Computational Pattern Making from 3D Garment Models

NICO PIETRONI, University of Technology Sydney, Australia
CORENTIN DUMERY, ETH Zurich, Switzerland
RAPHAEL FALQUE, University of Technology Sydney, Australia
MARK LIU, University of Technology Sydney, Australia
TERESA VIDAL-CALLEJA, University of Technology Sydney, Australia
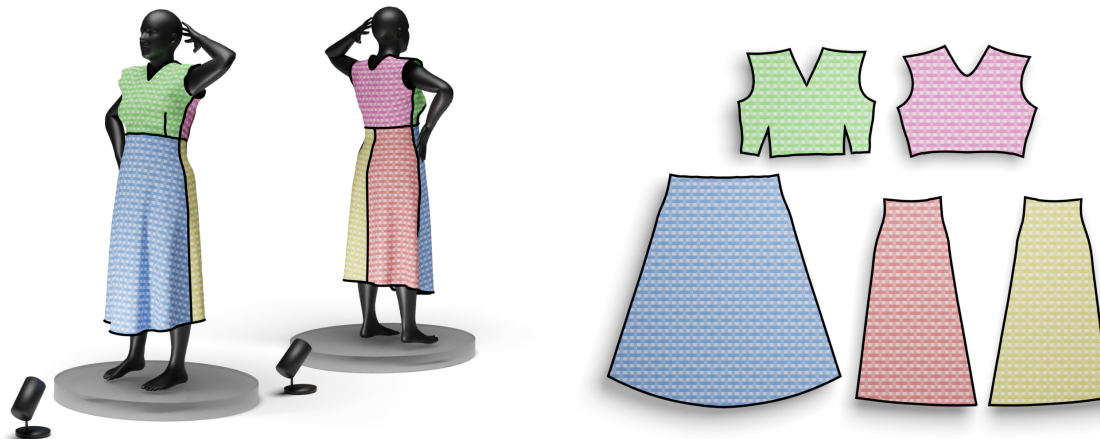OLGA SORKINE-HORNUNG, ETH Zurich, Switzerland

Fig. 1. Our technique receives a 3D mesh of a given garment as input, automatically segments the shape into patches (left and middle) and computes the 2D parametrerization required to produce a pattern (right) that can be cut out and sewn to fabricate the actual garment.

We propose a method for computing a sewing pattern of a given 3D garment model. Our algorithm segments an input 3D garment shape into patches and computes their 2D parameterization, resulting in pattern pieces that can be cut out of fabric and sewn together to manufacture the garment. Unlike the general state-of-the-art approaches for surface cutting and flattening, our method explicitly targets garment fabrication. It accounts for the unique properties and constraints of tailoring, such as seam symmetry, the usage of darts, fabric grain alignment, and a flattening distortion measure that models woven fabric deformation, respecting its anisotropic behavior. We bootstrap a recent patch layout approach developed for quadrilateral remeshing and adapt it to the purpose of computational pattern making, ensuring that the deformation of each pattern piece stays within prescribed bounds of cloth stress. While our algorithm can automatically produce the sewing patterns, it is fast enough to admit user input to creatively iterate on the pattern design. Our method can take several target poses of the 3D garment into account and integrate them into the sewing pattern design. We demonstrate results on both skintight and loose garments, showcasing the versatile application possibilities of our approach.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; **Shape modeling**; **Mesh geometry models**.

Additional Key Words and Phrases: pattern making, cloth parameterization, patch layout, garment fabrication

Authors' addresses: Nico Pietroni, University of Technology Sydney, Sydney, Australia, nico.pietroni@uts.edu.au; Corentin Dumery, ETH Zurich, Zurich, Switzerland, corentin.dumery@inf.ethz.ch; Raphael Falque, University of Technology Sydney, Sydney, Australia, raphael.guenot-falque@uts.edu.au; Mark Liu, University of Technology Sydney, Sydney, Australia, Mark.Liu@uts.edu.au; Teresa Vidal-Calleja, University of Technology Sydney, Sydney, Australia, teresa.vidalcalleja@uts.edu.au; Olga Sorkine-Hornung, ETH Zurich, Zurich, Switzerland, sorkine@inf.ethz.ch.

1 INTRODUCTION

In this work, we propose a method for automatically creating a sewing pattern for a given 3D model of a garment.

In the fashion industry, the garment creation process starts from the 2D domain: The pattern maker creates the 2D sewing pattern using traditional, often tacit knowledge [Chen 1998], established templates and a few standard measurements, such as waist circumference, shoulder width, etc. The cut fabric pieces are then sewn together to form the garment. Designers may work with mannequins to experiment with the desired shape and draping of the fabric in the physical 3D space, but the ultimate determination of the garment shape comes from the 2D pattern. Also, in digital garment design

tools such as Clo3D [2022], the 2D pattern is required to simulate and drape the garment on a virtual mannequin or avatar, and thus the design process is centered around the sewing pattern. Pattern making is currently a manual, time-consuming task requiring much experience and skill. One of the consequences is that custom made clothing that perfectly fits the intended wearer is a luxury available to the privileged few. The majority of clothing is mass-produced using standard sizes based on averaged measurements that do not fit well for most people [SizeGermany 2020].

There are many powerful techniques to model shapes directly in 3D, with various approaches specifically developed for 3D garment modeling, e.g. with sketch based user interfaces [Cani et al. 2007; Decaudin et al. 2006; Robson et al. 2011; Rose et al. 2007; Wolff et al. 2021], input gestures in the physical [Wibowo et al. 2012] or virtual reality [TiltBrush 2022], via 3D scanning [Pons-Moll et al. 2017] or data driven modeling [Wang et al. 2018]. Advances in human body modeling [Alldieck et al. 2021; Bogo et al. 2014; Loper et al. 2015; Osman et al. 2020] facilitate custom digital tailoring of garments that fit the personalized body avatar without having to rely on standard sizes. However, the 2D sewing pattern of the garment is still necessary for accurate cloth simulation and the manufacturing of the physical garment.

To create the 2D pattern, the 3D garment surface is divided into patches, also called panels, and each panel is flattened onto the 2D domain. Many methods exist for surface segmentation and low-distortion parameterization, as we discuss in Sec. 2. State-of-the-art approaches are able to compute optimal cuts that balance the flattening distortion and cut length [Li et al. 2018a; Poranne et al. 2017; Sharp and Crane 2018]. Recent work [Wolff et al. 2021] applies such a technique to create the sewing pattern, but the resulting panel shapes are visually far from common practice in fashion and challenging to sew. The problem is that most segmentation techniques are general and do not account for the specific setting of cloth pattern making and the fabrication constraints unique to it. In particular, parameterizing a 3D shape to be made of woven textile requires a special deformation model aware of the thread properties and fabric grain alignment, as well as shape and length constraints on the seams and garment aesthetics. For example, to sew together two pieces of cloth, in practice, the two matching seams must be of the same length, as straight or at least smooth as possible, and ideally they should be reflection-symmetric in 2D, so that the tailor can put the panels one on top of the other and sew them together along one planar curve.

In this work, we propose a shape segmentation and patch flattening method specifically intended for pattern making (see Fig. 1). Our core contributions are a 3D patch layout creation approach informed by a geometric measure of woven fabric distortion and tailoring fabrication requirements, as well as a patch parameterization method that minimizes the textile deformation measure and ensures that the pattern pieces are sewable. Our method incorporates mechanisms of classical pattern making, such as the usage of darts, preference towards global symmetry and vertical grain alignment by default. Our algorithm can work fully automatically, but it is efficient enough to admit interactive input, allowing creative design iteration in real time. The designer can sketch on the 3D garment model to hint at desired seams and textile grain alignment and vary
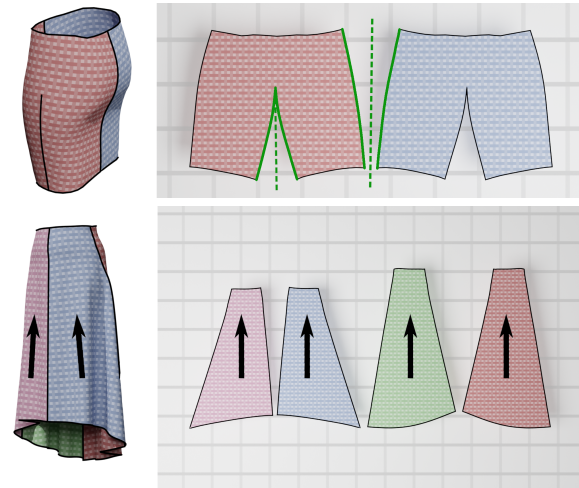

Fig. 2. Top: matching seams on two patches to be stitched together, as well as the sides of a dart must be of equal length and ideally reflection-symmetric. Bottom: the fabric grain of pattern pieces aims to align with the vertical direction on the worn garment.

the flattening parameters according to the physical properties of the desired fabric. As an additional option, if multiple poses of the garment corresponding to various body poses are available, our algorithm can adapt the pattern by integrating the information from all the poses. We demonstrate results on a number of tight-fitting and loose garments, showcasing the usability and versatility of our method. To foster future research on digital fashion, we publicly release our software implementation.

2 RELATED WORK

Digital garment design and fabrication pose several fundamental, interconnected research problems in geometry processing and physics based simulation. We give a brief review of the methods for surface modeling, patch decomposition and flattening in the context of computational pattern making.

Garment shape design. The digitalization of the fashion industry in general and garment design in particular bears multiple economical, ecological and societal advantages and poses fascinating research challenges, sparking significant interest [Nayak and Padhye 2017]. Clo3D [2022] and Optitex [2022] are examples of common CAD tools used in the industry. Such interactive CAD editors allow the designer to create sewing patterns in 2D and simulate their physical appearance and draping on an avatar in 3D. Adjusting the 3D garment shape requires changing the 2D pattern via a trial-and-error process. In the research community, Umetani et al. [2011] propose bidirectional interactive garment editing, leveraging fast cloth simulation to enable users to work in 2D and 3D simultaneously and observe the effects of changes in both modes.

Various computational approaches create 3D garment shapes from contours, seam-, fold- and boundary curves, sketched by the user on an avatar, see e.g. [Cani et al. 2007; Decaudin et al. 2006; Robson et al. 2011; Rose et al. 2007; Wolff et al. 2021]. Other approaches

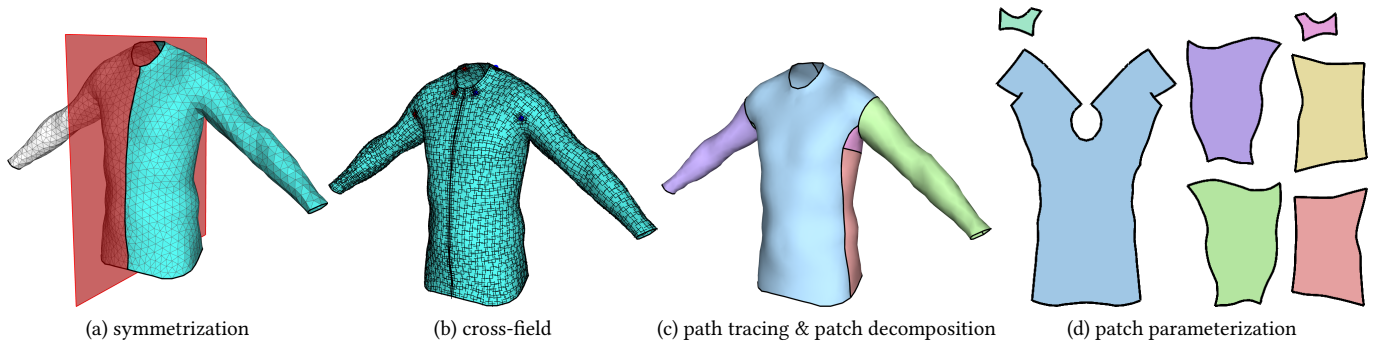| (a) symmetrization | (b) cross-field | (c) path tracing & patch decomposition | (d) patch parameterization |

Fig. 3. An overview of the different steps of our pipeline. The shape is first symmetrized (a), a smooth cross-field is computed on the mesh (b), paths are then traced on the shape by following the cross-field (c), and the resulting patches are then flattened onto the *UV* plane to create a 2D sewing pattern (d).

leverage data and neural networks to construct 3D garments from 3D scans [Bang et al. 2021; Chen et al. 2015; Pons-Moll et al. 2017] or by using parametric models and learning [Vidaurre et al. 2020; Wang et al. 2018]; a recent data set of 3D garments with corresponding 2D patterns can be used for supervision [Korosteleva and Lee 2021]. To simulate and further process the garment models, most methods require 2D patterns, which are either provided as input or computed using surface segmentation and parameterization, as discussed below. Dedicated approaches to knitwear design and fabrication are explored in [McCann et al. 2016; Narayanan et al. 2018; Narayanan* et al. 2019; Wu et al. 2019; Yuksel et al. 2012], where a sewing pattern is not needed since knitting machines can continuously knit various topologies and optimize for comfort during body motion [Liu et al. 2021]. By contrast, in this paper we focus on pattern based garment construction from woven fabric (or knitted fabric such as jersey, treated similarly), which is an overwhelmingly widespread practice in the garment industry.

Sewing pattern design and optimization. Several methods *modify given* garment patterns to fit a particular body shape [Bang et al. 2021; Bartle et al. 2016; Brouet et al. 2012; Cordier et al. 2003; Liu et al. 2018; Meng et al. 2012; Wang et al. 2005b; Wang 2018], optimize stress, pressure and seam traction [Montes et al. 2020], use parametric, predefined pattern designs [Kim and Park 2007] or generate user-defined target folds [Li et al. 2018b]. They do not modify the given topological patch layout. In contrast, our method *generates* a sewing pattern directly from the 3D garment model while respecting fabric stress bounds and fabrication constraints. The evolutionary algorithm by Kwok et al. [2015] generates a sewing pattern from 3D body geometry, but it does not account for fit or fabric stress and manufacturing requirements.

Patch layout decomposition. General surface segmentation methods for the purpose of low-distortion flattening aim at creating a small number of patches (or short cuts) to reduce Gaussian curvature. A number of works compute the patches and cuts jointly with the parameterization in order to optimize or bound both [lga Sorkine et al. 2002; Li et al. 2018a; Poranne et al. 2017]. Variational surface cutting [Sharp and Crane 2018] optimizes the tradeoff between a given distortion measure and cut length. Another class of works computes approximations of a given shape with (nearly)

developable surfaces [Binninger* et al. 2021; Ion et al. 2020; Julius et al. 2005; Stein et al. 2018]. However, the seams generated by all such methods are not appropriate for garment fabrication because they are far too complex and difficult to sew and do not necessarily fulfill even the basic requirement of equal length for matching seams, nor symmetry. Huang et al. [2012] use predefined cuts based on an anthropomorphic subdivision, which do not generalize to other garments or body shapes.

Methods for patch layout generation for the purpose of globally smooth parameterization and semiregular remeshing focus on the quality of the layout and its combinatorial structure [Campen 2017]. State-of-the-art techniques compute a smooth tangent vector field on the surface [Vaxman et al. 2017] aligned to features and principal directions as the basis for tracing the patch decomposition [Livesu et al. 2020; Nuvoli et al. 2019; Pietroni et al. 2021, 2016; Razafind-razaka et al. 2015]. The topology of these layouts is restricted by the remeshing application, e.g. four corners per patch for quadrangulation, which is too stringent for pattern making. We adapt the method by Pietroni et al. [2021] to suit the different setting of garment design and fabrication, where the topology requirements are different and textile based distortion measures must be bounded.

Surface parameterization and textile distortion measures. General flattening methods take a surface of disk topology and find its mapping to 2D that minimizes a distortion measure [Hormann et al. 2007]. The used objective most typically measures conformal distortion (e.g. [Lévy et al. 2002; Sheffer et al. 2004]) or deviation from isometry (e.g. [Liu et al. 2008; Rabinovich et al. 2017]). Guaranteeing locally and globally injective parameterization is possible [Jiang et al. 2017; Sawhney and Crane 2017] but can be computationally costly. Most importantly, the general distortion measures are isotropic and do not account for the particular behavior of woven fabric, which is almost inextensible along the yarn directions and more stretchable in the diagonal direction. In other words, developable surfaces are global minimizers of all these distortion measures, but non-vanishing locally minimal energy states do not necessarily model plausible fabric configurations.

McCartney et al. [2005; 2000] and Wang et al. [2005a] look specifically at models for woven fabric and flatten the mesh by discriminatively optimizing the yarn stretch and shear. Wang et al. [2005a] create a discrete orthogonal grid to represent the textile and wrap

it around a target 3D mesh, thereby establishing the 2D-to-3D mapping. McCartney et al. opt for a continuous approach, where the intextensible yarn directions are represented by the UV-axes in the flat domain. Their energy optimization is nonlinear and its optimization could be impractical for realtime interactive design iteration framework. Our textile deformation model is inspired by these works, but we use a different optimization strategy to achieve fast performance, and we incorporate seam symmetry [Wolff et al. 2019] and grain alignment constraints for fabrication feasibility.

3 OVERVIEW

Our pipeline takes as input a triangle mesh representing the target garment in 3D and automatically produces a 2D sewing pattern. The designer can influence the final layout by sketching some curves on the surface, indicating desired seams. In addition to the rest shape in 3D, our approach can also consider additional target poses that the garment should assume when worn.

Our method is intended for woven fabrics that consist of warp and weft yarns. The warp direction is the main (longitudinal) direction of the fabric, also termed *grain*, and the weft direction is orthogonal to it. Depending on the material, the textile threads can be elastic or nearly inextensible, determining the stretching resistance of the fabric along the warp and weft directions. The fabric can usually stretch more along the diagonal direction, resulting in shear of the woven structure. The main parameters of our framework are therefore the maximal admissible stretch and shear, determining the maximum deformation that the material undergoes from the flat configuration to the draped 3D shape. Note that excessive shear is undesirable, even if physically possible, because it leads to wrinkles and impression of bad fit. Ideally, the deformation of the pattern pieces is close to isometry. We assume the fabric is sufficiently thin and its resistance to bending too small influence the pattern making.

3.1 Requirements

To compute a usable, manufacturable sewing pattern for a garment, our algorithm should fulfill several (interdependent) requirements.

Patch shape. The pattern should preferably consist of few large patches with straight or at least smooth boundary segments. Practical patterns pieces have few corners (between 6 and 8), and their angles should be approximately orthogonal. If a patch contains darts, their opening angles should also be sufficiently large, and the dart length usually should not exceed a fraction of the patch length, as it is then easier in practice to cut through and sew two separate pieces together.

Bounded fabric strain. According to the mentioned fabric parameters, the textile cannot exceed the prescribed deformation threshold when draped into the target 3D shape. The lower this bound, the more pattern pieces are required for doubly curved target shapes.

Seam sewing feasibility. Matching seams to be sewn together must have equal length. In practice, straight seams are easiest to sew, and it is desirable that seams can be sewn flat, meaning that the matching seams must be reflection-symmetric, so that the two pieces of fabric can be placed on top of each other to sew (Fig. 2, top).

Layout symmetry. Aesthetics is strongly associated with symmetry, and therefore symmetric sewing patterns are desirable, in particular when the targeted garment design is symmetric.

Grain alignment. A pattern piece should be oriented in 2D such that the fabric grain is roughly aligned with the vertical (gravity) direction when the garment is worn, or with some other axis, such as along the arm (see Fig. 2, bottom). Grain alignment ensures predictable, symmetric behavior when the garment is draped and subject to gravity, and pressure of the body, as well as when shrinkage due to washing occurs. Sometimes the designer may wish to prescribe a special grain direction, as in bias cut, where the grain is at 45 degrees to the vertical direction, e.g. in some skirt designs.

Efficiency. Constructing an efficient pipeline allows the designer to creatively iterate and adjust the patch layout by providing simple input by interactively sketching on the 3D surface.

3.2 Pipeline structure

Fig. 3 summarizes our processing pipeline.

Input. We assume the input garment mesh to be a manifold and with well shaped triangles (otherwise we uniformly remesh with standard tools). If additional poses of the 3D garment are available, we assume their meshes share the connectivity and are in full correspondence with the rest pose. If the input garment is symmetric, we can enforce symmetry throughout the pipeline by splitting the mesh along the symmetry plane, running the method on one side, and then reflecting the result (Fig. 3a).

Cross-field construction. We compute a smooth 4-rotational-symmetric (4-RoSy) tangent vector field on the surface (Fig. 3b) aligned with principal curvature directions and boundaries. If the input includes multiple poses, we integrate the contribution of the curvature of all the poses in the derived field (see Sec. 4.1).

Layout construction. With the goal of producing pattern pieces that fulfill the requirements above, we trace a set of *paths* across the mesh to partition the surface into the different panels (see Sec. 4.2 and Fig. 3c). The paths are oriented to follow the underlying cross-field and include the curves sketched by the designer. The patch formation is controlled by the textile distortion measure, and the designer can specify the maximum number of corners allowed in a patch, to control patch complexity. At the end of this step, the seam on the symmetry plane can be optionally removed, if possible.

Patch flattening. Finally, each patch is flattened onto the 2D space and packed with the others into a single textile sheet (see Fig. 3d). This step is achieved by a novel parameterization method that mimics textile physics using geometric measures. We impose constraints on the seams to ensure that they can be physically sewn. The parameterization step is also deployed during patch layout construction to individually uphold the distortion threshold of each patch.

4 METHOD

The following sections detail each phase of our processing pipeline.

4.1 Cross-field construction

Given a proper input, we initially construct a cross-field (a 4-RoSy tangent-vector field [Vaxman et al. 2017]) on the input surface. The cross-field is crucial in our framework as it drives the entire tracing
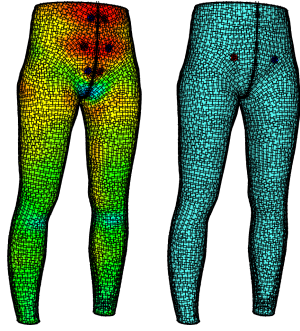
Fig. 4. Cross-field computation: The multi-scale main curvature directions used as soft constraint to retrieve a smooth field.

process; the cuts follow the directions expressed by the cross-field. Intuitively, aligning the seams to the main curvature directions is an excellent strategy to maximize the developability of the resulting patches. A significant collection of quadrangulation methods have demonstrated the strong correlation that exists between developability and curvature alignment [Bommes et al. 2013, 2009; Pietroni et al. 2021].

We initialize the field by extracting curvature directions at a low scale using the method proposed in [Panozzo et al. 2010]. Similarly to [Bommes et al. 2009] we use the anisotropy of the curvature directions to detect the regions where these curvatures are essential (see Fig. 4, left).

Then we run the globally smooth method proposed by Diamanti et al. [2014], adding soft constraints to align the field to the principal curvature directions, similarly to [Panozzo et al. 2012]. The anisotropy weighs each soft constraint value we previously computed to adhere to the principal curvature directions where those are relevant, and smooth on the rest. In addition to these soft constraints, we impose the field to align simultaneously with the user-defined constraints and the boundary edges of the 3D mesh. This way, the path will hit the boundary orthogonally, and we implicitly avoid creating artifacts or long stripes (Fig. 9). The resulting field is shown in Fig. 4, right.

When multiple frames are provided, we first compute the multi-scale curvature and the anisotropy for each frame. Then, we average the field directions for each face in the rest shape (we transport the field on the best shape and perform $\pi/2$ invariant interpolation of the cross-field, weighted by the anisotropy values). Lastly, we smooth all the fields together in the rest position and average the anisotropy among all frames.

4.2 Patch layout creation

This pipeline step obtains a patch layout as the byproduct of an iterative process that traces field-aligned paths over the target surface. The network of paths designs a patch layout composed of rectangular or non-rectangular patches. Distinct paths can only intersect orthogonally on the surface. Two intersecting paths can cross or stop when they meet, forming a T-junction. We aim at inserting the optimal number of paths in the right locations to obtain patches that satisfy the requirements we introduced in Sec. 3.
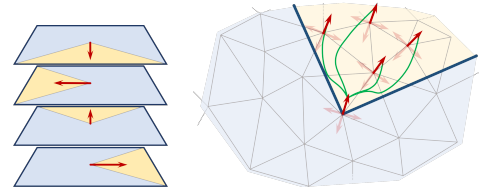

Fig. 5. The construction of the tracing graph. Each mesh vertex spawns 4 nodes in the graph, corresponding to the 4 components of the cross-field. Edges in the graph connect nodes of adjacent vertices that represent matching cross-field directions.
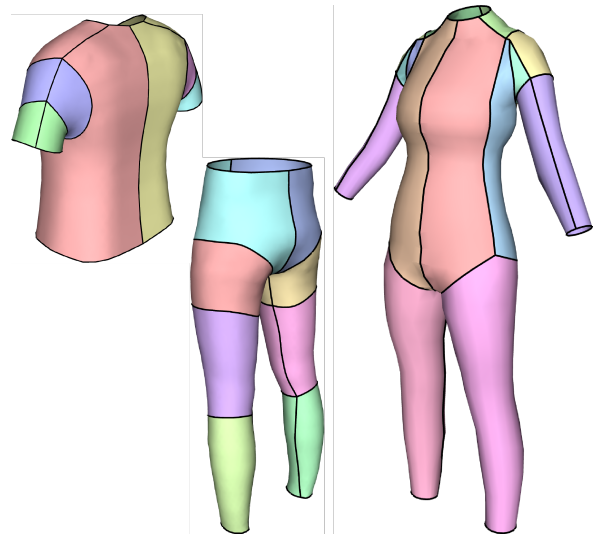

Fig. 6. Loops and border-to-border paths are both useful in most of the cases to obtain a proper patch decomposition.
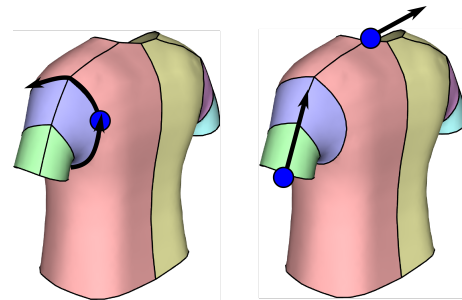

Fig. 7. Loop tracing (left) and border-to-border tracing (right).

Path tracing. To trace field-oriented paths, we use the graph-based approach proposed by Nuvoli et al. [2019] and Pietroni et al. [2021]. First, we interpolate the cross-field on vertices (considering their invariance to 90-degree rotations). We create a graph having four nodes for each mesh vertex, one for each direction of its cross-field. Then, we connect adjacent nodes with matching cross-field direction. Finally, we associate to each connection of the graph
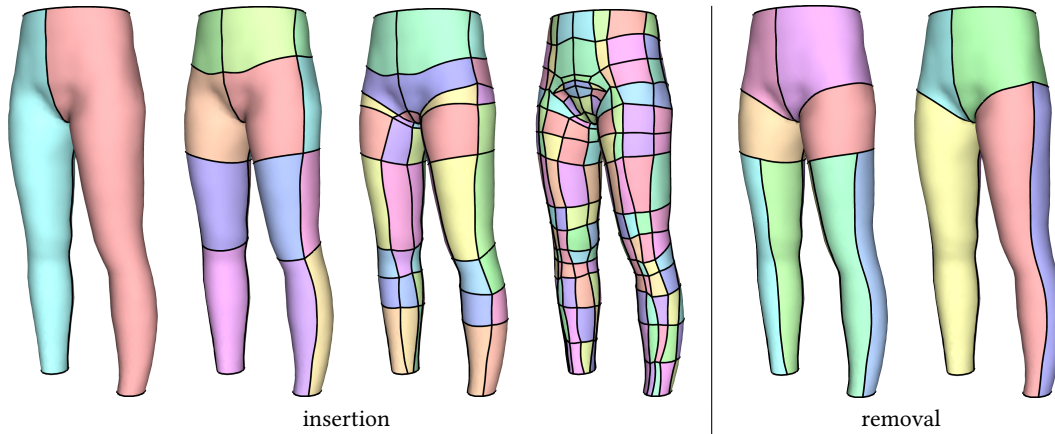
Fig. 8. Loops and border-to-border paths are inserted iteratively until the goals for each patch are satisfied. In the removal step, patches are fused by removing paths if the fused patch still satisfies our goals.
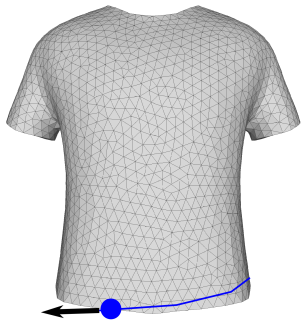

Fig. 9. The insertion of border-to-border paths that would generate long thin strips, such as the one shown in blue, is avoided by selecting vertices that *enter* and *exit* the mesh w.r.t. the cross-field.

a weight that depends on how much it drifts from the direction defined at the adjacent nodes (see Fig. 5). For details on the graph creation we refer to [Pietroni et al. 2021, 2016].

Path classification. We trace two classes of paths: paths connecting border-to-border and paths connecting to themselves, i.e., loops. We found that both categories of these paths are fundamental to assembling a proper patch layout (see Fig. 6). We formulate the problem of path tracing as a shortest-path search between a given *source node* and a set of potential *destination nodes* in the graph. In particular, when tracing a *loop* from a specific vertex, we select an internal node as the source and we aim at coming back to the same node (see Fig. 7, left).

Similarly, for border-to-border paths, we select as source a node lying on a boundary vertex whose direction *enters* the mesh, and as a destination all the border nodes whose directions *exit* the mesh (see Fig. 7, right). This trick avoids any path to form strange, long strips when touching the border (see Fig. 9). In a preprocessing step, we mark each boundary node with a label to determine if it is an exit or an entrance node. Since we align the field to boundaries, we always have these two kinds of nodes for each boundary vertex.

Paths are sequences of adjacent edges that belong to the triangle mesh. So, they usually have irregular shapes. We smooth the paths and reproject them over the surface to make them more regular and we update the rest of the mesh as a consequence.

Path insertion. We initially sample a set of candidate paths by tracing border-to-border paths and loops. For the sake of efficiency, we subsample uniformly the source nodes on the border and in the interior of the mesh. Similarly to [Livesu et al. 2020], we insert a path using a greedy strategy that favors the furthest path from the previously inserted one. As in [Pietroni et al. 2021, 2016], the distance is computed for each node using the M4 stratification of the graph [Campen et al. 2012], and averaged for each path. Notice that two paths crossing orthogonally can be very far. In contrast, parallel paths tend to be close. This simple strategy avoids conglomerations of paths.

We keep the complete patch layout updated at every insertion step, and we mark all the patches that do not satisfy the goals specified in Sec. 3.1. If a candidate path splits one patch that does not meet our goals and does not intersect tangentially with any previously inserted path, then we insert it. As in [Pietroni et al. 2016], given two candidate paths, it is trivial to check whether they intersect tangentially by simply testing whether they pass the same vertex through non-orthogonal directions.

We repeat this insertion strategy until all the patches match our goals. We perform a recursive step on each patch if needed. Notice that this step needs new patches to be parameterized at every insertion step to check whether the produced mapping fulfills the bijectivity and bounded distortion requirements, hence the need for a fast and reliable method to parameterize and test the produced patches. We illustrate the steps of this sampling procedure in Fig. 8, left.

Path removal. Once the path insertion is complete, we have a patch layout satisfying our goals. However, during its construction, there is no easy way to predict whether a candidate path is essential in the final layout in its entirety (since some goals can only be satisfied by a combination of multiple paths). We remove redundant
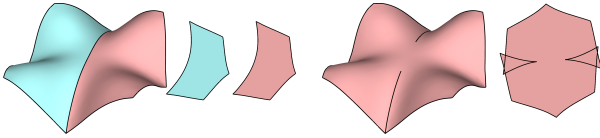
Fig. 10. Creating darts starting from areas with negative curvature tends to generate overlaps in the UV mapping.
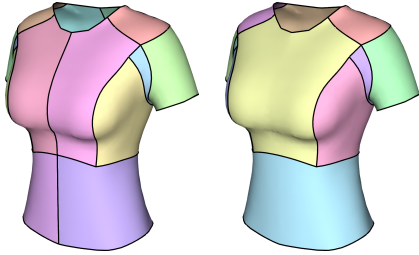

Fig. 11. The seam on the global symmetry plane can be safely removed at the end of the layout generation process.

path segments starting from the last inserted path in reverse order. Fig. 8 (right) shows the effect of the removal procedure. In this step, we disable the removal of path segments that form T-junctions to avoid creating darts, as these are treated explicitly in the next step.

Dart creation. We removed all the redundant path segments at the end of the previous step. However, there might be adjacent patches that we can *partially* glue together, maintaining the distortion under the predefined threshold. Such partial cuts can be found in fashion design and are usually referred to as *darts*. In traditional pattern making, a tailor introduces darts to better shape the body's curves. By removing a wedge-shaped piece of fabric and sewing both sides together, an experienced tailor effectively introduces angle deficiency on the garment and thus creates a curved shape from a flat pattern.

To introduce darts, we first split all paths into their segments, at the intersections with the other paths. Then we sort the different path segments by considering the Gaussian curvature of the mesh region they span (as in field computation, we extract curvature at a low scale using the method proposed by Panozzo et al. [2010]). Ideally, we want to merge starting from the regions with lower Gaussian curvature, as such seams are more likely to be merged with low distortion. Similarly, areas with negative Gaussian curvature (saddles) should be merged later, as they most likely generate overlaps in the UV domain (see Fig. 10). Then we select the first path segment and split it uniformly into a number of subpaths. Following the same intuition, we start merging from the lower curvature side, and we continue as long as the produced distortion is below the threshold.

Symmetry. The generation of the patch layout is conducted on one side of the mesh using a symmetry plane (shown in red in Fig. 3a). We first copy the patch layout on the other symmetric side, then we start the path removal and dart insertion process for the paths laying on the symmetry plane. This way, we keep the

symmetric distribution and, at the same time, remove unwanted seams along the symmetry plane (see Fig. 11).

4.3 Anisotropic textile parameterization

Woven fabric can be modeled as a regular grid of threads, where the two axes are called warp and weft (or collectively 'grain'). We assume the common case where the grid is orthogonal, and thus represent it by the UV axes in the flat domain, but it is possible to model arbitrary angles between the warp and the weft directions [McCartney et al. 2005, 2000]. Ideally, the cloth undergoes only bending when the flat pattern is draped on the 3D body, but this implies that the 3D garment shape is piecewise developable, which is not always possible. We hence need to measure deviation from developability in a way that is consistent with the woven structure properties. General parameterization distortion measures are isotropic, or invariant to rotations, but this ignores the fact that the threads are nearly inextensible, while a certain, limited angular distortion is permitted, which means that the cloth can stretch diagonally to the warp and weft (UV) directions, but much less so along those directions. Hence the intrinsic parameterization distortion measure needs to be anisotropic. We penalize stretch along the U and V directions separately from shear. Note that by shear we mean angular distortion of the fabric grid, while the thread lengths are preserved, so it is not an area-preserving shear transformation (see Fig. 12).

Consider a reference triangle $A'B'C'$ in XYZ coordinates mapped to a distorted triangle ABC in uv coordinates, see Fig. 13. A unique affine transformation f maps $A'B'C'$ to ABC. To penalize stretch of the fabric grain, we can consider the scaling induced by the inverse transformation $f^{-1} : \mathbb{R}^2 \to \mathbb{R}^3$, which can be interpreted as the stretch of a canonical unit frame $\mathcal{G}$ in the UV plane, undergoing the transformation f^{-1}, as illustrated in Fig. 13. The stretch values s_u and s_v can be expressed using the columns of the Jacobian of f^{-1} (i.e., the tangents of the parameterization), denoted $J = [J_1 \ J_2] \in \mathbb{R}^{3 \times 2}$:

$$s_u = \|J_1\| , \ s_v = \|J_2\|. \tag{1}$$

Stretch. We could define the stretch penalty as the deviations of s_u, s_v from 1. However, these scaling factors are non-linear functions of the variables in our problem, namely the UV coordinates of the triangle vertices ABC, leading to a non-quadratic stretch measure. We therefore invert the problem and consider the stretching of the *image* of frame $\mathcal{G}$, namely $J(\mathcal{G})$, when f is applied.

For simplicity, we express the frame $\mathcal{G}$ explicitly by three points: the 2D triangle centroid G, and the points $G_u = G + (1, 0)$, $G_v = G + (0, 1)$ (see Fig. 13). The 3D grain directions $J(\mathcal{G})$ can be expressed
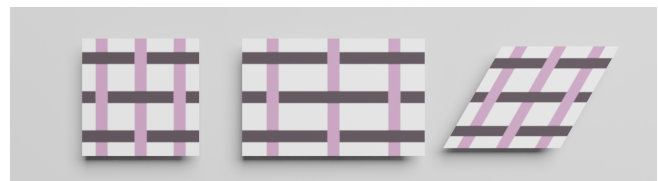

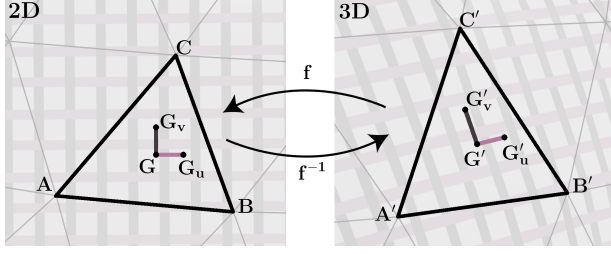Fig. 12. Woven net (left) undergoing stretch (middle) and shear (right).

Fig. 13. Per-triangle flattening, modeled as an affine map f.

by transforming these three points by f into G', G'_u, G'_v. We denote by $(\alpha_X, \beta_X, \gamma_X)$ the barycentric coordinates of any point X w.r.t. triangle ABC. Then we have:

$$G' = \alpha_G A' + \beta_G B' + \gamma_G C', \qquad (2)$$

$$G'_u = \alpha_{G_u} A' + \beta_{G_u} B' + \gamma_{G_u} C', \qquad (3)$$

$$G'_v = \alpha_{G_v} A' + \beta_{G_v} B' + \gamma_{G_v} C'. \qquad (4)$$

This allows us to measure the lengths $s_u = \|G'_u - G'\|$ and $s_v = \|G'_v - G'\|$ on the 3D triangle. If the mapping f does not stretch the J_1 direction, then $\|G_u - G\| = s_u$, and equivalently for J_2 and v. We denote the UV coordinates of our helper points as $G = (u_G, v_G), G_u = (u_{G_u}, v_{G_u}), G_v = (u_{G_v}, v_{G_v})$, and then the stretch of J_1 caused by f is $\|G_u - G\|$, which is equal to $|u_{G_u} - u_G| = (\alpha_{G_u} - \alpha_G)u_A + (\beta_{G_u} - \beta_G)u_B + (\gamma_{G_u} - \gamma_G)u_C$, and similarly for the other direction. We thus define the per-triangle stretch energy terms as follows:

$$E_{\text{stretch},u}(ABC) = \qquad (5)$$
$$\omega_{\text{stretch}} \left[s_u - \left((\alpha_{G_u} - \alpha_G)u_A + (\beta_{G_u} - \beta_G)u_B + (\gamma_{G_u} - \gamma_G)u_C \right) \right]^2,$$
$$E_{\text{stretch},v}(ABC) =$$
$$\omega_{\text{stretch}} \left[s_v - \left((\alpha_{G_v} - \alpha_G)v_A + (\beta_{G_v} - \beta_G)v_B + (\gamma_{G_v} - \gamma_G)v_C \right) \right]^2.$$

The total energy $E_{\text{stretch},u}$ is then defined as the sum of per-triangle energies, and similarly for $E_{\text{stretch},v}$.

Shear and rigidity. A direct expression for shear is the angle between the parameterization tangents J_1 and J_2, with the constraint that the tangents maintain their length, but this is again not a quadratic in the variables $u_A, v_A, \ldots$ Instead, we propose to measure deviation from isometry, implicitly penalizing shear in the same manner as in as-rigid-as-possible (ARAP) parameterization [Liu et al. 2008]:

$$E_{\text{rigid}}(ABC) = \omega_{\text{rigid}} \sum_{e \in ABC} (u_e - u_{e'})^2 + (v_e - v_{e'})^2, \qquad (6)$$

where e denotes a triangle edge in triangle ABC, e' is the corresponding edge in triangle $M(A'B'C')$ and M is the best-fit rigid transformation that aligns $A'B'C'$ to ABC, computed using Procrustes [Sorkine-Hornung and Rabinovich 2016]. The total rigidity measure E_{rigid} sums up Eq. (6) over all triangles. Although this ARAP measure mixes shear with isotropic stretch, it tends to produce well conditioned results thanks to the Procrustean step and works well for our purpose.

Seam reflection symmetry. Pattern pieces are sewn together along seams by placing one part onto the corresponding part and stitching along a curve. This implies the existence of a perfect reflection between the matching borders that get sewn together. We represented matching seams in the UV domain as two sets of duplicated vertices $\mathcal{P} = (p_1, \ldots, p_n)$ and $\mathcal{Q} = (q_1, \ldots, q_n)$, where p_i and q_i correspond to the same 3D location but are mapped to different locations in 2D.

To measure the deviation of the matching seams from a perfect reflection, we first compute the best-fit reflection transformation M between $\mathcal{P}$ and $\mathcal{Q}$, similarly to Wolff et al. [2019]. We achieve this by simply switching the sign of the determinant during procrustean analysis. If the UV seams are reflection-symmetric, $M\mathcal{P}$ and $\mathcal{Q}$ coincide, as well as $\mathcal{P}$ and $M^{-1}\mathcal{Q}$. Otherwise, we define target points as:

$$\forall i \in [1 \ldots n], \quad q_{t,i} = \frac{Mp_i + q_i}{2}, \; p_{t,i} = \frac{p_i + M^{-1}q_i}{2}. \qquad (7)$$

The seam's reflection symmetry energy E_{seam} is then defined as:

$$E_{\text{seam}} = \omega_{\text{seam}} \left(\sum_{p_i \in \mathcal{P}} (p_i - p_{t,i})^2 + \sum_{q_i \in \mathcal{Q}} (q_i - q_{t,i})^2 \right). \qquad (8)$$

In our general framework, patches are cut and flattened progressively. Consequently, upon flattening a given patch, the energy E_{seam} may not be defined for all seams if the mirroring patch was not flattened previously. To address this, we jointly optimize E_{seam} for all patches during the final parameterization.

Dart symmetry. Darts are similar to seams, but the two matching parts belong to the same pattern piece and meet at the tip, so the tip vertex is not duplicated. Ideally, this vertex lies on the dart's symmetry axis in 2D. We use a similar energy to Eq. (8) for darts but modify the target point definition. We define the set of midpoints $\mathcal{R}$, consisting of $r_i = (p_i + q_i)/2$, and find the target symmetry axis by computing the best fitting line to $\mathcal{R}$ while being constrained to pass through the tip vertex. Then, we compute symmetric point sets $\overline{\mathcal{P}}$ and $\overline{\mathcal{Q}}$ and define an energy term E_{dart} equivalently to E_{seam}, weighted by ω_{dart} and with target positions

$$\forall i \in [1 \ldots n], \quad q_{t,i} = \frac{\overline{p_i} + q_i}{2}, \; p_{t,i} = \frac{p_i + \overline{q_i}}{2}. \qquad (9)$$

Grain alignment. Traditional garment patterns align the grain (warp) direction with the vertical axis in the final garment, or the "centerline" direction of a body part, such as along the sleeves. This ensures predictable and symmetric draping. In some cases, the tailor may perform a *bias cut* instead, favoring another alignment axis in specific regions in order to allow shearing on the main stress direction. In our framework, we allow the definition of a desired 3D alignment axis a' and rotate the 2D patches such that their warp aligns with a'. This is first performed on a per-triangle basis by computing the projection of a' on the triangle plane in 3D and transporting it to a on the 2D triangle. This defines a per-triangle desired axis, and we compute the best-fit global axis a_{opt} that the V direction should align to. Instead of including this in the energy formulation, we simply perform this alignment step in each iteration. Fig. 16 shows an example of grain alignment.
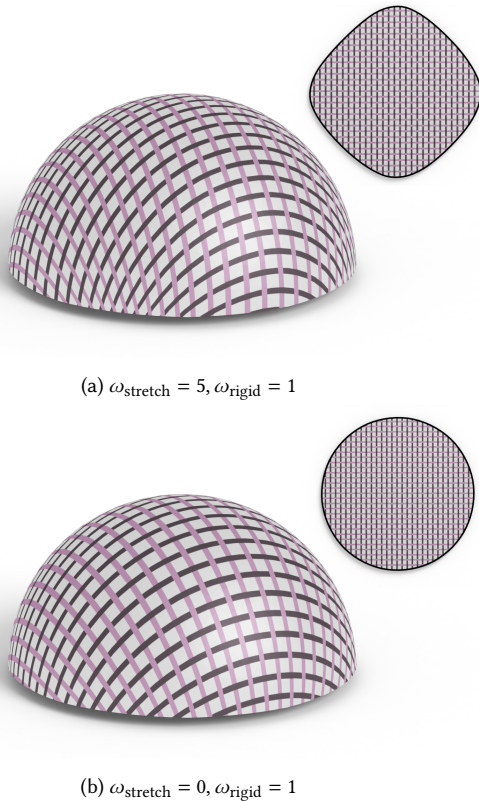
(a) $\omega_{\text{stretch}} = 5, \omega_{\text{rigid}} = 1$

(b) $\omega_{\text{stretch}} = 0, \omega_{\text{rigid}} = 1$

Fig. 14. Influence of the energy terms weights on the parameterization.

$\omega_{\text{dart}} = 0$, $\omega_{\text{seam}} = 0$ $\omega_{\text{dart}} = 5$, $\omega_{\text{seam}} = 5$
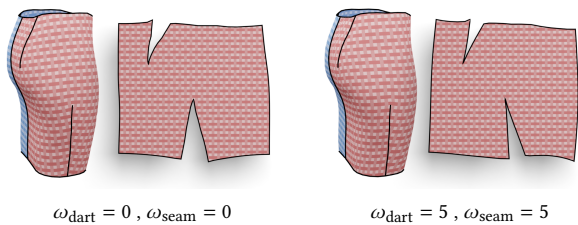
Fig. 15. Dart and seam reflection symmetry (Sec. 4.3) is essential for producing workable patterns with reflective cuts.

Optimization. We employ a local-global iteration strategy similar to [Liu et al. 2008; Sorkine and Alexa 2007] to compute a *UV* mapping of a given pattern piece by minimizing the distortion measure

$$E_{\text{textile}} = E_{\text{stretch,u}} + E_{\text{stretch,v}} + E_{\text{rigid}} + E_{\text{seam}} + E_{\text{dart}} \quad (10)$$

and accounting for grain alignment. As the initial guess we use the least squares conformal map of [Lévy et al. 2002], orienting the 2D patch such that the *V* axis matches the prescribed grain direction. We then compute the expressions in all the energy terms as described above (the local step), and solve for the updated parameterization of the pattern piece by minimizing E_{textile} w.r.t. the *UV* coordinates of the mesh vertices. This global step amounts to efficiently solving a sparse linear system, because E_{textile} is quadratic in the *UV*'s. We
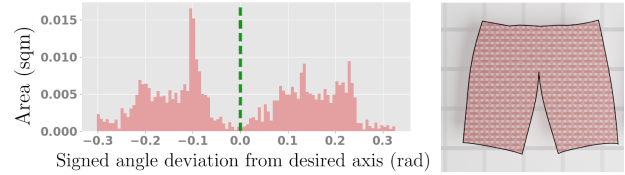

Fig. 16. A skirt pattern piece (Fig. 2, top) is split by a dart, resulting in conflicting desired grain alignment on each side. The global rotation chosen by our parameterization accommodates for both sides, despite being optimal for only a small area. On the left we show a histogram of the signed angle between the warp direction and the prescribed desired alignment.

iterate the local and global steps, reinstating the grain alignment in each local step. For all tested meshes, 5 iterations were sufficient to compute a satisfactory estimate and 20 iterations to converge.

The fabric properties are expressed by the weights ω; the default values in our examples are $(\omega_{\text{stretch}}, \omega_{\text{rigid}}, \omega_{\text{dart}}, \omega_{\text{seam}}) = (5, 1, 5, 5)$. Fig. 14 shows the effects of varying the distortion weights, while Fig. 15 illustrates the seam and dart reflection symmetry constraints.

Our parameterization naturally generalizes to the case where multiple poses are considered, as only the right side of the system in the global step depends on the 3D mesh. At each iteration, we thus compute the right-hand side for each pose considered, and use the average in the solve.

Note that our parameterization method does not guarantee bijectivity. Self-intersections can thus occur during patch layout computation when cuts are introduced in non-satisfactory locations. Our patch decomposition method takes advantage of this by discarding cuts that lead to non-injective patches. This significantly speeds up our exploration of possible cuts, as only promising solutions are considered. The final pattern produced is guaranteed to contain no overlaps, as all the cuts passed our self-intersection test.

5 RESULTS

We integrate our framework into an interactive editor that allows the designer to control the result and explore the different sewing patterns generated by our system. The main two main parameters exposed to the user are the maximum number of corners per pattern piece, C, and the maximum allowed stretch s_{max}. Fig. 19 shows the effect of these parameters on a simple example. The higher the number of permitted corners, the fewer patches appear in the patch decomposition. However, the patches are then more complex than the ones generated with fewer corners and potentially require more handling when sewing. Similarly, the maximum allowed distortion is also related to the number of patches inserted. Intuitively, more patches are needed to comply with a stricter distortion threshold, as seen in Fig. 19.

Fig. 21 demonstrates the advantage of our textile parameterization approach compared to standard, isotropic flattening using the as-rigid-as-possible (ARAP) method [Liu et al. 2008]. When plugged into our sewing pattern creation pipeline, isotropic flattening leads to very different patterns and a greater number of seams, compared with our parameterization method.

In Fig. 24, we show how our system is able to match the maximal prescribed stretch, and we also measure the parameterization
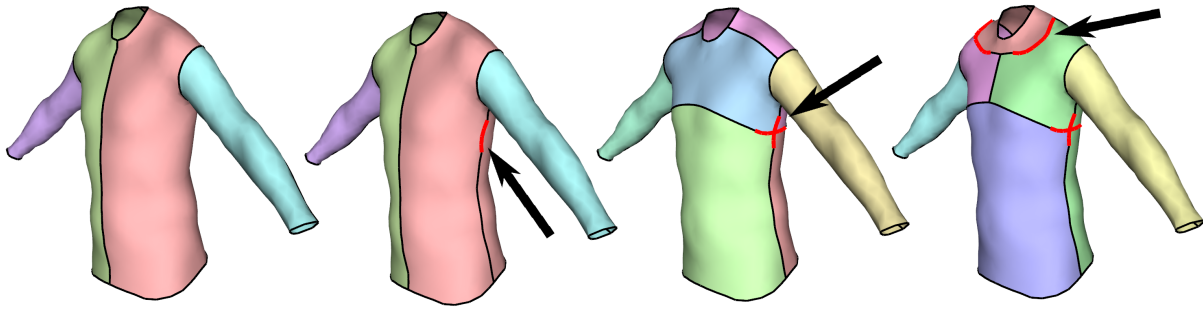
Fig. 17. A sequence of interactive adjustments of to the patch layout. The designer adds sketches (in red), and our method computes seams that interpolate the sketched curves while generating a valid sewing pattern.
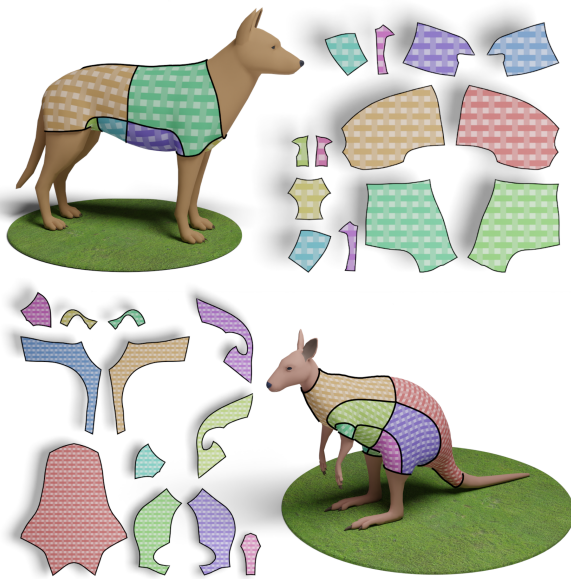

Fig. 18. Our method works also for non-humanoid shapes, such as a dog (top) or a kangaroo (bottom).

distortion for each pattern piece using the ARAP energy E_{rigid} [Liu et al. 2008], showing that we achieve good results also in terms of isometry.

The designer can also adjust the patch layout using sketches on the 3D garment surface, see Fig. 17 and the accompanying video. The initial cross-field is updated after each sketch to allow the patch decomposition to conform to the newly inserted constraints. We handle sketches in the same manner as Pietroni et al. [2021] handle sharp features. Our method allows real-time editing for small models (up to 3000 triangles) and takes a few seconds to process larger models.

Our method works on arbitrary input garments, including loose (Figures 1, 23), tight-fitting (Fig. 20) or with additional geometric features, like collars (Fig. 22). Our framework can be used to fabricate personalized garments based on 3D scans (see Figures 26, 27) or even

garments for animals (see Fig. 18). When multiple target poses of the 3D garment are available, our method adapts the sewing pattern to accommodate the motion, see Fig. 25.

We validate one of our patterns with the physically based simulation provided by CLO [CLO 2022] (see Fig. 23). Finally, we physically fabricate two examples: a wetsuit and a pair of leggings (see Figures 26 and 27). Both manufactured garments exhibit an excellent fit, demonstrating the feasibility of our approach. The wetsuit is made of neoprene and assembled with an overlocker machine, whereas the leggings are made from 90% polyester and 10% spandex. The wetsuit pattern has an intricate structure and took around 5 hours for a single person to cut and sew, whereas the leggings only took about an hour. To fabricate the garments, we compute seam allowance, including the additional margin necessary for sewing, and we add markers to help the seam matching (Fig. 28).

6 CONCLUSIONS

We present a pipeline that generates computational sewing patterns directly from 3D digital garments. The key novelty is our algorithm for shape segmentation and patch flattening that explicitly focuses on garment fabrication. By the usage of darts and taking into account the anisotropic properties of woven fabric, grain alignment and seam symmetry, our approach gives the designer the tools to produce (automatically and interactively) the 2D patterns to make the desired garment. We show that our algorithm produces patterns that are indeed suitable for fabrication of a custom, digitally designed garment. Our results also show that the pipeline can be applied to generate dewing patterns for tight-fitting and loose garments, for all types of body shapes, including animals.

Limitations and future work. Although our approach is quite robust, it has some limitations. If the input mesh is the result of a simulation or a scan of a physical garment, the wrinkles can introduce noise in the guiding field. Currently, we do not manage non-manifold data explicitly. Instead, we separate the input into manifold parts before processing. Usually, in fashion design, non-manifold edges occur where the separation between the different elements is a given (such as the main garment and a sewn-on pocket). In our future work, we could expand our method to non-manifold inputs and determine the separation automatically, enabling us to handle ruffles, pleats and gathers. Moreover, since the field tracing

$C = 8 , s_{max} = 0.05$ $C = 8 , s_{max} = 0.04$ $C = 8 , s_{max} = 0.02$

$C = 6 , s_{max} = 0.05$ $C = 6 , s_{max} = 0.04$ $C = 6 , s_{max} = 0.02$
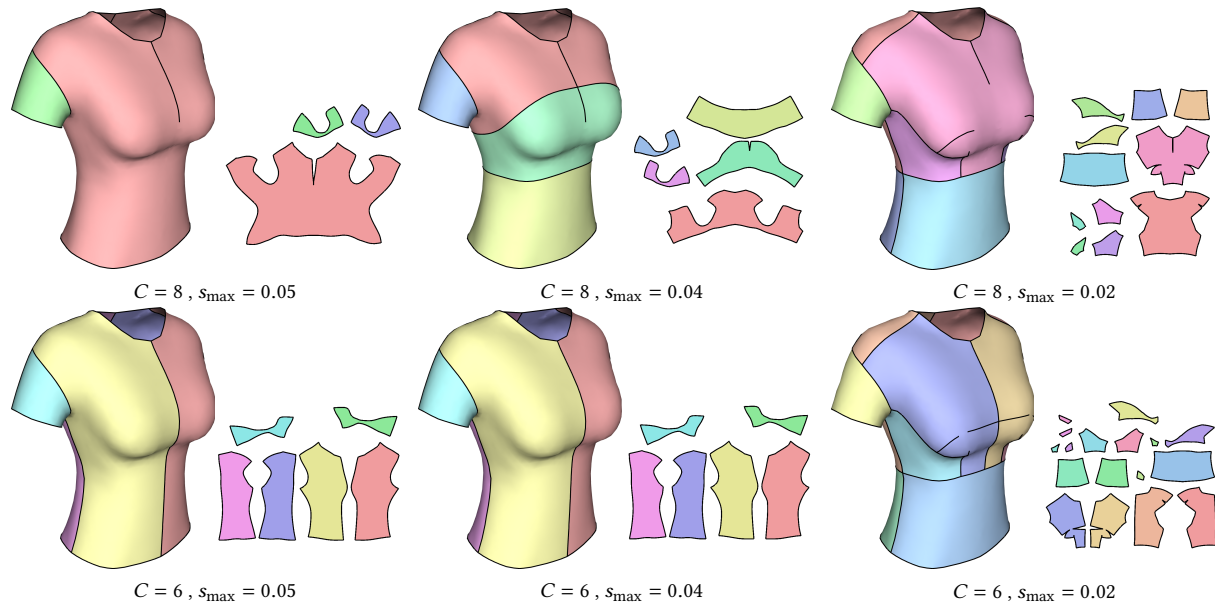
Fig. 19. Automatic pattern layout generation for a shirt model, varying the two parameters that control the decomposition: the maximum number of corners per patch C and the maximum stretch s_{max}.
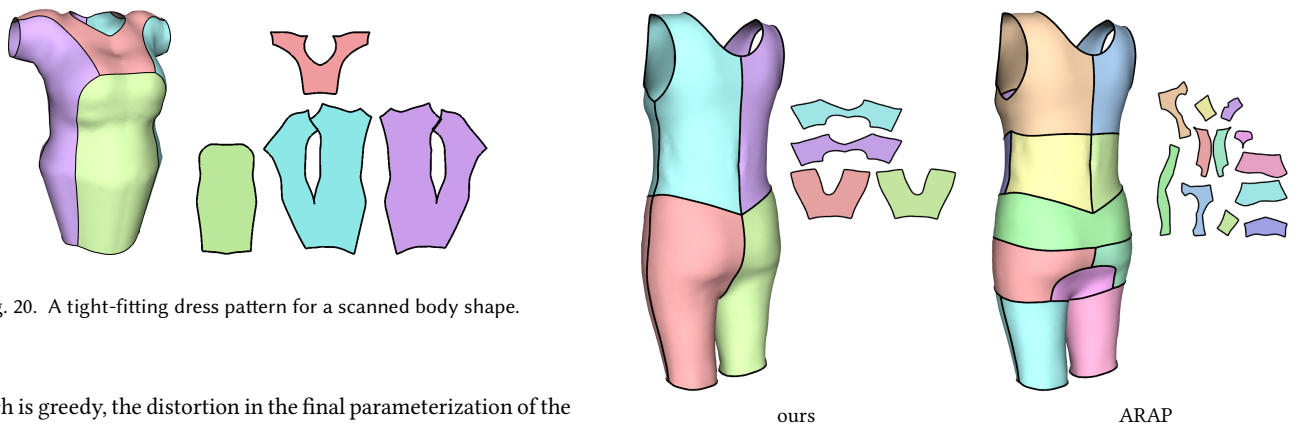

Fig. 20. A tight-fitting dress pattern for a scanned body shape.

ours   ARAP

Fig. 21. A comparison between the sewing patterns obtained using our pipeline with our textile parameterization method vs. ARAP parameterization [Liu et al. 2008].

approach is greedy, the distortion in the final parameterization of the pattern pieces may differ from the one computed during the patch decomposition stage. This is because the seam reflection constraints cannot be incorporated before all patches are computed. In practice, we never encountered this problem. Also owing to the greedy nature of our method, a slight difference in the input garment or user constraints can result in a significantly different pattern. In future work, it would be interesting to find a better optimization approach and also to include seam allowance constraints in our framework, to further secure easy and robust fabrication of the resulting sewing patterns.

ACKNOWLEDGMENTS

REFERENCES

Thiemo Alldieck, Hongyi Xu, and Cristian Sminchisescu. 2021. imGHUM: Implicit Generative Models of 3D Human Shape and Articulated Pose. arXiv:2108.10842 https://arxiv.org/abs/2108.10842

Seungbae Bang, Maria Korosteleva, and Sung-Hee Lee. 2021. Estimating Garment Patterns from Static Scan Data. *Computer Graphics Forum* 40 (05 2021). https://doi.org/10.1111/cgf.14272

Aric Bartle, Alla Sheffer, Vladimir G. Kim, Danny M. Kaufman, Nicholas Vining, and Floraine Berthouzoz. 2016. Physics-Driven Pattern Adjustment for Direct 3D Garment Editing. *ACM Trans. Graph.* 35, 4, Article 50 (jul 2016), 11 pages. https://doi.org/10.1145/2897824.2925896
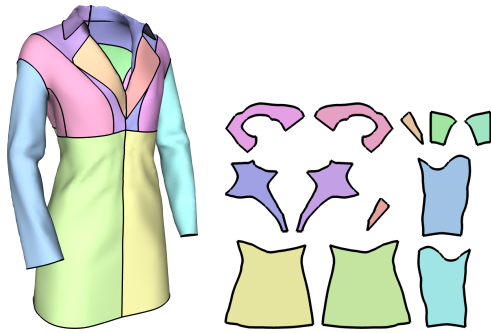
Fig. 22. Our method works on garment models with specific geometric features, such as collars.
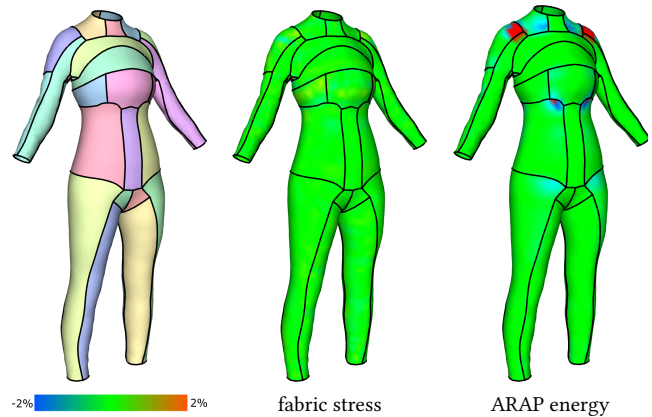

fabric stress   ARAP energy

Fig. 24. We measure the fabric stress and the ARAP energy on one of our patterns, showing that we achieve good results in terms of both measures.

Fig. 23. Our method computes sewing patterns (right) for the input meshes of a shirt and pants (left). To validate our result, we virtually sew our resulting patterns and then drape them using the physically based simulation of CLO [CLO 2022] (center).

27, 01 (jan 2007), 72–81. https://doi.org/10.1109/MCG.2007.1

Jocelyn Hua-Chu Chen. 1998. *An investigation into 3-Dimensional Garment Pattern Design.* Ph.D. Dissertation. Nottingham Trent University, UK.

Xiaowu Chen, Bin Zhou, Feixiang Lu, Lin Wang, Lang Bi, and Ping Tan. 2015. Garment Modeling with a Depth Camera. *ACM Trans. Graph.* 34, 6, Article 203 (oct 2015), 12 pages. https://doi.org/10.1145/2816795.2818059

CLO. 2022. clo3d.com. https://www.clo3d.com.

Frederic Cordier, Hyewon Seo, and Nadia Magnenat-Thalmann. 2003. Made-to-measure technologies for an online clothing store. *IEEE Computer graphics and Applications* 23, 1 (2003), 38–48.

Philippe Decaudin, Dan Julius, Jamie Wither, Laurence Boissieux, Alla Sheffer, and Marie-Paule Cani. 2006. Virtual garments: A fully geometric approach for clothing design. *Comput. Graph. Forum* 25, 3 (2006), 625–634.

Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Designing *N*-PolyVector Fields with Complex Polynomials. *Comput. Graph. Forum* 33, 5 (2014), 1–11.

Kai Hormann, Bruno Lévy, and Alla Sheffer. 2007. Mesh Parameterization: Theory and Practice. https://hal.inria.fr/inria-00186795 This document is the support of a course given at SIGGRAPH 2007.

H.Q. Huang, P.Y. Mok, Y.L. Kwok, and J.S. Au. 2012. Block pattern generation: From parameterizing human bodies to fit feature-aligned and flattenable 3D garments. *Computers in Industry* 63, 7 (2012), 680–691. https://doi.org/10.1016/j.compind.2012.04.001

Alexandra Ion, Michael Rabinovich, Philipp Herholz, and Olga Sorkine-Hornung. 2020. Shape Approximation by Developable Wrapping. *ACM Transactions on Graphics (proceedings of SIGGRAPH ASIA)* 39, 6 (2020). https://doi.org/10.1145/3414685.3417835

Zhongshi Jiang, Scott Schaefer, and Daniele Panozzo. 2017. Simplicial Complex Augmentation Framework for Bijective Maps. *ACM Trans. Graph.* 36, 6, Article 186 (nov 2017), 9 pages. https://doi.org/10.1145/3130800.3130895

Dan Julius, Vladislav Kraevoy, and Alla Sheffer. 2005. D-charts: Quasi-developable mesh segmentation. *Computer Graphics Forum* 24, 3 (2005), 581–590.

Sungmin Kim and Chang Park. 2007. Basic garment pattern generation using geometric modeling method. *International Journal of Clothing Science and Technology* 19 (01 2007), 7–17. https://doi.org/10.1108/09556220710717017

Maria Korosteleva and Sung-Hee Lee. 2021. Generating Datasets of 3D Garments with Sewing Patterns. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung (Eds.), Vol. 1. https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/013d407166ec4fa56eb1e1f8cbe183b9-Paper-round1.pdf

Tsz Ho Kwok, Yan-Qiu Zhang, Charlie Wang, Yong-Jin Liu, and Kai Tang. 2015. Styling Evolution for Tight-Fitting Garments. *IEEE Transactions on Visualization and Computer Graphics* 22 (01 2015). https://doi.org/10.1109/TVCG.2015.2446472

Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. 2002. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Trans. Graph.* 21, 3 (jul 2002), 362–371. https://doi.org/10.1145/566654.566590

Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérôme Maillot. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.* 21, 3 (2002), 362–371.

lga Sorkine, Daniel Cohen-Or, Rony Goldenthal, and Dani Lischinski. 2002. Bounded-distortion Piecewise Mesh Parameterization. In *Proceedings of IEEE Visualization*

Alexandre Binninger*, Floor Verhoeven*, Philipp Herholz, and Olga Sorkine-Hornung. 2021. Developable Approximation via Gauss Image Thinning. *Computer Graphics Forum (proceedings of SGP 2021)* 40, 5 (2021), 289–300. https://doi.org/10.1111/cgf.14374

Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Piscataway, NJ, USA, 3794–3801.

David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Cláudio T. Silva, Marco Tarini, and Denis Zorin. 2013. Quad-Mesh Generation and Processing: A Survey. *Comput. Graph. Forum* 32, 6 (2013), 51–76.

David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 3 (2009), 77.

Remi Brouet, Alla Sheffer, Laurence Boissieux, and Marie-Paule Cani. 2012. Design Preserving Garment Transfer. *ACM Trans. Graph.* 31, 4, Article 36 (jul 2012), 11 pages. https://doi.org/10.1145/2185520.2185532

Marcel Campen. 2017. Partitioning Surfaces Into Quadrilateral Patches: A Survey. *Comput. Graph. Forum* 36, 8 (2017), 567–588.

Marcel Campen, David Bommes, and Leif Kobbelt. 2012. Dual loops meshing: quality quad layouts on manifolds. *ACM Trans. Graph.* 31, 4 (2012), 110:1–110:11.

M. Cani, J. F. Hughes, E. Turquin, L. Boissieux, and J. Wither. 2007. A Sketch-Based Interface for Clothing Virtual Characters. *IEEE Computer Graphics and Applications*
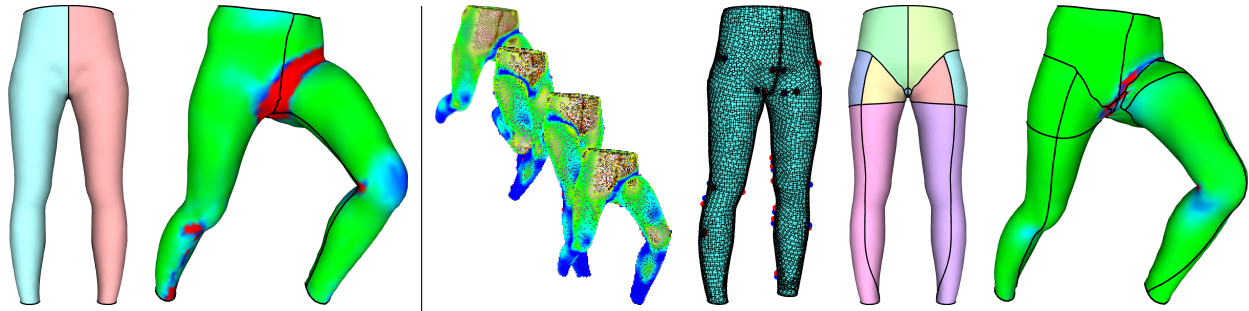
Fig. 25. For animated models, we use the multi-scale principal curvature directions integrated though the frames to derive the cross-field. Similarly, we add per-frame energy terms in the patch parameterization. Having multiple target poses of the garment helps the method to place seams that adapt to highly deforming areas, such as the area where the leg connects to the torso (see the result on the right), which would otherwise not be noticed (see the result computed based on the rest shape alone on the left).

(a) fabrication from our generated pattern

(b) final garment

Fig. 26. A fabricated wetsuit. The input 3D garment model is based on a 3D scan of the subject. The open collar will be replaced with a zipper in the final garment. It took around 5 hours for a single person to prepare the pieces, cut, and sew.
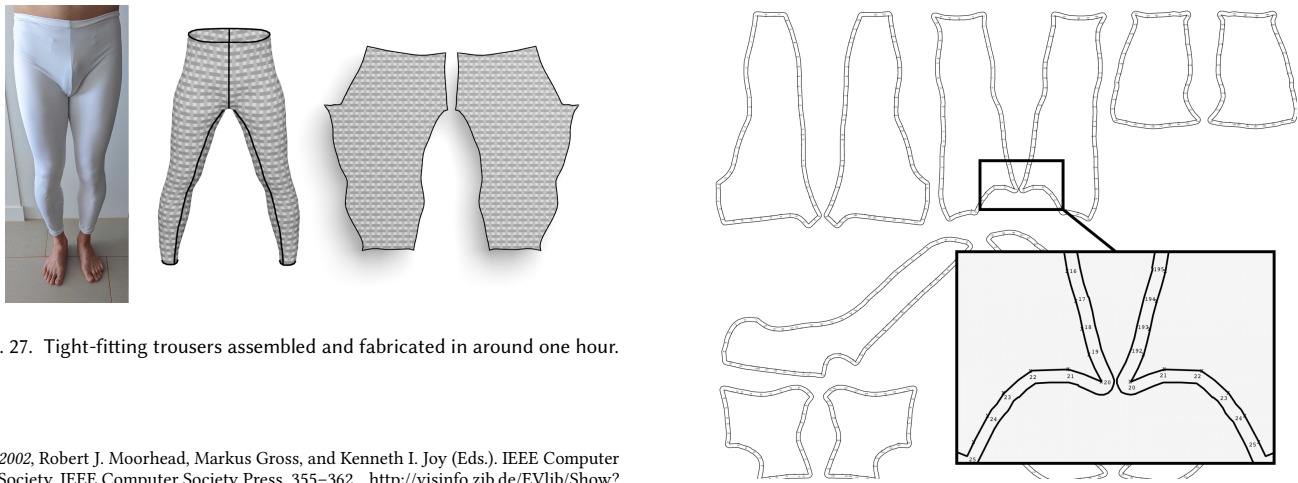

Fig. 27. Tight-fitting trousers assembled and fabricated in around one hour.

2002, Robert J. Moorhead, Markus Gross, and Kenneth I. Joy (Eds.). IEEE Computer Society, IEEE Computer Society Press, 355–362. http://visinfo.zib.de/EVlib/Show?EVL-2002-356

Minchen Li, Danny M. Kaufman, Vladimir G. Kim, Justin Solomon 0001, and Alla Sheffer. 2018a. OptCuts: joint optimization of surface cuts and parameterization. *ACM Trans. Graph* 37, 6 (2018), 247:1–247:13.

Minchen Li, Alla Sheffer, Eitan Grinspun, and Nicholas Vining. 2018b. Foldsketch: enriching garments with physically reproducible folds. *ACM Trans. Graph* 37, 4 (2018), 133:1–133:13.

Kaixuan Liu, Xianyi Zeng, Pascal Bruniaux, Xuyuan Tao, Xiaofeng Yao, Victoria Li, and Jianping Wang. 2018. 3D interactive garment pattern-making technology. *Computer-Aided Design* 104 (2018), 113–124. https://doi.org/10.1016/j.cad.2018.07.003

Fig. 28. The fabricated patterns (shown here: the wetsuit from Fig. 26) include additional margins and markers to allow for sewing.

Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J. Gortler. 2008. A Local/Global Approach to Mesh Parameterization. In *Proceedings of the Symposium on Geometry Processing* (Copenhagen, Denmark) *(SGP '08)*. Eurographics Association, Goslar, DEU, 1495–1504.

Zishun Liu, Xingjian Han, Yuchen Zhang, Xiangjia Chen, Yu-Kun Lai, Eugeni L. Doubrovski, Emily Whiting, and Charlie C. L. Wang. 2021. Knitting 4D Garments with Elasticity Controlled for Body Motion. *ACM Trans. Graph.* 40, 4, Article 62 (jul 2021), 16 pages. https://doi.org/10.1145/3450626.3459868

Marco Livesu, Nico Pietroni, Enrico Puppo, Alla Sheffer, and Paolo Cignoni. 2020. LoopyCuts: practical feature-preserving block decomposition for strongly hex-dominant meshing. *ACM Trans. Graph.* 39, 4 (2020), 121.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graph.* 34, 6 (Oct. 2015), 248:1–248:16.

James McCann, Lea Albaugh, Vidya Narayanan, April Grow, Wojciech Matusik, Jennifer Mankoff, and Jessica Hodgins. 2016. A Compiler for 3D Machine Knitting. *ACM Trans. Graph.* 35, 4, Article 49 (jul 2016), 11 pages. https://doi.org/10.1145/2897824.2925940

J. McCartney, B.K. Hinds, and K.W. Chong. 2005. Pattern flattening for orthotropic materials. *Computer-Aided Design* 37, 6 (2005), 631–644. https://doi.org/10.1016/j.cad.2004.09.006 CAD Methods in Garment Design.

J. McCartney, B.K. Hinds, B.L. Seow, and D. Gong. 2000. An energy based model for the flattening of woven fabrics. *Journal of Materials Processing Tech.* 107, 1-3 (2000), 312–318.

Yuwei Meng, Charlie CL Wang, and Xiaogang Jin. 2012. Flexible shape control for automatic resizing of apparel products. *Computer-Aided Design* 44, 1 (2012), 68–76.

Juan Montes, Bernhard Thomaszewski, Sudhir Mudur, and Tiberiu Popa. 2020. Computational Design of Skintight Clothing. *ACM Trans. Graph.* 39, 4, Article 105 (jul 2020), 12 pages. https://doi.org/10.1145/3386569.3392477

Vidya Narayanan, Lea Albaugh, Jessica Hodgins, Stelian Coros, and James McCann. 2018. Automatic Machine Knitting of 3D Meshes. *ACM Trans. Graph.* 37, 3, Article 35 (Aug. 2018), 15 pages. https://doi.org/10.1145/3186265

Vidya Narayanan*, Kui Wu*, Cem Yuksel, and Jim McCann. 2019. Visual Knitting Machine Programming. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)* 38, 4, Article 63 (jul 2019), 13 pages. https://doi.org/10.1145/3306346.3322995 (*Joint First Authors).

Rajkishore Nayak and Rajiv Padhye. 2017. *Automation in Garment Manufacturing*. Woodhead Publishing. 1–414 pages.

Stefano Nuvoli, Alex Hernandez, Claudio Esperança, Riccardo Scateni, Paolo Cignoni, and Nico Pietroni. 2019. QuadMixer: layout preserving blending of quadrilateral meshes. *ACM Trans. Graph* 38, 6 (2019), 180:1–180:13.

Optitex. 2022. optitex.com. https://optitex.com.

Ahmed A. A. Osman, Timo Bolkart, and Michael J. Black. 2020. STAR: Sparse Trained Articulated Human Body Regressor. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI (Lecture Notes in Computer Science, Vol. 12351)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 598–613.

Daniele Panozzo, Yaron Lipman, Enrico Puppo, and Denis Zorin. 2012. Fields on symmetric surfaces. *ACM Trans. Graph* 31, 4 (2012), 111:1–111:12.

D. Panozzo, Enrico Puppo, and L. Rocca. 2010. Efficient multi-scale curvature and crease estimation. *2nd International Workshop on Computer Graphics, Computer Vision and Mathematics, GraVisMa 2010 - Workshop Proceedings* (01 2010), 9–16.

Nico Pietroni, Stefano Nuvoli, Thomas Alderighi, Paolo Cignoni, and Marco Tarini. 2021. Reliable Feature-Line Driven Quad-Remeshing. *ACM Trans. Graph.* 40, 4, Article 155 (jul 2021), 17 pages. https://doi.org/10.1145/3450626.3459919

Nico Pietroni, Enrico Puppo, Giorgio Marcias, Roberto Scopigno, and Paolo Cignoni. 2016. Tracing Field-Coherent Quad Layouts. *Comput. Graph. Forum* 35, 7 (2016), 485–496.

Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. 2017. ClothCap: Seamless 4D clothing capture and retargeting. *ACM Trans. Graph.* 36, 4 (2017), 1–15.

Roi Poranne, Marco Tarini, Sandro Huber, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Autocuts: Simultaneous Distortion and Cut Optimization for UV Mapping. *ACM Trans. Graph.* 36, 6, Article 215 (nov 2017), 11 pages. https://doi.org/10.1145/3130800.3130845

Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable Locally Injective Mappings. *ACM Transactions on Graphics* 36, 2 (April 2017), 16:1–16:16.

Faniry H. Razafindrazaka, Ulrich Reitebuch, and Konrad Polthier. 2015. Perfect Matching Quad Layouts for Manifold Meshes. *Comput. Graph. Forum* 34, 5 (2015), 219–228.

C. Robson, R. Maharik, A. Sheffer, and N. Carr. 2011. Context-Aware Garment Modeling from Sketches. *Computers & Graphics (Proc. SMI 2011)* 35, 3 (2011), 604–613.

Kenneth Rose, Alla Sheffer, Jamie Wither, Marie-Paule Cani, and Boris Thibert. 2007. Developable Surfaces from Arbitrary Sketched Boundaries. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Barcelona, Spain) *(SGP '07)*. Eurographics Association, Goslar, DEU, 163–172.

Rohan Sawhney and Keenan Crane. 2017. Boundary First Flattening. arXiv:1704.06873 http://arxiv.org/abs/1704.06873

Nicholas Sharp and Keenan Crane. 2018. Variational Surface Cutting. *ACM Trans. Graph.* 37, 4, Article 156 (jul 2018), 13 pages. https://doi.org/10.1145/3197517.3201356

Alla Sheffer, Bruno Lévy, Maxim Mogilnitsky, and Alexander Bogomyakov. 2004. ABF++ : Fast and Robust Angle Based Flattening. *ACM Trans. Graph.* 24, 2 (2004), 311–330. https://hal.inria.fr/inria-00105689

SizeGermany. 2020. SizeGermany. https://portal.sizegermany.de.

Olga Sorkine and Marc Alexa. 2007. As-Rigid-as-Possible Surface Modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Barcelona, Spain) *(SGP '07)*. Eurographics Association, Goslar, DEU, 109–116.

Olga Sorkine-Hornung and Michael Rabinovich. 2016. Least-Squares Rigid Motion Using SVD. Technical note.

Oded Stein, Eitan Grinspun, and Keenan Crane. 2018. Developability of Triangle Meshes. *ACM Trans. Graph.* 37, 4, Article 77 (jul 2018), 14 pages. https://doi.org/10.1145/3197517.3201303

TiltBrush. 2022. www.tiltbrush.com. https://www.tiltbrush.com/.

Nobuyuki Umetani, Danny M. Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Transactions on Graphics (SIGGRAPH 2011)* 30, 4 (2011), 90.

Amir Vaxman, Marcel Campen, Olga Diamanti, David Bommes, Klaus Hildebrandt, Mirela Ben-Chen Technion, and Daniele Panozzo. 2017. Directional Field Synthesis, Design, and Processing. In *ACM SIGGRAPH 2017 Courses* (Los Angeles, California) *(SIGGRAPH '17)*. Association for Computing Machinery, New York, NY, USA, Article 12, 30 pages. https://doi.org/10.1145/3084873.3084921

Raquel Vidaurre, Igor Santesteban, Elena Garces, and Dan Casas. 2020. Fully Convolutional Graph Neural Networks for Parametric Virtual Try-On. *Computer Graphics Forum* 39, 8 (Dec. 2020), 145–156. https://doi.org/10.1111/cgf.14109

Charlie C.L. Wang, Yu Wang, and Matthew M.F. Yuen. 2005b. Design automation for customized apparel products. *Computer-Aided Design* 37, 7 (1 June 2005), 675–691. https://doi.org/10.1016/j.cad.2004.08.007

Charlie CL Wang, Kai Tang, and Benjamin ML Yeung. 2005a. Freeform surface flattening based on fitting a woven mesh model. *Computer-Aided Design* 37, 8 (2005), 799–814.

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

Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popovic, and Niloy J. Mitra. 2018. Learning a Shared Shape Space for Multimodal Garment Design. *ACM Trans. Graph.* 37, 6 (2018), 1:1–1:14. https://doi.org/10.1145/3272127.3275074

Amy Wibowo, Daisuke Sakamoto, Jun Mitani, and Takeo Igarashi. 2012. DressUp: a 3D interface for clothing design with a physical mannequin. In *Proceedings of the 6th International Conference on Tangible and Embedded Interaction 2012, Kingston, Ontario, Canada, February 19-22, 2012*, Roel Vertegaal, Stephen N. Spencer, Ylva Fernaeus, Audrey Girouard, and Sergi Jordà (Eds.). ACM, 99–102.

Katja Wolff, Philipp Herholz, and Olga Sorkine-Hornung. 2019. Reflection Symmetry in Textured Sewing Patterns. In *24th International Symposium on Vision, Modeling, and Visualization, VMV 2019, Rostock, Germany, September 30 - October 2, 2019*, Hans-Jörg Schulz, Matthias Teschner, and Michael Wimmer 0001 (Eds.). Eurographics Association, 11–18.

Katja Wolff, Philipp Herholz, Verena Ziegler, Frauke Link, Nico Brügel, and Olga Sorkine-Hornung. 2021. 3D Custom Fit Garment Design with Body Movement. arXiv:2102.05462 https://arxiv.org/abs/2102.05462

Kui Wu, Hannah Swan, and Cem Yuksel. 2019. Knittable Stitch Meshes. *ACM Transactions on Graphics* 38, 1, Article 10 (jan 2019), 13 pages. https://doi.org/10.1145/3292481

Cem Yuksel, Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2012. Stitch Meshes for Modeling Knitted Clothing with Yarn-Level Detail. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012)* 31, 3, Article 37 (2012), 12 pages. https://doi.org/10.1145/2185520.2185533