Knitting 4D Garments with Elasticity Controlled for Body Motion

ZISHUN LIU, Delft University of Technology, The Netherlands / Centre for Perceptual and Interactive Intelligence (CPII) Limited, Hong Kong, China
XINGJIAN HAN, Boston University, USA
YUCHEN ZHANG, Centre for Perceptual and Interactive Intelligence (CPII) Limited, Hong Kong, China
XIANGJIA CHEN, Centre for Perceptual and Interactive Intelligence (CPII) Limited, Hong Kong, China
YU-KUN LAI, Cardiff University, United Kingdom
EUGENI L. DOUBROVSKI, Delft University of Technology, The Netherlands
EMILY WHITING, Boston University, USA
CHARLIE C. L. WANG, The University of Manchester, United Kingdom

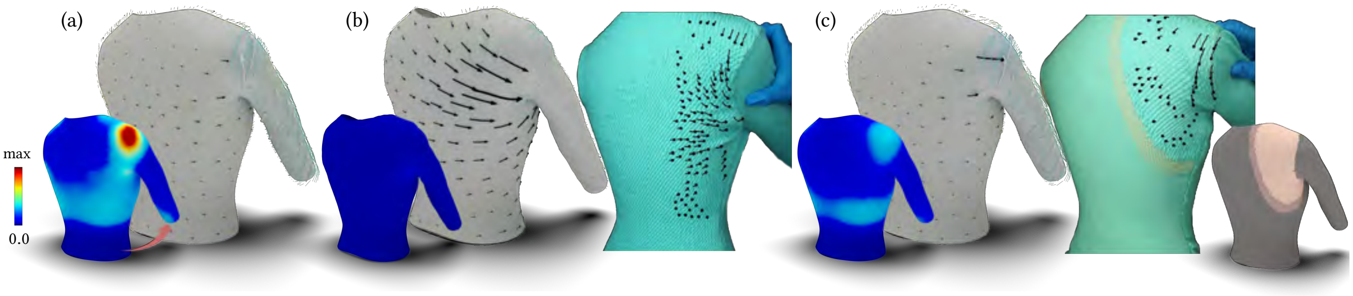


Fig. 1. During the body motion of swinging arms, a perfect-fit 3D garment can have: (a) large stress when using firm materials – leading to uncomfortable pressure or (b) large sliding when using soft materials – resulting in unwanted wrinkles. Both factors are considered in an integrated way on a knitwear with optimized distribution of elasticity as a 4D garment (c) that minimizes the stress and controls the maximal sliding during body motion. Stresses and displacements are visualized as color maps and black arrows respectively, where the maximal stress is 40.75kPa. Our work enables a computational framework for designing 4D garments and automatically fabricating them on digital knitting machines. A knitwear as 4D garment is physically fabricated by knitting different 'percentages' of firm and soft yarns in different regions. We make the regions of different elasticity visible by using firm yarns in light-blue and soft yarns in white. Sliding trajectories on physical specimens are evaluated by a vision-based method and displayed as black curves.

In this paper, we present a new computational pipeline for designing and fabricating 4D garments as knitwear that considers comfort during body movement. This is achieved by careful control of elasticity distribution to reduce uncomfortable pressure and unwanted sliding caused by body motion. We exploit the ability to knit patterns in different elastic levels by *single-jersey jacquard* (SJJ) with two yarns. We design the distribution of elasticity for a garment by physics-based computation, the optimized elasticity on the garment is then converted into instructions for a digital knitting machine by two algorithms proposed in this paper. Specifically, a graph-based algorithm is proposed to generate knittable stitch meshes that can accurately capture the 3D shape of a garment, and a tiling algorithm is employed to assign SJJ patterns on the stitch mesh to realize the designed distribution of elasticity. The effectiveness of our approach is verified on simulation results and on specimens physically fabricated by knitting machines.

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

Additional Key Words and Phrases: knitting, elasticity control, 4D garment, computational fabrication

ACM Reference Format:
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 (August 2021), 16 pages. https://doi.org/10.1145/3450626.3459868

1 INTRODUCTION

Designing clothing that considers the movement of the body is an age-old problem in the garment industry. Computer graphics research has advanced many aspects of garment design, enabling

Authors' addresses: Zishun Liu, Delft University of Technology, The Netherlands / Centre for Perceptual and Interactive Intelligence (CPII) Limited, Hong Kong, China, z.liu-4@tudelft.nl; Xingjian Han, Boston University, USA, xjhan@bu.edu; Yuchen Zhang, Centre for Perceptual and Interactive Intelligence (CPII) Limited, Hong Kong, China, yczhang@cpii.hk; Xiangjia Chen, Centre for Perceptual and Interactive Intelligence (CPII) Limited, Hong Kong, China, xjchen@cpii.hk; Yu-Kun Lai, Cardiff University, United Kingdom, laiy4@cardiff.ac.uk; Eugeni L. Doubrovski, Delft University of Technology, The Netherlands, E.L.Doubrovski@tudelft.nl; Emily Whiting, Computer Science Department, Boston University, USA, whiting@bu.edu; Charlie C. L. Wang, Department of Mechanical, Aerospace and Civil Engineering, The University of Manchester, United Kingdom, changling.wang@manchester.ac.uk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2021 Association for Computing Machinery.
0730-0301/2021/8-ART62 $15.00
https://doi.org/10.1145/3450626.3459868

perfect-fit 3D garments for varied body shapes (e.g., [Brouet et al. 2012; Montes et al. 2020]). Physics-based optimizations have also been employed to support both aesthetic design and comfort. Recently, the apparel industry has introduced innovations to vary elasticity in knitwear allowing customization for body motion, which is referred to as *4D garment design*. However, existing methods mainly rely on experience and intuition. Our work presented in this paper enables the computational design and fabrication of 4D garments as knitwear. In industry [Ladevèze et al. 1860], the quality of a garment can be evaluated as the perfect-fit between the cloth and body. According to interviews with garment experts, relative sliding is the most direct measurement when garments under motion are considered. As explained in Fig. 1, the stress in the garment and the sliding of the garment over the body need to be controlled to achieve desired comfort and perfect-fit. On the one hand, large stresses in the garment caused by firm materials can lead to uncomfortable stretch and compression. On the other hand, when using soft materials, sliding of the garment over the body can lead to unwanted wrinkles. We introduce a computational pipeline to control stress and sliding on clothes by distributed elasticity for improving the comfort in motion, with a focus on the challenging case of tight fitting garments.

Computer-controlled knitting has increasingly been capturing the attention of the computational design and fabrication community. Researchers have sought to reduce the barrier of programming a knitting machine [McCann et al. 2016]. Knitting instructions can now be automatically generated from 3D geometric input [Narayanan et al. 2018]. High-level visual programming interfaces have also been developed to improve the efficiency of generating knitting instructions [Narayanan et al. 2019]. Compared to conventional 3D garment production techniques, digitally knitted clothes show the following advantages:

- The capability to generate garments with numerous 'darts', which are automatically 'sewed' together on knitted clothes and therefore fabricate garments with complex 3D freeform surfaces.
- The capability to use mixed yarns in different regions that can locally control the level of elasticity to absorb deformations on human bodies in motion (see Fig. 1).

In our approach we utilize these two capabilities to produce 4D garments.

1.1 Our Method

To enable the design and fabrication of knitwear with controlled elasticity distribution, we first precisely fabricate the designed 3D shape by digital knitting, and then realize the elasticity variation in different regions by *single-jersey jacquard* (SJJ) with two yarns. These two components are strongly coupled. Specifically, without a fabrication method to perfectly realize the designed 3D shape, the garment will deform when worn on the body. In our paper we aim for the default pose to have uniform stresses (achieved by 3D shaping); then the elasticity assignment is only concerned with stresses and deformations that arise from motion. The stitch meshes generated by our algorithm use only the short-row knitting strategy to form 3D geometry, which can be executed on a knitting machine more efficiently. A tiling algorithm is employed to assign SJJ patterns on the stitch mesh to realize the designed distribution of elasticity.

In summary, we present the following technical contributions:

- A graph-based method to generate a knittable stitch mesh that can accurately capture the 3D shape of a surface patch. Our approach uses only the short-row knitting technique while controlling distortion at *apexes* (end stitches of short-rows).
- A method to generate machine knitting code for 3D garments with locally varying levels of elasticity, using different SJJ patterns with two yarns (one soft and one firm).
- An iterative algorithm to assign different levels of elasticity in different regions of a garment so that the deformation under body motion can be optimized. We consider reduced stress and controlled sliding to achieve 4D garment design.

To our knowledge, this is the first approach to simultaneously optimizing the comfort (reducing stress) and the perfect-fit (avoiding wrinkles caused by large sliding) under motion by tuning the elasticity of knitwear that can be automatically fabricated by digital knitting machines.

1.2 Related Work

1.2.1 *Cloth Simulation and Garment Design.*

After the pioneering work of simulating 3D clothes as deformable objects [Volino et al. 1995], many computational methods have been developed in computer graphics to simulate the physical behavior of clothes on human bodies in motion (e.g., [Baraff and Witkin 1998; Goldenthal et al. 2007]). The method of Berthouzoz et al. [2013] can parse patterns made by professional designers and automatically generate virtual fitting results. With the help of these simulation and forward design techniques, inverse design of garments can be generated by optimizing the planar panels to achieve the desired effect of clothes fitting [Bartle et al. 2016; Li et al. 2018b; Umetani et al. 2011]. In another aspect, the so-called design transfer function [Brouet et al. 2012; Meng et al. 2012; Wang et al. 2007] can be realized by generating the same fitting results when applied to human bodies in different 3D shapes and poses [Anguelov et al. 2005]. Limited by the means of fabrication, these methods often compute planar panels as the outcome. When sewn back into 3D garments, the elasticity is not tailor-made. In contrast, our approach enables programmable elasticity on 3D knitwear which can be digitally fabricated.

In order to simulate the physical behavior of knitted fabrics more precisely, computational models have been developed to simulate knitted cloth at the yarn level [Kaldor et al. 2008; Nocent et al. 2001]. The concept of the stitch mesh is introduced by Yuksel et al. [2012] to provide a canvas-like abstraction of the yarn model although it is not guaranteed to be knittable. With the help of these two techniques, Leaf et al. [2018] developed an interactive tool for designing yarn-level patterns for both knitted and woven cloth. Although these computational tools at the yarn-level are available in the literature, we however, simulate garments with distributed elasticity at the level of triangular meshes for the sake of computational efficiency, where the orthotropic material properties are calibrated experimentally (ref. [Wang et al. 2011]) and assigned to each triangular element. The cloth simulator, ARCSim [Narain et al. 2013b], embedded with the function of adaptive anisotropic remeshing [Narain et al. 2013a, 2012] can capture physical behavior in more detail while using a relatively coarse mesh as input. In our algorithm of inverse design, we modify the simulator to enable the assignment of different elasticity on different triangular elements.

Skintight garments applied to human bodies with tight-fitting clothing have been widely used in casual fashion, sportswear and medical treatment applications [Kwok et al. 2016; Montes et al. 2020; Wang and Tang 2010]. The work of Kwok et al. [2016] has considered aesthetic factors to develop an evolution process for inverse design of panel layout on 3D human bodies, which can be fabricated from planar panels generated by flattening algorithms (e.g., [Liu et al. 2008; Wang et al. 2010]). Physics-based computation is introduced in [Wang and Tang 2010] to generate optimized 2D panels according to prescribed pressure distribution on a 3D body shape. Recently, Montes et al. [2020] proposed a physics-based method to compute 2D panels of skintight garments by incorporating body deformation in multiple poses, pressure distribution, and seam traction into design objectives. Differently, our method does not have restrictions of 2D panels, enabling more accurate capture of the 3D shape. Moreover, we optimize the distribution of fabric elasticity based on physical simulation of garments during body motion, and propose a computational pipeline to enable automatic fabrication on digital knitting machines.

1.2.2 Computational Fabrication of Knitwear. Igarashi et al. [2008] introduced some of the earliest work on generating knitting code from an input 3D model. Their algorithm consists of two major steps: 1) segmenting the surface of a model into parallel strips with constant width and 2) sampling the boundary of each strip into knitting patterns. Constraints of machine knitting (e.g., the maximal allowed variation of stitches between rows) were not considered, which are investigated comprehensively and incorporated into the algorithm of [Narayanan et al. 2018]. Wu et al. [2019] have proposed algorithms to convert a stitch mesh into a knittable one, where stitch meshes as isotropic quad-meshes are constructed on input 3D models by a global parameterization for simulating knitwear [Wu et al. 2018]. With the help of prior work (e.g., [McCann et al. 2016]), knitting machines can automatically execute instructions from the knitting maps as shown in Fig. 15. However, none of these approaches exploit the shape distortion on knitted clothes caused by different numbers of stitches between rows. Details will be discussed in Section 4. Moreover, the realization of distributed elasticity has not been considered yet.

More research efforts have been made to provide high-level visual programming interfaces [Kaspar et al. 2019; Narayanan et al. 2019]. An augmented data structure of stitch mesh is presented in [Narayanan et al. 2019] as the core of their interface for interactively editing codes for machine knitting. We also employ their data structure to represent knittable stitch meshes, which can be easily converted into stitch maps to be executed on knitting machines. The interface developed in [Kaspar et al. 2019] allows users to specify both the knitting primitives and the stitch patterns. Different stitch patterns are designed in a way like image editing. However, the difference in elasticity observed on different knitting patterns has not been adopted for functional usage. In short, no prior research has been performed to enable the programmable tailor-control of elasticity on a knitwear. Note that 4D garments mentioned in this paper are different from the soft actuated garments [Albaugh et al. 2019] and self-folding textiles [Knittel et al. 2015], where the deformation is generated on clothes by knitted actuators and residual strains respectively.

1.2.3 Inverse Design of Deformable Objects. Designing distributed elasticity on garments is related to a wider range of inverse design applications that consider structural strength and deformation behavior of deformable objects. An inverse design approach was proposed in [Ly et al. 2018] for garments made from non-flat rest shape. Combinations of different materials and therefore tailor-made elasticity were adopted as design variables in [Bickel et al. 2010; Skouras et al. 2013; Xu et al. 2015] to achieve a desired deformation behavior. Besides multiple materials, structural strength [Chen et al. 2014; Pérez et al. 2015], cellular micro-structures [Panetta et al. 2015; Schumacher et al. 2015] and composite silicone [Zehnder et al. 2017] have also been inversely designed for desired deformation in prior research. Because of complex physical phenomena that are difficult to simulate accurately, the properties of physical models are often captured by a data-driven methodology in these approaches (e.g., [Bickel et al. 2010; Panetta et al. 2015; Schumacher et al. 2015; Zehnder et al. 2017; Zhang et al. 2016]). The same strategy is employed in our approach to capture the elasticity of different SJJ patterns by knitting two yarns – one soft and the other firm.

The elasticity on woven fabrics has been employed in applications of computational fabrication. Small structures [Guseinov et al. 2017] and layouts of inextensible polymer [Pérez et al. 2017] are 3D printed on stretched fabrics to form a desired shape after releasing stretch. Planar panels of fabric containers inflated by pressurized air [Skouras et al. 2014] and pressures of injected viscous fluid [Zhang et al. 2019] are inversely designed using orthotropic material properties of fabrics. The same material model is employed in our physics-based inverse design, where the material parameters are measured from physically fabricated SJJ patterns.

2 KNITTING STRATEGY

Knitting is an ancient method to fabricate clothes by manipulating loops of yarns. In this section, we define the basic terminology and exploit the strategy of knitting a 3D surface patch with distributed elasticity. We focus on weft knitting, which is conceptually the same as hand knitting and can be conducted in 1) a fully automatic way on industry-level machines with two beds of slide needles [Seiki 2019] (see Fig. 15) or 2) a semi-automatic manner on a household knitting machine [Brother 2013] with only one needle-bed. For weft knitting, the loops of yarns are formed row by row, where each loop is called a *stitch* and a row of stitches is called a *course*. Loops are formed on the current course by switching the sliders on needles and trapping the yarn onto the corresponding loops of the previous course [Albaugh et al. 2019].

Knitting can be arranged into sheets or tubes when different machines are used. In this work, we focus on knitting curved sheets which can be realized on general industrial knitting machines with two needle-beds or even one needle-bed. Although tubular knitting can also be realized on a machine with two needle-beds, it is not preferred in the context of this work as only limited patterns of

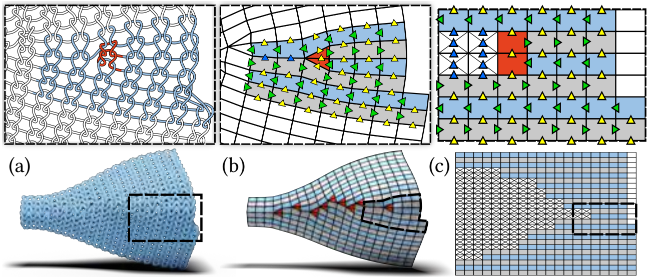


Fig. 2. A surface patch in 3D can be seamlessly knitted by using the short-row shaping strategy. Three representations are used to illustrate the knitted stitches: (a) a yarn-level model which depicts the stitches faithfully, (b) a stitch mesh as an abstract representation at the fabric level where each triangle / quadrangle corresponds to a stitch in the yarn-level model, and (c) the knitting map as a 2D grid which actually provides the knitting instructions – the yellow (and blue) arrows indicate the connected loops and the green arrows provide the knitting direction of yarn-carrier. The boundaries of a 'dart' (indicated by blue arrows) are automatically knitted together as indicated by the blue arrows.

stitches can be applied [Spencer 2001]. Other surface topologies can be converted to a disk-topology by adding cuts [Zhang et al. 2005].

2.1 3D Shaping by Short-Row

Knitted fabrics in general have constant width s_w and height s_h for all stitches (as illustrated in Fig. 2(a)), which are determined according to both the type of yarn and the mechanical design of needle-beds. Knitting a regular grid of stitches produces a rectangular (planar) patch of fabric. To form a 3D surface, different numbers of stitches are used for different courses. Two strategies are usually conducted, which are 'conjugate' to each other (Fig. 3):

(1) *Increase / decrease shaping* changes the local width of the fabric by adding an extra loop in a course (or trapping a loop to more than one loop in the previous course).
(2) *Short-row shaping* changes the local height of the fabric by knitting partial rows in regions needing more stitches.

Partially knitting rows with fewer stitches has the outcome of collapsing local regions (indicated by diagonal cross marks in Fig. 2(c)). This is analogous to adding darts when forming 3D non-developable surfaces by inextensible planar materials [Decaudin et al. 2006; Mitani and Suzuki 2004]. Differently, the knitting process can automatically 'sew' a 'dart' by connecting its boundary stitches (e.g., the stitches connected by the blue arrow in Figs. 2(b) and (c)).

The benefit of using the short-row shaping strategy is twofold. Increasing / decreasing stitches in the middle of a course needs more machinery operations (e.g., switching the loops between two beds of needles), which is less efficient than knitting short-rows only – see the comparison in Fig. 3. Moreover, short-row shaping can be realized on machines with only one needle-bed (e.g., the low-cost household machines that are widely used by makers). Note that, interior ending stitches of short-rows are represented with triangles on a stitch mesh (see Fig. 2(b)). The same stitch pattern is applied to the triangular faces and the quadrangular faces, where the only difference is that the stitches in a triangular face are somewhat collapsed by its neighbors – therefore causing local shape distortion on the physical knitting results. The corresponding stitches of triangles on a knittable stitch mesh are referred to as *apexes* in the rest of this paper. In Section 4, we develop an algorithm to knit general 3D surfaces by only using short-row shaping, while controlling the accumulation of the distortions on apexes.

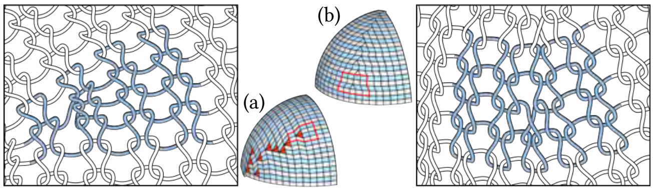
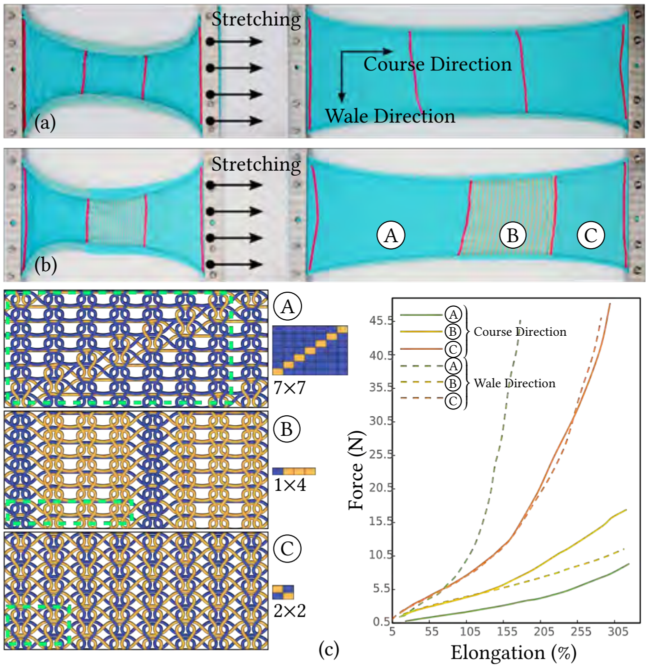


Fig. 3. Two different strategies to knit the same 3D shape: (a) short-row shaping and (b) increase / decrease shaping, where increasing / decreasing stitches between courses take more operations on a machine with two needle-beds. For this example, knitting by only using short-row shaping can improve the knitting speed by 39.5% on the same machine.



Stitch Size (Unit: mm)	Ⓐ	Ⓑ	Ⓒ
Width × Height	0.125 × 0.0780	0.0984 × 0.0680	0.0812 × 0.105

Fig. 4. Different from stretching a fabric knitted by using a single SJJ pattern (a), different elongations can be observed when using three different SJJ patterns (b). Yellow denotes soft yarns and blue is used for firm yarns. Three different SJJ patterns represented by yarn-level models and their corresponding strain-force curves (in horizontal and vertical tensile tests) are given in (c). When designing different SJJ patterns, floats are controlled within the lengths allowed by reliable machine knitting. The average stitch sizes of different SJJ patterns are also given.

2.2 Elasticity by Single-Jersey Jacquard

We adopt the SJJ technique to generate distributed elasticity for the same reason that we use short-row 3D shaping: efficiency and

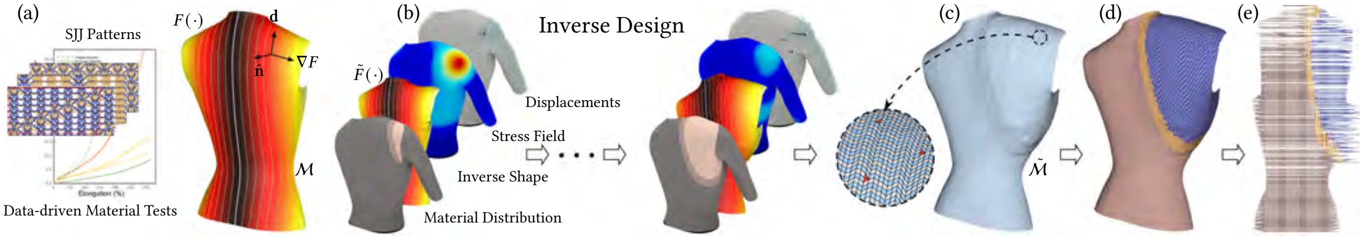


Fig. 5. Our computational design and fabrication pipeline to produce 4D knitwear with elasticity controlled for body motion: (a) data-driven material tests of SJJ patterns and the geodesic distance-field $F(\cdot)$ on a garment $\mathcal{M}$ for assigning orientations for knitting stitches and orthotropic material simulation, (b) progressive updating of the soft / firm material distribution, (c) knittable stitch mesh generated on the inverse geometry, (d) SJJ patterns assigned on the stitch mesh and (e) the resultant knitting map. Note that, after computing the inverse geometry $\tilde{\mathcal{M}}$ of the garment to compensate for the variation of shrinkage ratios, the orientation of each triangle in orthotropic material simulation is re-evaluated on an updated geodesic distance-field $\tilde{F}(\cdot)$.

feasibility on low-cost machines (i.e., even for those with only one needle-bed). SJJ is a technique to knit two or more yarns together when moving the carrier horizontally. Specifically, two yarns with different elasticity levels (soft and firm) are employed in our approach. The soft and firm yarns are led at different heights by the carrier, and the needle hooks are shifted to their corresponding height to determine which yarn is used to form the loop.

When altering the arrangement of the two yarns, different elasticities can be formed by changing the periodic pattern. See Fig. 4 for three example patterns that result in different strain-force curves. In single-jersey jacquard fabrics, the same knitting map (as given in Fig. 2(c)) can still be used to govern the machine operation. A remaining problem is how to assign the SJJ information to a knitting map. We present a tiling algorithm in Section 5.2 for this purpose.

When starting to knit a new course, loops in the previous course are taken off the hooks of the needles. Although the distances between needles are constant, loops formed by different yarns will shrink into different sizes. The elasticity control on a knitted garment needs to take this variation of shrinkage into consideration. We compute the inverse geometry of a 3D surface for incorporating this factor (see Section 5.1).

3 DESIGN AND FABRICATION OF 4D KNITWEAR

Given a 3D garment $\mathcal{M}$, the shape of which is designed around the human body $\mathcal{H}$ in a static pose, we evaluate the performance of $\mathcal{M}$ under a prescribed body motion of $\mathcal{H}$ for comfort on a physical simulator. Specifically, for comfort evaluation the maximal stress $\sigma_{\max}$ is measured on $\mathcal{M}$ among all elements at all time steps during the motion. The maximal sliding $d_{\max}$ between $\mathcal{M}$ and $\mathcal{H}$ in a user-specified region Ω is also detected during the whole motion sequence. The computational design and fabrication problem to be solved here is 1) to find a distribution $\mathcal{P}$ of elasticity that minimizes $\sigma_{\max}$ while controlling the allowed maximal sliding distance as $d_{\max} \leq \bar{d}$, and 2) to determine a knitting map $\mathcal{K}$ that can realize $\mathcal{P}$ by SJJ patterns with two yarns. $\bar{d}$ is a user-defined threshold.

We tackle the problem by first computing the distribution of elasticity $\mathcal{P}^e$ at the element level according to predefined SJJ patterns and then generating the knitting map $\mathcal{K}$ that can realize $\mathcal{P}^e$ by these SJJ patterns. The pipeline is illustrated in Fig. 5.

3.1 Inverse Design of Distributed Elasticity

The physical properties of elasticity that can be realized by SJJ patterns are determined by a data-driven method similar to Wang et al. [2011]. Our flat sheets knitted from uniform SJJ patterns are simulated effectively by an orthotropic fabric model with physical properties measured as in Wang et al. [2011]. The first step of our computational pipeline determines the orientation of knitting structures for each piece of a garment $\mathcal{M}$. A base line that indicates the knitting direction is defined by the user. The base line can be selected from the feature curves of the garment (e.g., the central line of a front / back piece) or specified according to a desired direction of knit stitches. A geodesic distance-field $F(\cdot)$ of the base line is computed to determine the orientation of stitches by assigning 1) the weft direction as the gradient of $F(\cdot)$, ∇F, and 2) the wale direction as $\hat{\mathbf{n}} \times \nabla F$ with $\hat{\mathbf{n}}$ being the surface normal. With these assigned weft / warp directions (see the illustration in Fig. 5(b)), the material orientation in the orthotropic model can be determined and used in the physical simulation below.

The elasticity distribution is determined on $\mathcal{M}$ at the triangular element level by a greedy algorithm.

1. First, two simulations are conducted with the soft and the firm materials to obtain their corresponding maximal stress and sliding as $(\sigma^s_{\max}, d^s_{\max})$ and $(\sigma^f_{\max}, d^f_{\max})$ respectively.
2. We then initialize with a uniform distribution of firm material by assigning the material level as 0. Here we adopt different material levels as: 0-firm, 1-medium, 2-soft.
3. Regions with large stresses are progressively assigned into soft and medium by using the following steps (see Fig. 5(b)).
   - Regions of the designed domain with stress value larger than 90% of the current maximal stress are assigned as soft using material level 2.
   - The filtering step is applied to perform 5 iterations of Laplacian smoothing on the material levels as floating-point values on all triangular elements.
   - Quantization is conducted by rounding the material level from floating-point value back to integer as {0, 1, 2}, to generate a new material distribution in discrete levels.
4. After updating the material distribution, the simulation is run again to obtain the updated $\sigma_{\max}$ and $d_{\max}$.
5. Go back to (3) or stop the iteration when $d_{\max} > \bar{d}$.

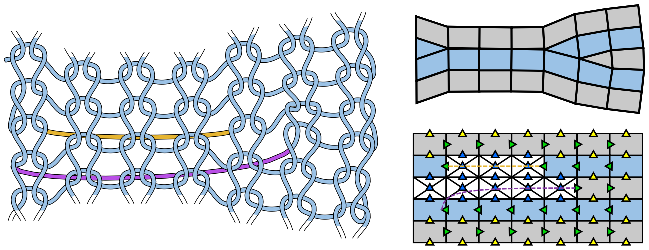


Fig. 6. Long floats will be formed when discontinuity occurs in the same course (shown in yellow) and between courses (shown in purple). Both can lead to unstable knitting with incorrect stitches formed.

3.2 Enabling Machine Knitting

Given a distribution of elasticity $\mathcal{P}^e$ assigned on $\mathcal{M}$, we compute the knitting map $\mathcal{K}$ in four steps to realize this 4D garment.

1. **Inverse geometry computation.** To compensate for the variation of shrinkage ratios among different SJJ patterns, an inverse geometry of $\mathcal{M}$ is computed as $\tilde{\mathcal{M}}$ (see Fig. 5(c)). Computing stitch meshes on $\tilde{\mathcal{M}}$ will then incorporate the pattern-specific shrinkage ratio on the rest shape $\mathcal{M}$ (Section 5.1). As the geodesic distance-field on $\tilde{\mathcal{M}}$ is changed to $\tilde{F}(\cdot)$, this step is also included in the routine of inverse design to re-assign the material orientation of each triangle.
2. **Knittable stitch meshes** are computed on the inverse geometry $\tilde{\mathcal{M}}$ by using the geodesic distance-field $\tilde{F}(\cdot)$ (see Fig. 5(c)). Quadrangles / triangles of a stitch mesh are constructed between $\tilde{F}$'s isocurves by optimizing the shape of facets and repulsing the location of triangles (i.e., apexes), which is formulated as finding a shortest path on a graph. Sections 4.2 & 4.3 present the details of our algorithm.
3. **SJJ patterns** are assigned to the knittable stitch meshes to realize the distributed elasticity (see Fig. 5(d)). A flooding algorithm is employed to determine the local row / column indices of tiling. An algorithm based on *minimal spanning tree* (MST) is developed to minimize the discontinuity in Section 5.2.
4. **The knitting map** that can be executed on a machine is generated from a stitch mesh (see Fig. 5(e)). The location of each stitch (quadrangle or triangle) on a knitting map can be determined by assigning it with the sorted indices of course and wale (see Section 4.4).

4 DISTORTION-CONTROLLED 3D KNITTING

When conducting the short-row shaping strategy to knit a 3D surface patch, the following rules are applied to generate the knitting maps considering manufacturing constraints [Spencer 2001].

1. *Course ordering:* The connected stitches are knitted in an order queue of courses from bottom to top.
2. *In-course direction:* Stitches in the same course are knitted in an alternating left-to-right and right-to-left order[1].
3. *In-course continuity:* All stitches in the same course should be neighboring connected.
4. *Between-course continuity:* The ending stitch of the current course should be neighboring to the starting stitch of the next courses.

The first two rules are derived according to the working principle of weft knitting. The third and the fourth rules are defined for imposing the continuity in knitting, without which unwanted long floats will be generated (see Fig. 6). The needle hanged with long floats will form a wrong stitch while knitting the next course. All these four rules are incorporated into our algorithms to generate knittable stitch meshes only using the short-row 3D shaping technique.

Given the per stitch constant height s_h and width s_w, we need to ideally generate a stitch mesh on the given 3D surface $\mathcal{M}$ using quadrangles of size $s_w \times s_h$ and triangles as the end of short-rows. To achieve this goal, we first generate wale curves with equal distance s_w, which are sampled into segments with length s_h. The final stitch meshes are constructed by connecting neighboring wale curves with optimized quadrangles / triangles.

4.1 Generation of Wale Curves

Given an input 3D surface $\mathcal{M}$ that is homeomorphic to a 2D disk, the geodesic distance-field $F(\cdot)$ is first computed from the user-defined source (point or curve) on the input surface (see the left of Fig. 7(a)). Wale curves are generated from $F(\cdot)$'s isocurves that are evenly distributed with distance s_w.

A valid wale cannot be a closed curve. Therefore, if the source is a point, it should lie on the boundary of $\mathcal{M}$. If the source is a curve, it should be a curve cutting $\mathcal{M}$ into two pieces or a curve completely laying on $\mathcal{M}$'s boundary. In our project of garment design, pieces of an input garment usually have reflectional symmetry. The axial curve of symmetry that goes through the input mesh is adopted as the geodesic source. However, isocurves can still form closed loops in some extreme cases (see Fig. 7(a) for an example), which should be split. Inside a closed isocurve, there is a point $\mathbf{p}$ containing the local maximum of $F(\cdot)$. We can split the closed loop by computing a shortest path from $\mathbf{p}$ to the surface's boundary and cutting the surface along the path (see Fig. 7(a) for such a cut). This process is repeated until no closed loop is found on wales.

All wale curves generated in this way can be easily sorted by their geodesic distances, where the sign of distance can be assigned when the source curve separates an input surface into two regions – i.e., one side positive while the other side as negative. The direction of a wale curve can be determined as $\hat{\mathbf{n}} \times \nabla F$ with $\hat{\mathbf{n}}$ being the surface normal (see $\mathbf{d}$ shown in Fig. 7(a)). Vertices of a stitch mesh are generated on wale curves by sampling them into segments with length s_h. In the following step of mesh generation, quads and triangles are always constructed in pairs to ensure knittability (see details in Section 4.2). Therefore, even number of segments are required for each wale. To achieve this, a wale curve's two ends are slightly extended when odd segments are obtained.

4.2 Constructing Meshes between Wales

Given two neighboring wale curves, there are numerous ways to connect the sampled points on them to form a stitch mesh. However, not all are knittable. We propose a graph based algorithm to determine an optimized mesh that is knittable (Fig. 7(b)).

[1] Therefore, there is a starting / ending stitch for each course.

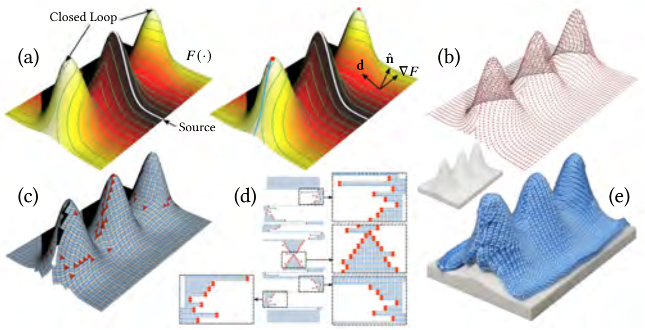


Fig. 7. Steps for generating knittable stitch meshes on a given surface: (a) Isocurves of a geodesic distance-field $F(\cdot)$ are used as wale curves with segments having nearly equal length. The input mesh will be cut if there are closed wale curves. The resultant wale curves are sorted according to their values of $F(\cdot)$. (b) Optimal quadrangles / triangles are constructed between neighboring wale curves. (c) Courses are traced out and sorted on a stitch mesh to generate the final knitting map. (d) The physical knitting result can be generated by following the knitting map, where the stitch size is 3.5mm × 2.8mm in this example. (e) The physical knitting result.

By using the short-row shaping technique, a knitting machine produces fabrics course-by-course from bottom to top in a zigzag manner while meeting both the in-course continuity and the continuity between courses. Without loss of generality, we assume the yarn carrier goes from left to right on courses with even indices (e.g., gray courses in Figs. 6, 7) and from right to left on odd courses (e.g., blue courses in Figs. 6, 7). Then, we impose the constraints of short-row knitting by defining where and how quads and triangles are presented in a stitch mesh.

A triangle defines the interior end (either head or tail) of a short-row, which is only located in the interior of a stitch mesh. After imposing the constraint of between-course continuity, triangles always appear in pairs and are one of two types:

- *Left-End Pair*: the lower triangle serves as the tail of a right-to-left knitting course while the upper one is the head of a left-to-right knitting course (e.g., the apexes shown in Fig. 2(b)).
- *Right-end pair*: the lower triangle in the pair is the tail of a left-to-right knitting course and the upper one is the head of a right-to-left course.

Operators are defined below to ensure that triangles are only presented in one of these two configurations. Similarly, quads are also constructed in pairs in our algorithm to ensure the knitting directions and the continuity of courses.

We employ a graph-based method to generate the mesh between neighboring wale curves, inspired by Wang and Tang [2005]. Constructing a wale mesh is equivalent to finding the set of bridge edges (or simply called *bridges*) that connect vertices between two wale curves. Given two wale curves A and B that are sampled into $2n_a$ and $2n_b$ segments, our algorithm computes the optimal and knittable bridges connecting their vertices $\{a_i\}$ $(i = 0, 1, \ldots, 2n_a)$ and $\{b_j\}$ $(j = 0, 1, \ldots, 2n_b)$. At initialization (see Fig. 8(a)), we define candidate *anchor* bridges between wale vertices with even indices, where the bridge length is less than λ. Having a very long bridge will lead to stitch meshes with large distortion, we chose $\lambda = s_w + 2s_h$ as a threshold in our implementation. Our graph-based method represents the anchor bridges as nodes on a graph Γ (see Fig. 8(c)). We then generate a knittable stitch mesh by finding an optimal path that connects a selected set of nodes via operators.

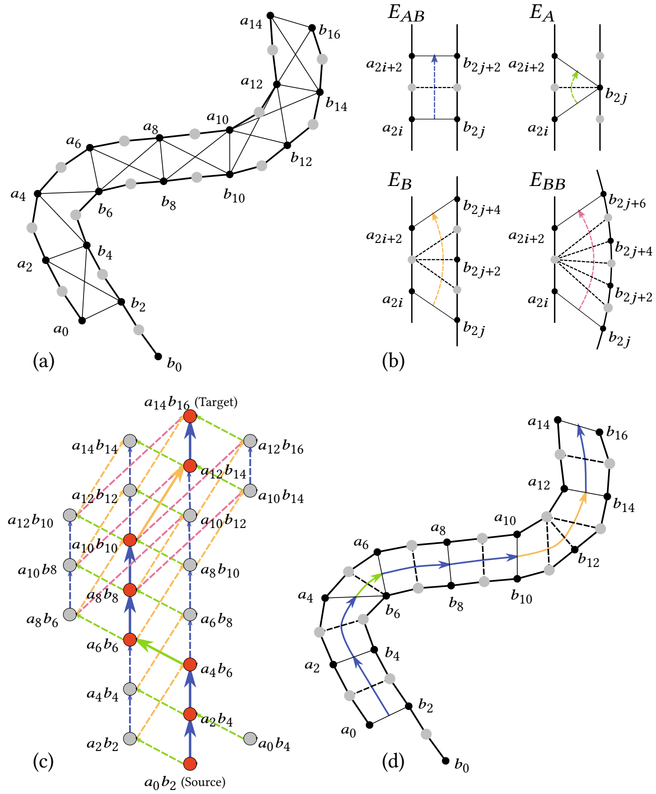


Fig. 8. Graph-based generation of knittable stitch meshes between wale curves. (a) Points are sampled on two wale curves (even indices shown in black, odd indices in gray). Candidate anchor bridges connecting the wale curves are illustrated by line segments. (b) Four operators are introduced in our algorithm to form the mesh. These operators insert intermediate bridges (dashed lines) to construct pairs of triangles and quads between consecutive anchor bridges (solid lines). (c) The graph Γ shows all possible configurations to tessellate the two wale curves. Anchor bridges are represented as nodes of Γ and operators are denoted by edges of Γ (different colors denote different types). The problem of finding an optimized tessellation between two wale curves is converted to finding a shortest path on Γ (i.e., the bold arrows). (d) The resultant mesh tessellation corresponding to the shortest path on Γ.

Four different operators are defined to insert *intermediate* bridges between the anchors:

- *AB-advance*: Two quadrangles are formed between $a_{2i}b_{2j}$ and $a_{2i+2}b_{2j+2}$ by adding two new bridges $a_{2i+1}b_{2j+1}$ and $a_{2i+2}b_{2j+2}$ (illustrated as E_{AB} in Fig. 8(b)).
- *A-advance*: Two triangles are constructed between $a_{2i}b_{2j}$ and $a_{2i+2}b_{2j}$ by adding a right-end pair (see E_A in Fig. 8(b)).
- *B-advance*: The purpose of this operator is to add two triangles as a left-end pair. However, as a left-end pair of triangles can only start from an odd course, we add one quad, two triangles

and the other quad all together between $a_{2i}b_{2j}$ to $a_{2i+2}b_{2j+4}$ (see E_B shown in Fig. 8(b)).

- *BB-advance*: While right-end pairs can be consecutively presented in the stitch mesh by repeatedly applying the *A-advance* operator, the consecutive left-end pairs cannot be realized by the *B-advance* operator. We define a new operator by adding one quad, four triangles and the other quad between $a_{2i}b_{2j}$ to $a_{2i+2}b_{2j+6}$ (see E_{BB} shown in Fig. 8(b)).

Each operator defines a directed edge in Γ with the weight defined as the total length of newly added bridges. If the length of any bridge introduced by an operator is longer than λ, we exclude its corresponding edge from Γ.

Given a source node and a target node, an optimal mesh can be generated by computing a shortest path between them on the graph Γ (see Figs. 8(c) and (d)). Dijkstra's algorithm is employed. By using the bridge's length as weight (i.e., the metric to evaluate the quality of mesh), quads and triangles with less distortion can be generated (ref. [Fuchs et al. 1977]). When the wales are located near the boundary of an input surface, the lengths of two wale curves can have significant difference. We then determine the source and the target nodes by the following method. Two vertices a_{2i} and b_{2j} are considered a bijective pair if b_{2j} is a_{2i}'s closest neighbor on B and vice versa. The first and the last bijective pairs are selected as the source and the target nodes in our approach, which results in stitch meshes with quads in good quality even on highly curved surfaces (e.g., the result shown in Fig. 7(c)). The graph-based algorithm introduced here can always generate a knittable stitch mesh after applying it to all the strips between wale curves, which is guaranteed by the four carefully defined operators.

4.3 Distortion Control by Apex Diffusion

Besides the shape of elements, we observe another source of distortion on stitch meshes – that is the location of apexes. As discussed in Section 2, each apex in physical knitting is in fact formed by collapsing the corresponding stitch at the end of each course. If there are many apexes placed together, the distortion errors generated by this collapse are accumulated and can lead to significant distortion in the global shape of the knitted patch. For example, in the top row of Fig. 9, the global distortion of the hemisphere can be clearly observed from the results of the FEA simulation and also the physical fabrication.

To solve this problem, we introduce a *diffusion term* for the weight of an edge in the graph Γ. This term will penalize the apexes close to each other. Specifically, for an operator introducing apexes, the weight on its corresponding edge in the graph is defined as $L + w \exp(-D^2)$ to penalize large L and small D with L being the total length of the newly added bridges and D being the distance between the newly added apex and the apexes already existing while tessellating previous wales. We selected $w = 100s_w$ to reflect the influence of stitch size s_w. With the help of this diffusion term, scattered distribution of apexes will be generated when constructing meshes between wale curves. The distortion errors caused by apexes are therefore diffused into different regions. As a result, the 3D shape of input model can be better preserved on the knitting result (see the bottom row of Fig. 9).

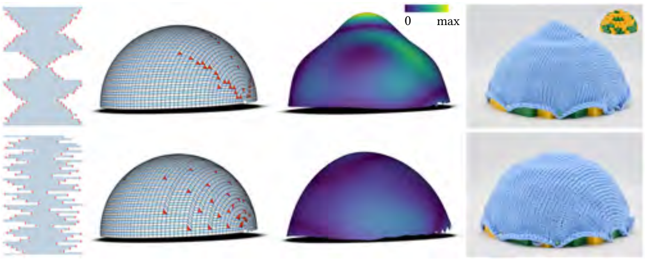


Fig. 9. Knitted hemisphere. (Left-to-right) Knitting map, stitch mesh, 3D shape of knitting result simulated by FEA, and the physical result of knitting. (Top row) The knitted hemisphere is distorted when the apexes (red elements) are crowded together. The color map visualizes the shape approximation error between the simulation result and the input 3D surface in terms of the hemisphere's radius r (max = $0.15r$). (Bottom row) After applying the diffusion term onto the apexes, the global shape error is significantly reduced. Note that the geometry of support for physical results is shown at the top-right corner. Stitch size used in physical knitting is 3.2mm × 2.2mm.

4.4 Knitting Map Generation

The knitting map can be extracted from a stitch mesh assigning each quad / triangle with its corresponding wale and course indices. The wale index of an element can be easily obtained from the indices of wale curves, which have been sorted. Now the problem is how to assign the course index for each element on a stitch mesh.

Considering elements adjacent to the same wale curve, two elements belong to the same course if and only if they share an edge on the wale curve. We determine elements in the same course by a flooding algorithm. After that, the course indices are determined by 1) converting neighboring courses into connected nodes on a graph and 2) applying a topological sorting algorithm on the graph to determine the order of courses in knitting (see Appendix B).

5 KNITTING FOR DISTRIBUTED ELASTICITY

This section presents how SJJ patterns can be realized on the stitch meshes. A method to compensate for the variation of stitch sizes is first introduced. After that, we present a tiling algorithm that minimizes the discontinuity while assigning SJJ patterns.

5.1 Stitch-Size Compensation

As analyzed in Section 2, stitch size is not uniform on a jacquard fabric. The variation of stitch sizes is not negligible if two SJJ patterns with different elasticity are knitted on the same fabric (as illustrated in Fig. 10). Similar to the strategy of homogenization applied for evaluating the elasticity on different SJJ patterns, we measure their average stitch sizes on the physical specimens. The size variation is compensated on the input triangular mesh $\mathcal{M}$ by computing an inversely deformed shape $\tilde{\mathcal{M}}$. The basic idea is that a region should be enlarged when it is assigned with a pattern having stitch-size smaller than the standard size $s_w \times s_h$. Inversely, the surface area should be reduced if the assigned pattern's stitch-size is larger than the standard size. In our implementation, the standard size is obtained by measuring the average of stitches on a knitted fabric using completely firm material. Note that, the stitch size variation of

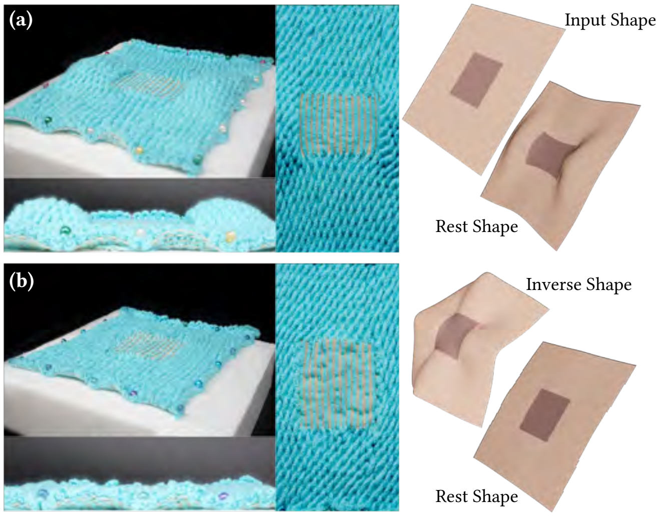


Fig. 10. Caused by the different average stitch sizes presented on the knitted regions with different elasticity, unwanted wrinkles can be observed on both the physical knitting result and the rest shape obtained from simulation (a). The problem can be solved by computing an inverse shape to generate the stitch mesh and the knitting map – see (b) for the improved physical and simulation results.

SJJ patterns is orthotropic, i.e., it could be larger than the standard size along one direction but smaller along the other direction.

By assigning the required elasticity (i.e., the corresponding SJJ pattern) to each triangle $f \in \mathcal{M}$, f's shape on the inverse geometry $\tilde{\mathcal{M}}$ can be determined as $\tilde{f}$. Specifically, a local frame of course and wale directions can be established at the center of f according to the geodesic distance-field $F(\cdot)$. When the average stitch-size of SJJ pattern assigned to f is $\tilde{s}_w \times \tilde{s}_h$, the scaling ratios $s_w/\tilde{s}_w$ and $s_h/\tilde{s}_h$ are applied along the course and the wale directions respectively to obtain $\tilde{f}$. A global shape of $\tilde{\mathcal{M}}$ can be obtained by blending these locally scaled triangles and following the iteration routine proposed by Bouaziz et al. [2012]. The source curve of geodesic distance-field is obtained on $\tilde{\mathcal{M}}$ by keeping the barycentric coordinates of its vertices on $\mathcal{M}$. Then, an updated geodesic distance-field $\tilde{F}(\cdot)$ can be computed on $\tilde{\mathcal{M}}$.

5.2 Tiling

The designed SJJ patterns are tiled on the stitch mesh to realize the designed distribution of elasticity in physical fabrication. First, the stitch mesh is segmented into different regions according to the assigned patterns. After that, each region with the same SJJ pattern is tiled independently.

To tile a SJJ pattern with dimension $m \times n$ onto a region of a stitch mesh is to determine a pattern coordinate $(\bar{i}, \bar{j})$ for each stitch $F_{i,j}$ in the region, where (i, j) denotes the course and the wale indices for a stitch. A proper tiling of a local neighborhood on the stitch mesh should be identical with neighborhood on the SJJ pattern. For two stitches $F_{i,j}$ and $F_{p,q}$ that are edge-neighbor to each other, the difference between their course-wale indices should be consistent with their pattern coordinates (by considering the winding). Specifically, the following conditions should be satisfied:

$$p - i \equiv \bar{p} - \bar{i} \pmod{m}, \qquad q - j \equiv \bar{q} - \bar{j} \pmod{n}.$$

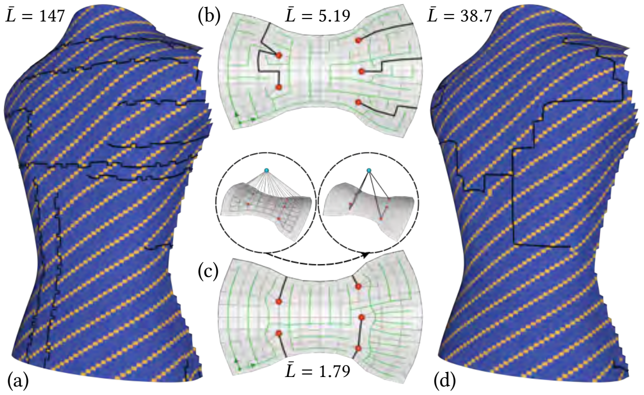


Fig. 11. Tiling the designed SJJ patterns onto a stitch mesh. Tiling by direct flooding ((a) & (b)), where the order of flooding has been illustrated as green paths shown on the patch in (b). The bold black curves illustrate the boundaries of neighboring stitches with incompatible SJJ patterns – discontinuity. To reduce the length of incompatible curves, we compute them by constructing a *minimal spanning tree* (MST) to connect all apexes (shown as red points in (b) & (c)) and use all boundary nodes as a common virtual node (see the blue point shown inside the circles of (c)). As a result, the discontinuity can be significantly reduced (see the tiling result shown in (d)). The SJJ pattern shown in Fig. 4 is employed in (a) and (d) to demonstrate the discontinuity. The total length of incompatible boundary curves $\bar{L}$ is also given for every model.

When these are not satisfied, the pattern coordinates assigned to $F_{i,j}$ and $F_{p,q}$ are considered as *incompatible*. All the boundaries of incompatible neighbors (see the bold black curves shown in Fig. 11) form a seam of tiling, where the discontinuity of patterns will lead to errors in elasticity and thus needs to be minimized. Source of the tiling incompatibility is the irregular vertices at the apexes, where the number of adjacent elements is not four. Moreover, the incompatibility at one apex will also propagate to 1) the boundary of a patch or 2) the other apexes (see the black curves shown in Figs. 11(a) and (b). Therefore, it cannot be entirely eliminated when apexes are present in a stitch mesh.

A flooding algorithm can be employed to assign the pattern coordinate of each stitch progressively by incorporating the above condition of compatibility. The order of flooding can be either *breadth-first search* (BFS) or any other orders that ensures to visit every stitch once. However, randomly determined order (and also BFS) can generate long seams (see Fig. 11(a) for an example).

Study shows that the incompatibility is caused by apexes and propagated along some seams into other apexes or the surface boundary. To reduce the incompatibility on a stitch mesh, we introduce an algorithm based on *minimal spanning tree* (MST) to minimize the length of seams. First of all, a graph Υ is constructed by considering every apex as a node. All boundary vertices of the region to tile are considered as a common virtual node in Υ. An edge between two nodes on the graph Υ is constructed by computing the shortest path between them that travels along the edges of the stitch mesh. The length of this path is employed as the weight of the graph edge. The MST computed on Υ gives a tree of connected paths with minimal total length. When applying the flooding algorithm while avoiding crossing the paths determined by the MST tree, the total length of seams is minimized (see Figs. 11(c) and (d)).

6 RESULTS AND DISCUSSIONS

6.1 Hardware and Implementation Details

We have tested our computational pipeline to design and fabricate a variety of examples with controlled shape and elasticity. All examples presented in this section are knitted on a fully computerized Shima Seiki SVR093SPSV machine, which is a widely used industrial level machine having two needle beds with 14 needles per inch. Moreover, the knitting maps generated by our method are also used on a Brother KH868 household machine to fabricate knitwear in a semi-automatic manner (e.g., the triple-peaks in Fig. 7 and the hemisphere in Fig. 9). Two different yarns – 40S cotton yarn (firm) and 75D rubber yarn (soft) are employed to realize different elasticity by SJJ patterns on the Shima Seiki machine. We generated the SJJ patterns manually considering the maximally allowed float length – that is less than 12 stitches on the Shima Seiki machine we used. We tested 13 patterns and selected the A-B-C patterns in Fig. 4 to give maximal, medium and minimal elongation in the course direction. When all stitches are knitted by 40S cotton yarn, we obtain the standard stitch size as s_w = 1.25mm and s_h = 1.00mm. The average stitch sizes of different SJJ patterns can be found in Fig. 4.

When designing the distribution of elasticity, the cloth simulator ARCSim [Narain et al. 2013b] that supports anisotropic materials, is employed to evaluate stresses and displacements during the whole sequence of body motion. We modify their code to permit assignment of different material properties at the triangle level, which enables the simulation of clothes with distributed elasticity. Accurate friction is difficult to model since aside from contact forces, it is also affected by skin roughness, local curvature, etc., and may vary in different regions for the same person. For example, elbows may produce higher friction than upper arms. We used the Coulomb friction model [Bridson et al. 2002] (provided within ARCSim) to approximate the friction between cloth and human body. Specifically, we adopted a ball draping experiment to calibrate friction, thickness, density, bending and stretching. The friction coefficient remains unchanged during simulation. The homogenized material properties of different SJJ patterns can be obtained by the data-driven method presented in [Wang et al. 2011].

The 3D models of mannequins are all reconstructed by a structured light based system [Song et al. 2013]. Motion data of mannequins, i.e., the long-sleeve example in Fig. 1 and the sock example in Fig. 16, are captured by using markers for registration [Loper et al. 2014]. The legging example in Fig. 13 is extracted from the deformable human body in SMPL [Loper et al. 2015].

6.2 3D Shaping

We tested the performance of the short-row based 3D shaping technique proposed in our pipeline. When generating knitting maps for production, a common industrial approach is to flatten 3D surfaces into 2D panels and then generate grids of stitches on the 2D panels.

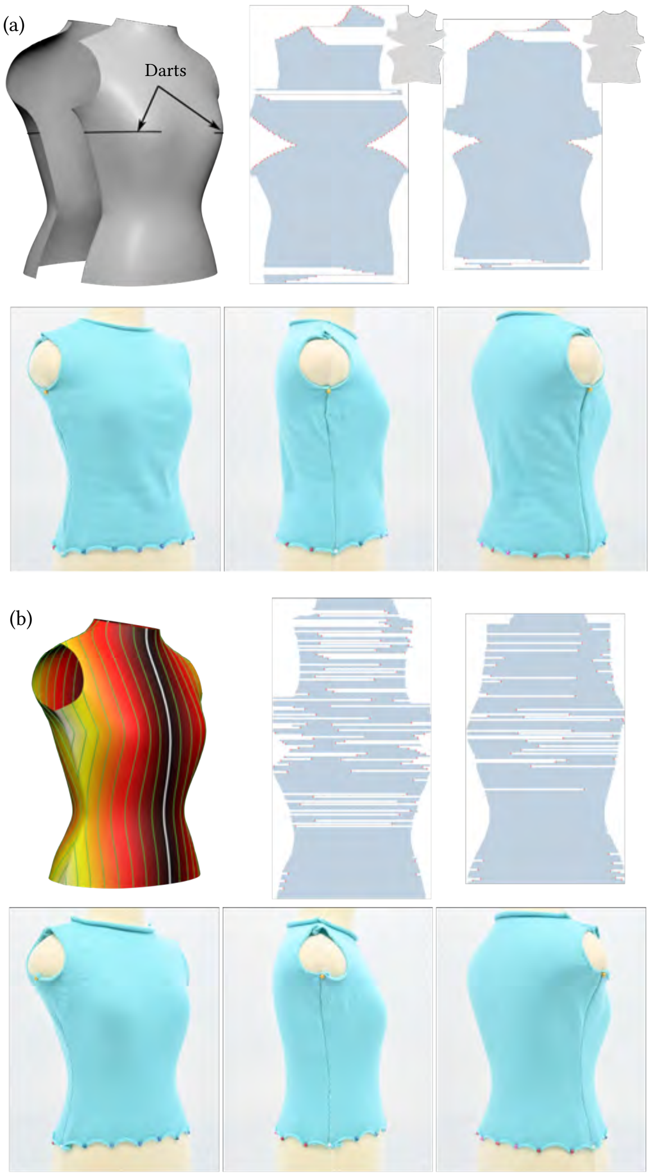


Fig. 12. Comparison of knitting maps generated from the flattened 2D panels with manually added darts (a) vs. from the 3D surface with the help of geodesic distance-field (b). The same set of pin holds are used for the two tests. The physical knitting results show the advantage of our approach with improved fit of the final garment. Note that the 2D panel results are generated by [Liu et al. 2008] without applying any scaling. Although scaling down the 2D panels can improve the fit at the waist, it will also make other parts of the garment too tight.

To flatten regions with high curvature, darts are added manually during the flattening step (see Fig. 12(a)). Different from clothes made by sewing, darts on knitwear will automatically be knitted together. However, the surface flattening process still introduces shape distortion errors. The result obtained from our short-row based 3D shaping technique shows significant improvement in fit – see Fig. 12(b), with the most pronounced difference in the waist

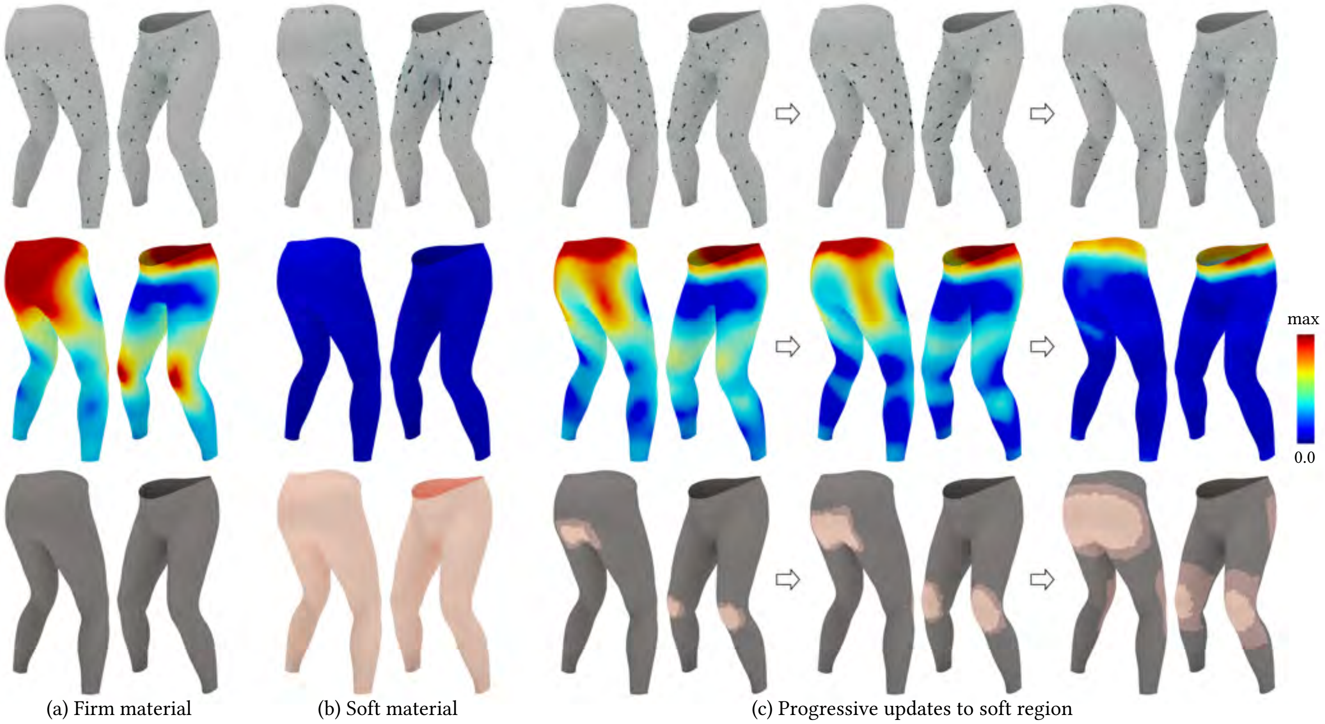


Fig. 13. Designing leggings with elasticity controlled for squat motion: (a) result by using firm material – with large stresses but small displacements, (b) the result by only using soft material – with small stresses but large displacements, and (c) progressive results by enlarging the soft region to reduce stresses while keeping small displacements. Stresses are displayed as a color map in all results (max = 3.9464kPa), and the displacements are visualized as a vector field in which the magnitudes are rendered as arrows with different lengths. $\bar{d}$ = 2.2cm is employed as the terminal threshold of inverse design.

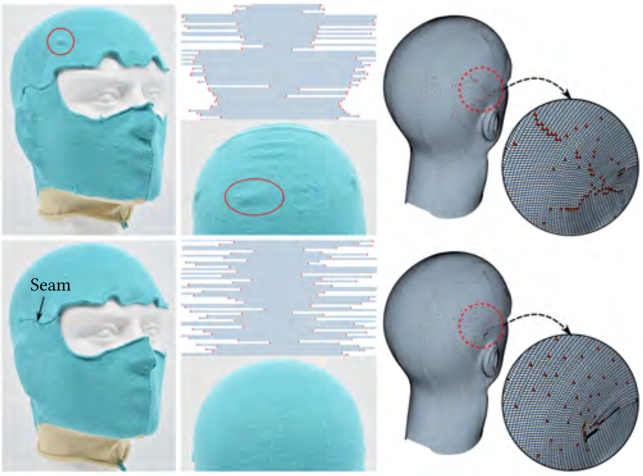


Fig. 14. Knitted skull cap. (Top) Without apex diffusion: bumps form when the shape approximation errors at apexes are accumulated together (see red circles). (Bottom) With apex diffusion: the bumps can be completely eliminated. Accessory seams are added from the top of the ear to canthus to open the closed-loops formed on isocurves of the geodesic distance-fields.

region. When the central front / back lines are selected as the source to compute the geodesic distance-field $F(\cdot)$, the stitches generated by our approach are nearly symmetric.

As demonstrated in Fig. 9, we control the distortion of knitting a 3D surface by diffusing the apexes on the stitch mesh. We also verified our distortion control method on real garments like the skull cap shown in Fig. 14. The knitting map generated by short-row approach without apex diffusion produces many unwanted bumps (top row of Fig. 14), which can be completely eliminated by using the apex diffusion (bottom row of Fig. 14). A few pre-processing steps are applied to generate the stitch mesh and the knitting map for the skull cap. First, the topology of the original design with eye hole is not homeomorphic to a disk. It is unwrapped by adding a seam from the nose to neck. In both cases, the intersection curve between the model and its symmetrical plane is used as the geodesic source. Closed-loops are found on the isocurves generated from this geodesic source; therefore, the method presented in Section 4.1 is applied to open closed-loops by adding new seams (see the ones from ear to canthus). Moreover, to make it easier to take on / off, a highly elastic yarn is employed to knit the neck part.

6.3 Knitwear as 4D garment

We present results for our proposed method on controlled distribution of elasticity to produce 4D garments. We first apply the inverse design algorithm presented in Section 3.1 on an example of leggings. As shown in Fig. 13, when fabric with a single material is applied, the motion of a half squat either 1) results in large stresses in the hip and knee regions when using firm material (see Fig. 13(a)) – causing

Table 1. Computational statistics of our computational pipeline of knitting 4D garments with controlled elasticity.

			Time (sec.) of Inverse Design			Time (sec.) of Knitting Enabler			Result # of
Model	Fig.	# Tri.	Per 4D Sim.†	Inv. Geom.	Material Ort.‡	Stitch Mesh	Tiling	Knitting Map	Stitches
Long-sleeve	1, 15	23,465	1567.31 (24 poses)	4.04	8.24	576.59	34.47	116.48	817,792
Leggings	13, 18	16,247	631.15 (72 poses)	3.84	9.68	1225.14	180.94	246.18	1,805,756
Sock	16, 18	13,883	327.23 (29 poses)	0.72	3.97	276.23	31.72	22.15	166,646
One-shoulder	17, 18	15,416	44.46 (static§)	0.94	3.44	116.47	37.85	40.25	204.866

† The time is reported for each simulation cycle with body motion, which is demonstrated in the supplementary video of our paper. ‡ This time includes the steps of generating geodesic distance-field and assigning material orientations. § No motion is involved in the one-shoulder top example.

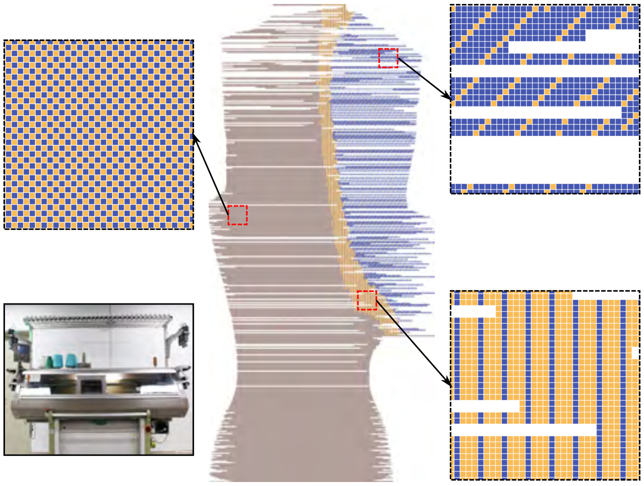


Fig. 15. The knitting map for the back piece of the 4D garment shown in Fig. 1. The zoomed-in views show the tiled SJJ patterns in different regions. The knitting machine used in physical fabrication is also shown.

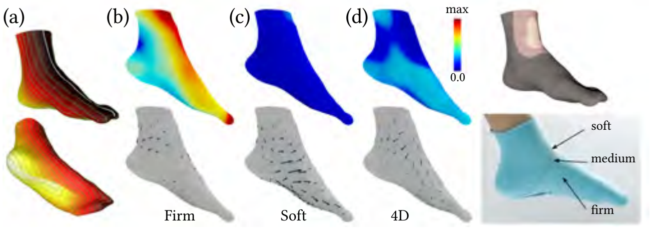


Fig. 16. Sock with distributed elasticity: (a) geodesic distance-field for generating the stitch mesh, (b) simulation of 3D knitting with firm material, (c) simulation of 3D knitting with soft material, (d) resulting 4D design with simulation and physical experiments. Both the stresses and the displacements have been significantly reduced. The maximal stress in this example is 0.6602kPa. The try-on process of this 4D sock can be found in the supplementary video. In this example, $\bar{d} = 0.8$cm is employed as the terminal threshold of inverse design.

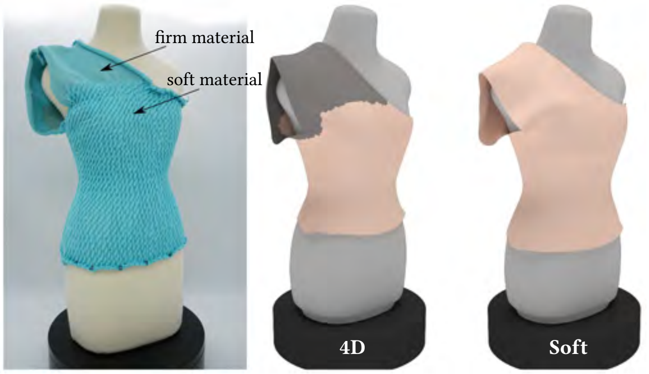


Fig. 17. One-shoulder knitted top with prescribed material distribution to achieve a desired shape. Firm material on the shoulder produces a stiff style, while soft material on the torso ensures comfort. From the simulation, it can be found that the style cannot be achieved using a uniformly soft material.

discomfort in these regions; or 2) generates large sliding in the thigh region when applying uniformly soft material (see Fig. 13(b)) – thus forming unwanted wrinkles by friction. These simulation results are very close to our daily experience. Starting from the region with large stress, our inverse design algorithm progressively enlarges the area of soft material while controlling the maximally allowed sliding (displacements) during the body motion. The progressive results of our algorithm are shown in Fig. 13(c).

Similar progressive results to generate 4D garments are shown for a long sleeve garment (see Figs. 1 and 5). The deformation caused by lifting the right arm is propagated from the arm to a large area on the back when single soft material is used for the whole garment. After using different elasticity in different regions, the deformation decays within a smaller region. Using soft material in the shoulder region also helps to reduce the stress and therefore makes the garment more comfortable. The knitting map with tiled SJJ patterns for this garment is given in Fig. 15.

Fig. 16 shows a 4D sock design with distributed soft, medium and firm elasticity achieved by SJJ patterns. When comparing with the 3D designs using a single elasticity, both the stresses and the amounts of sliding (in terms of displacements) have been significantly reduced. Note that, although the sock shape is homeomorphic to a disk, we still add a cutting line at the back to make it easier to knit. In industrial practice, socks are usually knitted by courses in loops. However, it is more challenging to knit in this way on a machine with two needle-beds when SJJ patterns are applied.

Fig. 17 shows a one-shoulder top design. The distribution of elasticity is specified by a designer, who adopts firm material in the shoulder region for styling purposes. Our pipeline provides a useful tool for designers to enable their ideas, achieving both the desired shape and the desired comfort. See the simulation and physical

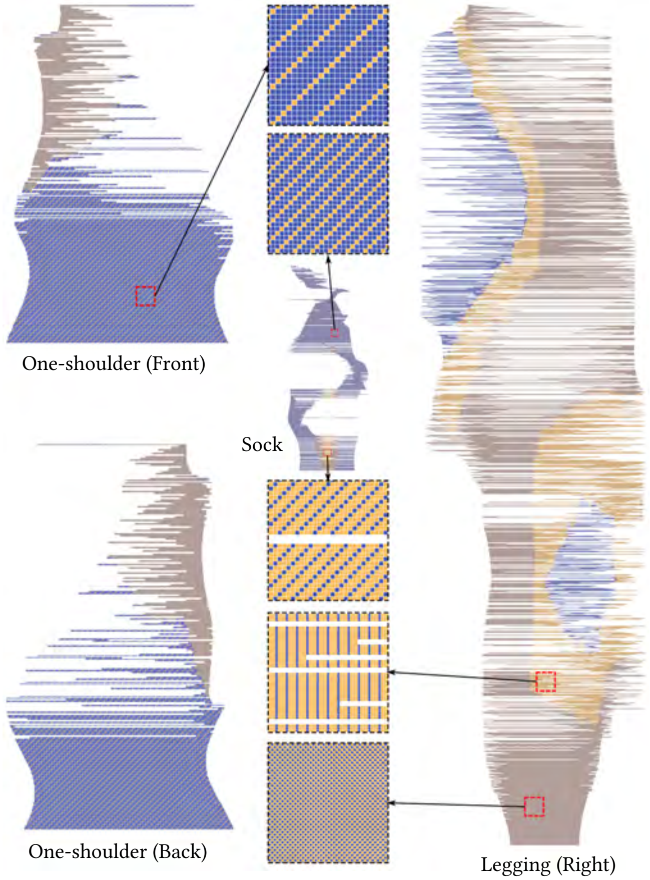
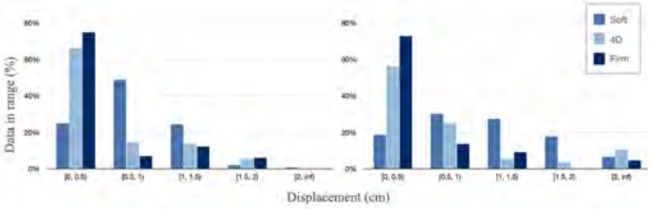


Fig. 18. Knitting maps of 4D examples: leggings, sock and one-shoulder top. For the symmetric pair of leggings, only the right part is given here.



(a) Simulation (b) Physical Experiment

Fig. 19. Statistical analysis for displacements on the long-sleeve example in Fig. 1. It can be observed that displacements on the 4D design are successfully reduced compared to the garment using soft material. Results obtained from (b) physical experiment can be successfully predicted by (a) simulation in our framework.

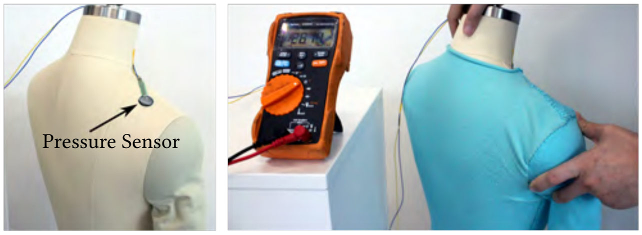


Fig. 20. A pressure sensor is used to measure pressure at the shoulder point on the mannequin. Pressures can be calibrated from the readings of a universal meter (middle) that measures electric resistance.

fabrication results in Fig. 17. Knitting maps for all 4D examples can be found in Fig. 18.

Table 1 lists statistics for each step of our computational pipeline for 4D knitted garments. The inverse design is implemented in C++, including physical simulation with ARCSim [Narain et al. 2013b]. The rest of the pipeline is implemented in Python. All computational experiments reported in Table 1 are obtained on a desktop PC with an Intel® Core™ i7-9700 CPU (8 cores @ 3.00GHz) + 32GB RAM, running Linux. Note that even after applying homogenization, i.e., not taking at the yarn level, the simulation is still very time-consuming as there are many poses to be processed in body motion.

6.4 Physical verification

We conducted physical experiments to verify the performance of simulation and optimization conducted in our work. We directly extracted displacements at points on the knitwear. For the long-sleeve garment shown in Fig. 1, the statistical measurements of the simulated displacements on the 3D design using soft material and the 4D design are

- Maximum: 2.84cm (soft 3D), 1.38cm (4D)
- Percentage (<0.5cm): 24.7% (soft 3D), 66.2% (4D)

where both the maximal displacement and the percentage of nearly static points (i.e., less than 0.5cm) are reported. During physical experiment, we obtain displacement-measurements on 40 visually traceable points from captured videos as follows.

- Maximum: 2.56cm (soft 3D), 2.30cm (4D)
- Percentage (<0.5cm): 18.7% (soft 3D), 56.1% (4D)

The significantly reduced displacements of our 4D design on these points verify the effectiveness of our approach. All these data are obtained from histograms of displacement-distribution as shown in Fig. 19.

We also conducted a physical test to verify the effectiveness of our approach in comfort improvement. Measuring stresses directly on knitwear is difficult; instead, we measured pressure at specific points on a mannequin. As shown in Fig. 20, we attach a FSR 402 pressure sensor at the shoulder point for this test. By using the method in [Montes et al. 2020], pressures at this shoulder point can be converted from stresses generated by our simulation. The simulated pressures are compared with the results obtained from physical experiment as follows.

- Physical Experiment: 50kPa (firm), 8kPa (soft), 12kPa (4D)
- Simulation: 40.75kPa (firm), 6.76kPa (soft), 9.25kPa (4D)

Our simulation successfully predicts the trend of physical measurement's change, and therefore provides a valid reference for material distribution design.

6.5 Discussion

The main limitation of our approach is the simplification of conducting physical simulation at the triangular mesh level instead of the yarn level (i.e., homogenization is applied). As a consequence, we cannot optimize the distribution of elasticity at the yarn level. The distribution is also updated by heuristically assigning soft material at regions with large stresses instead of sensitivity analysis, which relies on fast and accurate computation of physical models. Moreover, the orientation of orthotropic materials used in simulation is assigned according to the geodesic distance-field $F(\cdot)$. After assigning different elasticity and computing the inverse geometry, the geodesic distance-field will be changed. Therefore, the orientations of triangles in the material space can also change. In short, the fabric orientation in simulation and the varied inverse geometry caused by different distribution of elasticity are coupled. This is the other reason why we can only use the simulator to 'validate' a distribution of elasticity rather than conduct an optimization based on sensitivity analysis. It will be interesting to explore the technique of homogenized yarn-level simulation [Sperl et al. 2020] in future work.

In the graph-based stitch mesh generation between wale curves, there may be cases where no path exists on Γ connecting the source and target nodes. For example, Γ may be split into disconnected graphs when using a very conservative threshold λ. In these scenarios we incrementally enlarge λ by 10% until a path can be found. But note that enlarging λ may lead to significantly stretched yarns during knitting which are at risk of breaking.

In extreme cases of complex shapes (e.g., Fig. 7), topology with closed loops may be formed by the isocurves of the geodesic distance-field (although this was not found in our garment examples). Additional seams will need to be introduced, which may not be acceptable in specific applications. Models must also have disk topology, potentially requiring conversion using additional seams or by the method of [Narayanan et al. 2018] using segmentation. In this work, we propose a method to generate machine knitting code by only using the short-row shaping strategy, which is more efficient as demonstrated in Fig. 3. Moreover, the barrier of short-row shaping is much lower while the increase / decrease shaping strategy requires knitting machines equipped with more needle beds and more complicated mechanical structures. In future work, opportunities could be explored in distortion-minimized knitting for complicated surfaces by using increase / decrease shaping together with short-row shaping. We plan to study this alternative solution on high-end knitting machines.

When controlling the shape approximation error of a 3D knitted fabric, we introduced a heuristic to generate scattered apexes. Possible future work is to incorporate physical simulation in the loop of stitch mesh generation so that the locations of apexes can be optimized to further reduce the shape approximation error. Moreover, a simple friction model was employed in our simulation cycle. We plan to explore whether more precise prediction of sliding is possible when sophisticated friction models such as ARGUS [Li et al. 2018a] are used.

Constrained by float length, the maximal tile size used in our framework is 12 × 12. Within this range, the tile sizes are allowed to change freely in our pipeline as the variation is very small compared to the dimension of a full-size garment. Strong deformation is not observed at the discontinuous boundary of tiles. However, it has significant influence on the appearance and increases the chance of generating floats longer than the maximally allowed length.

7 CONCLUSION

We present a computational pipeline to enable the design and fabrication of 4D garments as knitwear with elasticity controlled for body motion. A distribution of elasticity that can be realized by SJJ patterns using two yarns is computed by a physics-driven method. The optimized elasticity on the garment is converted into a knitting map to be executed on digital knitting machines. We have developed new algorithms for this computational pipeline, including distortion-controlled 3D knitting by a short-row shaping technique, tiling to generate SJJ patterns with higher continuity, and shape compensation for incompatible stitch sizes. We verified the performance of our approach on a variety of examples in both simulation and physical experiments. The results are encouraging: both the large stresses that cause pressure and the large sliding that can lead to discomfort are significantly reduced on the 4D garments produced by our pipeline.

ACKNOWLEDGMENTS

The authors would like to thank Alexandre Kung for his professional comments on garment engineering and the anonymous reviewers for their valuable comments. The authors also thank Zhan Song and his group for their help on 3D scanning. This project is financially supported by the Centre for Perceptual and Interactive Intelligence (CPII) Limited at Hong Kong, the National Science Foundation under Grant No. #2047342, the Royal Society (Grant No. IES\R3\170164) and the Alfred P. Sloan Foundation: Sloan Research Fellowship. Zishun Liu is a PhD student supervised by E.L. Doubrovski and C.C.L. Wang at TU Delft, and he is also partially supported by the China Scholarship Council.

REFERENCES

Lea Albaugh, Scott Hudson, and Lining Yao. 2019. Digital Fabrication of Soft Actuated Objects by Machine Knitting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) *(CHI'19)*. ACM, New York, NY, USA, Article 184, 13 pages. https://doi.org/10.1145/3290605.3300414

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: Shape Completion and Animation of People. *ACM Trans. Graph.* 24, 3 (July 2005), 408–416. https://doi.org/10.1145/1073204.1073207

David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. ACM, New York, NY, USA, 43–54. https://doi.org/10.1145/280814.280821

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 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925896

Floraine Berthouzoz, Akash Garg, Danny M. Kaufman, Eitan Grinspun, and Maneesh Agrawala. 2013. Parsing Sewing Patterns into 3D Garments. *ACM Trans. Graph.* 32, 4, Article 85 (July 2013), 12 pages. https://doi.org/10.1145/2461912.2461975

Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. 2010. Design and Fabrication of Materials with Desired Deformation Behavior. *ACM Trans. Graph.* 29, 4, Article 63 (July 2010), 10 pages. https://doi.org/10.1145/1778765.1778800

Sofien Bouaziz, Mario Deuss, Yuliy Schwartzburg, Thibaut Weise, and Mark Pauly. 2012. Shape-Up: Shaping Discrete Geometry with Projections. *Computer Graphics Forum* 31, 5 (2012), 1657–1667. https://doi.org/10.1111/j.1467-8659.2012.03171.x

Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust Treatment of Collisions, Contact and Friction for Cloth Animation. *ACM Trans. Graph.* 21, 3 (July 2002), 594–603. https://doi.org/10.1145/566654.566623

Brother. 2013. KH970/KR850. https://youtu.be/HrQ2_pGFzvM

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

Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. 2014. An Asymptotic Numerical Method for Inverse Elastic Shape Design. *ACM Trans. Graph.* 33, 4, Article 95 (July 2014), 11 pages. https://doi.org/10.1145/2601097.2601189

Philippe Decaudin, Dan Julius, Jamie Wither, Laurence Boissieux, Alla Sheffer, and Marie-Paule Cani. 2006. Virtual Garments: A Fully Geometric Approach for Clothing Design. *Computer Graphics Forum* 25, 3 (2006), 625–634. https://doi.org/10.1111/j.1467-8659.2006.00982.x

Henry Fuchs, Zvi Meir Kedem, and Samuel Parker Uselton. 1977. Optimal Surface Reconstruction from Planar Contours. *Commun. ACM* 20, 10 (Oct. 1977), 693–702. https://doi.org/10.1145/359842.359846

Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. 2007. Efficient Simulation of Inextensible Cloth. *ACM Trans. Graph.* 26, 3 (July 2007), 49–es. https://doi.org/10.1145/1276377.1276438

Ruslan Guseinov, Eder Miguel, and Bernd Bickel. 2017. CurveUps: Shaping Objects from Flat Plates with Tension-Actuated Curvature. *ACM Trans. Graph.* 36, 4, Article 64 (July 2017), 12 pages. https://doi.org/10.1145/3072959.3073709

Yuki Igarashi, Takeo Igarashi, and Hiromasa Suzuki. 2008. Knitting a 3D Model. *Computer Graphics Forum* 27 (Oct. 2008), 1737–1743. Issue 7. https://doi.org/10.1111/j.1467-8659.2008.01318.x

Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2008. Simulating Knitted Cloth at the Yarn Level. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 1–9. https://doi.org/10.1145/1360612.1360664

Alexandre Kaspar, Liane Makatura, and Wojciech Matusik. 2019. Knitting Skeletons: A Computer-Aided Design Tool for Shaping and Patterning of Knitted Garments. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. ACM, New York, NY, USA, 53–65. https://doi.org/10.1145/3332165.3347879

Chelsea E. Knittel, Diana S. Nicholas, Reva M. Street, Caroline L. Schauer, and Genevieve Dion. 2015. Self-Folding Textiles through Manipulation of Knit Stitch Architecture. *Fibers* 3, 4 (2015), 575–587. https://doi.org/10.3390/fib3040575

Tsz-Ho Kwok, Yan-Qiu Zhang, Charlie C. L. Wang, Yong-Jin Liu, and Kai Tang. 2016. Styling Evolution for Tight-Fitting Garments. *IEEE Trans. on Vis. and Comp. Graph.* 22, 5 (2016), 1580–1591. https://doi.org/10.1109/TVCG.2015.2446472

François Ladevèze, Alfred Darroux, and Jacques Darroux. 1860. *Méthode de coupe, pour hommes: systém Vauclair-Darroux (1ére èdition)*. Académie Internationale de Coupe de Paris.

Jonathan Leaf, Rundong Wu, Eston Schweickart, Doug L. James, and Steve Marschner. 2018. Interactive Design of Periodic Yarn-Level Cloth Patterns. *ACM Trans. Graph.* 37, 6, Article 202 (Dec. 2018), 15 pages. https://doi.org/10.1145/3272127.3275105

Jie Li, Gilles Daviet, Rahul Narain, Florence Bertails-Descoubes, Matthew Overby, George E. Brown, and Laurence Boissieux. 2018a. An Implicit Frictional Contact Solver for Adaptive Cloth Simulation. *ACM Trans. Graph.* 37, 4, Article 52 (July 2018), 15 pages. https://doi.org/10.1145/3197517.3201308

Minchen Li, Alla Sheffer, Eitan Grinspun, and Nicholas Vining. 2018b. Foldsketch: Enriching Garments with Physically Reproducible Folds. *ACM Trans. Graph.* 37, 4, Article 133 (July 2018), 13 pages. https://doi.org/10.1145/3197517.3201310

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.

Matthew Loper, Naureen Mahmood, and Michael J. Black. 2014. MoSh: Motion and Shape Capture from Sparse Markers. *ACM Trans. Graph.* 33, 6, Article 220 (Nov. 2014), 13 pages. https://doi.org/10.1145/2661229.2661273

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, Article 248 (Oct. 2015), 16 pages. https://doi.org/10.1145/2816795.2818013

Mickaël Ly, Romain Casati, Florence Bertails-Descoubes, Mélina Skouras, and Laurence Boissieux. 2018. Inverse Elastic Shell Design with Contact and Friction. *ACM Trans. Graph.* 37, 6, Article 201 (Dec. 2018), 16 pages. https://doi.org/10.1145/3272127.3275036

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 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925940

Yuwei Meng, Charlie C. L. Wang, and Xiaogang Jin. 2012. Flexible Shape Control for Automatic Resizing of Apparel Products. *Comput. Aided Des.* 44, 1 (Jan. 2012), 68–76. https://doi.org/10.1016/j.cad.2010.11.008

Jun Mitani and Hiromasa Suzuki. 2004. Making Papercraft Toys from Meshes Using Strip-Based Approximate Unfolding. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 259–263. https://doi.org/10.1145/1015706.1015711

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

Rahul Narain, Tobias Pfaff, and James F. O'Brien. 2013a. Folding and Crumpling Adaptive Sheets. *ACM Trans. Graph.* 32, 4, Article 51 (July 2013), 8 pages. https://doi.org/10.1145/2461912.2462010

Rahul Narain, Armin Samii, and James F. O'Brien. 2012. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Trans. Graph.* 31, 6, Article 152 (Nov. 2012), 10 pages. https://doi.org/10.1145/2366145.2366171

Rahul Narain, Armin Samii, Tobias Pfaff, and James F. O'Brien. 2013b. *ARCSim: Adaptive Refining and Coarsening Simulator.* UC Berkeley. Retrieved March 16, 2020 from http://graphics.berkeley.edu/resources/ARCSim/

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 James McCann. 2019. Visual Knitting Machine Programming. *ACM Trans. Graph.* 38, 4, Article 63 (July 2019), 13 pages. https://doi.org/10.1145/3306346.3322995

Olivier Nocent, Jean-Michel Nourrit, and Yannick Remion. 2001. Towards Mechanical Level of Detail for Knitwear Simulation. In *The 9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*. 252–259.

Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic Textures for Additive Fabrication. *ACM Trans. Graph.* 34, 4, Article 135 (July 2015), 12 pages. https://doi.org/10.1145/2766937

Jesús Pérez, Miguel A. Otaduy, and Bernhard Thomaszewski. 2017. Computational Design and Automated Fabrication of Kirchhoff-Plateau Surfaces. *ACM Trans. Graph.* 36, 4, Article 62 (July 2017), 12 pages. https://doi.org/10.1145/3072959.3073695

Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy. 2015. Design and Fabrication of Flexible Rod Meshes. *ACM Trans. Graph.* 34, 4, Article 138 (July 2015), 12 pages. https://doi.org/10.1145/2766998

Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. 2015. Microstructures to Control Elasticity in 3D Printing. *ACM Trans. Graph.* 34, 4, Article 136 (July 2015), 13 pages. https://doi.org/10.1145/2766926

Shima Seiki. 2019. Mach 2XS153. https://www.shimaseiki.com/

Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. 2013. Computational Design of Actuated Deformable Characters. *ACM Trans. Graph.* 32, 4, Article 82 (July 2013), 10 pages. https://doi.org/10.1145/2461912.2461979

Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. 2014. Designing Inflatable Structures. *ACM Trans. Graph.* 33, 4, Article 63 (July 2014), 10 pages. https://doi.org/10.1145/2601097.2601166

Zhan Song, Ronald Chung, and Xiao-Ting Zhang. 2013. An Accurate and Robust Strip-Edge-Based Structured Light Means for Shiny Surface Micromeasurement in 3-D. *IEEE Transactions on Industrial Electronics* 60, 3 (2013), 1023–1032. https://doi.org/10.1109/TIE.2012.2188875

David J. Spencer. 2001. *Knitting Technology: A Comprehensive Handbook and Practical Guide* (third edition ed.). Woodhead Publishing, Cambridge, England. https://doi.org/10.1016/B978-1-85573-333-6.50004-0

Georg Sperl, Rahul Narain, and Chris Wojtan. 2020. Homogenized Yarn-Level Cloth. *ACM Trans. Graph.* 39, 4, Article 48 (July 2020), 16 pages. https://doi.org/10.1145/3386569.3392412

Nobuyuki Umetani, Danny M. Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive Couture for Interactive Garment Modeling and Editing. In *ACM SIGGRAPH 2011 Papers* (Vancouver, British Columbia, Canada) *(SIGGRAPH '11)*. ACM, New York, NY, USA, Article 90, 12 pages. https://doi.org/10.1145/1964921.1964985

Pascal Volino, Martin Courchesne, and Nadia Magnenat Thalmann. 1995. Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. ACM, New York, NY, USA, 137–144. https://doi.org/10.1145/218380.218432

Charlie C. L. Wang, Kin-Chuen Hui, and Kai-Man Tong. 2007. Volume Parameterization for Design Automation of Customized Free-Form Products. *IEEE Trans. on Auto. Sci. and Eng.* 4, 1 (2007), 11–21. https://doi.org/10.1109/TASE.2006.872112

Charlie C. L. Wang and Kai Tang. 2005. Optimal Boundary Triangulations of an Interpolating Ruled Surface. *Journal of Computing and Information Science in Engineering* 5, 4 (02 2005), 291–301. https://doi.org/10.1115/1.2052850

Charlie C. L. Wang and Kai Tang. 2010. Pattern Computation for Compression Garment by a Physical/Geometric Approach. *Comput. Aided Des.* 42, 2 (Feb. 2010), 78–86. https://doi.org/10.1016/j.cad.2009.02.018

Charlie C L Wang, Yunbo Zhang, and Hoi Sheung. 2010. From Designing Products to Fabricating Them from Planar Materials. *IEEE Computer Graphics and Applications* 30, 06 (nov 2010), 74–85. https://doi.org/10.1109/MCG.2009.155

Huamin Wang, James F. O'Brien, and Ravi Ramamoorthi. 2011. Data-Driven Elastic Models for Cloth: Modeling and Measurement. *ACM Trans. Graph.* 30, 4, Article 71 (July 2011), 12 pages. https://doi.org/10.1145/2010324.1964966

Kui Wu, Xifeng Gao, Zachary Ferguson, Daniele Panozzo, and Cem Yuksel. 2018. Stitch Meshing. *ACM Trans. Graph.* 37, 4, Article 130 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201360

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

Hongyi Xu, Yijing Li, Yong Chen, and Jernej Barbič. 2015. Interactive Material Design Using Model Reduction. *ACM Trans. Graph.* 34, 2, Article 18 (March 2015), 14 pages. https://doi.org/10.1145/2699648

Cem Yuksel, Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2012. Stitch Meshes for Modeling Knitted Clothing with Yarn-Level Detail. *ACM Trans. Graph.* 31, 4, Article 37 (July 2012), 12 pages. https://doi.org/10.1145/2185520.2185533

Jonas Zehnder, Espen Knoop, Moritz Bächer, and Bernhard Thomaszewski. 2017. Metasilicone: Design and Fabrication of Composite Silicone with Desired Mechanical Properties. *ACM Trans. Graph.* 36, 6, Article 240 (Nov. 2017), 13 pages. https://doi.org/10.1145/3130800.3130881

Eugene Zhang, Konstantin Mischaikow, and Greg Turk. 2005. Feature-Based Surface Parameterization and Texture Mapping. *ACM Trans. Graph.* 24, 1 (Jan. 2005), 1–27. https://doi.org/10.1145/1037957.1037958

Xiaoting Zhang, Guoxin Fang, Melina Skouras, Gwenda Gieseler, Charlie C. L. Wang, and Emily Whiting. 2019. Computational Design of Fabric Formwork. *ACM Trans. Graph.* 38, 4, Article 109 (July 2019), 13 pages. https://doi.org/10.1145/3306346.3322988

Xiaoting Zhang, Xinyi Le, Zihao Wu, Emily Whiting, and Charlie C. L. Wang. 2016. Data-Driven Bending Elasticity Design by Shell Thickness. In *Proceedings of the Symposium on Geometry Processing* (Berlin, Germany) *(SGP '16)*. Eurographics Association, Goslar, DEU, 157–166.

APPENDIX A: POST-PROCESSING OF STITCH MESH

After generating the wale curves by a geodesic distance field and constructing the stitch mesh between wale curves by a graph-based algorithm, a knittable stitch mesh can be obtained. To make it perfect at the boundary, two more operators are introduced for post-processing the stitch mesh.

- The first one is to correct pairs of triangles with their apex-vertices located at the boundary of a patch (as illustrated in Fig. 21(a)). During physical knitting, regular stitches will be formed in these regions. To make the stitch mesh consistent with the knitting result, we replace triangles in these cases by quads.
- The second one is to correct the discontinuity between courses at the boundary of patch (as illustrated in Fig. 21(b)). If the end of a boundary course is not aligned with the beginning of the next course, we add a few support stitches as accessory into the next course to avoid the long float.

Both operators are only applied to the boundary, which will not influence the interior global shape of a knitted patch.

APPENDIX B: TOPOLOGICAL SORTING OF COURSES

A topological sorting algorithm is employed in our implementation to determine the order of courses in knitting process by considering the between-rows continuity.

Two elements belong to the same course if and only if they share an edge on the wale curve. First, we determine elements in the same course by a flooding algorithm. After that, graph-based *topological sorting* is used to find the order of courses. A *directed acyclic graph* (DAG) can be constructed by defining each course as a node on the graph (see Fig. 22 for an example). Directed edges are defined between neighboring courses only when the between-row continuity is satisfied. For example, when knitting courses in the bottom-up order, the edge between c and e in Fig. 22 is invalid although they are neighboring to each other. Considering the knitting direction of c (right to left) and e (left to right) are different, the ending stitch of c does not connect to the starting stitch of e – i.e., violating between-row continuity. Similarly, the edges $e \longrightarrow h$ and $h \longrightarrow i$ are not added for the same reason. Building a graph in this way will automatically generate the order of courses and segment a stitch mesh into multiple regions on a knitting map where each region can be knitted by a continuous yarn. Therefore, three yarns are needed for the example shown in Fig. 22.

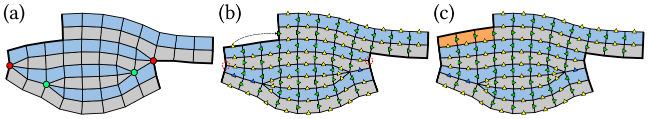


Fig. 21. Two operators of boundary correction: (a) A pair of triangles is replaced by quads if their apex vertex is located on the boundary (denoted by red dots and circles in dash lines), and the triangles will remain when their apex vertex is in the interior (denoted by green dots). (b) The discontinuity between courses is corrected by adding support stitches as quads (displayed in orange) in the course next to a boundary course.

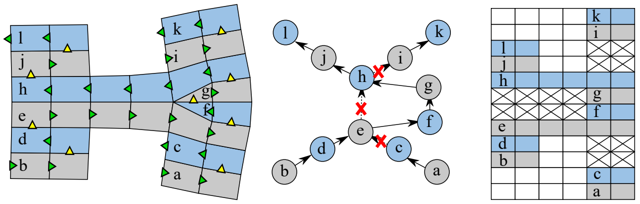


Fig. 22. Given a mesh with courses determined by flooding (left), each course is represented as a node on a directed graph (middle). The order of courses can be determined by the depth-first search on this graph. After sorting, the stitch mesh can be converted into a knitting map (right).

Note that, two post-processing operators introduced in Fig. 21 are applied to the boundary of a stitch mesh after the DAG-based course sorting. The second operator (in Fig. 21(b-c)) fixes the discontinuity only when

(1) there is no other course connected to the second course in the graph, and
(2) the first course has not been connected to any other course.

For example, it cannot be applied to connect h and i in Fig. 22 unless the course j does not exist (i.e., prevented by the first condition mentioned here). Similarly, the operator cannot be applied to connect c to e according to the second condition. In short, we never form any tree in the graph.