengths, and changing garment tightness, or fit. To apply these envisioned edits designers need to generate the 2D patterns, or outlines of cloth panels, that yield the desired 3D appearance once stitched together and draped around a person, bending and stretching. Consequently, much of the garment editing workflow is traditionally performed in 2D space and involves meticulous editing of 2D patterns to achieve the desired 3D look; this process requires significant time and expertise. We replace this 2D-dominated workflow with an editing framework that allows users to specify the desired edits directly in 3D space, and then automatically generates the target 3D geometry and corresponding patterns (Figure 1), speeding up the process and reducing the design effort.

Our framework consists of two components: a 3D editor that generates the designer-envisioned 3D *target* garment geometry, and a pattern maker that creates the 2D patterns that replicate this target geometry when lifted into 3D via a physical draping simulation correctly accounting for the intrinsic deformation the cloth undergoes during draping. Our system takes as input one or more source garments, draped around a mannequin with user-selected simulation software, and their corresponding patterns. The user can alter individual garments, changing length or fit, or mix together components from different garments. Garment design literature indicates that when modifying or combining garments, users seek to maximally retain their original style subject to the specific modification, and when mixing garments aim for a smooth, unified appearance [Brown and Rice 2001; Assembil 2013]. A garment's style is defined by a combination of three factors [Brown and Rice 2001; Assembil 2013; Brouet et al. 2012]: *proportionality*, which describes the relative location of different garment elements with respect to the wearer's body; *shape*, which describes the orientation of the garment contours, and hence reflects the normal directions across the garment surface; and *fit*, which encodes the distance between the garment and the wearer's body. Our 3D editor allows users to schematically specify the garment modifications they wish to apply (Figure 1, left) and then automatically generates a 3D gar-

---

SIGGRAPH '16 Technical Paper,, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925896

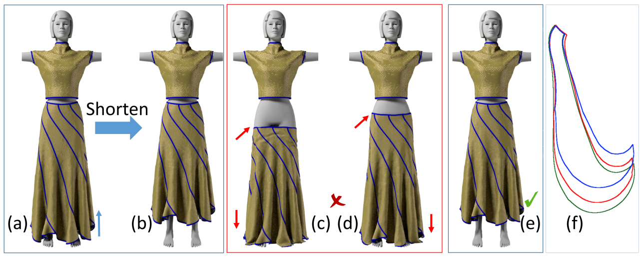


Figure 2: *Our editor takes the original simulated skirt (a); and shortens it directly in 3D (b). The original simulation stretches the input 2D panels (f, green). Consequently, using the shortened skirt as a rest shape causes the resimulated skirt (c) to slip off; using its 2D parameterization as a rest shape the resimulated skirt barely stays on but does not reproduce the target (d). Simulation using our physics-aware patterns faithfully reproduces the target shape (e); (f) original patterns (green) and patterns produced by stages (d) and (e) in red and blue respectively (8 identical panels). Note how the adjustment non-linearly shortens the original panels.*

ment that preserves the original input style and conforms with designer expectations. We compute this *target* garment by extending a formulation of garment style originally designed solely for grading purposes [Brouet et al. 2012] to other garment editing operations, and show it to be equally effective in these settings. Using this new formulation and aligning components of different garments along proportionality preserving transition boundaries our 3D editor preserves source style better than alternative mixing techniques (Section 5, Figure 1b).

While the resulting target 3D garments are visually pleasing, they cannot be used as-is for 3D simulation, nor for manufacturing. Garment manufacturing and most simulation methods require 2D patterns. While some simulators can use 3D rest-shapes, a simulation using a target garment as a starting point will not reproduce its shape, as illustrated in Figure 2. Here the source skirt stretches at the hips during draping, and this stretched shape is maintained after shortening. Using this stretched skirt as rest shape, results in a garment which is too loose and which slips down during simulation. Generating 2D patterns by parameterizing the edited 3D garment panels using standard parameterization techniques, e.g. [Liu et al. 2008; Sheffer et al. 2005], as done in existing literature ignores this "baked in" intrinsic distortion. A garment generated by stitching and draping such *flattening-based* patterns onto the original mannequin typically looks quite different from the target one (Figures 1e, 2d). Previous garment processing frameworks that operated in 3D space ignored the intrinsic distortion that occurs during draping and were unable to create patterns that allow for faithful target resimulation (Section 2).

We propose a *physics-aware* method to generate patterns that replicate the target geometry under simulation. The general inverse problem of seeking a rest shape that matches a target geometry under simulation is challenging and often ill-posed; it has not been addressed for draped garments. We address the generation of patterns that match a given target in the context of garment editing and derive a *simulator independent* pattern-making approach by leveraging several observations about the physics of draped cloth. Our key observation is that the shape of a triangulated 2D pattern is fully determined by the shape of its triangles, and in turn that the shape of a draped garment is affected by the pattern shape but is independent of both pattern location and orientation. We therefore avoid explicitly optimizing pattern layout and focus our efforts on finding optimal rest triangle shapes. Starting from an initial guess, we seek a sequence of linear transformations that result in a set of rest triangles that, once simulated, form the desired target garment. Our initial guesses are designed to be sufficiently close to the solution and the discrepancy between the target and the drape simulated using this initial guess is dominated by intrinsic stretch due to contact, gravity and other forces. Consequently, our optimization focuses on minimizing this intrinsic distortion and encodes deviation between our intermediate solutions and the target draped garment in terms of the shape of their mesh triangles. This formulation lets us design a garment rest-shape optimization using a local-global fixed-point iteration algorithm that operates in the space of linear transformations. Our method typically converges in under five iterations, and results in resimulated garments that are visually indistinguishable from the target (Figure 2e) and corresponding manufacture-ready patterns (Figure 2f, blue).

This pattern generation method is the key technical contribution of our paper, making the direct 3D editor a practical and appealing solution for real-life and virtual garment modeling. For typical inputs pattern adjustment reduces the maximal deviation (Hausdorff distance) between a resimulated and target garments from up to 10% of mannequin height to under 1%.

We validate our framework in a number of ways. Throughout the paper we demonstrate a range of examples of new garment designs and associated patterns generated with our 3D interface using just a handful of mouse-clicks. The resulting target garments consistently conform to designer expectation; and our pattern adjustment technique produces patterns that, when resimulated, reproduce these targets. Professional designers confirmed that our output, post-simulation, garments preserve the input style, conform with aesthetic constraints, and are consistent with results a professional would produce. We compare our results to baseline alternatives (which consistently fail where our method succeeds), confirm our method's invariance to the choice of simulation techniques by using it with different simulators, and demonstrate the real-life manufacturability of the generated designs by creating a real garment from one of our hybrid designs.

2 Related Work

Garment design is an intricate and time-consuming task that requires an intimate understanding of the complex and nonintuitive mapping between 2D flat fabric panels and the resultant 3D garments. Moving in either direction must account for how stitching panels, and draping stretchable material over the human form, is affected by contacts, friction and gravity.

To create 2D panels that match a desired 3D form we must account for the stretching, sagging, and wrinkling that physics will impose. Traditionally, to account for these factors, even the most proficient garment designers repeatedly adjust their patterns via a time-consuming, iterative process of cutting, pinning and draping designs on mannequins. The primary challenge in replacing this workflow with an intuitive, purely 3D manipulation is to account for the coupled *geometric* and *physical* constraints at play in forming garment shapes. Computational tools are beginning to address garment design needs. Until now, however, enabling free-form 3D shape edits that reflect garment design criteria and converting these 3D shapes into realizable 2D garment designs have remained outstanding hurdles. Before explaining our tool and methods in detail, we first review recent developments towards the promise of an interactive and intuitive digital garment design process.

3D Shape Editing. A large body of work addresses editing, e.g. [Harmon et al. 2011; Kraevoy et al. 2008; Yumer et al. 2015] and mixing, e.g. [Funkhouser et al. 2004] of natural and engineered shapes. While our work uses a conceptually similar approach to generate new garment designs, the underlying technical challenges we address are very different. Volumetric deformation methods [Kraevoy et al. 2008; Yumer et al. 2015] cannot easily account for garment-mannequin interaction such as maintaining fit or proportions. Typical surface deformation methods (see [Botsch et al. 2010] for a review) penalize surface shearing while allowing changes in surface normals; such formulations are unsuited for our

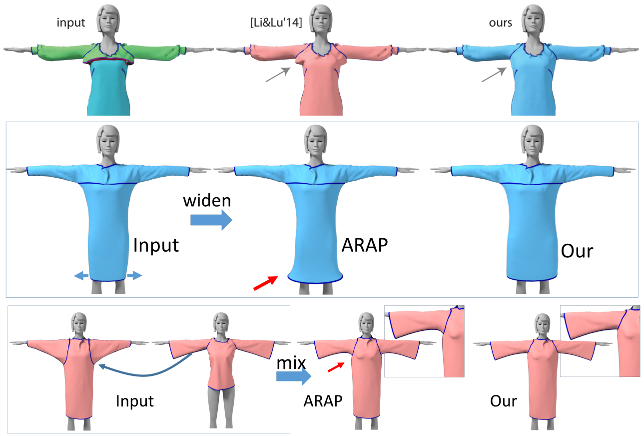


Figure 3: *Keeping component geometry fixed during mixing (top row) creates artifacts when transition boundaries have different length and fit (top). Standard surface deformation (here [Sorkine and Alexa 2007]) undesirably changes the input garment's silhouettes and look during fit loosening (second row) and garment mixing (last row). It also frequently creates non-manufacturable doubly curved surfaces (see the "lip" on the dress on in row two, or the toroidal transition region on the bottom). Our 3D edits preserve the input style and manufacturability.*

needs as they lead to undesirable and unnatural changes in the garment's look (Figure 3). Furthermore, these methods are purely *geometric* in focus and do not consider manufacturability nor drape with respect to physical constraints. In contrast, we seek to enable an intuitive garment design tool that supports *both* meaningful free-form geometric operations on garments *and* produces a physically realizable design at the end of the process.

2D→3D Garment Modeling and Editing. The traditional garment design process begins with the creation and stitching of 2D fabric patterns. A range of commercial tools, such as ClothAssembler [Fontana et al. 2005], Optitex PDS (Pattern Design Software), Marvelous Designer, and Pattern Works Intl. support virtual pattern assembly and use physics-based simulation to generate and analyze the resulting 3D garment shape. Use of these CAD packages requires significant time and expertise to create and modify garments.

Interactive 2D pattern-based garment modeling systems allow users to edit patterns in 2D and provide instant 3D feedback in response to pattern changes by using a real-time physics-based simulation, e.g. [Volino et al. 2005]. These systems rely on domain-specific user knowledge to create patterns that achieve a desired form when lifted to 3D. While faster than modeling garments from scratch, they require technical tailoring skills and a significant time investment in a workflow similar to traditional drape and pin iterations.

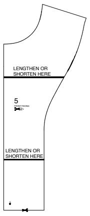
3D Geometric Garment Modeling and Editing. Over the years, researchers have proposed a range of sketch-based virtual garment modeling interfaces that allow users to trace garment silhouettes on top of a mannequin and then automatically infer plausible 3D garment shape [Wang et al. 2003; Decaudin et al. 2006; Turquin et al. 2007; Robson et al. 2011]. The models created by these systems are frequently non-physical and hence unrealizable.

Automatic grading of garments [Wang et al. 2005; Meng et al. 2012; Cordier et al. 2003; Brouet et al. 2012] aims to non-uniformly scale a garment designed for one person to fit a person with different body shape and proportions, while retaining the original garment's design. Brouet et al [2012] introduce a style-preserving 3D grading formulation that agrees with designer criteria. We extend this formulation to handle other editing operations, such as as garment mixing, and fit and length adjustments.

Several authors investigated tools for intuitive garment mixing. Mixing requires no pattern edits when the transition between the combined components is along shared, same-length seams. Berthouzoz et al. [2013] locate such shared seams on selected patterns. Zheng et al. [2014] proposed a recommendation system for detecting and combining such mixable components and for adding decorative garment elements. Li and Lu [2014] mix garment components by connecting them using zippering or Coons patches. These methods result in visible artifacts when the transition boundaries have different length and offset to the body (Figure 3, top). Kwok et al. [2016] present a mixing method for tight garments; applying this approach to loose garments would lead to similar artifacts. Our mixing algorithm (Section 5) is freeform, allowing users to specify arbitrary mid-panel transitions, and correctly handles transition boundaries of different lengths (such as the sleeve replacement in Figure 3).

All of the above 3D modeling and editing methods are *purely geometric* and do not model the physical forces that determine the final shape of a garment drape. These techniques either stop once a 3D garment is created, e.g. [Meng et al. 2012; Robson et al. 2011], or else apply off-the-shelf parameterization methods to generate 2D patterns for the target garment, e.g. [Decaudin et al. 2006; Brouet et al. 2012]. Such solutions are not sufficient to realize the target garment geometry when draped (see Figures 2, 10).

3D→2D Physical Garment Editing. While effective methods exist to *physically* map 2D patterns to 3D draped garments, to enable a complete and intuitive digital workflow we seek a comparable 3D→2D *physical* mapping of realizable 3D shapes to physically valid designs. As the first step towards this promising vision, the Sensitive Couture system [Umetani et al. 2011] supports interactive 3D preview of changes in 2D design, and partial 3D→2D workflows by using *slopers*, pre-existing, parameterized panel templates, commonly used by designers when modifying standard patterns (see inset). Sensitive Couture maps 3D user interaction to sloper parameter changes, e.g., for lengthening or widening, but does not support arbitrary 3D garment edits. We enable seamless 2D↔3D workflow by developing a tool that interactively supports unrestricted free-form 3D→2D *physical* garment editing operations.



Inverse Simulation. Physically valid 3D→2D garment editing is a special case of the *inverse statics* problem. In this general formulation we optimize to find a rest state that matches a target configuration while maintaining force balance between elastic energies, frictional contact, and gravity. Inverse statics are broadly treated in shape optimization [Zolésio 1992], animation [Derouet-Jourdan et al. 2010; Twigg and Kačić-Alesić 2011], hair simulation [Derouet-Jourdan et al. 2013], masonry analysis (e.g., [Shin et al. 2016]) and fabrication [Mori and Igarashi 2007; Furuta et al. 2010; Skouras et al. 2014; Chen et al. 2014]. Stoll et al. [2010] attack a related problem, seeking physical cloth parameters (e.g., stiffness) to best match simulation to captured video, but do not solve for rest shape nor physically realizable garment patterns. The static inverse problem is generally underdetermined and nonconvex [Schittkowski 2002]. Inverse problems that must account for contact forces are additionally challenged by nonsmoothness [Derouet-Jourdan et al. 2013]. Recent inverse methods handling frictional contact have focused on hair [Derouet-Jourdan et al. 2013] and masonry [Shin et al. 2016] and have attacked the combined challenges of the problem with domain-specific strategies that leverage the particular physical and geometric properties of the system of interest. In a similar fashion, we construct our algorithm by taking advantage of properties specific to the problem of 3D→2D inverse garment editing. This allows us to construct, for the first time, a physics-aware 3D space garment editor with a modular, swappable back-end cloth simulation component. The resulting algorithm is agnostic to choice of cloth simulator, simple to add to pre-existing cloth simulation pipelines, and can quickly be customized to suit animation and/or garment fabrication needs.

3 Direct 3D Garment Editing Tool

We perform user-specified edits directly in 3D space, and use one or more simulated draped garments as an input. Designers indicate the desired alteration in 3D space using a simple UI (see Figure 1) and do not need to account for patterns during the editing process. Our editing toolbox includes representative alterations that are commonly performed by garment designers such as fit and length adjustments, and garment mixing. We support free-form control-handles, not constrained to garment seams, and allow users to produce garments that significantly deviate from the original design.

Our supported edits lead to non-local garment deformations, and as we perform them we aim to preserve the style of the input garment(s) - their *proportions, fit,* and *shape* - and avoid visual artifacts. The formulation proposed by Brouet et al. [2012] for garment grading directly codifies the three style components above, including a strong preference for shape, or normal, preservation. We therefore adopt this formulation, and optimize the style energy they suggested subject to the constraints imposed by our editing tasks. The resulting outputs provide the desired balance between preserving the original garment look and satisfying the user constraints, outperforming possible alternatives (Figures 3).

Alterations to a single garment. We support a suite of frequently performed design modifications (Figure 1), each of which uses a loop of garment mesh edges encircling the mannequin as a control handle. The user can move the control loops along the body to elongate or shorten portions of the garment, such as sleeves or skirts (Figures 2, 6), and can offset points on this loop toward or away from the body (Figures 1, 3) to change the fit of the garment by loosening or tightening it around the loop.

Once the user specifies the new handle position, our algorithm updates the garment shape by optimizing the garment style energy described in [Brouet et al. 2012], subject to preserving the post-alteration positions of the control handle vertices (Figure 3, top). The resulting outputs satisfy the handle locations, maintain the original style, and have no visible artifacts.

For completeness we also provide support for grading, implemented directly via the original method of Brouet et al [2012] (Figure 10). The framework can be extended to support other operations using similar principles.

Garment Mixing. Mixing is a very popular garment editing operation [Brown and Rice 2001] as it enables users to create new complex designs by combining parts from existing garments. To generate a garment mix in our editor, users choose two garment designs and specify the parts they wish to combine. Generating 3D hybrid, or mixed, garment geometry is non-trivial; users expect the style of individual garment parts to be well preserved, yet expect the transition between them to be seamless, or essentially invisible (e.g. Figure 5c). In particular, while users expect the part geometry to change somewhat to allow for the smooth transition, they expect the relative location of these parts with respect to the body to remain fixed. We detail our mixing algorithm that produces the desired results in Section 5.

4 Pattern Adjustment

Given a desired *target* garment geometry created via our direct 3D editing tool, we require a rest garment shape that will reproduce this geometry when draped around a mannequin under gravity. Since, at rest, garment panels are developable, we realize garment rest shape with a 2D pattern. The *rest* and *draped* geometries are related by static equilibrium of physical forces

$$\mathbf{E}(\mathbf{X},\mathbf{x}) = \mathbf{L}_{int}(\mathbf{X},\mathbf{x}) + \mathbf{L}_{ext}(\mathbf{X},\mathbf{x}) = 0, \tag{1}$$

where $\mathbf{E}$ is the residual energy of the system, $\mathbf{X}$ is the vector of rest-shape mesh vertices, $\mathbf{x}$ is the vector of draped material vertices in 3D , and $\mathbf{L}_{int}$ and $\mathbf{L}_{ext}$ are the internal (e.g., elastic) and external (e.g., gravitational) forces respectively. We seek a rest shape $\mathbf{X}$ where correspondence with respect to a desired 3D target garment shape $\mathbf{x}^g$ is ensured under equilibrium by $\|\mathbf{E}(\mathbf{X},\mathbf{x}^g)\| < \epsilon$.

A potential approach to find equilibrating $\mathbf{X}$ in Equation 1 would be to iteratively linearize the system and seek force balance via the spatial gradients $\nabla_{\mathbf{X}}\mathbf{E}$ and/or the sensitivity matrix $\nabla_{\mathbf{x}}\mathbf{E}^{-1}\nabla_{\mathbf{X}}\mathbf{E}$; however, for the garment drape problem this has not been done. The presence of contact constraints, friction and strong material nonlinearities make the computation of these gradients expensive, challenging and, especially due to frictional contact forces, numerically unstable. Moreover, such computations would have to be customized for each choice of simulation code. Deriving and computing appropriate gradients consistent with simulator-specific forces must account for the exact formulations used, while the computation of gradients for many contact, collision and friction forces preclude automatic differentiation and may not be possible at all for many cloth simulation codes.

We avoid the difficulties inherent in trying to implement such an approach and instead derive a *gradient free, simulator independent* approach for rest shape computation that leverages a number of observations about the physics of draped cloth and the editing setup we operate in. Keeping all parameters for the fabric and the mannequin fixed, the equilibrium geometry of a draped cloth largely depends on two key factors: the geometry, or shape, of the 2D patterns, and the initial draping conditions. As we *a priori* seek a final draped garment that replicates our target geometry, we use this target geometry to set the initial conditions at each step in our search. Our solution is thus absolutely independent of the pattern 2D location or orientation. In turn, the shape of triangulated 2D patterns is fully defined by the shape of their constituent triangles. We consequently can cast our search for optimal pattern geometry in terms of optimizing the shape of these triangles.

We use 2D parameterization of the target *revised garment's 3D* panels, as an initial guess for the desired patterns. These initial patterns capture the garment cut but do not account for physical forces that stretch the patterns during draping. Garments resimulated using such patterns typically exhibit significant errors in proportions and fit when compared to the target, but have triangle normals largely similar to the target ones (e.g., Figure 1e). The large deviation in fit and proportions they exhibit is the result of different intrinsic geometry, i.e., difference in shape between corresponding triangles on the two garment meshes. Our optimization consequently focuses on minimizing this intrinsic shape difference, seeking for isometric output and target. Due to draping constraints, once the two surfaces are (near-)isometric the Euclidean distance between them becomes similarly negligible.

Our last observation is that the intrinsic change in mesh geometry during draping is both *bounded* and largely *one-sided*. Specifically, fabric practically never compresses and its stretch during draping is limited – at the extreme end knits exhibit up to 100% stretch while typical woven cloth exhibits less than 10% stretch compared to its rest state [Hu 2004].

The above observations allow us to cast garment rest-shape computation as a local-global, fixed-point iteration algorithm that operates in the space of linear transformations. Specifically we optimize in the space of triangle shape encodings rather than Euclidean space. Our algorithm applies simulation code intended to physically model the final garment in order to find equilibrium drape given the input garment patterns and an initializer. Our method works with most standard simulation codes and only requires that

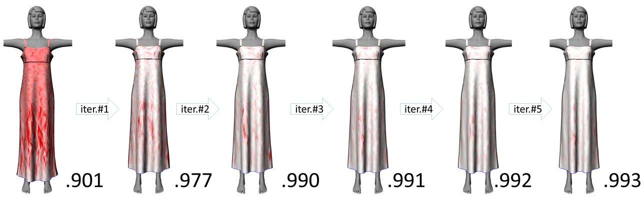
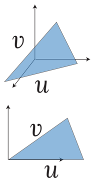


Figure 4: *Pattern adjustment following the hem elongation in Figure 6. Left to right: iterations zero to 5. Color (red to white) shows intrinsic triangle stretch w.r.t. to corresponding target geometry (scale from 0.8 to 1 with 1 being optimal). The number next to each garment is the average stretch.*

a simulator takes as input a set of patterns $\mathbf{X}$, allows us to initialize the solver's starting drape with our target $\mathbf{x}^g$, and gives as output a final drape at static equilibrium

$$\Phi(\mathbf{X}, \mathbf{x}^g) \rightarrow \mathbf{x}. \tag{2}$$

Reformulating our goal, we now seek a set of mesh garment patterns $\mathbf{X}$ with corresponding triangle faces $\mathbf{S}$ such that, after applying simulation, the *intrinsic* transformation between the triangles of the output simulated drape mesh, $\mathbf{s}$, and the triangles of the target mesh, $\mathbf{s}^g$, is a pure identity; in other words the 3D transformation between them, per triangle i, is rigid so that $\mathbf{s}_i^g = \mathbf{R}_i\mathbf{s}_i + \mathbf{c}_i$, where $\mathbf{R}_i$ and $\mathbf{c}_i$ are respectively a rotation and translation. Note that the mapping between the pattern triangles $\mathbf{S}_i$ and simulated drape triangles $\mathbf{s}_i$ is not, itself, rigid. Given this goal, and an initial guess $\mathbf{S}^0$, we derive an algorithm that seeks optimality in this form via an update of iterations $\mathbf{S}^k$ on the rest shape (Figure 4).



Setup. We rotate triangles $\mathbf{S}_i, \mathbf{s}_i$, and $\mathbf{s}_i^g$ to the x-y plane; co-align them so that a designated edge vector, $\mathbf{u}$, consistently chosen, is aligned with the x-axis; and then represent them in translation-free, 2D matrix form as the linear operators $\mathbf{T}_i, \mathbf{t}_i$, and $\mathbf{t}_i^g$ respectively [1]. The 2D *intrinsic* action of a simulation step in Equation 2, per triangle i is then

$$\Psi(\mathbf{T}_i) = \mathbf{t}_i = \mathbf{A}_i\mathbf{T}_i, \tag{3}$$

where

$$\mathbf{A}_i = [\mathbf{t}_i{\mathbf{T}_i}^{-1}] \tag{4}$$

gives the *local* change in triangle shape due to the globally coupled simulation step.

Update. At each iteration k of our algorithm we start from a 2D pattern with triangles $\mathbf{T}_i^k$. We first apply a global simulation step from target to simulated equilibrium mesh obtaining updated triangles $\mathbf{t}_i^k = \mathbf{A}_i\mathbf{T}_i^k$. We then update each triangle $\mathbf{T}_i^k$ independently and apply a global 2D embedding to stitch all updated triangles into a continuous 2D pattern suitable for the next simulation step.

Our goal is to minimize the intrinsic shape difference between the simulated triangles $\mathbf{t}_i^k$ at step k and target triangles $\mathbf{t}_i^g$. To do this, we seek optimal rest-shape triangles $\mathbf{T}^*$ that, when mapped by the forward simulation, satisfy

$$\mathbf{t}_i^g = \mathbf{A}_i^*\mathbf{T}_i^*. \tag{5}$$

$$\begin{tikzcd} \mathrm{t}_i^g & \mathrm{t}_i^k \arrow[l, "\mathrm{t}_i^g(\mathrm{t}_i^k)^{-1}"'] \\ \mathrm{T}_i^* \arrow[u, "\mathrm{A}_i^*"] & \mathrm{T}_i^k \arrow[u, "\mathrm{A}_i"'] \arrow[l, "?"] \end{tikzcd}$$

While $\mathbf{A}_i^*$ is unknown we can iteratively approximate it by pattern to drape transformations $\mathbf{t}_i^k = \mathbf{A}_i\mathbf{T}_i^k$. We can then restate the above relationship as the following fixed-point condition on triangles $\mathbf{T}_i^*$

$$\mathbf{T}_i^* = \mathbf{A}_i^{-1}\mathbf{t}_i^g = [\mathbf{T}_i^k(\mathbf{t}_i^k)^{-1}]\mathbf{t}_i^g = \mathbf{T}_i^k[(\mathbf{t}_i^k)^{-1}\mathbf{t}_i^g] = \mathbf{T}_i^k\mathbf{B}_i. \tag{6}$$

As triangles $\mathbf{t}_i^k$ and $\mathbf{t}_i^g$ both have a common vertex origin and an edge aligned with the x axis, the change of basis matrix $\mathbf{B}_i$ is a shearing matrix with a horizontal and vertical components. Following Equation (6) we then update rest triangles to reduce intrinsic deformation by applying a step of the fixed point iteration

$$\mathbf{T}_i^{k+1} = \mathbf{T}_i^k\mathbf{B}_i, \tag{7}$$

followed by a global simulation update step detailed below.

Embedding In processing triangles independently, each fixed-point update step in Equation 7 produces a set of disconnected 2D triangles. We then update from $\mathbf{T}_i^{k+1}$ to a consistent set of mesh patterns $\mathbf{X}^{k+1}$, suitable for simulation. We compute a 2D embedding as the connected solution using the seams of the original patterns as our panel boundaries. As we minimize stretch locally and update our equilibrium solution globally, embedding provides the necessary connection to relax the updated triangles to a nearby complete mesh. To maintain convergent behavior the embedding should introduce minimal distortion and maintain contraction.

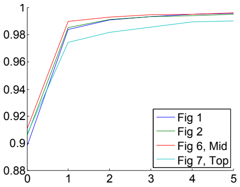
Embedding a set of triangles in 2D with minimal distortion is a standard mesh parameterization problem. We experimented with both ARAP [Liu et al. 2008] and ABF++ [Sheffer et al. 2005]. While ABF++ is frequently used for pattern making (e.g., Brouet et al. [2012]), we found the ARAP embedding more suitable for our needs as it balances length and angle preservation, and thus distributes scale distortion more uniformly in the resulting embedded rest mesh $\mathbf{X}^{k+1}$. We then apply the next simulation step with the new rest mesh $\Phi(\mathbf{X}^{k+1}, \mathbf{x}^g) \rightarrow \mathbf{x}^{k+1}$ to obtain a global update of $\mathbf{A}_i$.

Within the 3D optimization process target garment panels are typically nearly developable and the intrinsic distortion in each simulation step is bounded. Thus the shear transformations B_i do not dramatically change the shape of the *a priori* compatible pattern triangles. A typical distortion, i.e. change in triangle shape, introduced by the 2D embedding, measured using the angle and area distortion formulas of [Liu et al. 2008] is less than 0.003 (both angle and area)–a minuscule number.

Initialization. We have briefly discussed our choice of using target drape to initialize equilibrium solves, and now discuss this initialization step in greater detail. When constructing our starting point to initialize each equilibrium solve, we recall that our fixed-point iterations converge when there is a neighborhood containing the optimal rest shape $\mathbf{X}^*$ where each step is contractive [Bertsekas 1999], so that the per-triangle transformations we apply satisfy $\|\mathbf{B}_i\|_2 < 1$. Our target geometry, $\mathbf{x}^g$, serves as a useful initial drape guess, as we generally expect it to stretch under gravity during each simulated equilibrium solve; consequently, for each update step, we expect the corresponding fixed point transform to compress and hence satisfy contraction. We then construct our optimization's starting rest mesh, $\mathbf{X}^0$ as the corresponding 2D embedding of the target geometry processed with the same 2D embedding as described above.

---

[1] Per triangle, edge vectors are $\mathbf{u}$,$\mathbf{v}$ with internal angle θ, so that the 2D triangle matrix representation is simply $\begin{pmatrix} |\mathbf{u}| & |\mathbf{v}|\sin(\theta) \\ 0 & |\mathbf{v}|\cos(\theta) \end{pmatrix}$.



Termination and Convergence We define per triangle stretch computed between the ith current and target triangles as

$$S_i = 1 - \sqrt{(\lambda_i^1 - 1)^2 + (\lambda_i^2 - 1)^2} \tag{8}$$

where λ_i^1, λ_i^2 are the eigenvalues of the matrix $\mathbf{t}_i^g(\mathbf{t}_i^k)^{-1}$, which captures the current deviation from the target. We terminate iterations when the average triangle stretch is sufficiently small or when the change in stretch between iterations drops below a given threshold. We observe that the first iteration consistently brings the biggest improvement and the algorithm typically converges to a stable solution with a simulated drape visually indistinguishable from the target geometry in under 5 iterations (Figure 4). The inset plots the stretch values through iterations for a number of typical examples.

While the physical properties of cloth (discussed above) suggest that fixed-point steps will largely be strictly contractive, this is of course not guaranteed. To better understand the convergence behavior of the pattern adjustment algorithm, we evaluate $\mathbf{B}_i$ across the adjustment iterations for the four examples in the plot above in which our analysis finds that over 97% of triangle updates have eigenvalues satisfying contraction.

Undoing pattern compression. While a real fabric is close to incompressible, simulation codes sometimes compress patterns during a geometric preprocess step prior to draping. A common example is treatment of different-length shared boundaries between input mesh patterns; rather than generate fabric bunching, many simulators resolve this disparity by compressing the triangles along the longer boundary. Such compression in the input garments carries over to our target geometry, and negatively affects pattern adjustment, as our contraction assumption no longer holds. To overcome this preprocessing artifact, we update our initial guess to take this compression into account by using the original input garments and their patterns. For each triangle $\mathbf{t}_i^g$ on the target mesh, we compute the intrinsic transformation $\mathbf{M}_i$ from its corresponding originating pattern triangle to the original 3D garment which satisfies $\mathbf{t}_i^g = \mathbf{M}_i \mathbf{t}^{\text{original}}$. We then extract the eigenvalues λ_1, λ_2 of $\mathbf{M}_i$, which measure the stretch induced by the simulation. To reverse the compression artifacts we compute the initial guess triangles $\mathbf{t}^0$ as

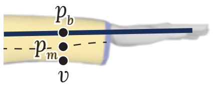
$$\mathbf{t}^0 = \begin{pmatrix} \max(1, \frac{1}{\lambda_1}) & 0 \\ 0 & \max(1, \frac{1}{\lambda_2}) \end{pmatrix} \cdot \mathbf{t}^g \tag{9}$$

5 Garment Mixing

Garment mixing allows users to create new designs by combining together style elements from different designs that complement one another in terms of relative locations, jointly forming a complete garment, see examples in Figures 1, 7). During mixing designers seek to maintain these locations as well as other style properties of the individual parts, and to have a smooth, essentially invisible, transition between them [Brown and Rice 2001]. Our framework satisfies these expectations and allows users to combine parts along arbitrary user-specified transition boundaries, enabling them to create hybrids with mid-panel transitions (e.g. Figure 5) and to mix parts with a priori very different transition boundary lengths (e.g. Figure 3).

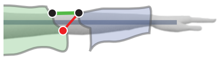
To generate a garment mix in our editor, users choose two garment designs and specify the part on one of them they want to keep, while replacing the rest with the corresponding part from the other. To delineate the part of interest they can either identify a specific edge loop as a transition boundary, or specify an approximate transition region using a brush interface and have the system optimize the boundary location to facilitate a visually seamless transition. We found that the former interface is useful when users want a transition along a particular seam, such as when replacing sleeves (Figure 3). In contrast, when seeking a mid-panel transition, users typically have a more vague sense of where this transition should occur, and seek a location which will best satisfy style and smoothness. Instead of forcing users to make this choice manually, we therefore allow them to specify a transition region and automatically select the best boundary location within this region (Figure 5). The corresponding boundary and the complementary part on the second garment are typically computed automatically based on the common parameterization but may also be specified manually.

Relative Parameterization To parameterize two garments with respect to one another we first parameterize each individual garment with respect to the common mannequin skeleton. We compute the relative locations of garment vertices with respect to the skeleton loosely following [Brouet et al. 2012]. For each vertex we select the nearest bone as a reference; for vertices close to two bones (e.g. in the center of the skirt) we consistently select the same bone as reference, prioritizing the bone to the right of the vertex in a frontal view. For each vertex v we compute both the nearest location p_b on the bone and the intersection p_m between the segment v, p_b and the mannequin (see inset). While p_b encodes the relative axial position of the vertex along the bone, the surface point p_m encodes its radial or angular location around the bone. To establish a proportion preserving mapping between garments we map vertices on each garment to locations with maximally similar matching bone and surface points on the other.



Given the user input we utilize the relative parameterization to optimize the transition boundary location when it is not fully specified, and determine the dense correspondence between the two boundaries. If the user specifies the boundary on only one garment, our algorithm computes the corresponding boundary on the second garment, with a goal of optimizing for proportionality, by mapping each vertex on the specified boundary with relative parameters p_b, p_m to the point on the other garment with the closest parameters p'_b, p'_m. It then defines the second boundary by connecting these mapped points and uses the correspondence established by the mapping for the subsequent deformation step.

When users manually select both boundaries, we expect them to surround the same bones (replacing a sleeve with a skirt is not a likely operation) but at possibly different relative locations. Since both boundaries form closed loops we induce the same radial parameterization along them, to avoid undesirable radial surface shifts (red in inset) Specifically, for each vertex on one boundary with parameters p_b, p_m we compute the corresponding location on the second boundary (green ray in inset) by seeking for a boundary point p' whose corresponding mannequin point p'_m minimizes $\sqrt{\|p_m - p'_m\|^2 - ((p_m - p'_m) \cdot b)^2}$, where b is the bone direction.



Optimizing Transition Boundaries. If the user only provides an approximate transition region, we compute matching boundaries that lie within this region and its matching region on the second garment (using the match computed via the proportional parameterization). To preserve proportionality, we use the mapping above to map any selected boundary on one garment to the other. The simplest way to define a part boundary within a region is to compute the minimal cut that separates its two sides. Using length alone however is sub-optimal, as the computed boundary and its matching counterpart may have very dissimilar normals. Mixing garments along such boundaries would either significantly change the part shape, or normals, near the joined boundaries or create visible discontinuities in the output hybrid (Figure 5,b). To avoid such unde-

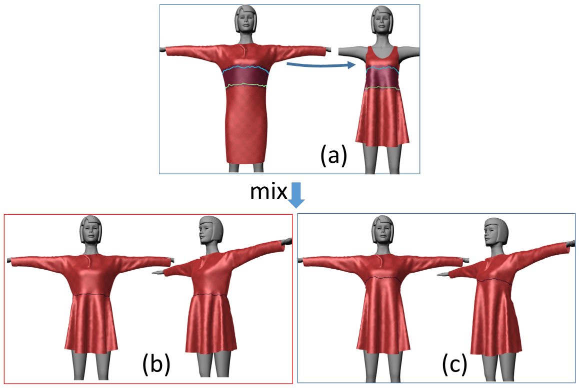


Figure 5: *Given the two input garments and an approximate transition region (a), using the shortest loop inside the region as the transition boundary (green) results in an undesirable discontinuous transition (b). Taking normals into account generates a longer boundary loop (blue), but leads to a smoother combination of the two garments (c).*

sirable artifacts we optimize for a boundary that while still short, balances length against normal similarity. We formulate the computation of the desired boundary as a min-cut problem, where each edge e in the transition region is associated with a cost based on normal similarity and cut-length:

$$E(e) = \tau_{\text{n}} \|1 - n_e \cdot n'_e\| + \|e\|, \tag{10}$$

The first term aims to minimize normal differences between corresponding points (n_e is the edge normal on the first garment, and n'_e is the normal on the corresponding line segment on the second garment), and the second term favors shorter boundaries. We empirically set $\tau_{\text{n}} = .4$, prioritizing boundary length over normal similarity. Figure 5 shows the impact of using normal similarity on the final garment.

Joining Garment Components. Once a dense boundary correspondence is computed, we smoothly join these boundaries together while maximally preserving the style of the participating parts. To achieve this we optimize the style energy [Brouet et al. 2012] with respect to the original garments while constraining the boundary vertex positions to coincide. We seek for an aesthetically pleasing, and hence visibly seamless, transition; therefore for vertices in the vicinity of the boundary we compute corresponding vertices on the other garment using the mannequin based correspondence and softly constrain them to align. For each given vertex correspondence, (v, v'), we set the alignment weight to

$$w = \max\left(0, 1 - \frac{1}{\tau_{\text{blend}}^2} \max(d, d')^2\right)$$

where d is the distance to the closest point of the transition boundary of the first garment, and d' is the distance to the transition boundary of the second garment. We empirically set $\tau_{\text{blend}} = 0.01h$, where h is the mannequin height. The weight drops to zero for vertices further from the boundary than τ_{blend}. After joining the components together, we need to update pattern topology – if the Gaussian curvature along the transition boundary is small, we join the patterns on its two sides together, otherwise we retain the boundary as a seam.

6 Results

Throughout the paper we exhibit a range of garments created using our 3D editing framework. We use input garment patterns purchased from *burdastyle.com*, a popular do-it-yourself garment making website.

Simulation We have tested our pipeline with two forward dynamics simulators that we run from target drape to equilibrium at each evaluation. The first employs a standard mass-spring cloth system [Bridson et al. 2003], with position-based strain limiting and collision processing [Müller et al. 2007] that biases towards speed over accuracy and on-average takes 3 seconds per resimulation step. The second, a FEM-based simulator, ARCSim [Narain et al. 2012; Narain et al. 2013] biases towards accuracy over efficiency where each resimulation step took on average five minutes to complete. In all examples we have used the mass-spring cloth simulator unless ARCSim is specified. As demonstrated by Figure 8 our framework performs equally well with both simulators. Noticeably the two simulators produce different artifacts when starting from flattening based patterns (Figure 8, middle and bottom): using the mass-spring simulator the hybrid dress created with such patterns drags on the floor, while with ARCSim it exhibits unexpected seam shifting on the chest. In both cases thanks to the physics-aware pattern adjustment method, our resimulated outputs successfully replicate their targets.

Alterations. We used our system to perform a range of alterations of individual garments, including hem lengthening and shortening (Figures 2, 6), fit adjustment (Figures 3, 6), and grading (Figure 10). In all these examples we were able to substantially alter the input garments with just a few mouse clicks, and our system produced both target and resimulated outputs that conform with user expectations.

Garment Hybrids. We tested our garment mixing algorithm on a range of complex inputs, including cases where the two transition boundaries have vastly different lengths (e.g.Figures 3bottom, 7bottom), transition along diversely shaped and oriented seams (e.g. Figures 7 top, 8 bottom), as well as transitions in the middle of existing panels (e.g., Figures 5, 12). We tested both mixes of garment tops and bottoms, as well as swapping sleeves (Figure 3) and even replacing a collar (Figure 9). Our method successfully handled all these tasks, with both target and resimulated outputs preserving the style of the input components while smoothly transitioning between them.

Comparison to prior art. As Figure 3 demonstrates, standard mesh editing techniques are inadequate for garment editing, while our style-energy based solution generates the desired 3D outputs.

Most previous 3D garment editing and modeling methods use flattening to create patterns, with ABF++ [Sheffer et al. 2005] used by both [Brouet et al. 2012] and [Decaudin et al. 2006]. While with very low-stretch fabric, this solution exhibits no major visible artifacts after resimulation, as Figure 10 shows it is inadequate in more general simulation setups, where our pattern adjustment succeeds.

While Sensitive Couture [Umetani et al. 2011] provides some garment editing capabilities in 3D, they are restricted to pre-defined, specific parameters on user-provided deformable pattern slopers. Some of our length and fit adjustment operations can potential be implemented with such sloper templates, however the diverse set of pattern geometries and topologies we support (see Figures 1, 2) makes providing templates for each pattern family impractical. Slopers similarly cannot be used for free-form mixing operations as those require drastic changes in panel shape.

Material-Driven Pattern Adjustment. Similar to adapting to different simulators, our frameworks effortlessly adapts to different simulation parameters. Figure 11 shows how we can use this adaptability to generate different patterns for the same target garment geometry but different fabric stiffnesses.

Real Life Validation To validate that our patterns are suitable for manufacturing, we cut and stitched one of our hybrid designs, scaling it to fit a 30cm 3D-printed replica of our mannequin (Figure 12).

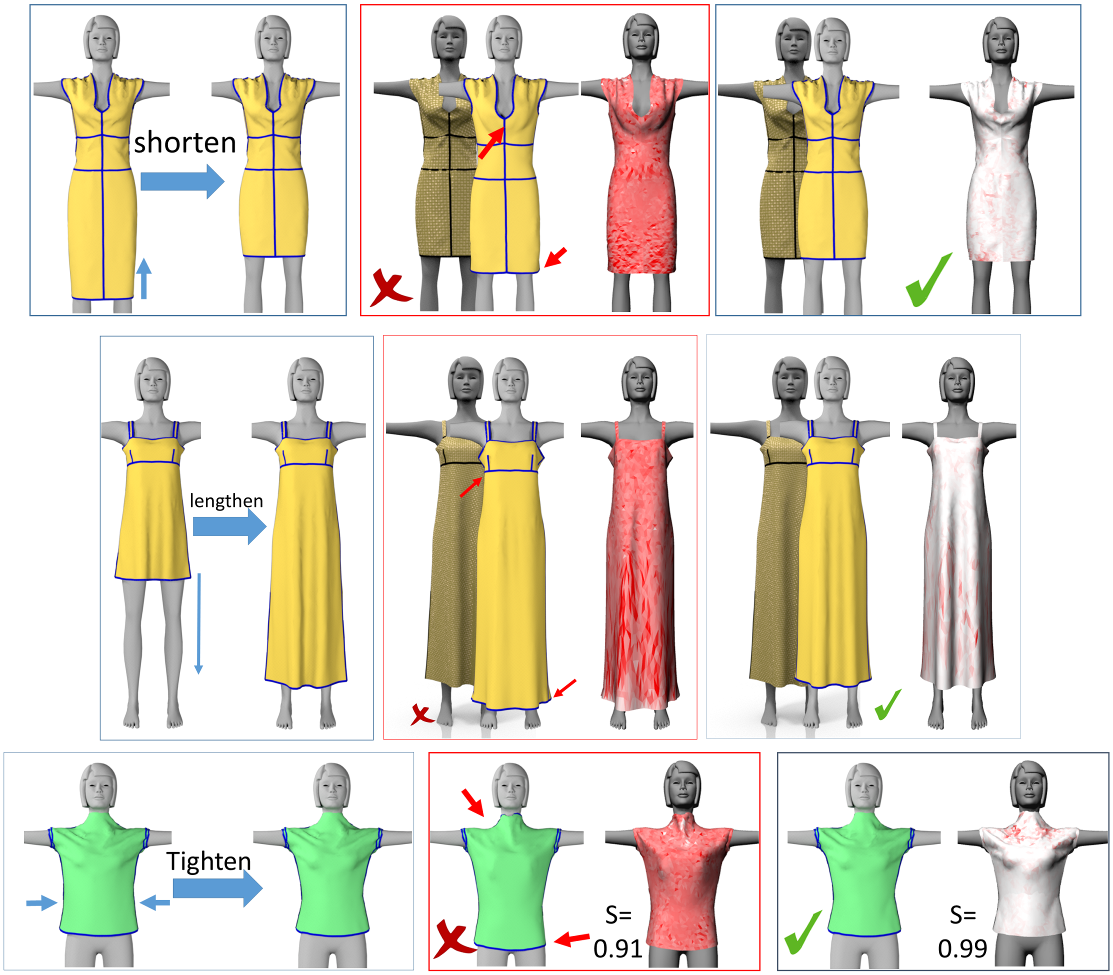


Figure 6: *Garment length and fit changes. (left) 3D editing input and output. (right) Simulation with flattening-based patterns (target rendered in the back for reference, rows one and two) and simulation result after pattern adjustment. For both simulations, we show the per-triangle stretch as compared to the target (red-white scale from 0.8 to 1, with 1 being best). The Hausdorff distances for the three models went down from 4.6%, 6%, 3.8% of mannequin height to 0.5% each.*

To sew the garment, we added a seam allowance to all panels. To drape it around the solid mannequin, we stitched the shoulder seams after draping using over-stitching. Despite these changes and the purely eyeballed fit between fabric and simulation parameters, our real replica (d) looks similar to the resimulated result (c), validating our approach.

Perceptual Validation. We validate our algorithm via feedback from two professional designers and 10 non-experts. To collect their input we devised a questionnaire (see supplementary material) that asks viewers to select between alternative outputs for a range of garment modifications. Each question contains one answer created with our system, and one alternative solution (resimulated drape with flattening-based patterns or classical surface deformation). Both experts selected our solution as the expected result in all 10 questions. They stated that our outputs preserve all the key elements of style. While none of them currently uses any computerized design tools, one was very interested in trying our software out to ideate and to create initial patterns. Both noted that the current simulators do not account for some parameters that impact final garment shape such as weave direction; so they expect to need to perform some minor manual edits to our patterns in setups where these matter. The non-expert participants were shown the same survey and selected our results in 89% of cases - a clear majority. They also commented that while results created by draping flattening-based patterns often violated input fit and proportions ours kept them. We used all female participants as all our examples are of women's apparel, thus we expect women to have a better sense of their style.

Times and Statistics. Typical input garment meshes contain between 5K to 27K triangles (Table 1). This number is consistent with those used in commercial garment design softwares, e.g. Marvelous Designer, and was chosen to provide a reasonable trade-of between speed and accuracy. The total processing time varies from 40 seconds for the small inputs to 2.5 minuted for the largest. Out of this time, 10 to 20 seconds are spent doing the 3D computations and 20 to 80 seconds are spent performing pattern updates; the rest of the time is spent in the simulator. Pattern adjustment typically takes under 5 iterations to converge (Figure 4), with over 80% reduction in error achieved in the first iteration.

Hausdorff distance. While our optimization method is formulated in terms of stretch, the metric we want to optimize is the distance between the target and resimulated garments. In particular we are most interested in the worst, or Hausdorff distance between them (Table 1). For most of the models tested the Hausdorff distance between the target and the drape simulated using flattening-based patterns was over 5% of mannequin height. After pattern adjustment it went down to between 0.5% and 0.8% a huge visible difference. The mean distance similarly went down by factor 10. All distances were measured using Metro [Cignoni et al. 1998].

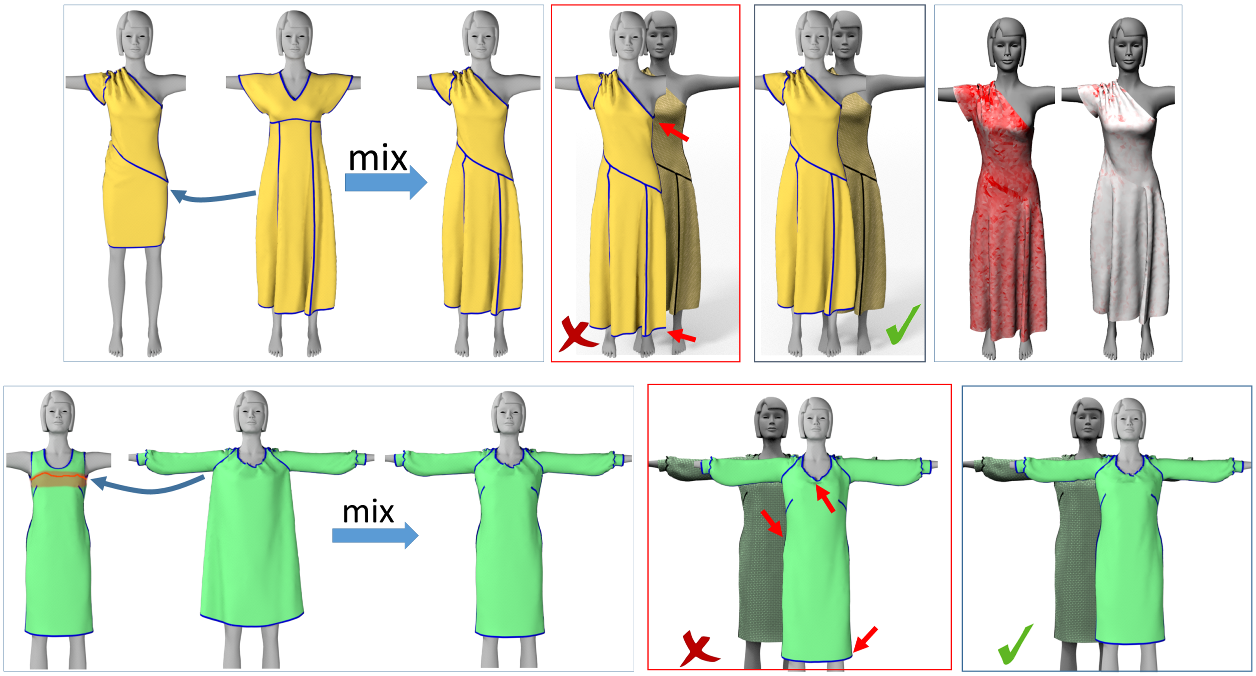


Figure 7: *Garment mixing: (left) input garments with user-specified transition seam (top) and region (bottom) highlighted and 3D editing output; (right) simulation using flattening-based patterns, and resimulated garment after pattern adjustment (target in back for reference).*

	# Δ	Stretch initial	Stretch final	Hausdorff initial (% h)	Hausdorff final (% h)
Fig 1:	27669	.896	.990	6.2	0.7
Fig 2:	8622	.897	.990	8.0	0.8
Fig 6 top:	8222	.899	.992	4.7	0.5
Fig 6 mid:	3076	.900	.989	6.0	0.5
Fig 6 bot:	5095	.909	.990	3.9	0.7
Fig 7 top:	15677	.905	.994	6.8	0.7
Fig 7 bot:	10911	.908	.988	5.6	0.8
Fig 8 top-right:	14725	.947	.993	6.6	0.8
Fig 8 top-left:	14725	.982	.992	5.6	0.9
Fig 8 mid:	18712	.909	.989	6.6	0.7
Fig 8 bot:	18712	.933	.986	5.5	0.6
Fig 10	11789	.923	1.0	4.9	0.3
Fig 12:	7776	.986	.996	2.3	0.7

Table 1: *Pattern adjustment statistics.*

Scalability. To test our algorithm's scalability we experimented with subdividing the input mesh in Figure 2 twice, creating a mesh with 65K triangles. Running this model through the system, predictably took significantly longer, 11 minutes total, but required the same number of iterations to achieve the same error bound.

7 Conclusions

We presented a system for direct editing of garments in 3D space, which, compared to the traditional 2D editing approaches, is dramatically faster to use and more intuitive for novice users who have no experience with 2D patterns. Our evaluation confirms that we can perform a range of popular garment modifications directly in 3D, and that our system produces simulation- and manufacturing-ready patterns. Key to our system is a novel pattern adjustment algorithm, which generates 2D patterns that match the desired 3D drape appearance. This algorithm is gradient-free and simulator independent. Expert designers confirmed that 3D designs and patterns produced by our system are comparable to what a professional would have made, and that direct 3D editing can drastically simplify their workflow. We believe that the simplicity and accessibility of our method will inspire future research on direct 3D garment design. Our system can be extended to handle other 3D edits, e.g ones requiring changes in pattern topology. An interesting future research would be to compute not only pattern geometry for a target shape, but to optimize fabric parameters as well.

Our framework does not prevent users from generating unrealizable garments–for instance, a user can loosen the fit of a garment to a point where it will slip down under gravity. In future work it would be interesting to investigate ways to restrict users from performing physically non-valid edits. As with all simulations of real-life materials, to be indicative of real-life behavior our framework requires exact fabric parameters. Lastly, while our pattern adjustment framework works well in practice, it has no theoretical convergence guarantees.

We dedicate this paper to the memory of our co-author Floraine Berthouzoz.

Acknowledgments

We thank designers A. Chavez and L. Shade for their valuable input. Sheffer and Vining thank NSERC for its ongoing financial support.

References

ASSEMBIL. 2013. *How Patterns Work: The Fundamental Principles of Pattern Making and Sewing in Fashion Design*. Assembil Books.

BERTHOUZOZ, F., GARG, A., KAUFMAN, D., GRINSPUN, E., AND AGRAWALA, M. 2013. Parsing sewing patterns into 3d garments. *ACM Transactions on Graphics 32*, 4, 85.

BERTSEKAS, D. P. 1999. *Nonlinear Programming*. Athena Scientific.

BOTSCH, M., KOBBELT, L., PAULY, M., ALLIEZ, P., AND LEVY, B. 2010. *Polygon Mesh Processing*. AK Peters.

BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *Proc. Symp. on Computer Animation*, 28–36.

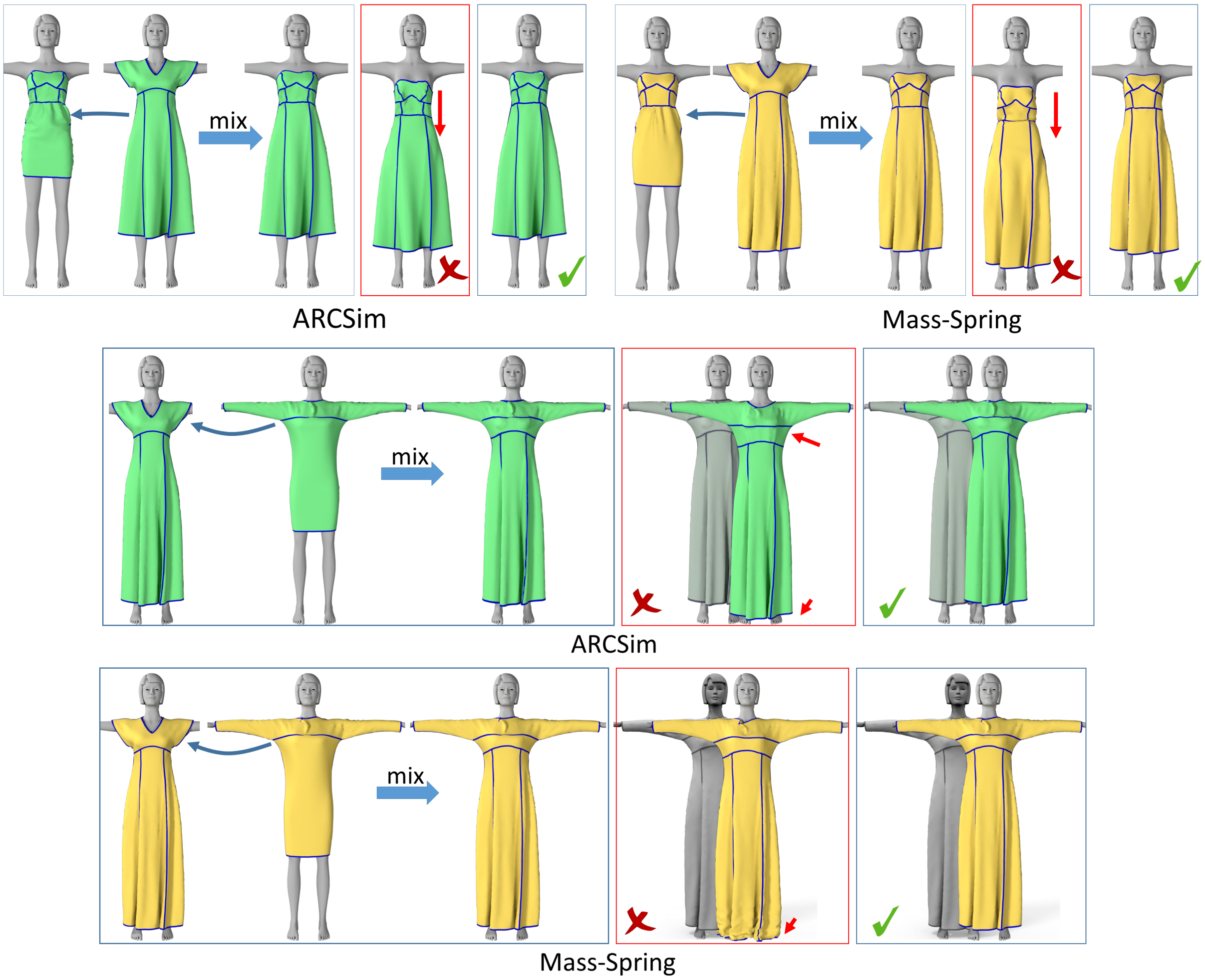


Figure 8: *Using both mass-spring and ARCSim simulators the hybrid strapless dress simulated from flattening-based patterns (top) slips off. In the bottom hybrid the use of different simulators leads to different, undesirable, artifacts in the garments simulated from the such patterns. Our method transparently adjusts all input patterns generating resimulated outputs that match the 3D targets with both simulators.*

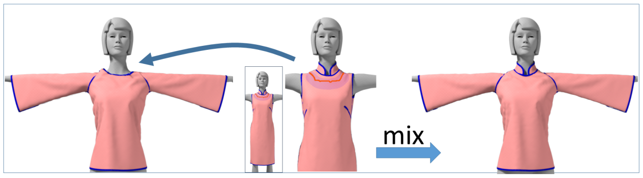


Figure 9: *Using mixing to add a collar. Transition region and final boundary highlighted.*

BROUET, R., SHEFFER, A., BOISSIEUX, L., AND CANI, M.-P. 2012. Design Preserving Garment Transfer. *SIGGRAPH*.

BROWN, P. K., AND RICE, J. 2001. *Ready-to-wear apparel analysis*. Prentice Hall.

CHEN, X., ZHENG, C., XU, W., AND ZHOU, K. 2014. An asymptotic numerical method for inverse elastic shape design. *ACM Transactions on Graphics 33*, 4.

CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. 1998. Metro: Measuring Error on Simplified Surfaces. *Comp. Graph. Forum*.

CORDIER, F., SEO, H., AND MAGNENAT-THALMANN, N. 2003. Made-to-measure technologies for an online clothing store. *Comp. Graph. and App. 23*, 38–48.

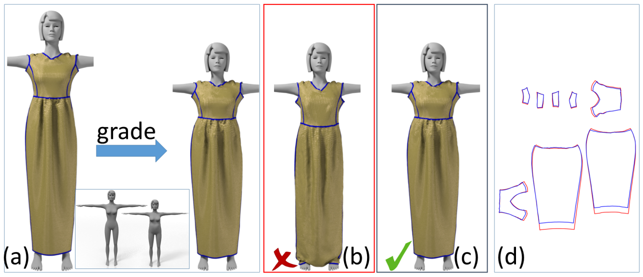


Figure 10: *Garment grading (a) followed by resimulation (b,c). (b) Patterns generated with ABF++ parameterization used by [Brouet et al. 2012; Decaudin et al. 2006] result in a resimulated graded garment that drags on the floor (a). Our solution (c). ABF++ (red) and our adjusted (blue) patterns (d).*

DECAUDIN, P., JULIUS, D., WITHER, J., BOISSIEUX, L., SHEFFER, A., AND CANI, M.-P. 2006. Virtual garments: A fully geometric approach for clothing design. In *CGF*, vol. 25.

DEROUET-JOURDAN, A., BERTAILS-DESCOUBES, F., AND THOLLOT, J. 2010. Stable inverse dynamic curves. *ACM Transactions on Graphics 29*, 6.

DEROUET-JOURDAN, A., BERTAILS-DESCOUBES, F., DAVIET,

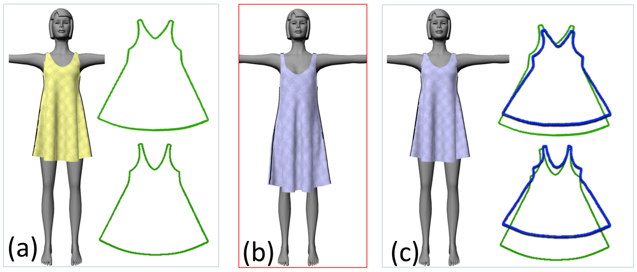


Figure 11: *Using different simulator parameters, we can automatically generate patterns suitable for different fabrics. We generate a target garment from original patterns (green) using stiff material (a); use these patterns to seed pattern adjustment with a much more stretchy fabric (b); our patten adjustment automatically shrinks the patterns (c, blue) to generate a resimulated visually identical garment from the stretchy material.*

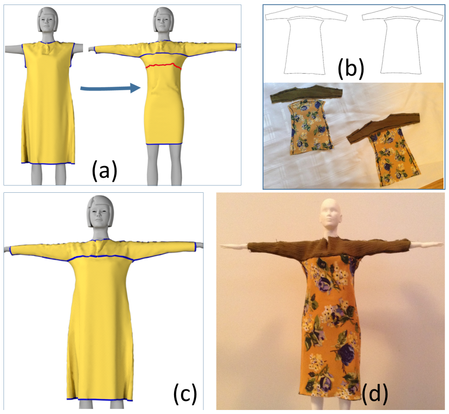


Figure 12: *Realization (d) of a hybrid design (c) using adjusted patterns (b). The result is fully manufacturable and retains target style up to inevitable impact of stitching and seam allowance (b).*

G., and Thollot, J. 2013. Inverse dynamic hair modeling with frictional contact. *ACM Transactions on Graphics 32*, 6.

Fontana, M., Carubelli, A., Rizzi, C., and Cugini, U. 2005. Clothassembler: a cad module for feature-based garment pattern assembly. *CAD & Applications 2*, 6.

Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., and Dobkin, D. 2004. Modeling by example. In *ACM Trans. on Graph.*, vol. 23.

Furuta, Y., Umetani, N., Mitani, J., Igarashi, T., and Fukui, Y. 2010. A Film Balloon Design System Integrated with Shell Element Simulation. In *Eurographics 2010*.

Harmon, D., Panozzo, D., Sorkine, O., and Zorin, D. 2011. Interference aware geometric modeling. *ACM Transactions on Graphics 30*, 6.

Hu, J. 2004. *Structure and Mechanics of Woven Fabrics*. Woodhead Publishing Series in Textiles. Elsevier Science.

Kraevoy, V., Sheffer, A., Cohen-Or, D., and Shamir, A. 2008. Non-homogeneous resizing of complex models. In *ACM Trans. Graphics*.

Kwok, T. H., Zhang, Y. Q., Wang, C. C. L., Liu, Y. J., and Tang, K. 2016. Styling evolution for tight-fitting garments. *IEEE Trans. Visualization & Computer Graphics 22*, 5.

Li, J., and Lu, G. 2014. Modeling 3d garments by examples. *Computer-Aided Design 49*, 28 – 41.

Liu, L., Zhang, L., Xu, Y., Gotsman, C., and Gortler, S. J. 2008. A local/global approach to mesh parameterization. *Computer Graphics Forum (Proc. SGP) 27*, 5, 1495–1504.

Meng, Y., Wang, C. C., and Jin, X. 2012. Flexible shape control for automatic resizing of apparel products. *CAD*.

Mori, Y., and Igarashi, T. 2007. Plushie: An interactive design system for plush toys. *ACM Transactions on Graphics 26*, 3.

Müller, M., Heidelberger, B., Hennix, M., and Ratcliff, J. 2007. Position based dynamics. *J. Vis. Comun. Image Represent. 18*, 2, 109–118.

Narain, R., Samii, A., and O'Brien, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. on Graphics 31*, 6, 147:1–10.

Narain, R., Pfaff, T., and O'Brien, J. F. 2013. Folding and crumpling adaptive sheets. *Trans. on Graphics 32*, 4.

Robson, C., Maharik, R., Sheffer, A., and Carr, N. 2011. Context-aware garment modeling from sketches. *Computers & Graphics 35*, 3, 604–613.

Schittkowski, K. 2002. *Numerical Data Fitting in Dynamical Systems: A Practical Introduction with Applications and Software*. No. v. 1 in Applied Optimization. Springer.

Sheffer, A., Lévy, B., Mogilnitsky, M., and Bogomyakov, A. 2005. Abf++: fast and robust angle based flattening. *ACM Trans. Graph. 24* (April), 311–330.

Shin, H. V., Porst, C. F., Vouga, E., Ochsendorf, J., and Durand, F. 2016. Reconciling elastic and equilibrium methods for static analysis. *ACM Transactions on Graphics*, To Appear.

Skouras, M., Thomaszewski, B., Kaufmann, P., Garg, A., Bickel, B., Grinspun, E., and Gross, M. 2014. Designing inflatable structures. *ACM Trans. Graph. 33*, 4.

Sorkine, O., and Alexa, M. 2007. As-rigid-as-possible surface modeling. *Proc. Symposium on Geometry Processing*.

Stoll, C., Gall, J., de Aguiar, E., Thrun, S., and Theobalt, C. 2010. Video-based reconstruction of animatable human characters. *ACM Trans. Graph. 29*, 6 (Dec.).

Turquin, E., Wither, J., Boissieux, L., Cani, M.-P., and Hughes, J. F. 2007. A sketch-based interface for clothing virtual characters. *CG & A, IEEE 27*, 1, 72–81.

Twigg, C. D., and Kačić-Alesić, Z. 2011. Optimization for sag-free simulations. In *Proc. Symp. on Computer Animation*.

Umetani, N., Kaufman, D. M., Igarashi, T., and Grinspun, E. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph. 30*, 4, 90.

Volino, P., Cordier, F., and Magnenat-thalmann, N. 2005. From early virtual garment simulation to interactive fashion design. *CAD 37*.

Wang, C. C., Wang, Y., and Yuen, M. M. 2003. Feature based 3d garment design through 2d sketches. *CAD*.

Wang, C. C. L., Wang, Y., and Yuen, M. M. F. 2005. Design automation for customized apparel products. *CAD 37*.

Yumer, M. E., Chaudhuri, S., Hodgins, J. K., and Kara, L. B. 2015. Semantic shape editing using deformation handles. *ACM Trans. Graphics 34*.

Zeng, S., Zhou, F., Wang, R., and Luo, X. 2014. Probabilistic model for virtual garment modeling. In *IGTA*, vol. 437.

Zolésio, J. S.-J. 1992. *Introduction to Shape Optimization: Shape Sensitivity Analysis*. Springer.