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

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; Yucheng Zhang, Centre for Perceptual and Interactive Intelligence (CPII) Limited, Hong Kong, China, yyczhang@cpii.hk; Xiangjia Chen, Centre for Perceptual and Interactive Intelligence (CPII) Limited, Hong Kong, China, xichen@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.

This is the author’s version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in ACM Transactions on Graphics.

ACM Trans. Graph., Vol. 0, No. 0, Article 0 (2021), 16 pages.

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 perfect-fit 3D garments for varied body shapes (e.g., [Brouet et al. 2012; Montes et al. 2020]). Physics-based optimizations have also

In this paper, we present a new computational pipeline for designing and fabricating 4D garments as knittedware 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 of SJJ patterns in different elastic levels by single-jersey jacquard (SJJ) with two yarns. We design the distribution of elasticity of 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.

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. 0, 0, Article 0 (2021), 16 pages.
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. 2018; 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 knit 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 SJ 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 SJ 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 stitches can be applied [Spencer 2001]. Other surface topologies can be converted to a disk-topology by adding cuts [Zhang et al. 2005].
This is analogous to adding darts when forming 3D non-developable surfaces by inextensible planar materials [Decaudin et al. 2006; Miani 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 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; Miani 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 changes the local height 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.

### 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.

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.

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

We tackle the problem by first computing the distribution of elasticity $P_c$ at the element level according to predefined SJJ patterns and then generating the knitting map $K$ that can realize $P_c$ 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 $M$. 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 $\hat{F}(-)$ of the base line is computed to determine the orientation of stitches by assigning 1) the weft direction as the gradient of $\hat{F}(-)$, $\nabla \hat{F}$, and 2) the wale direction as $\hat{n} \times \nabla F$ with $\hat{n}$ 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 $M$ 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 $(\sigma_{\text{max}}^s, d_{\text{max}}^s)$ and $(\sigma_{\text{max}}^f, d_{\text{max}}^f)$ 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 $\sigma_{\text{max}}$ and $d_{\text{max}}$.
5. Go back to (3) or stop the iteration when $d_{\text{max}} > d$.

#### 3.2 Enabling Machine Knitting

Given a distribution of elasticity $P_c$ assigned on $M$, we compute the knitting map $K$ in four steps to realize this 4D garment.

---

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 $\hat{F}(-)$ on a garment $M$ for assigning orientations for knitting stitches and orthotropic material simulation, (b) the maximal sliding as $d_{\text{max}}$, (c) knitting map $K$ for assigning the SJJ information to a knitting map, (d) SJJ patterns assigned on the stitch mesh and (e) the resultant knitting map. Note that, after computing the inverse geometry $\tilde{M}$ 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 $\tilde{F}(-)$. 

---

**References:**

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]. Computing stitch meshes on the rest shape \( M \) will then incorporate the pattern-specific shrinkage ratio on the rest shape \( M \) (Section 5.1). As the geodesic distance-field on \( M \) is changed to \( \hat{F}(\cdot) \), this step is also included in the routine of inverse design to re-assign the material orientation of each triangle.

(1) **Inverse geometry computation.** To compensate for the variation of shrinkage ratios among different SJJ patterns, an inverse geometry of \( M \) is computed as \( \hat{M} \) (see Fig. 5(e)).

(2) **Knittable stitch meshes** are computed on the inverse geometry \( \hat{M} \) by using the geodesic distance-field \( \hat{F}(\cdot) \) (see Fig. 5(e)). Quadrangles / triangles of a stitch mesh are constructed between \( \hat{F} \)'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.

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 \( 3.5 \times 2.8 \text{mm} \) in this example. (e) The physical knitting result.

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 \( M \) 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 \( M \) 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 \( M \). If the source is a curve, it should be a curve cutting \( M \) into two pieces or a curve completely laying on \( M \)'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 \( p \) containing the local maximum of \( F(\cdot) \). We can split the closed loop by computing a shortest path from \( p \) 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 \( n \times \nabla F \) with \( n \) being the surface normal (see \( d \) shown in Fig. 7(a)). Vertices of a stitch mesh are generated on wale curves by sampling them into segments with length \( s_k \). 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 Wale

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)).

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 of 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 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 \( \lambda \). Having a very long bridge will lead to stitch meshes with large distortion, we chose \( \lambda = s_w + 2s_k \) as a threshold in our implementation. Our graph-based method represents the anchor bridges as nodes on a graph \( \Gamma \) (see Fig. 8(e)).

![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 \( \Gamma \) shows all possible configurations to tessellate the two wale curves. Anchor bridges are represented as nodes of \( \Gamma \) and operators are denoted by edges of \( \Gamma \) (different colors denote different types). The problem of finding an optimized tessellation between two wale curves is converted to finding a shortest path on \( \Gamma \) (i.e., the bold arrows). (d) The resultant mesh tessellation corresponding to the shortest path on \( \Gamma \).](image)

We then generate a knittable stitch mesh by finding an optimal path that connects a selected set of nodes via operators.

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

- **AB-advance:** Two quadrangles are formed between \( a_{2i}b_{2j} \) and \( a_{2i+1}b_{2j+2} \) by adding two new bridges \( a_{2i+1}b_{2j} \) and \( a_{2i+2}b_{2j+1} \) (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)).
Each operator defines a directed edge in $\Gamma$ 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 $\lambda$, we exclude its corresponding edge from $\Gamma$.

Given a source node and a target node, an optimal mesh can be generated by computing a shortest path between them on the graph $\Gamma$ (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 $\Gamma$. 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).

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 $M$ by computing an inversely deformed shape $\tilde{M}$. 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
SJJ patterns are 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 SJ pattern) to each triangle \( f \in \mathcal{M} \), \( f \)'s shape on the inverse geometry \( \bar{\mathcal{M}} \) can be determined as \( \bar{f} \). 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 \( \bar{s}_c \times \bar{s}_w \), the scaling ratios \( s_c/\bar{s}_c \) and \( s_w/\bar{s}_w \) are applied along the course and the wale directions respectively to obtain \( \bar{f} \). A global shape of \( \bar{\mathcal{M}} \) 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 \( \bar{F}(\cdot) \) can be computed on \( \mathcal{M} \).

5.2 Tiling

The designed SJ 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 SJ pattern is tiled independently.

To tile a SJ pattern with dimension \( m \times n \) onto a region of a stitch mesh is to determine a pattern coordinate \( (i, j) \) for each stitch \( E_{i,j} \) 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 \( E_{i,j} \) and \( E_{p,q} \) 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:

\[
p - i \equiv \bar{p} - \bar{i} \pmod{m}, \quad q - j \equiv \bar{q} - \bar{j} \pmod{n}.
\]

When these are not satisfied, the pattern coordinates assigned to \( E_{i,j} \) and \( E_{p,q} \) 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 \( G \) 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 \( G \). An edge between two nodes on the graph \( G \) is constructed by computing the shortest path between them that travels along the edges of the stitch mesh. The

**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.

**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 SJ 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 SJ 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.

ACM Trans. Graph., Vol. 0, No. 0, Article 0. Publication date: 2021.
length of this path is employed as the weight of the graph edge. The MST computed on $\Upsilon$ 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 $s_w = 1.25$mm and $s_h = 1.00$mm. 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. 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 \((\max = 3.9464\text{kPa})\), and the displacements are visualized as a vector field in which the magnitudes are rendered as arrows with different lengths. \(\bar{d} = 2.2\text{cm}\) is employed as the terminal threshold of inverse design.

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.

<table>
<thead>
<tr>
<th>Model</th>
<th>Fig.</th>
<th># Tri.</th>
<th>Time (sec.) of Inverse Design</th>
<th>Time (sec.) of Knitting Enabler</th>
<th>Result # of Stitches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long-sleeve</td>
<td>1, 15</td>
<td>23,465</td>
<td>1567.31 (24 poses) 4.04 8.24</td>
<td>576.59 34.47 116.48</td>
<td>817,792</td>
</tr>
<tr>
<td>Leggings</td>
<td>13, 18</td>
<td>16,247</td>
<td>631.15 (72 poses) 3.84 9.68</td>
<td>1225.14 180.94 246.18</td>
<td>1,805,756</td>
</tr>
<tr>
<td>Sock</td>
<td>16, 18</td>
<td>13,883</td>
<td>327.23 (29 poses) 0.72 3.97</td>
<td>276.23 31.72 22.15</td>
<td>166,646</td>
</tr>
<tr>
<td>One-shoulder</td>
<td>17, 18</td>
<td>15,416</td>
<td>44.46 (static§) 0.94 3.44</td>
<td>116.47 37.85 40.25</td>
<td>204.866</td>
</tr>
</tbody>
</table>

† 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.

ACM Trans. Graph., Vol. 0, No. 0, Article 0. Publication date: 2021.
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.

---

ACM Trans. Graph., Vol. 0, No. 0, Article 0. Publication date: 2021.
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 \( \Gamma \) connecting the source and target nodes. For example, \( \Gamma \) may be split into disconnected graphs when using a very conservative threshold \( \lambda \). In these scenarios we incrementally enlarge \( \lambda \) by 10\% until a path can be found. But note that enlarging \( \lambda \) 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 introduce 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


ACM Trans. Graph., Vol. 0, No. 0, Article 0. Publication date: 2021
A topological sorting algorithm is employed in our implementation. After generating the wale curves by a geodesic distance field and constructing the directed acyclic graph (DAG) can be constructed by defining each course as a node on the graph. 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 \( e \to h \) and \( h \to i \) 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.

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.