Pacific Graphics 2024
R. Chen, T. Ritschel, and E. Whiting
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 7

Digital Garment Alteration

A. M. Eggler [1,2], R. Falque [2], M.Liu[2], T. Vidal-Calleja[2], O. Sorkine-Hornung[1], N. Pietroni[2]

[1]ETH Zurich, Switzerland
[2]University of Technology Sydney, Australia

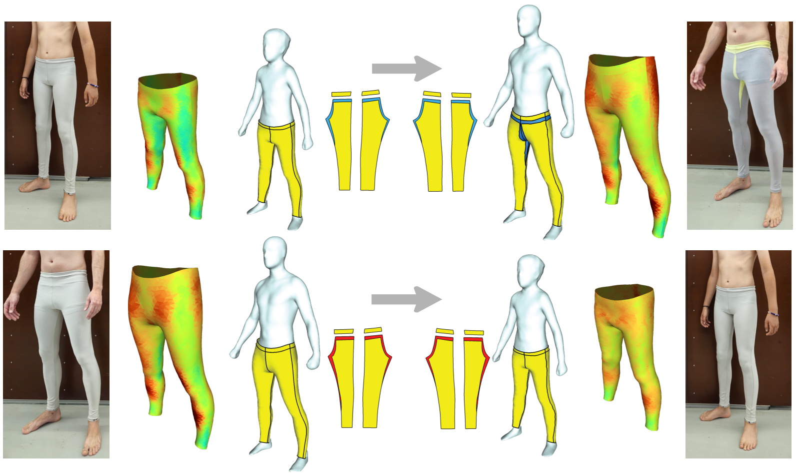

Figure 1: *Our system automatically computes a fabricable set of alterations to transport the drape of a garment from a template body to different body shapes. In this example, we compute the set of modifications to adapt a pair of child leggings to an adult, and vice-versa. The alterations are computed to transport the stress on the garment from the source to the target body. Alterations may include removal (shown in red on the sewing pattern) or insertion (shown in blue) of pieces of fabric, as well as the usage of darts.*

Abstract

Garment alteration is a practical technique to adapt an existing garment to fit a target body shape. Typically executed by skilled tailors, this process involves a series of strategic fabric operations—removing or adding material—to achieve the desired fit on a target body. We propose an innovative approach to automate this process by computing a set of practically feasible modifications that adapt an existing garment to fit a different body shape. We first assess the garment's fit on a reference body; then, we replicate this fit on the target by deriving a set of pattern modifications via a linear program. We compute these alterations by employing an iterative process that alternates between global geometric optimization and physical simulation. Our method utilizes geometry-based simulation of woven fabric's anisotropic behavior, accounts for tailoring details like seam matching, and incorporates elements such as darts or gussets. We validate our technique by producing digital and physical garments, demonstrating practical and achievable alterations.

1. Introduction

In the past decade, a new era of computational design tools has emerged to facilitate the digital creation and evaluation of garments. Interactive platforms like Clo 3D [CLO22] are actively employed in the apparel industry, seamlessly integrating sewing pattern design with physically-based simulation. Concurrently, academic research is contributing novel tools [PDF*22] that compute fabricable pattern layouts tailored to specific target bodies. While these tools have the potential for automation, their primary focus remains designing and fabricating *new* garments. Rather than design and fabricate a new garment from scratch, a more sustainable approach involves *modifying an existing garment* to fit a particular body shape faithfully. This process, commonly known as *garment alteration*, entails manually adding or removing fabric to a given garment to achieve the desired drape. Guided by the tailor's expertise and a few measurements, this procedure is often necessary to tailor newly bought clothing for a perfect fit or to refurbish old garments to accommodate changes in body shape over time.

In this paper, we introduce a novel digital pipeline for automating garment alteration. Our method determines a minimal set of modifications (involving fabric insertions and removals) to adapt a pre-existing sewing pattern from a source to a target body. The resulting alterations are grounded in reality, meeting the criteria for physical realization. The adjusted garment reproduces the intended drape of the original garment. We compute alterations by employing an iterative process alternating between global geometric optimization and physical simulation. Given a reference garment and its desired drape on a corresponding template avatar, we first adjust the initial pattern layout to match the desired fit on the intended target body. This iterative cycle deforms the pattern pieces while preserving their connectivity and structure. Then, we convert the disparity between the initial sewing pattern and the computed target pattern into a sequence of alterations expressed as a series of insertion and removal operations. The primary challenge in this phase lies in ensuring that the alterations are physically achievable and practical. Our pipeline is supported by a physically-based strategy that expands upon the widely recognized position-based dynamics method [MHHR07], incorporating the anisotropic behavior intrinsic to textiles. While many existing sewing pattern modification or creation methods primarily address tight-fitting garments [MTMP20], design preferences may lead to a more nuanced fit, perhaps tight in specific areas and loose in others. To capture and reproduce this behavior, we introduce the *fitting map*, which assigns a descriptor for each face of the reference garment reflecting the local deformation experienced during wear. The fitting map is a 2×2 tensor matrix that encodes the deformation of each triangle from its original shape in 2D space (where the textile is cut) to its deformed shape when the garment is draped over the body. Additional details are provided in Section 4. Throughout the paper, we visualize the fitting map using color coding, with red representing tension and blue representing compression (see Fig. 1 and 5). Since the tensor encodes length changes in two directions, we visualize the direction with the maximum magnitude.

To validate our approach, we conduct various experiments, including mapping between significantly different body shapes. Furthermore, we bring our methodology to life by fabricating physical garments, providing tangible evidence of the real-world applicability and efficacy of the proposed approach. Our pipeline accurately computes significant changes, such as adapting an adult garment for a child or vice versa (see Fig. 1) or adapting garments to different body shapes (see Fig. 13). It is essential to highlight that while various approaches have been proposed to grade template sewing patterns or freely deform them to match a different body shape [BSBC12, MWJ12, WWY05, Wan18]—thus generating an entirely new pattern layout—our approach stands out as the first to specifically address alterations of existing garments.

2. Related work

In the last decade, both the fashion industry and academia have shown considerable enthusiasm for emerging technologies that facilitate the digitization of garment design and production. Crafting computational techniques for designing, optimizing, or simulating virtual garment draping requires addressing a spectrum of interconnected challenges, ranging from differential geometry and physically-based simulation to user interaction.

2.1. Surface patch decomposition

Surface patch decomposition stands as a critical challenge with diverse applications in geometry processing and computer graphics [Cam17]. Existing techniques in this domain primarily focus on minimizing distortion during the 2D mapping of patches while maintaining a simple overall layout. Notable methods include bounded distortion parameterization [SCOGL02], Autocuts [PTH*17], and OptCuts [LKK*18]. While these approaches prove beneficial for UV mapping and texturing 3D shapes, their applicability to garment fabrication is limited. Alternative methods for surface patch approximation prioritize fabrication as the primary objective. Many aim to decompose the surface into a set of developable patches, facilitating fabrication using flat materials [JKS05, SGC18, IRHS20, BVHSH21]. However, the resulting patch layouts often fall short of being well-suited for garment fabrication due to the intricate seam structure in garment design. The common practice of decomposing a given shape into multiple patches is frequently employed to achieve a globally smooth parameterization and, in some instances, a semiregular quadrilateral remeshing [CK14, PPM*16]. This process typically involves computing a surface tangent vector field aligned with features and principal directions on the surface [VCD*17], followed by tracing the patch decomposition [RRP15, PPM*16, NHE*19, LPP*20, PNA*21]. However, the structural constraints imposed by these layouts often prove too restrictive for patternmaking purposes.

2.2. Sewing pattern modification

A new era of computational tools has revolutionized the fashion industry, significantly streamlining the production cycle and empowering designers to explore their creative choices rapidly and effortlessly. Tools such as Clo3D [CLO22] and Optitex [Opt22] have become industry standards, gaining widespread acceptance and incorporation into the production pipeline. Their popularity is such that these tools are now integral components of fashion school curricula. However, most industrial tools are CAD editors dedicated to iteratively adjusting patterns in 2D and 3D and assessing the resulting draping using simulation. A pioneering approach for these industrial tools was proposed by Umetani et al. [UKIG11], where fast cloth simulation enables users to work in 2D and 3D simultaneously and observe the effects of changes in both modes.

More sophisticated approaches allow the design of garments directly in 3D using a variational exploration of contours, fold- and boundary curves sketched by the user on a 3D avatar ([CHT*07, DJW*06, RSW*07, RMSC11, WHZ*21]). Other approaches use training data and machine learning methods to construct 3D garments starting from 3D scans [CZL*15, PMPHB17, BKL21] or explore the space of parametric generation models [WCPM18, VSGC20, KL21, KSH23, Fre24]. Various methods modify garment patterns to fit a particular body shape [CST03, WWY05, BSBC12, MWJ12, BSK*16, Wan18, BKL21, LZB*18]. Some of these meth-

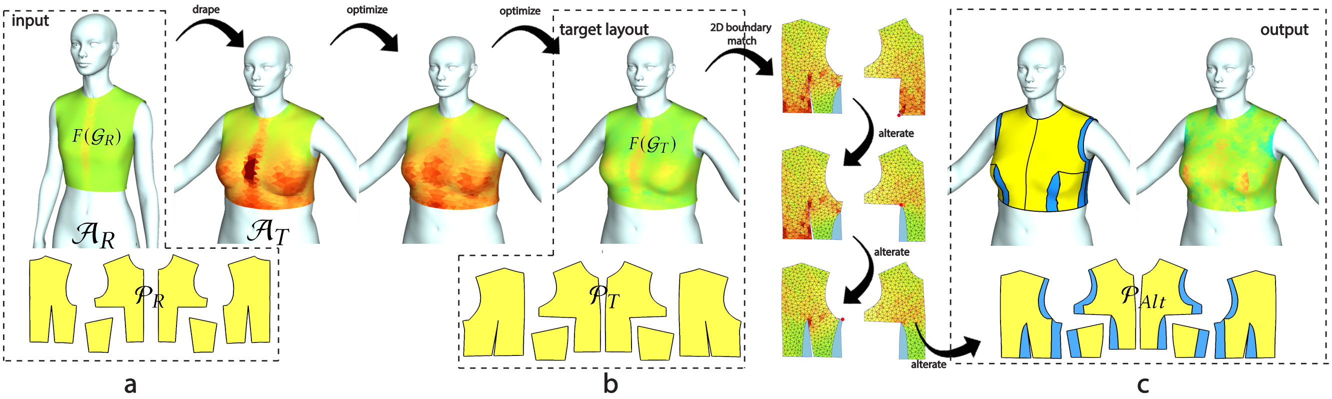

Figure 2: *(a) The initial sewing pattern and the target fitting map; (b) The fitting map is transported to the target body, guiding the derivation of a different sewing pattern; (c) The initial pattern is modified with a set of alteration operations, removing or adding pieces of fabric.*

ods can also optimize the resulting stress, pressure or seams traction [MTMP20]; they rely on predefined pattern designs [KP07] or let users design target folds [LSGV18] that have to be matched in the physical simulation. It is worth mentioning a different class of methods to design and fabricate knitwear [MAN*16, NWYM19, YKJM12, NAH*18, WSY19, LHZ*21]. However, the intrinsic nature of the knitting setup is different: here, the main task is to derive effective knitting paths and patterns. Pietroni et al. [PDF*22] recently proposed a new approach to patternmaking that utilizes cross-field tracing to produce a patch layout specifically optimized for woven fabric garment design. The split strategy uses a new distortion measure that ensures the textile stress of the produced garment remains below a threshold. Several innovative techniques have emerged for enhancing and customizing existing pattern layouts. For instance, Perfect Dart [dMQP*23] introduces optimizations by strategically incorporating and adjusting darts to enhance the overall fitness of the resultant dress. Another noteworthy approach, PerfectTailor [QI23], empowers users to modify a garment while preserving the fundamental structure of the underlying pattern layout. These advancements signify a dynamic shift in pattern design methodologies, offering greater flexibility and efficiency in garment customization. Recently [LyCL*24] introduced a method to recover simulation-ready garment and body assets including clothing patterns from a multi-view capture. Nevertheless, none of the preceding research efforts explicitly tackle the challenge of feasible and practical garment alterations.

Garment alterations differ significantly from freeform sewing pattern modifications. Previous works (like [BSK*16]) focus on deforming existing patterns to adapt to a different body shape, potentially creating an entirely new pattern. While these methods can be used to create new panels and fabricate a new garment from scratch, they do not address the specific challenges of altering existing garments. Garment alteration modifies an existing garment, which requires deriving a minimal set of fabricable and practically achievable modifications. We visualize this concept in Fig. 1. Our paper presents the first automatic solution for garment alteration, addressing this unique challenge.

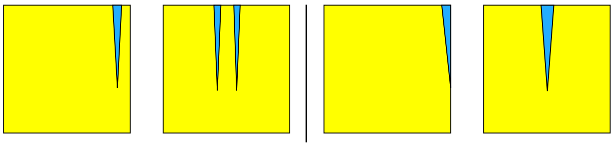

Figure 3: *Left: these two alterations are difficult to fabricate and sew together since they have a long thin sliver of fabric (left) or a fragmented boundary (right); Right: similar alterations that are easier to fabricate and assemble.*

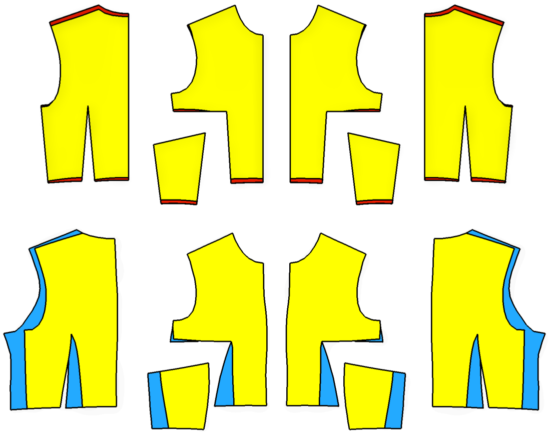

Figure 4: *A close-up view of pattern alteration of the 3D garment in Fig. 2. The first row shows the adaptation to a smaller body, and the second row to a larger body shape. Blue patches are added, while red ones are removed from the original garment.*

3. Overview

Our method is designed to achieve several key objectives.

Fit replication. The modified garment should fit the target body in the same way the original garment fits the reference body, with variations in tightness or looseness in different zones. Capturing

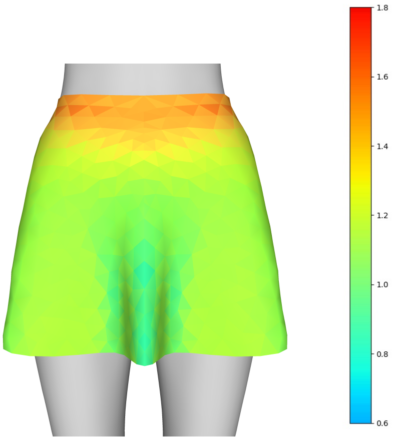

Figure 5: *An example fitting map calculated for a skirt with local variations. Red corresponds to stretch and blue to compression.*

this effect ensures the overall drape envisioned by the fashion designer is maintained in the modified garment.

Fabricability. The set of modifications should be practically feasible. For example, we want to avoid situations like the one depicted in Fig. 3 (left) and favor the ones shown in Fig. 3 (right). Promoting patch insertions in alignment with existing seams helps circumvent the creation of additional cuts, simplifying the overall fabrication process (an illustrative example of realizable pattern alteration is presented in Fig. 4).

Control. We aim to empower the user with control over the final result. This includes the ability to avoid modifications in areas corresponding to zippers or cuts intersecting with pockets. Additionally, users can enforce global constraints, such as symmetry.

At the end of the process, the original sewing pattern is enhanced with a set of modifications, including the addition of patches or the removal of existing fabric. A patch refers to a connected component within a sewing pattern. For the remainder of the paper, all alterations are represented in blue for additions and red for removals, as illustrated in Fig. 4. These alterations correspond to structural modifications of the original pattern, rather than simple variations of the existing components.

3.1. Structure of the processing pipeline

Fig. 2 provides an overview of the entire processing pipeline, which comprises multiple sequential steps.

Input. As input, we assume a reference garment $\mathcal{G}_R$ with corresponding sewing pattern $\mathcal{P}_R$ and its desired drape on a template avatar $\mathcal{A}_R$, as well as the target avatar $\mathcal{A}_T$ (see Fig. 2.a).

Target fitting map calculation. We compute the fitting map $F(\mathcal{G}_R)$ over the reference garment mesh to capture the desired drape as envisioned by the fashion designer. The fitting map serves to locally quantify the deformation experienced by the textile when worn (see Fig. 5) and provides the desired fitting for the target avatar mesh $F(\mathcal{G}_T)$ (see Fig. 2.a). Then, we dress the target avatar in the reference garment. To perform this step while maintaining proper draping, we smoothly morph between the reference and the target avatar and compute the drape using robust and efficient physically-based garment simulation. The details of this step are explained in Sec. 4.

Target sewing pattern derivation. We adapt the sewing pattern to the target avatar such that the resulting fitting map $F(\mathcal{G}_T)$ matches the reference fitting map $F(\mathcal{G}_R)$. The purpose of this step is to derive a new *target sewing pattern* $\mathcal{P}_T$ that mimics the drape of the reference avatar on the target one. This step involves an iterative optimization process, alternating between physically-based simulation and 2D patch flattening optimization (see Fig. 2.b). While allowing for flexible adjustments in the 2D sewing pattern, we impose global uniformity constraints, including symmetry, and adhere to fabricability constraints, such as matching across seams. A detailed explanation of this step is provided in Sec. 5.

Alteration operations. The final step consists of performing a sequence of adaptation operations to alter (by adding or removing fabric) the original sewing pattern $\mathcal{P}_R$ into the final one $\mathcal{P}_{Alt}$ in order to match the shapes indicated by the target pattern $\mathcal{P}_T$. We let physics guide the process of insertion or removal of the fabric (see Fig. 2.c). We stretch the input sewing pattern $\mathcal{P}_R$ to match the boundaries of the ideal sewing pattern $\mathcal{P}_T$. Intuitively, we want to let the stretched patch rip apart in the more stressed areas and then insert new pieces of textile in correspondence with the tearing. However, this strategy might lead to multiple uncontrolled alterations with fragmented boundaries, which are challenging to realize in practice. To cope with this problem, we solve a global integer linear programming (ILP) problem to decide where to propagate a fracture in the stretched patch. The ILP considers the practical issues involved in the alteration, for example, favoring the insertion of patches on the sides of existing ones. This way, we can exploit existing seams and avoid unnecessary cutting. We use a similar approach to determine the regions where we must remove pieces of textiles. We also provide the users with a degree of control to guide the process at a finer level, allowing them to avoid inserting or removing fabric from specific areas. We explain this step in Sec. 6.

4. Target fitting map

We define the *fitting map* $F(\mathcal{G}_R) : \mathcal{F} \to \mathbb{R}^{2\times 2}$ on the reference garment mesh, where $\mathcal{F}$ is the set of faces of the mesh $\mathcal{G}_R$. The fitting map encodes the deformation of a face from its 2D rest shape into its tangent space in 3D when the garment is worn. Since we only consider in-plane deformations of a single face, we factor out the component along the face normal, resulting in a 2×2 matrix per face. Given that our material, woven fabric, is strongly anisotropic, this matrix captures how two orthogonal vectors u and v in the tangent space change their lengths and relative angles when mapped to the 3D tangent space, and it is usually referred to as the Jacobian $\mathcal{J}$. This matrix captures the intrinsic deformation of textile fibers, including stretch, compression, and shearing. The fitting map is based on the simplifying assumption of Hooke's model, where stress is proportional to strain, which is typically accurate for small deformations. Fig. 5 illustrates an example fitting map. The color-coded signal showcases substantial variations across the garment surface. Throughout the paper, we consistently employ the following color scheme for fitting maps: red denotes stretching, blue indicates com-

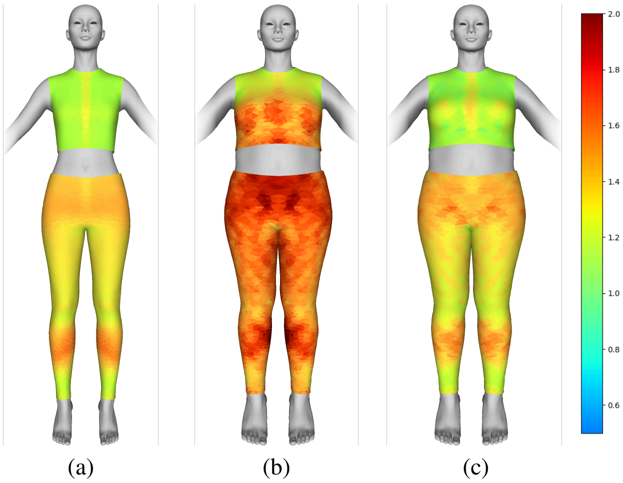

Figure 6: *(a) The reference fitting map computed over the reference body. (b) The garment is geometrically transported to the target body to derive a valid initial solution for the physically-based simulation. (c) The final simulation result is used to derive a physically plausible draping and the relative fitting map. The color panel on the right shows the stress on the surface. We use the same scale for all the examples in the paper.*

pression, and green signifies that the fibers maintain their length. Given that elongation changes can occur in both directions u and v, we represent this by choosing the maximum value between them.

After computing the reference fitting map $F(\mathcal{G}_R)$, we transport the reference garment $\mathcal{G}_R$ from the reference avatar $\mathcal{A}_R$ to the target avatar $\mathcal{A}_T$. The naive solution is to replace one body mesh with the other and then utilize a physically-based simulation to obtain the final draping. However, this approach only works for simple cases. If the two meshes vary considerably, replacing one with the other may generate an invalid starting solution for simulating the draping, which could result in significant intersections between the garment and the underlying body, leading to inaccuracies and artifacts.

4.1. Initial drape over target body

To correctly transport the garment from the source to the target body, we use a precomputed map $\mathcal{M} : \mathcal{A}_R \rightarrow \mathcal{A}_T$, which maps each point of the reference $\mathcal{A}_R$ to a corresponding point on the target avatar $\mathcal{A}_T$ (the details on how to obtain this mapping are provided in Sec. 8). We then express each vertex $v_i \in \mathcal{G}_R$ of the reference garment as the difference to its closest point on the reference avatar $c_i \in \mathcal{A}_R$. We can transport c_i from the reference to the target avatar using the mapping $\mathcal{M}$ and then displace the vertex along the calculated difference vector to map v_i to the target avatar. To factor out the local rotation, we multiply the difference vector by the smallest rotation matrix that aligns the normal of $c_i \in \mathcal{A}_R$ on the reference avatar to the corresponding normal of $\mathcal{M}(c_i) \in \mathcal{A}_T$.

This procedure allows us to derive a reasonable initial solution to bootstrap the physical simulation and compute a proper fitting map on the target. However, if there is a significant difference in size between the two avatars, the newly computed fitting map defined on the transported garment over the target body might fail to capture the intended draping. To understand this issue, imagine we aim at transporting a pair of trousers from a short person to a tall person. We would expect the trousers to become stretched in the process. In this case, the physical simulation might lead to sliding of the fabric over the body, and the expected stretch may vanish. To solve this issue, we can optionally constrain the physical simulation by fixing a few points on the extremities of the garment(e.g., sleeves of a shirt). The constrained position is the one produced by the initial mapping of the garment, since it is related to the underlying body.

5. Target sewing pattern derivation

The fitting map we obtained at the end of the previous step might differ significantly from the reference one. We show the difference in Figures 6 (a) and 6 (b). This difference arises from the fact that the original garment drapes differently over the source and the target body. To restore the original fitting map (and the original draping) over the target body, we need to deform the 2D sewing pattern patches to produce the desired fitting map.

Unlike the classical mesh parameterization where the overall distortion should be minimized, here we aim to find a patch flattening that achieves a specific distortion, identified by the fitting map. Our solution employs the widely used local-global strategy to minimize a non-linear energy, as is done in ARAP-based parameterizations [LZX*08, PDF*22].

5.1. Inverse Jacobian

In Sec. 4.1, we describe how we compute a draping of the original garment on the target avatar. We now wish to modify the 2D sewing pattern so that the produced fitting map matches the original over the reference avatar. To align the target fitting map $F(\mathcal{G}_T)$ with the reference fitting map $F(\mathcal{G}_R)$, it is necessary to adjust the garment in a manner that ensures that the stress exerted on the target avatar closely resembles the stress generated on the template avatar.

Intuitively, to solve this problem, we should change the 2D shape of each face f_{uv} composing the target sewing pattern $\mathcal{P}_T$, such that when the resulting garment is draped over the target avatar $\mathcal{A}_T$ and simulated, the deformation matches exactly the expected deformation expressed by the corresponding face Jacobian J_f from the reference fitting map $F(\mathcal{G}_R)$. In other words, we want a new target sewing pattern $\mathcal{P}_T$ that matches the desired deformation once draped over the target body. Specifically, each triangular face should target the deformation expressed by:

$$J_f(f_{uv}) = f_t, \tag{1}$$

where f_t is the final shape of a face produced by the physical simulation in 3D after the garment is draped on the target body. Next, let us consider the 3D shape of a face f_0 obtained after simulating the draping of the garment on the template body. To retrieve its optimal 2D vertex positions (which corresponds to the rest shape of the textile patches), we apply the inverse of the target Jacobian J_f^{-1} (specified by the reference fitting map $F(\mathcal{G}_R)$) to f_0. Then, similarly to [LZX*08], we perform a global stitching step to obtain vertex positions that best comply with the target configurations of all the faces (see Eq. (1) in [LZX*08]).

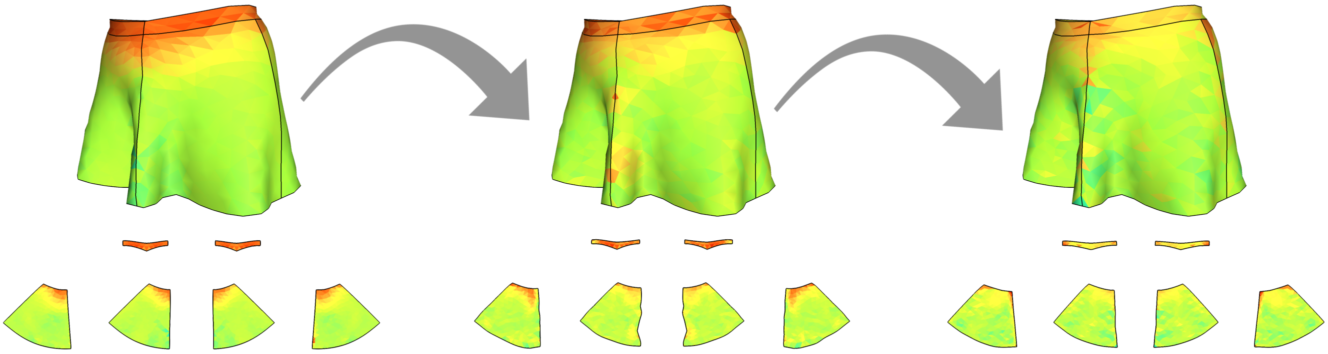

Figure 7: *Three steps of the inverse Jacobian optimization procedure.*

We iterate the local step of computing the inverse Jacobian and target 2D triangles, and the global stitching step until convergence. Fig. 7 shows some steps of the iteration process.

Seam reflection symmetry. To ensure the fabricability of the resulting patch layout, we must ensure that seams across adjacent patches are compatible with the physical sewing process, and we therefore require seam reflection symmetry. We minimize the deviation from a reflection transformation by using the same energy term as Pietroni et al. [PDF*22].

Accounting for physics. The procedure described above derives a sewing pattern that matches as well as possible the deformation expressed by a predefined fitting map. However, it assumes the 3D garment shape to be fixed. In reality, by changing the 2D pattern, we also change the physical behavior of the garment, and the produced drape might also be different. Theoretically, we should interleave our inverse Jacobian optimization step with a simulation step and repeat these two steps until convergence. We verified that the system converges to a correct solution after just one of these iterations. Hence, a second step of simulation and optimization is usually not necessary.

6. Pattern alteration

The previous step derived a new, ideal sewing pattern P_T from the initial sewing pattern P_R that transports the fitting map from the reference to the target avatar. Our final goal is to find a discrete set of practical alterations that transform the initial pattern P_R to closely match P_T. To address this challenge, we devise a simple physically-driven strategy. Consider a simple example with two patches of different sizes, and we aim to alter the smaller one into the bigger one. We can stretch the small patch until its boundaries align with the target patch. If the stretch exceeds a threshold, the patch will tear. We can exploit this behavior and add fabric in the areas created by the tearing (those areas are not part of the original textile anymore). Indeed, these are precisely the regions on the cloth that need additional material to conform to the stress. Conversely, if we map a larger patch into a smaller one, we introduce compression. However, since compression is the inverse of stretching, what causes compression in one mapping direction results in stretching in the opposite direction. In other words, the amount of fabric required to enlarge a smaller garment is equivalent to the amount of fabric we must remove from the larger garment to make it smaller. Therefore, we can invert the mapping and use the same procedure for the generation of tearing, which indicates where to remove fabric. The diagram in Fig. 8 summarizes this idea. Note that this process operates exclusively in 2D space and does not take into account the target's desired fitting maps, because these have already been addressed in the preceding step.

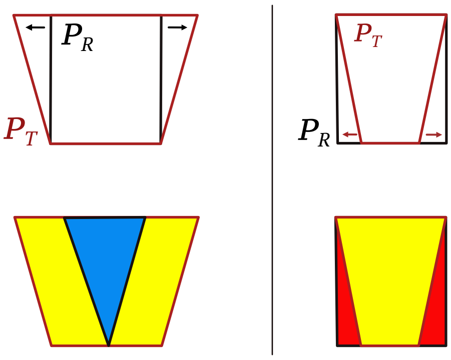

Figure 8: *A diagram illustrating the overall process of deriving pattern alterations. Left: We first stretch the reference pattern P_R (shown in black) such that the boundaries match the boundaries of the computed target pattern P_T (shown in red) to factor out the stretch. We then insert a new piece of material where the textile tears. Right: Similarly, we deform P_T to match the boundaries of P_R to factor out the compression, and we remove material from P_R in correspondence with the tearing.*

6.1. A fabricable set of alterations

We aim to promote adaptation operations that are more feasible to implement physically. We can summarize the main objective of this procedure as follows.

Limited alterations. The user can set the maximum number of alterations for each patch side. As previously mentioned, multiple nearby alterations are impractical (see Fig. 3).

Seam consistency. Given two adjacent patches (to be sewn to-

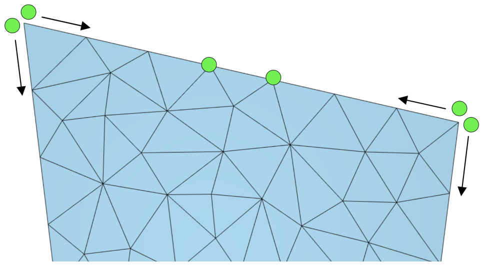

Figure 9: *Possible positions of tearing propagation and associated directions.*

gether), we want to ensure that alterations happen coherently along both seam sides.

No fragmentation of original patterns. We would like to avoid excessive fragmentation of the original sewing pattern, such as alterations running close and parallel to an existing seam. Therefore, we favor the creation of alterations in the middle of a seam.

Simple shape. We prefer alterations that have a simple shape, usually a strip. In some cases, multiple alterations can generate an L-shaped patch insertion/removal.

Pattern coherency. Our system should favor alterations that exploit pre-existing seams. For example, adding or removing material on the side of an existing seam is easier than on the middle (Fig. 3).

6.2. Deciding alterations

Instead of allowing the pattern pieces to rip in an uncontrolled way, we let only one tear to be inserted at a time, and propagate it as needed (explained below in Sec. 6.3). After each insertion, we recompute the new stress configuration using physically based simulation and potentially add a new tear. This strategy prevents the creation of multiple jagged boundaries and encourages the creation of a *clean and well-structured*, reduced set of cuts that result in a more practically manageable set of alterations.

To *avoid excessive fragmentation*, we allow the tears to start from a subset of boundary vertices close to the middle of each side. Additionally, the tear can also start from a corner and propagate along two possible directions (see Fig. 9). Propagating this kind of tearing is realized by disconnecting border vertices from their boundary constraints.

We determine the place of the next propagating tear by integer linear programming (ILP). We define a Boolean variable for each possible sampled location on patch sides $b_i \in \{0,1\}$. We also define a Boolean variable for each corner $c_i^u \in \{0,1\}$ and $c_i^v \in \{0,1\}$ (one for each direction). Corner variables are used to implement alterations along pre-existing seams and improve pattern coherency. We can use these variables to enforce a set of constraints and match our objectives:

- To *limit the set of alterations*, for each patch side s, we set the sum of the Boolean variables to be less than the global threshold Θ specified by the user, $\sum_{i\in s} b_i \leq \Theta$. We experimentally verify that 1 is a good value for Θ.
- To implement *consistency along seams*, we first compute the set $\mathcal{B}$ of pairs of boolean variables (b_i, b_j), $i \neq j$, that correspond to the same vertex in 3D, then we enforce equality for each pair, i.e., $b_i = b_j$. This forces the tear to be either propagated on both sides or none of the sides of a seam.
- To have a *simple shape* of the produced alteration, we impose a mutual selection of corner variables, $c_i^u + c_i^v \leq 1$. However, in some practical cases, disabling this constraint creates a better solution that includes L-shaped patches. While we allow the user to control this constraint, we prevent the selection of multiple subsequent corners that are closer than a certain threshold to avoid excessive fragmentation of the resulting alteration (see Fig. 10).

Fig. 10 presents a simple ablation study demonstrating that disabling the mentioned constraints leads to impractical alterations. In the example on the left we allow multiple alterations to be selected in each optimization step, resulting in unnecessary cuts to be propagated. On the right, we allow all corner variables to be selected simultaneously, producing a non-practical alteration set.

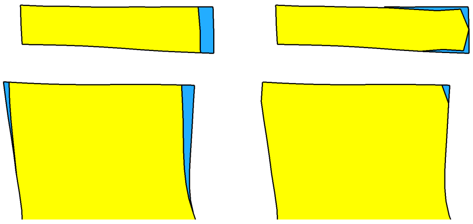

Figure 10: *Left: Not limiting the set of alterations for each patch side may result in impractical alteration. Right: Enabling multiple corner selections can lead to complex L-shaped corner alterations.*

Energy formulation. We want to propagate tears from the boundary areas with maximum stress. The intuition is that these areas benefit more from a tear that releases stress. We first associate to each possible location b_i and c_i a weight factor $w(b_i)$ and $w(c_i)$, which depends on the amount of stress tangential to the boundary. Intuitively, we want to propagate a tear orthogonally to the stress depending on the amount of tangential stress. Hence, given a boundary vertex v_i we can quantify the amount of tangential stress as the absolute dot product between the stretch of each fiber (ex-

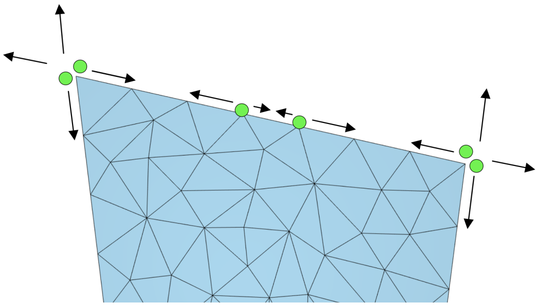

Figure 11: *Stress direction computation for each boundary (left) and corner points (right).*

pressed by the Jacobian) and the boundary direction (expressed using neighboring vertices). We illustrate this concept in Fig. 11.

Having defined the Boolean variables b_i and c_i and their corresponding weights $w(b_i)$ and $w(c_i)$, we can find a solution to the following problem:

$$\max_{b,c\in\{0,1\}} \quad \sum w(b_i)b_i + \gamma \sum w(c_i)c_i, \tag{2}$$

$$\forall s, \ \ \sum_{i\in s} b_i \leq \Theta \tag{3}$$

$$\forall (b_i,b_j) \in \mathcal{B}, \ \ b_i = b_j \tag{4}$$

$$\forall c_i, \ \ c_i^u + c_i^v \leq 1 \tag{5}$$

subject to the the constraints to limit the set of alterations (Eq. 3), the seams consistency (Eq. 4) and simple shape (Eq. 5). The parameter $\gamma \geq 1$ favors the insertion of tears (hence corresponding alterations) along existing seams and increases the *pattern coherence*. Solving this program, we determine a set of possible tear locations. However, we want to propagate only one tear at a time for each patch, since every modification can significantly change the overall stress distribution. Hence, we add an additional constraint to force only one Boolean variable for each patch to become 1. After solving the above ILP, we propagate the tear as much as required, update the physical simulation, recompute weights, and repeat this process to find a new fracture point. We stop insertion when the stress on each possible fracture starting point on patch boundaries is below a certain threshold μ.

Although iteratively inserting tears can reduce stress, some tears inserted early in the process may become redundant. Therefore, at the end of the process, we check each tear to see whether it is still necessary or has become obsolete due to later insertions. To address this, we test whether its removal is possible by reverting the tearing operation and then evaluating the resulting stress change. If the stress remains below μ, then the tear is redundant and can be reverted. This situation only occurred occasionally in our experiments.

6.3. Tearing

As previously explained, tears are created in regions of high stress and correspond to the areas where new fabric patches are inserted or removed. We allow two main kinds of tears: edge and corner.

Edge tear. This kind of tear can start from one point close to the middle of a patch side and requires more effort from a fabrication point of view, as inserting or removing a new patch in such a region involves cutting the original patches. Once the ILP decides that a fracture has to be propagated from a boundary vertex v, we first derive the propagation direction d. Ideally, the tear should propagate perpendicularly to the maximum stress direction. Given that we map one patch onto another, the stress of every triangular face can be described again as a 2×2 matrix J, we compute the direction of maximum stress using singular value decomposition (SVD). Then, we compute the average of these directions for each face surrounding v. Finally, the direction d is a unit vector orthogonal to this average stress direction. Notice that because the choice of the vertex on the boundary is determined by the stress along the boundary direction itself, the fracture will most likely propagate orthogonally to the boundary as well. As the stress direction at the tip of a tearing can change abruptly, we smooth the computed cut directions d_i with the previous one d_{i-1}, then $d_i = \alpha d_{i-1} + (1-\alpha)d_i$ for some $\alpha \in [0,1]$. If we start a new tear, no previous direction is available, and we set $\alpha = 0$. Fig. 12 (top) shows an example.

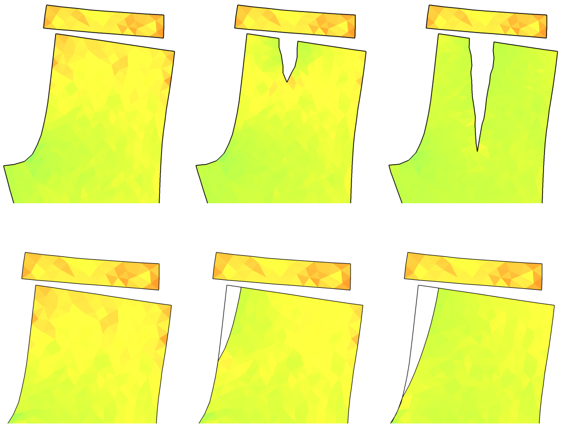

Figure 12: *A few steps of tear propagation from the middle of a side (top) or a corner (bottom).*

Corner tear. This type of tear starts at a corner and proceeds along the boundary. In practical applications, this tear corresponds to modifications that change existing seams, hence easier to realize. To create this kind of tear, we release boundary vertices from their constraints so they can move and relax stress. Fig. 12 (bottom) shows an example of tear propagation from a side.

Once the propagation direction d is computed, we propagate the fracture along the incident edge that aligns with it best. The procedure propagates the tear if the operation releases more stress than a given threshold. Otherwise, the propagation is stopped. While parameter α controls the cut directions' smoothness, the final boundary depends on the underlying triangulation and might be jagged. An irregular boundary is difficult to cut. At the end of tear propagation, we process the new borders using Taubin smoothing [Tau95]. At the end of this optimization procedure, we retrieve final alterations on $\mathcal{P}_{Alt}$ using 2D Boolean operations and triangulations.

7. Physically-based simulation

As outlined in Sec. 3, we formulate a simple method for cloth simulation using position-based dynamics (PBD) [MHHR07, MCKM14]. Position-based dynamics requires formulating a set of geometric constraints to define the mechanical behavior of the object at a small scale. Constraints are mostly defined over faces. Each face sets the target positions of its vertices that satisfy those constraints. Different constraints are then blended over vertices to derive the next position.

Usually, in position-based dynamics, cloth is simulated by simply constraining edge lengths. However, this formulation assumes that the textile has an isotropic behavior, which is usually false. Cloth offers much resistance to change of length along fiber's direction while, at the same time, allowing shear. A similar considera-

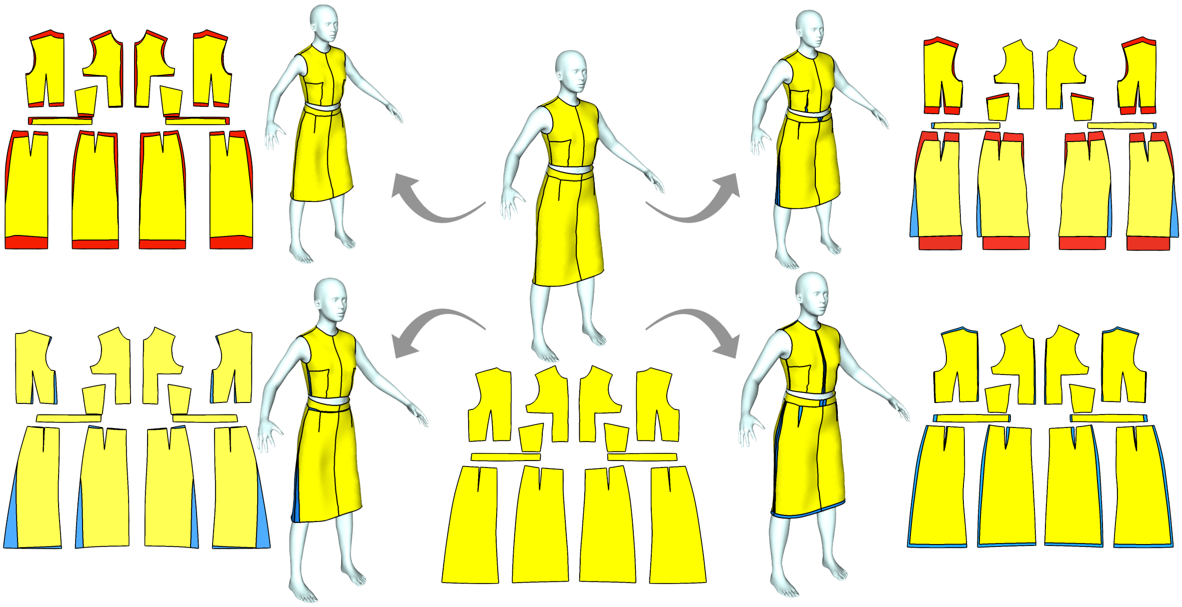

Figure 13: *Five variations of a female reference body. Notice the alteration in the top-right corner, which contains both patch insertion and removal.*

tion has been used for 2D flattening in [PDF*22] and in the context of cloth simulation in [GHF*07]. Here, we elaborate and extend such constraints to work with the PBD framework.

For each triangle, we compute the Jacobian J (as described in Sec. 5.1) that determines the in-plane deformation of each triangle, or, in physical terms, it quantifies the change of length of fibers and how much they deviate from being orthogonal (introducing shear). The columns of the 2×2 Jacobian matrix, J_u and J_v, indicate how two unit vectors aligned with the direction of the fibers deform from their rest shape (the 2D space of the patch) to the deformed state in 3D. We can define the geometric constraints that maintain *fiber inextensibility and orthogonality* by directly modifying J into a new $\bar{J}$ and adding constraints as follows:

Fiber inextensibility. By definition, J accounts for local deformation. To enforce the inextensibility of the fibers, it is sufficient to normalize the columns J_u and J_v. We then compute the target position of a vertex of the rest shape when applying the deformation without stretch: This includes local deformation, rotation and translation, whereas we find the latter two using procrustean analysis (cf. [SHR16]). This constraint models the anisotropic behavior of a textile that offers resistance along the direction of the fibers.

Fiber orthogonality. To control the shear of the fibers, we re-orthogonalize the vectors J_u and J_v and compute the target position of a vertex using the orthogonalized vectors. This constraint ensures that the unit frames remain perpendicular to each other, preventing the occurrence of excessive stretching along the diagonal direction.

To allow some degree of freedom to the deformation, we provide two parameters δ and ρ that control the maximum allowed amount of stretch and deviation from orthogonality. We then use fiber inextensibility for each vector J_u and J_v that deviate from unit length more than δ. Similarly, we use fiber orthogonality for each J_u and J_v whose angle differs from 90° by more than ρ and clamp them to an angle of $90^\circ \pm \rho$. These parameters govern the mechanical behavior and depend on the fabric properties. To account for invariance of rotation and translation in 3D, at each time step, we calculate the best-fitting rotation R and translation T using Procrustes analysis [SHR16], and we compose these transformations with the corrected $\bar{J}$ to recover the final target position of vertices.

Bending. The bending behavior of fabrics can vary significantly depending on their thickness and stiffness. We adopt the approach described in [MHHR07] to model bending. We add a positional constraint for every vertex belonging to pairs of adjacent faces where the angle between their normals exceeds a certain threshold.

Collisions. To manage the collisions between the garment mesh and the avatar mesh during simulation, we enforce a constraint to prevent the two meshes from intersecting. Similarly to many approaches in literature and industry [CLO22], we first pre-compute a distance field to the body avatar. At each simulation step, we first select vertices of the garment whose distance is below a certain threshold. Then we move away from the avatar mesh by translating them along the gradient direction of the distance field. We exclude self-collision detection for the garment in this phase.

8. Implementation details

In this section, we review the components of our pipeline that are not directly linked to the method itself. Yet, they remain essential tasks that must be addressed to derive the correct alteration.

Body-to-body mapping. The garment transportation algorithm discussed in Sec. 4 requires a smooth mapping between bodies

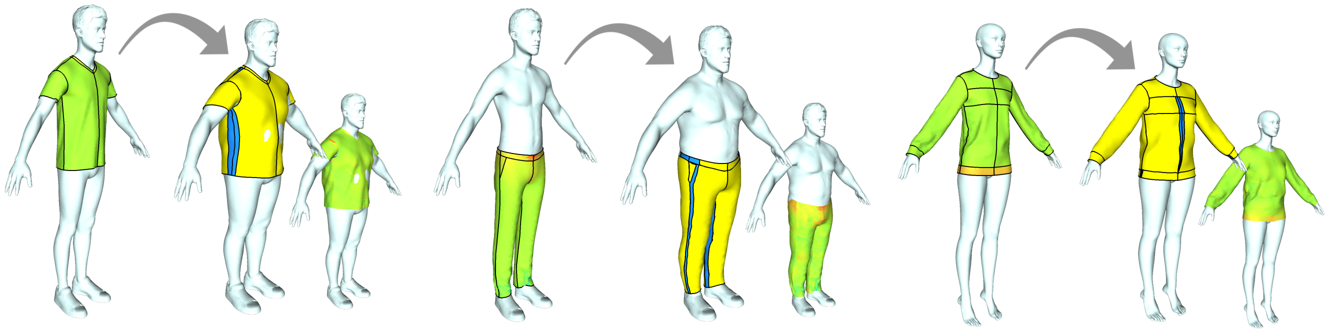

Figure 14: *Examples of garment alteration with different garments and fits.*

which is obtained by morphing the template body into the target body. We rely on a method similar to the one proposed by Falque et al. [FVCM*21]. The morphing is performed in a coarse-to-fine approach, where a rough alignment is first obtained between the two bodies using manual annotation. Then a refinement of the morphing is performed by searching for correspondence between the two surfaces at a vertex level. While the morphing automation is out of the scope of this paper, alternative fully automatic methods could be used by leveraging the literature on shape correspondence such as functional maps [OBCS*12], which have seen some recent improvement through the use of *zoomout* [MRR*19], additional regularization terms in the spectral domain [MO23], and the formulation of the method in a deep learning framework [LRR*17, DSO20].

Managing symmetry. In all the examples shown in the paper, the patterns are symmetric. Although the pipeline can handle non-symmetric garments, it is crucial to ensure symmetry during the process. Even though decorative elements like pockets may be asymmetrical, the overall structure of garments typically maintains symmetry. To achieve this, we enforce extrinsic symmetry by utilizing a plane. This approach allows us to flip the computation results from one side of the garment to the other while also updating the garment's geometry at each step.

9. Results

We implemented our pipeline C++ using the Eigen library [GJ*10] and libigl [JP*13]. To solve the linear programming problems for optimal alteration derivation (see Sec. 6.2), we used GUROBI [Gur12]. To derive the new patches for the alterations in correspondence with the tearing, we used Boolean operations in 2D in the Clipper library [Joh20]. To mesh the new areas created during alteration, we used Triangle [She96]. Our pipeline takes, on average, 30 seconds to fully process a dress composed of ∼10k triangles on a 2017 MacBook Pro with a 2.3 GHz Intel processor.

To demonstrate the validity of the proposed approach, we test the ability of our pipeline to generate a valid set of alterations and successfully replicate the initial fit to the target body. The results of these experiments are shown in Fig. 16. Next, we assess our method with various body and garment shapes. Then we demonstrate how the parameters affect the final set of alterations. Finally, we fabricate a few examples to show how the proposed method can be used in practice.

Body and garment variations. We assess our method by exploring different variations of body shapes. To generate a range of human body shapes, we use the procedural generation of MakeHuman [Fou20]. We start with an average female body shape, dress her in an input garment, and considered this our reference setup. Then we vary the body's size and height and let our method compute the different alterations. We show this experiment in Fig. 13. The reference body is in the middle of the image. The size varies on the horizontal axis, while height varies along the vertical axis. We report the final sewing pattern for each solution and highlight the alterations. As is possible to observe, our method derives a correct solution for all cases.

In Fig. 14, we show the results of our method on various kinds of garments with different fits, including loose fit (sweater), medium fit (pants), and tight fit (T-shirt). An additional example of the use of our algorithm for loose dresses is shown in Fig. 15.

In some cases, some part of the pattern has to be kept intact. For

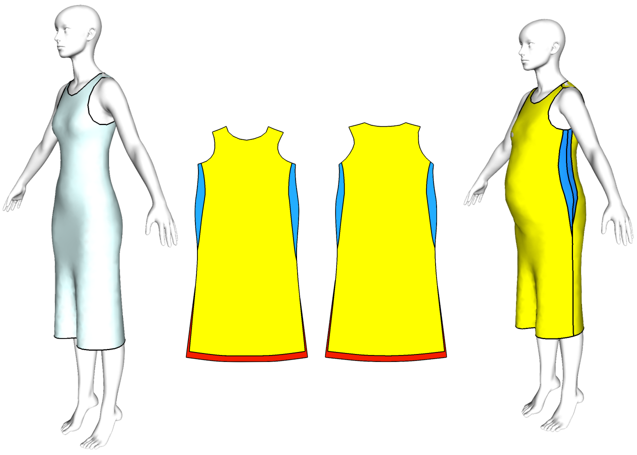

Figure 15: *Alteration of a loose garment requiring cloth insertion and removal.*

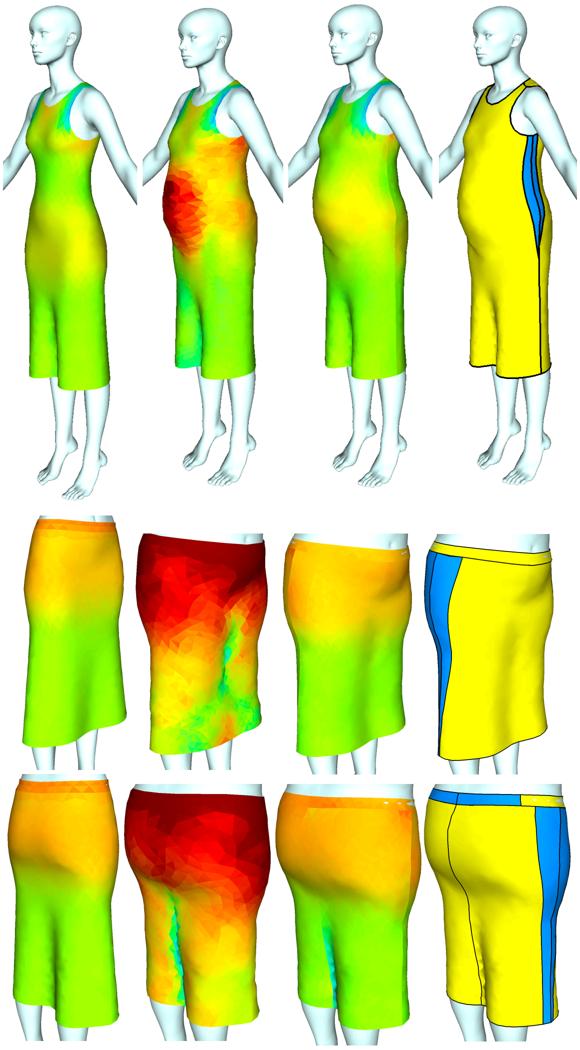

Figure 16: *Two different garments are altered to fit a different body shape in a similar way the garment fits the template body. In each experiment, the first column displays the fitting map for the template body, along with the resulting stress when the garment is simply draped onto the target body. The third column presents the fitting map after the alterations, while the final images depict the performed alterations, with blue indicating insertions.*

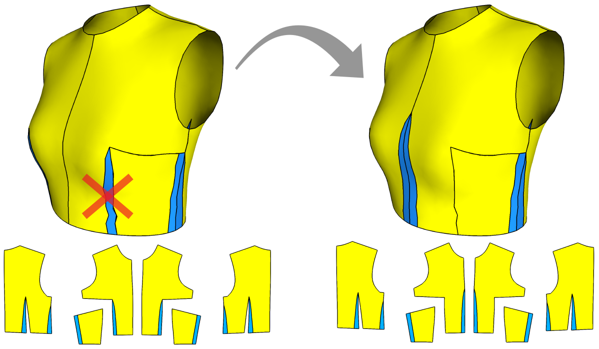

Figure 17: *Users can constrain the system to keep some sides intact in the final pattern. In this case, the system retrieves an alternative solution that avoids changes in the indicated side.*

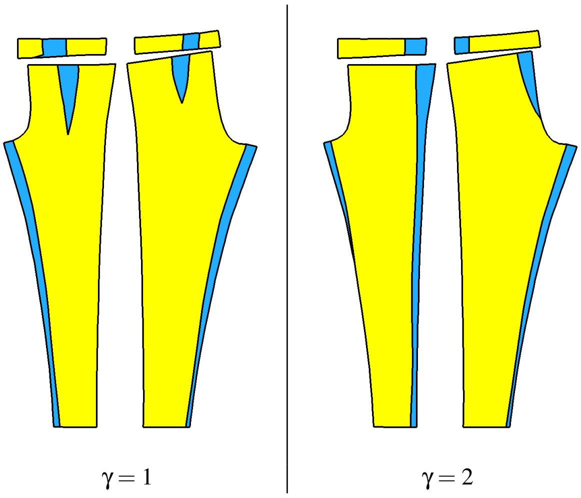

Figure 18: *The effect of changing the* γ *parameter in the final pattern; higher values of* γ *can be used to enforce seams coherency.*

example, we want to avoid modifying the side of a patch with a zipper; similarly, we cannot modify garment parts that correspond to pockets. Hence we allow the user to specify sides of pattern panels that have to remain intact. As shown in Fig. 17, the system converges to an alternative solution.

Parameter variations. A few parameters control the final sewing pattern. We control pattern coherency, as described in Sec. 6.2, via the parameter γ, which determines how many new cuts are generated or how much we rely on the pre-existing seams.

The user also can control the maximum allowed stretch on the final pattern. Intuitively, the higher the threshold, the fewer the alterations. As explained in Sec. 6.2, we control the maximum stretch with the parameter μ. Fig. 19 shows the effect on the final pattern for different values of this parameter.

Finally, Fig. 20 shows the effect of the parameter governing patch simplicity. As explained in Sec. 6.2, we can add a constraint to the linear program to allow only one corner tear to be selected and avoid L-shaped alterations.

Fabricated examples. We verify the fabricability and the accuracy of the derived sewing pattern. We acquire a real-world human body through 3D scanning and adapt existing leggings from a template avatar to fit the target person using our method. Then we cut and sew the obtained pattern. We show the result of this process in Fig. 21. In Fig. 22, we demonstrate how our method can be used to alter loose garments by adapting and fabricating a skirt for a 3D-scanned human model. In Fig. 23, we adapt a shirt to a 3D-scanned body,

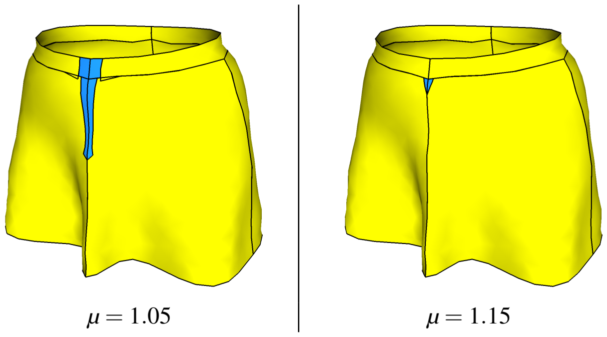

Figure 19: *The effect of changing the μ parameter in the final pattern.*

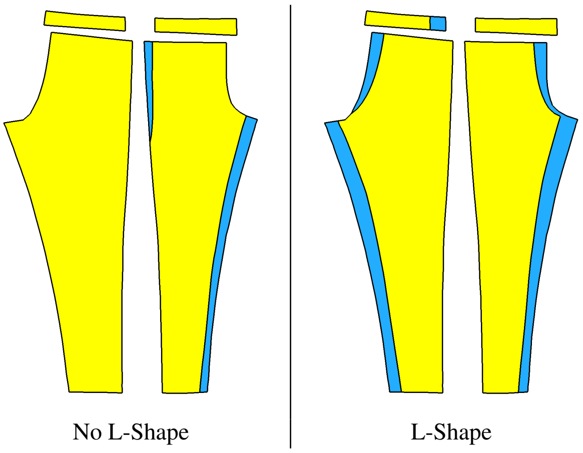

Figure 20: *The user can enable or disable the creation of L-shaped patches.*

but this time all the alterations are cloth removals. Finally, we show an extreme example of fabricated alteration in Fig. 1, where a pair of leggings for a child is adapted to an adult and vice-versa. Note that the material is added and removed correctly in the expected areas. We show a phase of the fabrication process in Fig. 24.

10. Conclusions

In this paper, we introduced a novel system designed to automatically compute a fabricable set of garment alterations, enabling users to alter an existing garment from a reference to a target body. Our pipeline exhibits robust capabilities in handling significant body variations and diverse drapes of input garments, ranging from tight to loose fits. Our approach involves a two-step process: initially, we optimize the sewing pattern to adapt to the new body, and subsequently, we derive a discrete set of operations to modify the initial sewing pattern for an optimal fit on the new body. The set of modifications is grounded in reality and physically achievable, as demonstrated by our results.

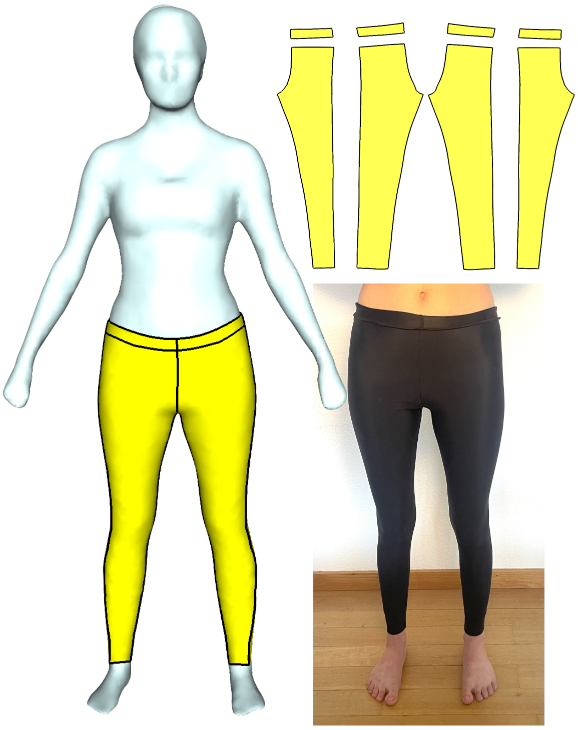

Figure 21: *Left: simulation and patterns; right: photo of worn leggings.*

Computational garment alteration is a relatively unexplored area in computer graphics and computational design. We have demonstrated the real-world applicability of our approach through several examples and physical fabrications of garments. Our system expands the horizons of sustainable garment design by leveraging computational methodologies for garment reuse through alteration.

Limitations and future work. Our method suffers from various limitations that can be addressed in the future. Firstly, our method requires an initial mapping between the reference and the target body. The mapping is used only to transport the garment from one body to another. We believe we can bypass this need with a more robust collision handling [HPSZ11]. The patch layout optimization does not guarantee bijectivity. The final mapping can have foldovers, or the garment can self-intersect in the 2D domain. However, we never experienced such a problem in practice. The overall framework can also benefit from robust management of self-intersections in the physical simulation. Our simulation can only model a limited range of physical behaviors, but it could be adapted to simulate more complex textiles. Finally, our ILP formulation allows for the potential insertion of multiple tears at once, which can expedite convergence. However, additional constraints might be needed to avoid intersections.

Acknowledgement

This work was supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (ERC Consolidator Grant, agreement No. 101003104, MYCLOTH) and by the Australian Space Research

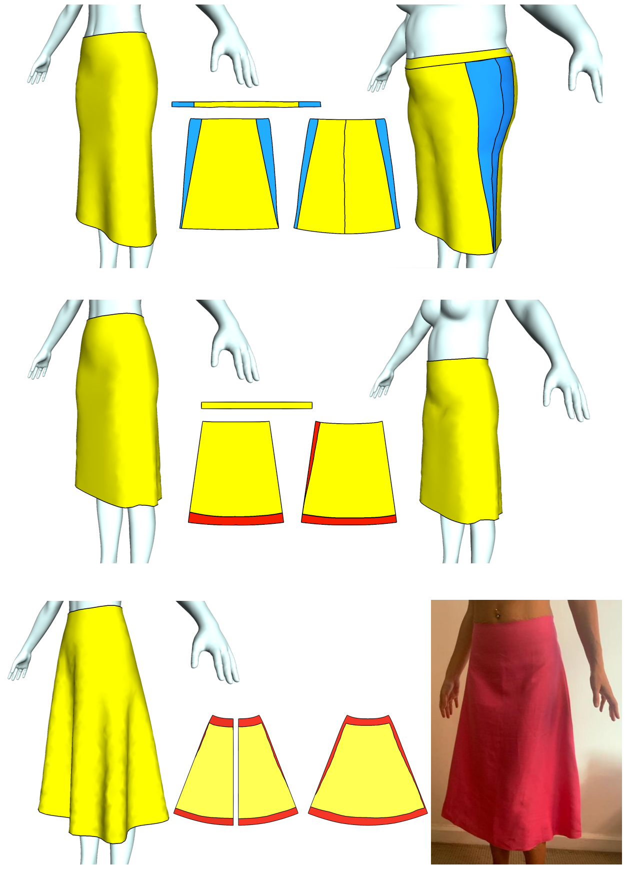

Figure 22: *Different adaptations of loose skirts and final fabrication. Left: the reference fit; right: the altered version.*

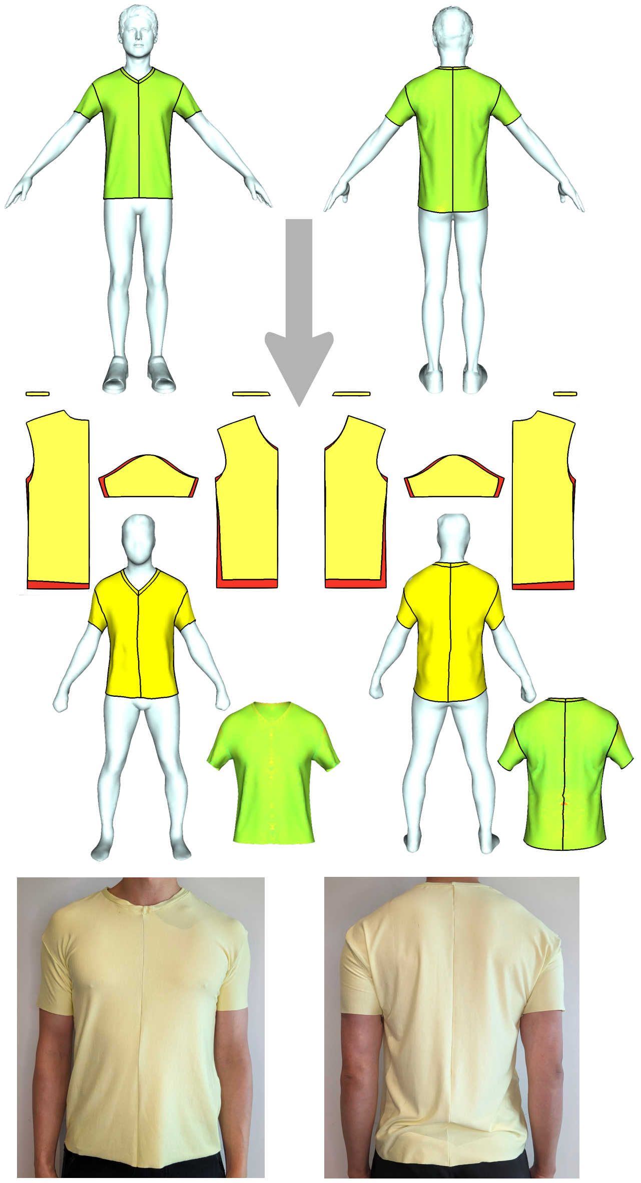

Figure 23: *Top: input digital avatar; bottom: photo of worn altered t-shirt.*

Network seed program (Automating the fit process for spacesuit design manufacturing).

References

[BKL21] Bang S., Korosteleva M., Lee S.-H.: Estimating garment patterns from static scan data. *Computer Graphics Forum 40* (05 2021). doi:10.1111/cgf.14272. 2

[BSBC12] Brouet R., Sheffer A., Boissieux L., Cani M.-P.: Design preserving garment transfer. *ACM Trans. Graph. 31*, 4 (jul 2012). URL: https://doi.org/10.1145/2185520.2185532, doi:10.1145/2185520.2185532. 2

[BSK*16] Bartle A., Sheffer A., Kim V. G., Kaufman D. M., Vining N., Berthouzoz F.: Physics-driven pattern adjustment for direct 3d garment editing. *ACM Trans. Graph. 35*, 4 (jul 2016). URL: https://doi.org/10.1145/2897824.2925896, doi:10.1145/2897824.2925896. 2, 3

[BVHSH21] Binninger* A., Verhoeven* F., Herholz P., Sorkine-Hornung O.: Developable approximation via gauss image thinning. *Computer Graphics Forum (proceedings of SGP 2021) 40*, 5 (2021), 289–300. doi:10.1111/cgf.14374. 2

[Cam17] Campen M.: Partitioning surfaces into quadrilateral patches: A survey. *Comput. Graph. Forum 36*, 8 (2017), 567–588. 2

[CHT*07] Cani M., Hughes J. F., Turquin E., Boissieux L., Wither J.: A sketch-based interface for clothing virtual characters. *IEEE Computer Graphics and Applications 27*, 01 (jan 2007), 72–81. doi:10.1109/MCG.2007.1. 2

[CK14] Campen M., Kobbelt L.: Dual strip weaving: interactive design of quad layouts using elastica strips. *ACM Trans. Graph. 33*, 6 (2014), 183:1–183:10. 2

[CLO22] CLO: clo3d.com. https://www.clo3d.com, Jan. 2022. 1, 2, 9

[CST03] Cordier F., Seo H., Thalmann N.: Made-to-measure technologies for an online clothing store. *Computer Graphics and Applications, IEEE 23* (02 2003), 38– 48. doi:10.1109/MCG.2003.1159612. 2

[CZL*15] Chen X., Zhou B., Lu F., Wang L., Bi L., Tan P.: Garment modeling with a depth camera. *ACM Trans. Graph. 34*,

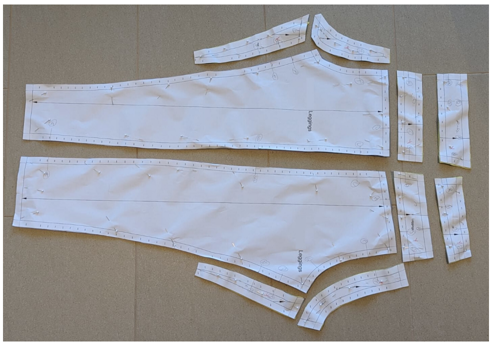

Figure 24: *The fabrication process of a pair of leggings. The printed patterns are equipped with extra space (seam allowance) to facilitate sewing. We also place labels to guide the sewing process.*

6 (oct 2015). URL: https://doi.org/10.1145/2816795.2818059, doi:10.1145/2816795.2818059. 2

[DJW*06] Decaudin P., Julius D., Wither J., Boissieux L., Sheffer A., Cani M.-P.: Virtual garments: A fully geometric approach for clothing design. *Comput. Graph. Forum 25*, 3 (2006), 625–634. doi:https://doi.org/10.1111/j.1467-8659.2006.00982.x. 2

[dMQP*23] de Malefette C., Qi A., Parakkat A. D., Cani M.-P., Igarashi T.: Perfectdart: Automatic dart design for garment fitting. In *SIGGRAPH Asia 2023 Technical Communications* (New York, NY, USA, 2023), SA '23, Association for Computing Machinery. URL: https://doi.org/10.1145/3610543.3626154, doi:10.1145/3610543.3626154. 3

[DSO20] Donati N., Sharma A., Ovsjanikov M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8592–8601. 10

[Fou20] Foundation B.: Makehuman, 2020. URL: http://www.makehumancommunity.org. 10

[Fre24] FreeSewing: freesewing.org, 2024. URL: http://freesewing.org. 2

[FVCM*21] Falque R., Vidal-Calleja T., McPhee M., Toohey E., Alempijevic A.: Virtualbutcher: Coarse-to-fine annotation transfer of cutting lines on noisy point cloud reconstruction. In *2021 IEEE 21st International Symposium on Computational Intelligence and Informatics (CINTI)* (2021), IEEE, pp. 000109–000114. doi:https://www.researchgate.net/publication/358123125_VirtualButcher_Coarse-to-fine_Annotation_Transfer_of_Cutting_Lines_on_Noisy_Point_Cloud_Reconstruction. 10

[GHF*07] Goldenthal R., Harmon D., Fattal R., Bercovier M., Grinspun E.: Efficient simulation of inextensible cloth. *ACM Trans. Graph. 26*, 3 (jul 2007), 49–es. URL: https://doi.org/10.1145/1276377.1276438, doi:10.1145/1276377.1276438. 9

[GJ*10] Guennebaud G., Jacob B., et al.: Eigen v3. http://eigen.tuxfamily.org, 2010. URL: http://eigen.tuxfamily.org. 10

[Gur12] Gurobi Optimization, Inc.: Gurobi optimizer reference manual, 2012. URL: http://www.gurobi.com. 10

[HPSZ11] Harmon D., Panozzo D., Sorkine O., Zorin D.: Interference-aware geometric modeling. *ACM Trans. Graph. 30*, 6 (dec 2011), 1–10. URL: https://doi.org/10.1145/2070781.2024171, doi:10.1145/2070781.2024171. 12

[IRHS20] Ion A., Rabinovich M., Herholz P., Sorkine-Hornung O.: Shape approximation by developable wrapping. *ACM Transactions on Graphics (proceedings of SIGGRAPH ASIA) 39*, 6 (2020). doi:10.1145/3414685.3417835. 2

[JKS05] Julius D., Kraevoy V., Sheffer A.: D-charts: Quasi-developable mesh segmentation. *Computer Graphics Forum 24*, 3 (2005), 581–590. 2

[Joh20] Johnson A.: Clipper2 - polygon clipping and offsetting library, 2020. URL: https://angusj.com/clipper2/Docs/Overview.htm. 10

[JP*13] Jacobson A., Panozzo D., et al.: libigl: A simple C++ geometry processing library, 2013. http://igl.ethz.ch/projects/libigl/. URL: http://igl.ethz.ch/projects/libigl/. 10

[KL21] Korosteleva M., Lee S.-H.: Generating datasets of 3d garments with sewing patterns. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks* (2021), Vanschoren J., Yeung S., (Eds.), vol. 1. URL: https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/013d407166ec4fa56eb1e1f8cbe183b9-Paper-round1.pdf. 2

[KP07] Kim S., Park C.: Basic garment pattern generation using geometric modeling method. *International Journal of Clothing Science and Technology 19* (01 2007), 7–17. doi:10.1108/09556220710717017. 3

[KSH23] Korosteleva M., Sorkine-Hornung O.: Garmentcode: Programming parametric sewing patterns. *ACM Transactions on Graphics 42*, 6 (Dec. 2023), 1–15. URL: http://dx.doi.org/10.1145/3618351, doi:10.1145/3618351. 2

[LHZ*21] Liu Z., Han X., Zhang Y., Chen X., Lai Y.-K., Doubrovski E. L., Whiting E., Wang C. C. L.: Knitting 4d garments with elasticity controlled for body motion. *ACM Trans. Graph. 40*, 4 (jul 2021). URL: https://doi.org/10.1145/3450626.3459868, doi:10.1145/3450626.3459868. 3

[LKK*18] Li M., Kaufman D. M., Kim V. G., 0001 J. S., Sheffer A.: Optcuts: joint optimization of surface cuts and parameterization. *ACM Trans. Graph 37*, 6 (2018), 247:1–247:13. 2

[LPP*20] Livesu M., Pietroni N., Puppo E., Sheffer A., Cignoni P.: Loopycuts: practical feature-preserving block decomposition for strongly hex-dominant meshing. *ACM Trans. Graph. 39*, 4 (2020), 121. URL: https://doi.org/10.1145/3386569.3392472, doi:10.1145/3386569.3392472. 2

[LRR*17] Litany O., Remez T., Rodola E., Bronstein A., Bronstein M.: Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 5659–5667. 10

[LSGV18] Li M., Sheffer A., Grinspun E., Vining N.: Foldsketch: enriching garments with physically reproducible folds. *ACM Trans. Graph 37*, 4 (2018), 133:1–133:13. URL: https://doi.org/10.1145/3197517.3201310, doi:10.1145/3197517.3201310. 3

[LyCL*24] Li Y., yu Chen H., Larionov E., Sarafianos N., Matusik W., Stuyck T.: Diffavatar: Simulation-ready garment optimization with differentiable simulation, 2024. URL: https://arxiv.org/abs/2311.12194, arXiv:2311.12194. 3

[LZB*18] Liu K., Zeng X., Bruniaux P., Tao X., Yao X., Li V., Wang J.: 3d interactive garment pattern-making technology. *Computer-Aided Design 104* (2018), 113–124. URL: https://www.sciencedirect.com/science/article/pii/S0010448518304093, doi:https://doi.org/10.1016/j.cad.2018.07.003. 2

[LZX*08] Liu L., Zhang L., Xu Y., Gotsman C., Gortler S. J.: A local/global approach to mesh parameterization. *Computer Graphics Forum 27*, 5 (2008), 1495–1504. URL:

https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2008.01290.x, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2008.01290.x, doi:https://doi.org/10.1111/j.1467-8659.2008.01290.x. 5

[MAN*16] McCann J., Albaugh L., Narayanan V., Grow A., Matusik W., Mankoff J., Hodgins J.: A compiler for 3d machine knitting. *ACM Trans. Graph. 35*, 4 (jul 2016). URL: https://doi.org/10.1145/2897824.2925940, doi:10.1145/2897824.2925940. 3

[MCKM14] Müller M., Chentanez N., Kim T.-Y., Macklin M.: Strain Based Dynamics. In *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation* (2014), Koltun V., Sifakis E., (Eds.), The Eurographics Association. doi:10.2312/sca.20141133. 8

[MHHR07] Müller M., Heidelberger B., Hennix M., Ratcliff J.: Position based dynamics. *Journal of Visual Communication and Image Representation 18*, 2 (2007), 109–118. URL: https://www.sciencedirect.com/science/article/pii/S1047320307000065, doi:https://doi.org/10.1016/j.jvcir.2007.01.005. 2, 8, 9

[MO23] Magnet R., Ovsjanikov M.: Scalable and efficient functional map computations on dense meshes. *Computer Graphics Forum 42*, 2 (2023), 89–101. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14746, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14746, doi:https://doi.org/10.1111/cgf.14746. 10

[MRR*19] Melzi S., Ren J., Rodolà E., Sharma A., Wonka P., Ovsjanikov M.: Zoomout: spectral upsampling for efficient shape correspondence. *ACM Trans. Graph. 38*, 6 (nov 2019). URL: https://doi.org/10.1145/3355089.3356524, doi:10.1145/3355089.3356524. 10

[MTMP20] Montes J., Thomaszewski B., Mudur S., Popa T.: Computational design of skintight clothing. *ACM Trans. Graph. 39*, 4 (jul 2020). URL: https://doi.org/10.1145/3386569.3392477, doi:10.1145/3386569.3392477. 2, 3

[MWJ12] Meng Y., Wang C. C., Jin X.: Flexible shape control for automatic resizing of apparel products. *Computer-Aided Design 44*, 1 (2012), 68–76. Digital Human Modeling in Product Design. URL: https://www.sciencedirect.com/science/article/pii/S0010448510002186, doi:https://doi.org/10.1016/j.cad.2010.11.008. 2

[NAH*18] Narayanan V., Albaugh L., Hodgins J., Coros S., McCann J.: Automatic machine knitting of 3d meshes. *ACM Trans. Graph. 37*, 3 (Aug. 2018), 35:1–35:15. URL: http://doi.acm.org/10.1145/3186265, doi:10.1145/3186265. 3

[NHE*19] Nuvoli S., Hernandez A., Esperança C., Scateni R., Cignoni P., Pietroni N.: Quadmixer: layout preserving blending of quadrilateral meshes. *ACM Trans. Graph. 38*, 6 (nov 2019). URL: https://doi.org/10.1145/3355089.3356542, doi:10.1145/3355089.3356542. 2

[NWYM19] Narayanan* V., Wu* K., Yuksel C., McCann J.: Visual knitting machine programming. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019) 38*, 4 (jul 2019), 63:1–63:13. (*Joint First Authors). URL: http://doi.acm.org/10.1145/3306346.3322995, doi:10.1145/3306346.3322995. 3

[OBCS*12] Ovsjanikov M., Ben-Chen M., Solomon J., Butscher A., Guibas L.: Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph. 31*, 4 (jul 2012). URL: https://doi.org/10.1145/2185520.2185526, doi:10.1145/2185520.2185526. 10

[Opt22] Optitex: optitex.com. https://optitex.com, Jan. 2022. 2

[PDF*22] Pietroni N., Dumery C., Falque R., Liu M., Vidal-Calleja T., Sorkine-Hornung O.: Computational pattern making from 3d garment models. *ACM Trans. Graph. 41*, 4 (jul 2022). URL: https://doi.org/10.1145/3528223.3530145, doi:10.1145/3528223.3530145. 1, 3, 5, 6, 9

[PMPHB17] Pons-Moll G., Pujades S., Hu S., Black M. J.: Clothcap: seamless 4d clothing capture and retargeting. *ACM Trans. Graph. 36*, 4 (jul 2017). URL: https://doi.org/10.1145/3072959.3073711, doi:10.1145/3072959.3073711. 2

[PNA*21] Pietroni N., Nuvoli S., Alderighi T., Cignoni P., Tarini M.: Reliable feature-line driven quad-remeshing. *ACM Trans. Graph. 40*, 4 (jul 2021). URL: https://doi.org/10.1145/3450626.3459941, doi:10.1145/3450626.3459941. 2

[PPM*16] Pietroni N., Puppo E., Marcias G., Scopigno R., Cignoni P.: Tracing field-coherent quad layouts. *Comput. Graph. Forum 35*, 7 (2016), 485–496. URL: https://doi.org/10.1111/cgf.13045, doi:10.1111/cgf.13045. 2

[PTH*17] Poranne R., Tarini M., Huber S., Panozzo D., Sorkine-Hornung O.: Autocuts: Simultaneous distortion and cut optimization for uv mapping. *ACM Trans. Graph. 36*, 6 (nov 2017). URL: https://doi.org/10.1145/3130800.3130845, doi:10.1145/3130800.3130845. 2

[QI23] Qi A., Igarashi T.: Perfecttailor: Scale-preserving 2d pattern adjustment driven by 3d garment editing, 2023. arXiv:2312.08386. 3

[RMSC11] Robson C., Maharik R., Sheffer A., Carr N.: Context-aware garment modeling from sketches. *Computers and Graphics 35*, 3 (2011), 604–613. Shape Modeling International (SMI) Conference 2011. URL: https://www.sciencedirect.com/science/article/pii/S0097849311000410, doi:https://doi.org/10.1016/j.cag.2011.03.002. 2

[RRP15] Razaflndrazaka F. H., Reitebuch U., Polthier K.: Perfect matching quad layouts for manifold meshes. In *Proceedings of the Eurographics Symposium on Geometry Processing* (Goslar, DEU, 2015), SGP '15, Eurographics Association, p. 219–228. URL: https://doi.org/10.1111/cgf.12710, doi:10.1111/cgf.12710. 2

[RSW*07] Rose K., Sheffer A., Wither J., Cani M.-P., Thibert B.: Developable surfaces from arbitrary sketched boundaries. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Goslar, DEU, 2007), SGP '07, Eurographics Association, p. 163–172. 2

[SCOGL02] Sorkine O., Cohen-Or D., Goldenthal R., Lischinski D.: Bounded-distortion piecewise mesh parameterization. In *IEEE Visualization, 2002. VIS 2002.* (2002), pp. 355–362. doi:10.1109/VISUAL.2002.1183795. 2

[SGC18] Stein O., Grinspun E., Crane K.: Developability of triangle meshes. *ACM Trans. Graph. 37*, 4 (jul 2018). URL: https://doi.org/10.1145/3197517.3201303, doi:10.1145/3197517.3201303. 2

[She96] Shewchuk J. R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Applied Computational Geometry: Towards Geometric Engineering*, Lin M. C., Manocha D., (Eds.), vol. 1148 of *Lecture Notes in Computer Science*. Springer-Verlag, May 1996, pp. 203–222. From the First ACM Workshop on Applied Computational Geometry. 10

[SHR16] Sorkine-Hornung O., Rabinovich M.: Least-squares rigid motion usingsvd, 2016. Technical note. 9

[Tau95] Taubin G.: Curve and surface smoothing without shrinkage. In *Proceedings of IEEE International Conference on Computer Vision* (1995), pp. 852–857. doi:10.1109/ICCV.1995.466848. 8

[UKIG11] Umetani N., Kaufman D. M., Igarashi T., Grinspun E.: Sensitive couture for interactive garment modeling and editing. *ACM Transactions on Graphics (SIGGRAPH 2011) 30*, 4 (2011), 90. 2

[VCD*17] Vaxman A., Campen M., Diamanti O., Bommes D., Hildebrandt K., Technion M. B.-C., Panozzo D.: Directional field synthesis, design, and processing. In *ACM SIGGRAPH 2017 Courses* (New York, NY, USA, 2017), SIGGRAPH '17, Association for Computing Machinery. URL: https://doi.org/10.1145/3084873.3084921, doi:10.1145/3084873.3084921. 2

[VSGC20] Vidaurre R., Santesteban I., Garces E., Casas D.: Fully convolutional graph neural networks for parametric virtual try-on. *Computer Graphics Forum 39*, 8 (Dec. 2020), 145–156. doi:https://doi.org/10.1111/cgf.14109. 2

[Wan18] Wang H.: Rule-free sewing pattern adjustment with precision and efficiency. *ACM Trans. Graph. 37*, 4 (jul 2018). URL: https://doi.org/10.1145/3197517.3201320, doi:10.1145/3197517.3201320. 2

[WCPM18] Wang T. Y., Ceylan D., Popovic J., Mitra N. J.: Learning a shared shape space for multimodal garment design. *ACM Trans. Graph. 37*, 6 (2018), 1:1–1:14. doi:10.1145/3272127.3275074. 2

[WHZ*21] Wolff K., Herholz P., Ziegler V., Link F., Brügel N., Sorkine-Hornung O.: 3D custom fit garment design with body movement, 2021. URL: https://arxiv.org/abs/2102.05462, arXiv:2102.05462. 2

[WSY19] Wu K., Swan H., Yuksel C.: Knittable stitch meshes. *ACM Transactions on Graphics 38*, 1 (jan 2019), 10:1–10:13. URL: http://doi.acm.org/10.1145/3292481, doi:10.1145/3292481. 3

[WWY05] Wang C., Wang Y., Yuen M.: Design automation for customized apparel products. *Computer-Aided Design 37*, 7 (June 2005), 675–691. doi:10.1016/j.cad.2004.08.007. 2

[YKJM12] Yuksel C., Kaldor J. M., James D. L., Marschner S.: Stitch meshes for modeling knitted clothing with yarn-level detail. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31*, 3 (2012), 37:1–37:12. URL: http://doi.acm.org/10.1145/2185520.2185533, doi:10.1145/2185520.2185533. 3

Appendix A: Appendix

The following pseudo-code provides a comprehensive overview of our entire processing pipeline. In step 1a, we simulate the draping of the garment on the template body and compute the target fitting map (as shown in Fig. 2.a). In step 1b, we drape the original garment and evaluate the resulting fitting map. In step 2, we freely modify the seams to derive the target patch layout that matches the target fitting map (see Fig. 2.b). Finally, we alternate between tearing propagation and ILP global solving to derive the final set of feasible fabric alterations (see Figure 2.c).

ALGORITHM 1: Garment Alteration Pseudocode

```
Data: $\mathcal{G}_R$ = reference garment
Data: $\mathcal{P}_R$ = reference sewing pattern
Data: $\mathcal{A}_R$ = reference avatar
Data: $\mathcal{A}_T$ = target avatar
/* Step 1a: Simulate reference garment on
   reference avatar and compute fitting map
   */
simulateGarmentOnBody($\mathcal{G}_R, \mathcal{A}_R, \mathcal{P}_R$);
$\mathcal{F}_{(\mathcal{G}_T)} \leftarrow$ computeFittingMap($\mathcal{G}_R, \mathcal{P}_R, \mathcal{A}_R$);
/* Step 1b: Drape garment on target avatar
   and simulate                                 */
$\mathcal{G}_T \leftarrow$ drapeGarmentOnTargetAvatar($\mathcal{A}_T, \mathcal{G}_R, \mathcal{P}_R$);
simulateGarmentOnBody($\mathcal{G}_T, \mathcal{A}_T, \mathcal{P}_T$);
/* Step 2: Iteratively compute the target
   sewing pattern                               */
while not converged do
    $\mathcal{P}_T \leftarrow$ inverseJacobianOptimization($\mathcal{F}_{(\mathcal{G}_T)}, \mathcal{G}_T$);
end
/* Step 3a: Addition of fabric                  */
/* Stretch the initial pattern to match the
   boundary of the target sewing pattern        */
$\mathcal{P}_S \leftarrow$ stretchPatternAndSimulate($\mathcal{P}_R, \mathcal{P}_T$);
$ILP \leftarrow$ setupIntegerLinearProblem($\mathcal{P}_S$);
tearLocations $\leftarrow$ solveILP($ILP$);
foreach $i \in$ tearLocations do
    while stress at tip $\geq$ threshold do
        propagateFracture($\mathcal{P}_S$);
    end
end
fabricToAdd $\leftarrow$ retriangulateDifference($\mathcal{P}_S, \mathcal{P}_T$);
/* Step 3b: Removal of fabric                   */
/* Stretch the target pattern to match the
   boundary of the initial sewing pattern       */
$\mathcal{P}_S \leftarrow$ stretchPatternAndSimulate($\mathcal{P}_T, \mathcal{P}_R$);
$ILP \leftarrow$ setupIntegerLinearProblem($\mathcal{P}_S$);
tearLocations$\leftarrow$ solveILP($ILP$);
foreach $i \in$ tearLocations do
    while stress at tip $\geq$ threshold do
        propagateFracture($\mathcal{P}_S$);
    end
end
fabricToRemove $\leftarrow$ retriangulateDifference($\mathcal{P}_S, \mathcal{P}_R$);
```