DOI: 10.1111/cgf.15030
EUROGRAPHICS 2024 / A. Bermano and E. Kalogerakis
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 2

Computational Smocking through Fabric-Thread Interaction

Ningfeng Zhou   Jing Ren   Olga Sorkine-Hornung

ETH Zurich, Switzerland

Abstract

We formalize Italian smocking, an intricate embroidery technique that gathers flat fabric into pleats along meandering lines of stitches, resulting in pleats that fold and gather where the stitching veers. In contrast to English smocking, characterized by colorful stitches decorating uniformly shaped pleats, and Canadian smocking, which uses localized knots to form voluminous pleats, Italian smocking permits the fabric to move freely along the stitched threads following curved paths, resulting in complex and unpredictable pleats with highly diverse, irregular structures, achieved simply by pulling on the threads. We introduce a novel method for digital previewing of Italian smocking results, given the thread stitching path as input. Our method uses a coarse-grained mass-spring system to simulate the interaction between the threads and the fabric. This configuration guides the fine-level fabric deformation through an adaptation of the state-of-the-art simulator, C-IPC [LKJ21]. Our method models the general problem of fabric-thread interaction and can be readily adapted to preview Canadian smocking as well. We compare our results to baseline approaches and physical fabrications to demonstrate the accuracy of our method.

CCS Concepts

• ***Computing methodologies*** → ***Mesh geometry models;***

1. Introduction

Embroidery, one of the oldest forms of art, produces exquisite decorations through the interplay of fabric and threads, and continues to captivate researchers in visual computing [EMZ16, MS22, RSSH23, ZPH*23]. A recent paper [RSSH23] delves into Canadian smocking, an embroidery technique renowned for its intricate and voluminous pleats, whose geometric shape is difficult to predict by looking at the smocking pattern alone. During the fabrication of Canadian smocking, multiple points along the same stitching line are gathered and secured with a knot. To compute a digital preview of a smocked result, Ren et al. [RSSH23] cast Canadian smocking as a graph embedding problem, merging multiple graph nodes into a single one, as fabric thickness has a negligible effect in this technique. However, this method is not applicable to *Italian smocking*, a different traditional smocking technique, in which fabric thickness significantly influences pleat formation and cannot be disregarded.

In Italian smocking, the fabric is "drawn up into close pleats on *rows* of gauged stitching with deviations that create *patterned irregularities*" [Wol96]. An Italian smocking pattern is typically divided into rows (thread paths, not necessarily straight); each row is stitched with a separate thread. The thread alternates between passing in and out of the fabric, creating front stitches and back stitches. After completing the stitching of all rows, the threads are gently pulled on, so that the fabric is pushed and gathered, creating folded pleats. Finally, the thread ends are tied to secure the pleats in the desired shape. See Fig. 1 for an example and Fig. 2 for a demonstration of the fabrication process.

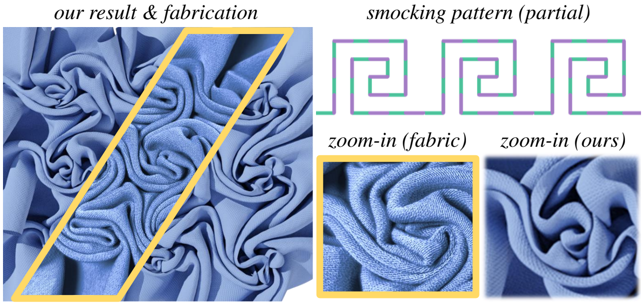

Figure 1: *Our simulated result and physical fabrication (highlighted in yellow) for an Italian smocking pattern.*

We can readily observe the distinctive characteristics of Italian smocking, setting it apart from English and Canadian smocking. In English smocking, the fabric is stitched in straight paths with regular spacing, yielding uniform pleats. The colorful rows of stitches running through these pleats are similar to standard 2D embroidery. In contrast, the stitching lines in Italian smocking do not follow a straight path. As the stitching line turns and curves, the fabric gathers and crinkles along the stitched thread, resulting in complex and unpredictable pleats with diverse, irregular structures, which play the main decorative role. Unlike the localized knots in Canadian smocking, which immobilize the stitching points and validate the assumption of negligible fabric thickness, Italian smocking per-

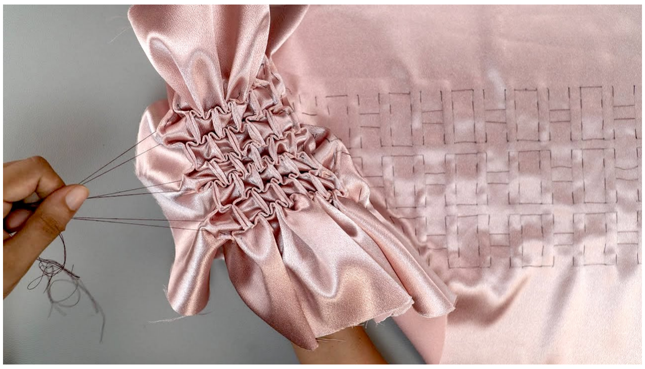

Figure 2: ***Italian smocking fabrication*** *[Han22]. A single thread is sewn through the fabric along one row of the pattern. Then, the free ends of the threads are pulled on, resulting in intricate pleat patterns. ©HandiWorks YouTube channel. Used with permission.*

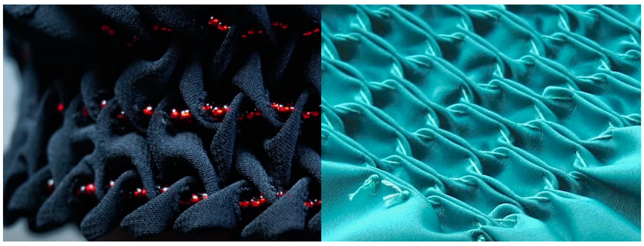

Figure 3: ***Loose threads****. An essential characteristic of Italian smocking is the deliberate avoidance of fully tightening the threads to achieve visually appealing pleats.* ***Left****: Beads are strategically employed to prevent the threads from being pulled taut [DIY18].* ***Right****: Visible threads provide structural support to the pleats and enhance the overall aesthetics in the final result [DIY17]. ©DIYstitching YouTube channel. Used with permission.*

mits the stitching points (i.e., the fabric) to move freely along the stitched thread, and the thread may not necessarily be pulled taut in the final fabrication (see Fig. 3 for some examples). We summarize the major differences between English, Canadian, and Italian smocking techniques in Table 1, and provide an illustrative comparison in Fig. 4.

In this work, we propose a novel method for previewing the result of Italian smocking given a stitching pattern by employing the general perspective of fabric-thread interaction. We model the fabric using a coarse-grained mass-spring system at a scale similar to the size of the front and back stitches on the fabric, as indicated in the input pattern. Subsequently, we simulate the dynamics of this coarse mass-spring system by incorporating adjustable stitching springs to model taut threads during the thread-pulling stage of Italian smocking fabrication. The resulting configuration of the mass-spring system is then used to guide the deformation of the fabric in a finer resolution by adapting C-IPC [LKJ21].

Contributions. (1) We introduce a formalization of Italian smocking pattern design and propose a simple method to preview the smocked results. (2) Our formulation is capable of handling continuous stitches on both the front and the back of the fabric, which can be used to facilitate free-form embroidery design. (3) We integrate non-zero sewing lengths and positional constraints into the state-of-the-art simulator C-IPC [LKJ21].

Table 1: *We compare three different types of smocking including English smocking, Canadian smocking, and Italian smocking.*

properties \ smocking type	English	Canadian	Italian
continuous, long *stitching paths?*	✔	✘	✔
localized *stitches and knots?*	✘	✔	✘
regular *spacing?*	✔	✘	✘
stitching paths ***pulled taut****?*	✔	✔	✘
decoration from ***colorful stitches****?*	✔	✘	✔
decoration from ***voluminous pleats****?*	✘	✔	✔

2. Related work

Fabric manipulation, such as embroidery, folding, and pleating, is a fascinating art form that poses challenging problems in computer graphics. These challenges include how to model the geometry of folds and how to efficiently and accurately simulate wrinkles. We therefore review the literature in these two aspects.

Modeling folds and pleats. Curves are popular for abstracting cloth folds. For example, Popa et al. [PZB*09] extract fold curves from input images to guide the 3D fabric deformation, achieving folds that closely match the reference. Jung et al. [JHR*15] propose a method to generate folds for garments or fashion accessories that are guaranteed to align with the curves drawn as silhouettes on a design sketch. The FoldSketch system [LSGV18] solves for an updated sewing pattern that aligns with user-specified curves, achieving expected folds through a combination of graph editing and cloth simulation. Zheng et al. [ZSS20] develop a system that allows users to generate folds on draped cloth by drawing curves and reconstructing the modified cloth surface for downstream simulation. Frequency-based models are investigated to simulate the fine-scale wrinkles and their evolution on coarse meshes [CCK*21, CKSV23]. Hermite radial basis functions can also be used to approximate the shape of wrinkles [MSYA21]. However, while curves and smooth basis functions have proven effective in modeling prominent folds, they are less suitable for capturing the intricate and structural folds found in embroidery, especially at a fine level of detail. In the special case of Canadian smocking, Kim [Kim20] accounts for the fold angle when performing step-by-step simulation manually set up using the Clo software [CLO23]. Ren et al. [RSSH23] propose using a coarse graph to abstract the pleat shape and solve for the fine-grained shape of the pleats via surface deformation guided by this coarse graph. However, none of the existing formulations can be applied to Italian smocking [Wol96, Shi13], where the intricate pleats are formed by the special fabric-thread interaction.

Cloth simulation. Various elastic models [TPBF87] have been explored to simulate cloth dynamics, such as finite element representation [BW98, NSO12], mass-spring system [CK02, LBOK13],

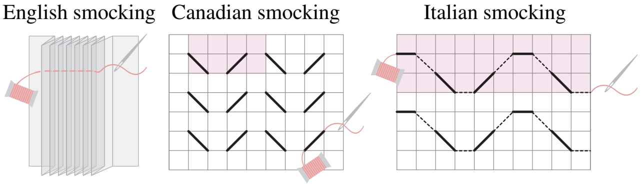

Figure 4: *Different types of smocking. For the Italian smocking pattern, the front and back stitches are delineated in solid and dashed line segments, respectively.*

and yarn-level model [KJM10, CLMMO14]. Advanced general-purpose simulators can be adapted to use in cloth simulation [LFS*20, LKJ21]. The codimensional incremental potential contact model (C-IPC) [LKJ21] is the state-of-the-art simulator designed for precision, efficiency, and stability, built upon [LFS*20]. C-IPC achieves intersection-free and strain-limit-satisfied simulation through the introduction of additional barrier models for strain-limiting potentials, thickness boundaries, and continuous collision detection. Coarse-to-fine approaches are introduced to capture the details of simulated cloth. Bergou et at. [BMWG07] augment coarse input with physically simulated details. Zhang et al. [ZDF*22] propose a progressive method to accelerate cloth simulation for detailed wrinkles. Existing physical simulators can produce plausible results without self-intersections for large-scale fabric manipulation, such as cloth draping. However, when applied to fine-scale embroidery like smocking without geometric priors or precise guidance, simulators often struggle to create regular and realistic pleats [RSSH23]. These challenges are likely due to factors such as excessive contraction under the assumption of zero sewing length, asymmetric motion solvers, and inappropriate stitched thread representation.

3. Preliminaries

An Italian smocking pattern $\mathcal{P}$ is usually drawn on a piece of fabric with a set of *stitching paths* $\mathcal{L} = \{\ell\}$. Each stitching path is a list of *consecutive* grid vertices with alternating front stitches and back stitches. Fig. 4 (right) shows an example of Italian smocking, which differs fundamentally from Canadian smocking in several ways: (1) Canadian smocking patterns consist of *disconnected* and short stitching lines, which are only annotated on *one side* of the fabric. In contrast, Italian smocking patterns feature *continuous* and long stitching paths that traverse both the *front and back* sides of the fabric alternately. (2) Creating a Canadian smocking pattern requires independently sewing each stitching line with localized knots. Conversely, Italian smocking patterns are fabricated along the stitching paths in a row-by-row fashion without localized knots. (3) The decorative pleats are formed differently as well. In Canadian smocking, the localized stitches exert force to push the fabric out of the initial fabric plane, resulting in voluminous pleats. Consequently, the fabric's thickness can be largely ignored, assuming that after stitching, several points merge into one [RSSH23]. In contrast, for Italian smocking, the fabric is gently compressed and folded, and the resulting pleats take shape as the thread is pulled at the free ends. These threads interweave a long sequence of points in the fabric and gather pleats. The thickness of the fabric plays a crucial role in maintaining the order of the created pleats and cannot be disregarded in physical simulation.

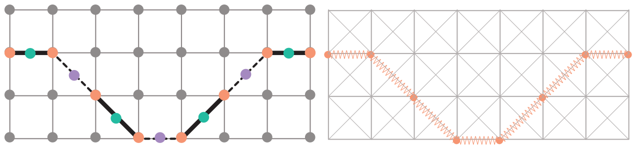

Figure 5: ***Mass-spring system**. For the Italian smocking pattern shown on the left, with front (resp. back) stitches annotated in solid (resp. dashed) line segments, we define a mass-spring system based on the original grid. **Left**: we highlight the stitching vertices $\mathcal{V}_s$, front midpoints $\mathcal{V}_f$, back midpoints $\mathcal{V}_b$, and pleat vertices $\mathcal{V}_p$, in orange, green, purple, and gray, respectively. **Right**: we color the fabric spring (stitching spring) in gray (orange).*

We model an input pattern $\mathcal{P}$ by a mass-spring system $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices from the original grid and $\mathcal{E}$ is the set of edges (springs) that connect the vertices in $\mathcal{V}$. Specifically, there are two types of springs in $\mathcal{E}$: (1) the **fabric springs** $\mathcal{E}_f$ that connect the adjacent grid vertices in $\mathcal{V}$, and (2) the **stitching springs** $\mathcal{E}_s$ that connect two stitching vertices of a line in $\mathcal{L}$.

We categorize the vertices in $\mathcal{V}$ into two groups, see also Fig. 5: (1) $\mathcal{V}_s$: the **stitching vertices** that lie on the stitching paths $\mathcal{L}$, and (2) $\mathcal{V}_p = \mathcal{V} \backslash \mathcal{V}_s$: the remaining vertices, which we refer to as **pleat vertices**. Note that the encoding of the front-back information is not possible within the stitching springs because there are no faces to distinguish the front or back side of a fabric in the mass-spring system. We therefore sample the midpoints of all front and back stitches in the stitching paths to encode the front-back information. Specifically, $\mathcal{V}_f$ denotes the midpoints of the *front stitches* in $\mathcal{L}$ (we call them **front midpoints** for simplicity) and $\mathcal{V}_b$ denotes the midpoints of the *back stitches* in $\mathcal{L}$ (called **back midpoints**). See Fig. 5 (left) for an illustration. For a vertex $v_i \in \mathcal{V} \cup \mathcal{V}_f \cup \mathcal{V}_b$, we denote its original 2D position in the flat fabric as $\bar{\mathbf{x}}_i \in \mathbb{R}^2$.

4. Method

Given an input Italian smocking pattern, our goal is to preview the corresponding fabricated result in the form of a surface mesh in 3D, showing the intricate folds and pleats. The main challenges of this problem include accounting for the fabric thickness, distinguishing between front and back stitches, and determining the non-vanishing length of stitched threads in the fabricated state. Existing methods such as [RSSH23] and C-IPC [LKJ21] do not support advanced stitching primitives. If the smocking stitches are considered as seams, these methods struggle with Italian smocking, since the distance between two stitching points is assumed to be zero.

To tackle the challenges, we propose an abstraction of a mass-spring system with front-back information from the input pattern. We then dynamically estimate the expected lengths of the stitching springs and find the 2D embedding of the stitching vertices via the simulated configuration of the mass-spring system (Sec. 4.1).

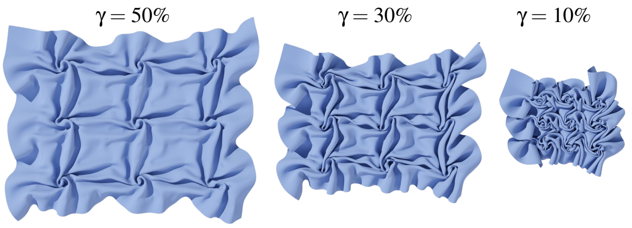

Figure 6: *Our results with different shrinkage γ.*

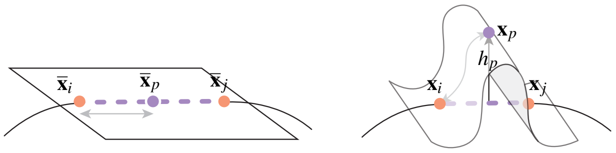

Figure 7: *Pulling the thread of a back stitch (left) causes the fabric to bend outward (right). We can estimate the height h_p of the midpoint $\mathbf{x}_p$ based on Pythagoras' theorem, as shown in Eq. (8).*

The resulting spring lengths and positions of the stitching vertices are fed into C-IPC to guide the deformation of the fabric in fine resolution, with collision handling (Sec. 4.2).

4.1. 2D simulation of the mass-spring system

Drawing inspiration from [RSSH23], we extract a low-resolution mass-spring system $(\mathcal{V}, \mathcal{E})$ to distill the geometric priors from the input Italian smocking pattern. Solving the dynamics of such a mass-spring system appears straightforward at first glance. However, the fact that the expected lengths for all fabric springs $\mathcal{E}_f$ and the stitching springs $\mathcal{E}_s$ are *unknown* makes the problem hard to formulate and solve. For example, as shown in Fig. 3, the final distance between two consecutive stitching vertices is not necessarily zero, leaving the expected length of a stitching spring unpredictable. Furthermore, in our setting, the length of a fabric spring in the final embedding is often much smaller than its original length in the fabric. Specifically, in typical cases where a high-resolution mass-spring system is employed to model a small fabric piece, it is common to assume that fabric springs preserve their original lengths during fabric deformation. Here the length indicates the anticipated Euclidean distance between two vertices. However, this assumption is not applicable to our low-resolution mass-spring system. For a fabric spring shown in Fig. 5 (right), with the thread tightened, its two endpoints would be closely embedded at a distance equal to the fabric thickness after smocking, which significantly deviates from the spring's original length within the fabric.

To address these challenges, we propose solving for a constrained *2D projection* of the dynamics of the abstracted mass-spring system, rather than dealing with the full 3D mass-spring system with unknown expected lengths. More precisely, we would like to find a 2D projection $\mathbf{X}$ for the mass-spring system by solving:

$$\max_{\mathbf{X}\in\mathbb{R}^{|\mathcal{V}|\times 2}} \sum_{(i,j)\in\mathcal{E}} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2, \tag{1a}$$

$$\text{s.t.} \quad \tau \leq \left\|\mathbf{x}_p - \mathbf{x}_q\right\|_2 \leq \left\|\bar{\mathbf{x}}_p - \bar{\mathbf{x}}_q\right\|_2, \quad \forall (p,q) \in \mathcal{E}_f, \tag{1b}$$

$$\tau\,|\mathcal{E}_s| \leq \mathbb{D}\left(\mathbf{X} \mid \mathcal{L}\right) \leq \gamma\,\mathbb{D}\left(\bar{\mathbf{X}} \mid \mathcal{L}\right), \tag{1c}$$

where τ is the fabric thickness, $|\mathcal{E}_s|$ is the number of edges (springs) in the stitching paths $\mathcal{L}$, $\bar{\mathbf{x}}_i$ is the original 2D position of the vertex v_i and $\bar{\mathbf{X}}$ is the set of all these original positions. The function $\mathbb{D}(\mathbf{X} \mid \mathcal{L})$ measures the total length of the threads that pass through the fabric along the stitching paths $\mathcal{L}$ given the current embedding $\mathbf{X}$:

$$\mathbb{D}\left(\mathbf{X} \mid \mathcal{L}\right) = \sum_{(i,j)\in\mathcal{E}_s} \|\mathbf{x}_i - \mathbf{x}_j\|_2. \tag{2}$$

The energy in Eq. (1a) encourages the vertices to stay away from each other to avoid cluttered pleats, similar to [RSSH23]. The first set of constraints, Eq. (1b), poses restriction on the length of a fabric spring $(p,q) \in \mathcal{E}_f$: once embedded in 2D, it must be at minimum the fabric thickness τ and at maximum equal to the original spring length of the initial flat fabric state. A violation of these constraints can result in fabric penetration or tearing.

The second set of constraints, Eq. (1c), poses restrictions on the stitching springs. Recall that when fabricating an Italian smocking pattern, the free ends of the threads (aligned with the stitching paths $\mathcal{L}$) are pulled. This action causes the fabric to fold and pleat along the stitching paths until the desired texture is achieved. The in-between stitching vertices have the freedom to slide along the thread. We therefore consider the total length of the threads $\mathbb{D}(\mathbf{X} \mid \mathcal{L})$ instead of each stitching spring independently. The minimal possible total length is $\tau\,|\mathcal{E}_s|$, which corresponds to the scenario when the threads are pulled completely taut, such that no stitching vertices can slide. At the same time, we propose a natural upper bound where the sum of the thread lengths is reduced to a γ fraction of their original total length. Here, the hyper-parameter γ models the pulling force applied to the threads. For example, when $\gamma = 1$, we can see that $\mathbf{X} = \bar{\mathbf{X}}$ gives the optimal solution. A smaller value $\gamma < 1$ drives the stitching vertices to move closer to meet the constraints, leading to folds and pleats. See Fig. 6 for an example.

Algorithm overview. The non-linearly constrained, non-convex problem in Eq. (1) is hard to optimize directly. We therefore propose to solve for the embedding $\mathbf{X}$ in an alternating scheme:

Step 1: initialize the embedding $\mathbf{X}^{(0)} = \bar{\mathbf{X}}$;

Step 2: estimate the expected distances $d_{ij}^{(k)}$ from the constraints based on $\mathbf{X}^{(k)}$ (detailed below);

Step 3: update the embedding $\mathbf{X}^{(k+1)}$ based on the expected distances $d_{ij}^{(k)}$, which is solved as an unconstrained problem:

$$\mathbf{X}^{(k+1)} = \operatorname*{arg\,min}_{\mathbf{X}\in\mathbb{R}^{|\mathcal{V}|\times 2}} \sum_{(i,j)\in\mathcal{E}} \left(\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2 - d_{ij}^{(k)}\right)^2; \tag{3}$$

Step 4: set $k \leftarrow k+1$ and go to Step 2, unless a stopping criterion is satisfied.

We now detail the estimation of $d_{ij}^{(k)}$ based on the current embedding $\mathbf{X}^{(k)}$ mentioned in Step 2. For a fabric spring $(i,j) \in \mathcal{E}_f$, $d_{ij}^{(k)}$ is determined to make sure the constraints in Eq. (1b) are satisfied:

$$d_{ij}^{(k)} = \max\left\{\tau,\ \min\left\{\|\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\|_2, \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|_2\right\}\right\}. \tag{4}$$

For stitching springs, we introduce a pulling direction to model their shrinkage speed. In physical fabrications, we observe that the stitches that align more closely with the pulling direction (i.e., with a larger projected length) shrink more rapidly (i.e., have a smaller expected length $d_{ij}^{(k)}$), and vice versa. Specifically, for a stitching spring $(i,j) \in \mathcal{E}_s(\ell)$, we compute

$$d_{ij}^{(k)} = \max\left\{\tau,\ \mathbb{P}_{\mathbf{d}}\left(\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\right)\right\}, \tag{5}$$

where $\mathbb{P}_{\mathbf{d}}(\mathbf{x})$ gives the projected length of a vector $\mathbf{x}$ onto the direction $\mathbf{d}$, which is orthogonal to the pulling direction. Here, for notation simplicity, instead of expressing the expected length as inversely proportional to the projection onto the pulling direction, we represent it as proportional to the projection onto the orthogonal of the pulling direction. The upper bound in Eq. (1c) is adopted as the stopping criterion in Step 4. Specifically, we stop the algorithm when the total length of the threads is below the preset shrinkage ratio γ. See Algorithm 1 for a detailed description and Appendix A for its convergence behavior analysis. In Appendix B, we provide a comparison to an existing solver, sequential least squares programming (SLSQP) [Kra88], where our algorithm shows significantly faster convergence and leads to better simulated results.

4.2. Mesh deformer via C-IPC

We use the solved 2D embedding $\mathbf{X}_s$ of the *stitching vertices* to guide the 3D deformation of the fabric, which is now represented in a much finer resolution. During the 2D simulation, the fabric springs $\mathcal{E}_f$ mainly serve as the inextensible constraints for the stitching springs $\mathcal{E}_s$, and their actual positions, depending on the final shape of the formed pleats, are undetermined at this stage. We therefore only use the stitching points to guide the deformation, since they are better constrained to provide more accurate control.

Algorithm 1 2D simulation of the mass-spring system

Hyper-parameters: shrinkage γ, projection direction $\mathbf{d}$, fabric thickness τ, time step $\Delta t = 0.1\,\text{sec}$, damping ratio $\alpha = 0.9$, fabric spring stiffness $k_f = 1$, stitching spring stiffness $k_s = 5$.

```
procedure 2D SIMULATION($\mathbf{X}, \overline{\mathbf{X}}, \mathcal{L}, \mathcal{E}_s, \mathcal{E}_f, \mathcal{E}$)
    $\mathbf{a} \leftarrow \mathbf{0}, \mathbf{v} \leftarrow \mathbf{0}$
    while $\mathbb{D}(\mathbf{X} \mid \mathcal{L}) > \gamma\,\mathbb{D}(\overline{\mathbf{X}} \mid \mathcal{L})$ do        ▷ Stopping criterion
        for $(i,j) \in \mathcal{E}_s$ do        ▷ Expected length for stitching springs
            $d_{ij} \leftarrow \max\{\tau,\ \mathbb{P}_{\mathbf{d}}(\mathbf{x}_i - \mathbf{x}_j)\}$        ▷ Eq. (4)
        end for
        for $(i,j) \in \mathcal{E}_f$ do        ▷ Expected length for fabric springs
            $d_{ij} \leftarrow \max\{\tau,\ \min\{\|\mathbf{x}_i - \mathbf{x}_j\|_2, \|\overline{\mathbf{x}}_i - \overline{\mathbf{x}}_j\|_2\}\}$        ▷ Eq. (5)
        end for
        for $(i,j) \in \mathcal{E}$ do        ▷ Calculate acceleration
            $\Delta\mathbf{d}_{ij} \leftarrow (\|\mathbf{x}_i - \mathbf{x}_j\|_2 - d_{ij})^{(\mathbf{x}_i - \mathbf{x}_j)}/_{\|\mathbf{x}_i - \mathbf{x}_j\|_2}$
            $\mathbf{a}_i \leftarrow \mathbf{a}_i - k\Delta\mathbf{d}_{ij}$
            $\mathbf{a}_j \leftarrow \mathbf{a}_j + k\Delta\mathbf{d}_{ij}$
        end for
        $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{a}\Delta t$
        $\mathbf{v} \leftarrow \alpha\mathbf{v}$
        $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}\Delta t$
        $\mathbf{a} \leftarrow \mathbf{0}$        ▷ Reset for next timestep
    end while
end procedure
```

Specifically, we first determine the 3D positions of the stitching vertices by assuming they all share the same height and experience no unbalanced external forces. Without loss of generality, we set the height to zero, i.e., the estimated 3D position for the stitching vertex, denoted $\mathbf{x}_s^c \in \mathbb{R}^3$, is:

$$\mathbf{x}_s^c \leftarrow (\mathbf{x}_s, 0), \quad v_s \in \mathcal{V}_s. \tag{6}$$

Next, we estimate the 3D position of the front/back midpoint v_m of the stitching spring $(i,j) \in \mathcal{E}_s$:

$$\mathbb{R}^3 \ni \mathbf{x}_m^c \leftarrow \begin{cases} \left({}^{(\mathbf{x}_i+\mathbf{x}_j)}/_2,\ -h_m\right) & \text{if } v_m \in \mathcal{V}_f \\ \left({}^{(\mathbf{x}_i+\mathbf{x}_j)}/_2,\ \ \ h_m\right) & \text{if } v_m \in \mathcal{V}_b \end{cases} \tag{7}$$

The superscript c abbreviates “constrained”, as these positions serve as positional constraints. Intuitively, the front stitches cause midpoints to fold inward, resulting in a negative height ($-h_m$), while the back stitches cause midpoints to fold outward, resulting in a positive height. A good approximation for h_m is (See Fig. 7):

$$h_m^2 + \left(\|\mathbf{x}_i - \mathbf{x}_j\|/2\right)^2 = \left(\|\overline{\mathbf{x}}_i - \overline{\mathbf{x}}_j\|/2\right)^2. \tag{8}$$

For simplicity one can also estimate h_m as $(\|\overline{\mathbf{x}}_i - \overline{\mathbf{x}}_j\| - \|\mathbf{x}_i - \mathbf{x}_j\|)/2$ when the fold is vertical with a flat crease.

We augment the original incremental potential of C-IPC [LKJ21] with the positional constraints of the embedded stitching vertices $\mathbf{X}_s$, the estimated midpoints $\mathbf{X}_m$, and the sewing length constraints to obtain the deformed fabric mesh in 3D:

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times 3}} E_{\text{cipc}}(\mathbf{X}) + w_s E_{\text{sew}}(\mathbf{X}) + w_p E_{\text{pos}}(\mathbf{X}), \tag{9}$$

where n is the total number of vertices in the finer representation of the fabric. The coefficients w_s and w_p are weights for these constraints. The term E_{sew} encourages the sewing length between two stitching vertices to be equal to our computed values. We add Δt^2, the squared timestep of C-IPC, to E_{sew}, incorporating it into the incremental potential as the sewing energy potential:

$$E_{\text{sew}}(\mathbf{X}) = \Delta t^2 \sum_{(p,q) \in \mathcal{E}_s} \left(\|\mathbf{x}_p - \mathbf{x}_q\|_2 - \|\mathbf{x}_p^c - \mathbf{x}_q^c\|_2\right)^2. \tag{10}$$

The term E_{pos} is the positional regularizer that encourages the stitching vertices and midpoints to remain close to the solved positions $\mathbf{X}_s \cup \mathbf{X}_m$:

$$E_{\text{pos}}(\mathbf{X}) = \sum_{i \in \mathcal{V}_s \cup \mathcal{V}_f \cup \mathcal{V}_b} \|\mathbf{x}_i - \mathbf{x}_i^c\|_2^2. \tag{11}$$

We consider E_{pos} as a soft positional constraint instead of a virtual energy potential. Therefore the timestep Δt^2 is not included. Integrating these two regularizers into C-IPC effectively guides the fabric from its initial flat state toward the desired pleat shapes, while the original configurations of C-IPC handle self-collisions during the cloth simulation.

5. Results

5.1. Comparisons to physical fabrications

To validate that our method produces faithful results, we manually fabricated several Italian smocking patterns. Using a laser cutter, we engraved the pattern onto the fabric, followed by sewing through the fabric along the pattern rows and pulling the threads

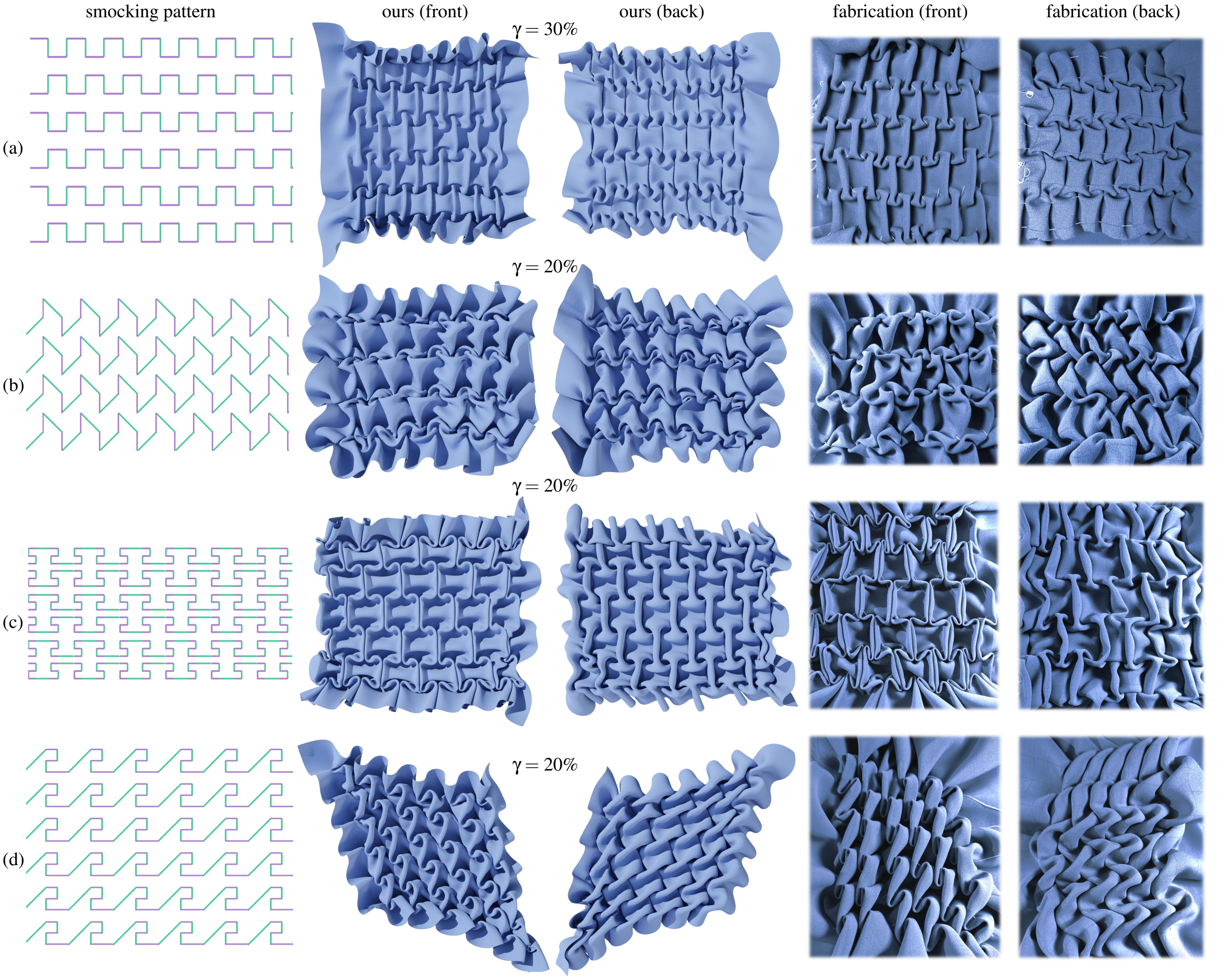

Figure 8: *We compare our simulated results to physical fabrications.*

to achieve a similar appearance to that shown in YouTube tutorials. We then measured the approximate thread shrinkage γ and executed our method using the same value. In Fig. 8 we show four examples of smocking patterns, where we color the front and back stitches in green and purple, respectively. We can see that our simulated results are close to the physical fabrications.

Italian smocking is relatively easy to fabricate, requiring merely the pulling of threads stitched through the fabric to create pleats. However, the complexity lies in pleat adjustment, as the resulting pleats after pulling are often irregular or distorted if the fabric does not properly slide along the threads (see Fig. 2). The fabricator has to adjust the pleats to achieve the desired shape. Therefore, in practice, designing a new Italian smocking pattern is challenging. If the fabricated result appears irregular or less pleasing, it is difficult to determine whether the issue lies in the pattern itself or the pleat refinement process was not adequately executed, since the pleat shape is unknown. In Fig. 9, we showcase two patterns ((b) and (c)), not available online, and utilize our computed results to assist in adjusting the physically fabricated pleats to improve their shapes. Specifically, we start from a known, classic pattern in Fig. 9 (a) and make local modifications to obtain the patterns (b) and (c), which retain the overall zigzag style. Despite the minor nature of these modifications, they lead to significant and unexpected changes in the final fabricated results. Our simulated results successfully helped us to adjust the shape of the physical pleats. We believe our algorithm can benefit artists in designing new patterns, as our results are consistently regular and reproducible in physical fabrications.

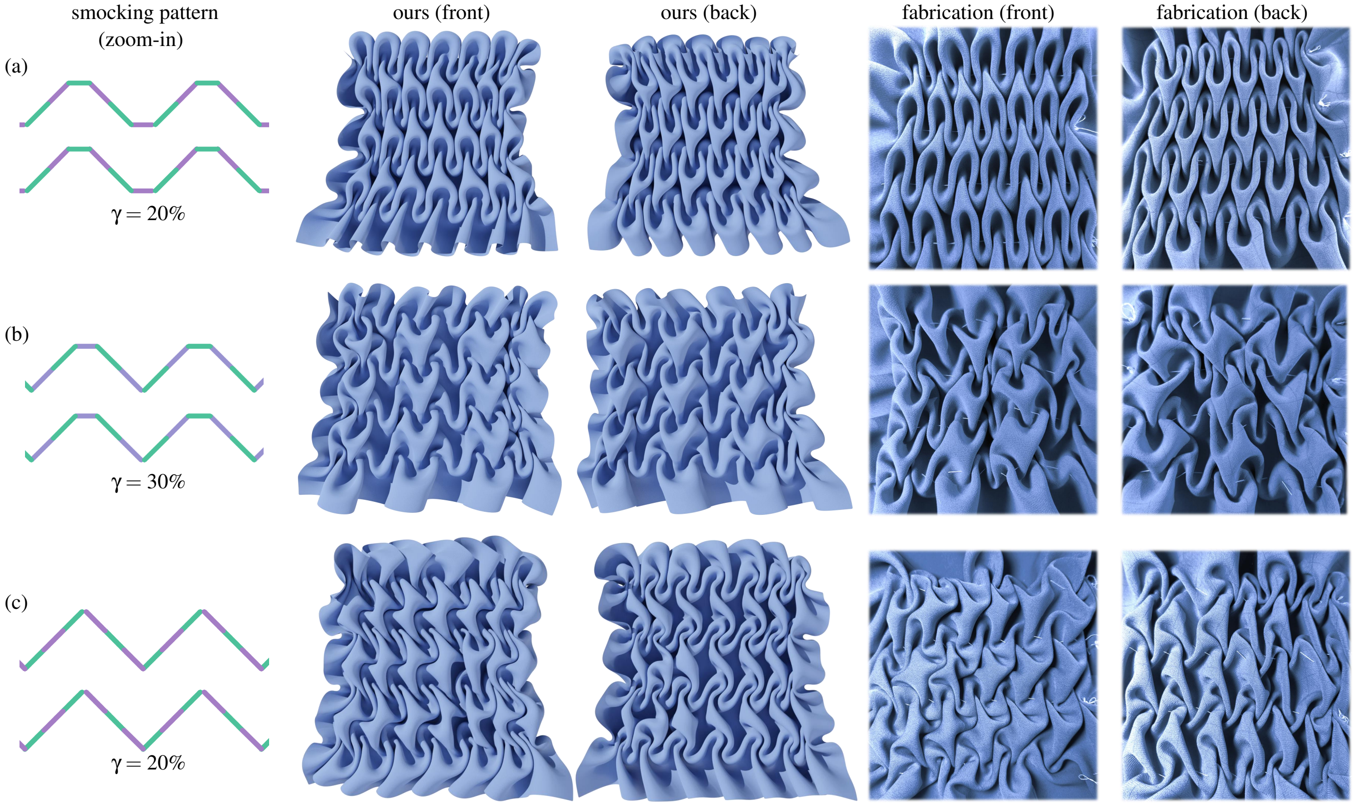

Figure 9: *Small changes in the smocking pattern can lead to significantly different pleats.*

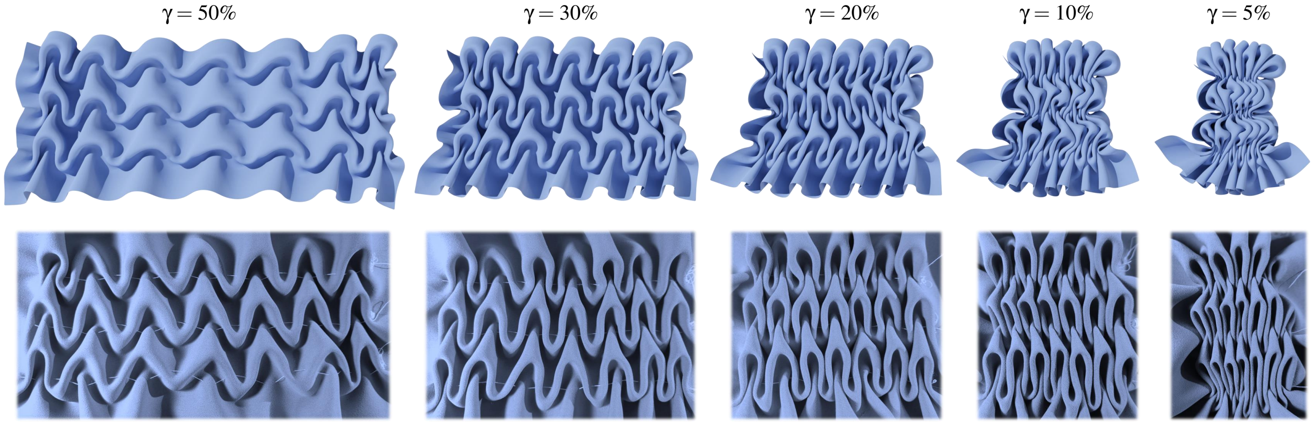

Figure 10: *Our results with different values of shrinkage γ (**top**) and the corresponding fabrications (**bottom**).*

5.2. Justifications of formulation

A key factor of our formulation is to model the gradual extraction of the thread during the fabric-thread interaction. As detailed in Eq. (1c), we achieve this by introducing the parameter γ to describe the shrinkage of the total length of the threads. In Fig. 10, we compare our results when using different values of γ to physical fabrications. It is evident that our results faithfully capture the smocking patterns, validating our choice of the parameter γ.

Our formulation is flexible and capable of accommodating scenarios where the expected sewing length distribution varies due to additional external constraints. For example, artists may incorporate beads into the fabrication process (see Fig. 3, left), which

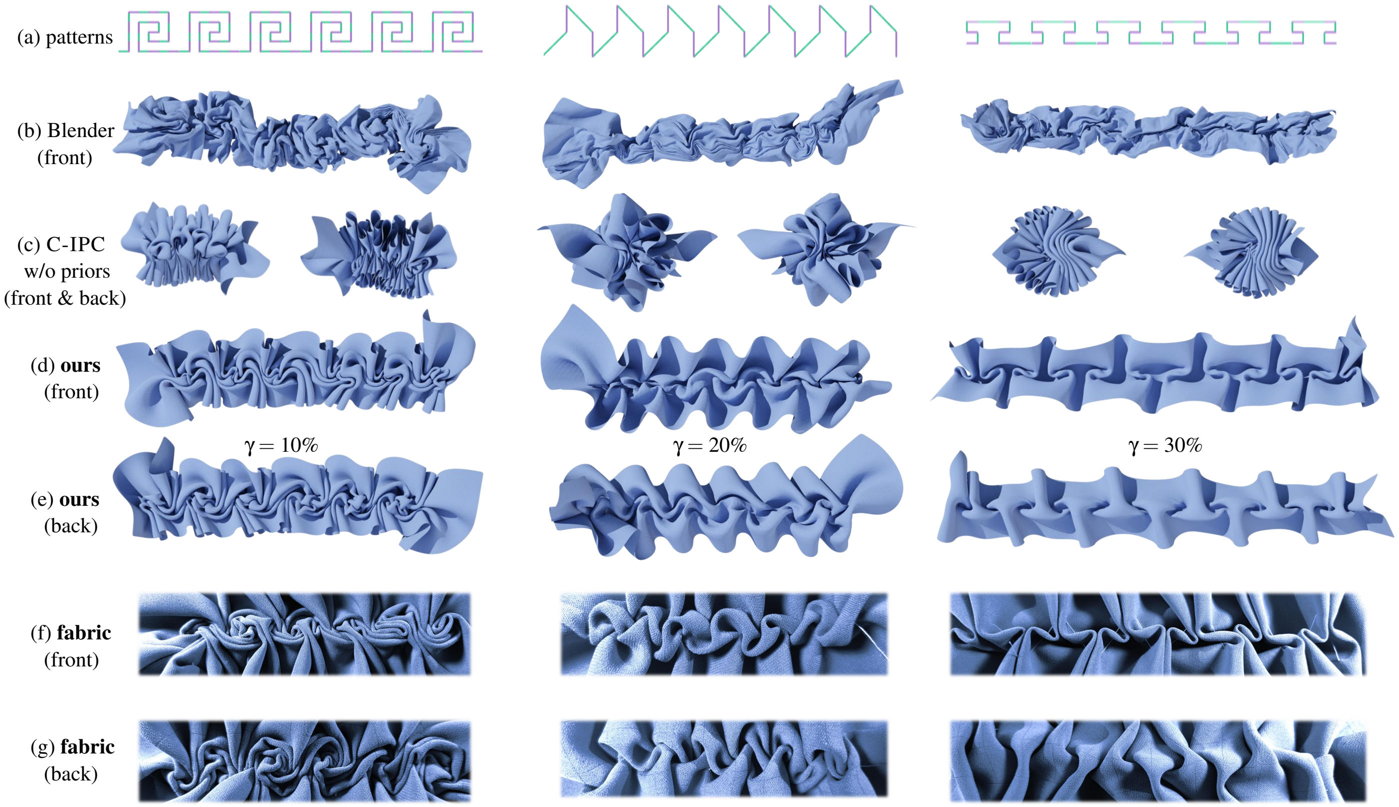

Figure 11: *We compare our method to Blender [Ble23],* C-IPC *[LKJ21] (without any priors), and physical fabrications.*

can be regarded as additional lower bound constraints for selected stitching springs. These constraints can be seamlessly integrated into our formulation, i.e., Eq. (1c). In Fig. 12, we show an example of an Italian smocking pattern with beads. The arrow indicates the placement of the beads during fabrication. For these selected stitching springs, we set the lower bound to the diameter of the beads. The resulting smocked design shows a distinct pleat pattern with an altered sewing length distribution imposed by these beads.

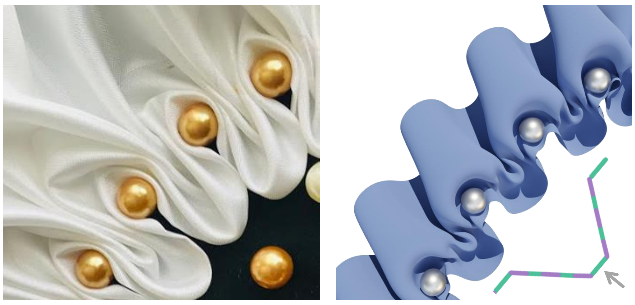

Figure 12: *Italian smocking with beads.* ***Left****: fabrication from ©FymsEmbroidery YouTube channel [Fym21]. Used with permission.* ***Right****: our result.*

5.3. Comparisons to cloth simulators

Baselines. We consider the closest setting in Blender [Ble23] (build-in cloth simulator), and C-IPC [LKJ21] (vanilla version without any prior knowledge) as baselines, where two stitching points are expected to have zero distance after sewing and the fabric is not allowed to slide along the threads. We admit that existing cloth simulators hold great potential to address the Italian smocking problem by modeling the threads with thin rods. However, this requires non-trivial considerations such as establishing boundary constraints and formulating pulling forces that guide the fabric to slide. We consider it a challenging open question and demonstrate our naive attempt at rod-fabric interaction in Appendix C.

Fig. 11 shows the comparison between our method, the considered baselines, and real fabrications. Specifically, we use the built-in cloth simulator in Blender to simulate the smocking process, treating each front and back stitch in the pattern as a sewing line. During the simulation, we manually halt the process when the total length of the sewing lines reaches γ fraction of their initial lengths. We particularly fine-tune the maximum sewing force parameter to mimic the real fabrication process, ensuring that stitched threads

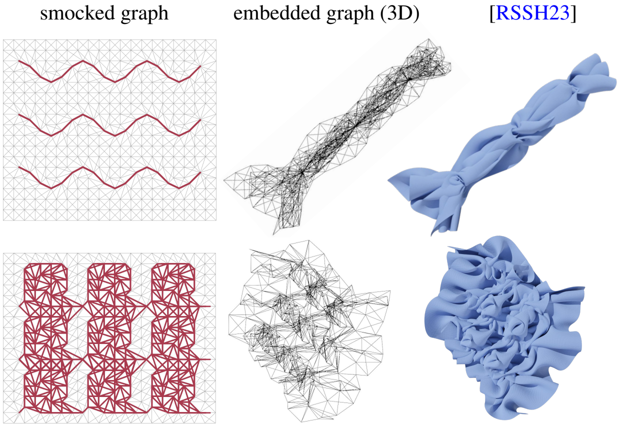

Figure 13: *[RSSH23] fails to produce reasonable results on the Italian smocking patterns shown in Fig. 9 (a) and Fig. 11 (left). The underlay edges in the smocked graphs are highlighted in red.*

are gently pulled out and avoiding rapid fabric shrinkage during simulation. We can see that Blender fails to produce reasonable results: the solver cannot distinguish between the front and back stitches and fails to produce correct and consistent bending directions for the fabric during the simulation. We also employ C-IPC without priors to simulate smocking, treating each front and back stitch as a sewing line. However, the simulated results often appear cluttered due to C-IPC's assumption of zero-length sewing lines. Furthermore, without distinguishing between front and back stitching lines, the generated pleats exhibit randomly oriented bulges in both upward and downward directions. In contrast, our preview results are more plausible and faithful in comparison with real fabrications.

5.4. Adaptation for Canadian smocking

Applying the method designed for Canadian smocking [RSSH23] to Italian smocking leads to disastrous results, since all the stitching points on the stitching line are simply merged into a single node. We therefore decompose a stitching path $\ell = (v_1, v_2, \cdots, v_k)$ in an Italian smocking pattern into a set of separate stitching lines $\{\ell_1 = (v_1, v_2), \ell_2 = (v_2, v_3), \cdots, \ell_{k-1} = (v_{k-1}, v_k)\}$ to avoid a degenerated smocked graph. However, even after this modification, the method in [RSSH23] still fails to produce reasonable results, as shown in Fig. 13. On the other hand, our method can be easily applied to Canadian smocking. We define the set of back midpoints as empty ($\mathcal{V}_b = \emptyset$), set the expected lengths for all stitching springs to zero, and run our algorithm. See Fig. 14 for some examples. Our method produces reasonable pleat shapes. Our results exhibit less regularity compared to [RSSH23], primarily because their approach incorporates structural geometry priors, whereas our method is simulation-based and offers less control over the regularity of the pleats.

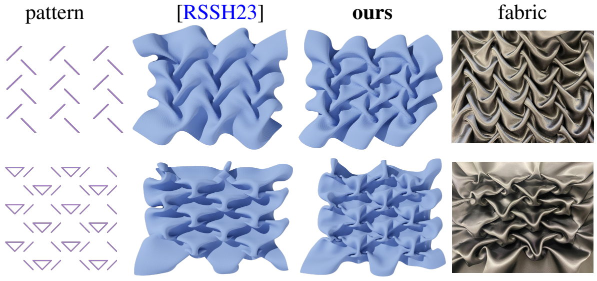

Figure 14: *Our algorithm can be adapted to Canadian smocking. Physical fabrications are from [RSSH23]. Used with permission.*

5.5. Ablation on geometric priors

In Fig. 15, we conduct an ablation study to justify the usefulness of the computed sewing lengths and positions for guiding C-IPC. In particular, we consider the following settings:

(a) E_{cipc} without priors: we use the released code of [LKJ21] for cloth draping with our fine-tuned parameters; the input stitching paths are considered standard sewing lines with an expected length of zero.
(b) E_{cipc} with offset initialization: the midpoints in $\mathcal{V}_b$ (resp. $\mathcal{V}_f$) on the planar fabric mesh are offset with uniform positive (resp. negative) height. This modified mesh is then loaded in C-IPC as the first frame. This configuration guides the simulator to deform the mesh with the corresponding bending direction of front and back stitches.
(c) $E_{\text{cipc}} + w_s E_{\text{sew}}$ with offset initialization: setting $w_p = 0$, we turn off the positional constraints and guide C-IPC only use the computed sewing lengths. Here we use the initialization with offsets, since the sewing lengths do not contain the labeling of front and back stitches.
(d) $E_{\text{cipc}} + w_p E_{\text{pos}}$: similarly, we set $w_s = 0$ to analyze the importance of the positional constraints. We do not use the offset initialization here, since our positional constraints already encode the front-back information.
(e) Ours, i.e., $E_{\text{cipc}} + w_s E_{\text{sew}} + w_p E_{\text{pos}}$.

When comparing (b) to (a), we can observe that the correct offsets in the initial configuration guide C-IPC to create more regularly folded pleats. However, the overall result appears cluttered and excessively shrunken. Incorporating the computed non-zero sewing lengths into C-IPC results in less cluttered but still unrealistic pleats, since the sewing length constraints still permit significant deformation freedom. On the other hand, positional constraints offer a stronger prior for C-IPC to achieve regular and realistic pleats. Note that the stitching vertices and the sampled midpoints are treated equally in the positional constraints. Incorporating the sewing length regularizer can impose stricter constraints on the stitching vertices alone. This combined energy results in more regular pleats, particularly in the boundary region.

5.6. Implementation, parameters & runtime

Implementation. We implement the 2D mass-spring system simulator in Python. The 3D deformer is implemented in C++ based on the

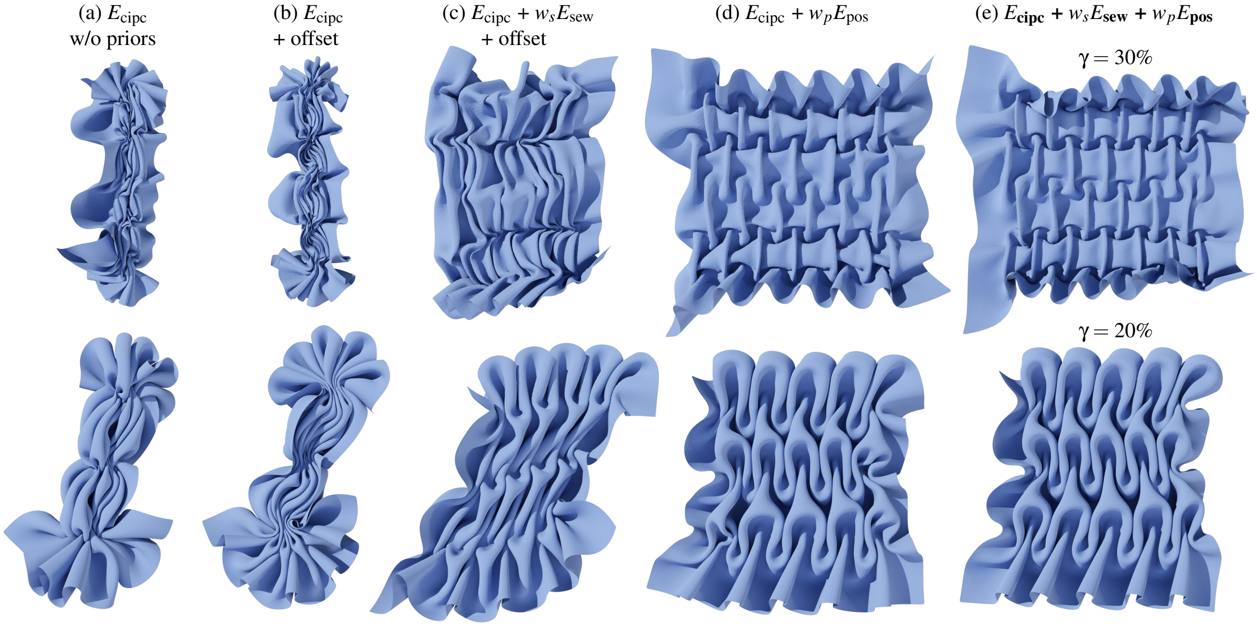
(a) E_{cipc}
w/o priors
(b) E_{cipc}
+ offset
(c) $E_{\text{cipc}} + w_s E_{\text{sew}}$
+ offset
(d) $E_{\text{cipc}} + w_p E_{\text{pos}}$
(e) $\mathbf{E_{cipc}} + w_s \mathbf{E_{sew}} + w_p \mathbf{E_{pos}}$
$\gamma = 30\%$
$\gamma = 20\%$

Figure 15: ***Ablation on geometric priors.*** *For two patterns with the same unit pattern shown in Fig. 8 (a) and Fig. 9 (a), we compare our results (shown in **(e)**) to four different settings: **(a)*** C-IPC *[LKJ21] without any priors, **(b)*** C-IPC *with offset initialization, where we additionally provide initial configuration with inward/outward offset to guide* C-IPC*, **(c)** we only provide the computed sewing lengths for* C-IPC *i.e., $w_p = 0$, with offset initialization, **(d)** we only provide the computed positional constraints for* C-IPC*, i.e., $w_s = 0$.*

released code of [LKJ21] with modifications to incorporate nonzero sewing lengths and positional constraints. All the experiments are run on an Ubuntu system with an 8-core 3.6 GHz Intel Core i7-9700K CPU and 32 GB of RAM. The full implementation can be found at `https://github.com/nifzhou/ItalianSmocking`.

Parameters. The input smocking pattern is scaled to have the longest side of 1 m. For all experiments, we use the *same* set of parameters: $w_s = 0.1$, $w_p = 0.01$. The shrinkage parameter γ is manually selected to replicate the textures demonstrated in online tutorials; we provide the values in the figures. Regarding the parameters for C-IPC, we retain most of the parameters provided for cotton material in the released code of [LKJ21]. We set the bending stiffness scaling parameter to 100 and use the isotropic membrane model without strain limiting. Our 3D deformer is executed for 100 frames with a timestep size of $\Delta t = 0.04$ sec, which is sufficient for convergence, starting from the fabric's initial planar state. For all the patterns except the CURVE pattern shown in Fig. 11 (left), we set the pulling direction (orthogonal to $\mathbf{d}$ discussed in Eq. (5)), to a unit horizontal vector, and the fabric thickness τ to 0.01. Altering the fabric thickness does not affect the appearance of the pleats, as these patterns rely on loose stitches to create their textures. The CURVE pattern is more challenging, since the fabric crinkles sharply along the curly stitching paths, whose accumulative rotation is more than 2π. The directions of stitching lines change significantly along the path and induce many local minima for the 2D spring system. We therefore align the pulling direction with each stitching spring instead of setting a uniform direction as in other patterns. We also adjust the thickness τ to 0.1 to better capture locally squeezed multi-layer folds of the pattern size used in physical fabrication.

Table 2: *For the examples shown in Figures 8 and 9, we report the input complexity (including #stitching vertices $|\mathcal{V}_s|$, #grid vertices $|\mathcal{V}|$, #vertices in the finer fabric mesh n, and shrinkage γ), the runtime for 2D simulation, 3D deformer, and total process.*

smocking pattern	*complexity*				*runtime (minutes)*		
	$\lvert\mathcal{V}_s\rvert$	$\lvert\mathcal{V}\rvert$	n	γ	2D sim.	3D sim.	total
Fig. 8 (a)	174	304	9116	30%	0.04	2.35	2.39
Fig. 8 (b)	116	288	8858	20%	0.06	2.49	2.55
Fig. 8 (c)	438	880	33649	20%	0.45	11.67	12.12
Fig. 8 (d)	222	638	21798	20%	0.18	7.21	7.39
Fig. 9 (a)	172	1175	42778	20%	1.77	11.92	13.69
Fig. 9 (b)	144	1000	35916	30%	0.70	8.63	9.33
Fig. 9 (c)	172	1363	50689	20%	2.23	16.04	18.27

Table 3: *The total runtime for the ablation study in Fig. 15.*

smocking pattern	*complexity*				*total runtime (minutes)*				
	$\lvert\mathcal{V}_s\rvert$	$\lvert\mathcal{V}\rvert$	n	γ	(a)	(b)	(c)	(d)	(e) **ours**
Fig. 15 (a)	135	252	7300	30%	5.46	4.98	1.88	1.19	1.79
Fig. 15 (b)	93	700	24069	20%	16.75	14.33	8.83	6.19	6.52

Runtime. In Table 2, we report the smocking pattern complexity and the runtime breakdown of our algorithm for the examples shown in Figures 8 and 9. The runtime of the 2D simulator depends on the number of grid vertices ($|\mathcal{V}|$), the number of stitching vertices ($|\mathcal{V}_s|$), and the shrinkage (γ). The runtime of the 3D deformer depends on the number of vertices in the finer fabric mesh (n) and the number of constraints added to C-IPC, closely related to $|\mathcal{V}_s|$.

Our 2D simulator is highly efficient, taking only seconds to converge even for smocking patterns with hundreds of stitching vertices. The 3D deformer can take minutes to deform the finer representation of the fabric with 10-50K vertices with collision handling. We also implemented a fast preview algorithm based on ARAP [SA07] with our computed positional constraints incorporated. This method is much more efficient and only takes a few seconds to preview the smocked pleats without material consideration. However, since there is no collision handling in ARAP, when the shrinkage is significant, the final result can have obvious self-intersections, as shown in Fig. 16 (see our result in Fig. 10 ($\gamma = 20\%$)).

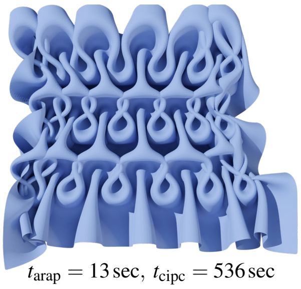
$t_{\text{arap}} = 13\,\text{sec}$, $t_{\text{cipc}} = 536\,\text{sec}$

Figure 16: *Preview via* ARAP.

The C-IPC-based deformer can handle self-collisions and produce more visually pleasing results, albeit at a higher computational cost. In practice, users or designers can use the ARAP-based deformer for a quick preview of the smocking results and switch to the C-IPC-based deformer to generate final results once satisfied with the smocking pattern design.

It is worth noting that incorporating additional non-zero sewing lengths and positional constraints actually speeds up the 3D simulation, contrary to intuition. In Table 3 we report the runtime comparison of the ablation study in Fig. 15, where we run each setting for 100 frames for a fair comparison. Our method, setting (e), is more than twice as fast as the original C-IPC without constraints (setting (a)). We believe this is because our accurate constraints reduce the search space, leading to faster convergence. Similar behavior is observed when comparing the setting (b) to (a), where adding the correct offset to the initialization helps C-IPC converge quicker.

6. Conclusion, limitations, & future work

In this work, we formalize the Italian smocking embroidery technique and propose a simple method to simulate the smocking results. Unlike Canadian smocking, Italian smocking involves continuous stitching lines that traverse the fabric. Pulling the free ends of the threads to gather the fabric along the stitching paths gives rise to intricate and complex pleat patterns. The distances between two stitching vertices in the final result are not necessarily zero, which makes existing methods, with zero-length stitching primitives, not directly suitable for the task. Our method consists of two main steps: firstly, we model the fabric as a coarse mass-spring system and solve for its projected 2D configuration by iteratively estimating the expected embedding length of fabric springs and stitching springs. Next, we incorporate the computed sewing lengths, along with the positions of the stitching vertices and their midpoints, into the C-IPC simulator to guide fabric deformation in finer resolution. Our method achieves more faithful results compared to baseline approaches. It is important to highlight that our smocked results closely resemble physical fabrications in structural and qualitative aspects, while not being completely identical. The inherent randomness in pleating persists, since the shape is not fully constrained, which is also a characteristic of smocking.

Our 2D simulator is efficient and offers accurate priors for 3D mesh deformation. However, running C-IPC to address self-collisions while integrating the computed constraints is time-consuming due to the fine level of details in the resulting pleats. A more efficient self-collision handling would greatly benefit interactive design, which we leave as future work. Another limitation of our formulation is that we use the shrinkage parameter γ with respect to the total length of threads to model the pulling process, implicitly assuming simultaneous shrinking for all stitches. In actual fabrication, it is common for stitches closer to the pulling end to shrink first, with the effect propagating to stitches farther away. While our final smocked results are reasonably faithful, the simulation process deviates from physical fabrication. It would be interesting to explore a more advanced and sophisticated formulation for accurately simulating the pulling process. In addition, we currently only address smocking in a planar configuration. Future work involving extensions to curved surfaces, such as adapting positional constraints through parameterization lifting, would contribute to digital textile design, providing realistic smocking patterns. Another interesting direction for future work is to develop an interactive system for pleated fabric. Many online smocking tutorials necessitate pleat adjustment by the fabricator to achieve the desired results, as pulling threads can result in irregular pleats. A digital equivalent in the form of an interactive system for digital design would be valuable.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful feedback. We extend our gratitude to M. Rifad (YouTube channel "DIY Stitching"), F. Shanas (YouTube channel "handiworks"), and S. Fyms (YouTube channel "FymsEmbroidery") for generously granting us permission to use the images of their remarkable fabrication results. This work was supported in part by the ERC Consolidator Grant No. 101003104 (MYCLOTH). Open access funding provided by Eidgenossische Technische Hochschule Zurich.

References

[Ble23] BLENDER FOUNDATION AND COMMUNITY: *Blender*, 2023. URL: `https://docs.blender.org/manual/en/latest/physics/cloth/index.html`. 8

[BMWG07] BERGOU M., MATHUR S., WARDETZKY M., GRINSPUN E.: Tracks: Toward directable thin shells. *ACM Transactions on Graphics 26*, 3 (July 2007), 50. 3

[BT95] BOGGS P. T., TOLLE J. W.: Sequential quadratic programming. *Acta Numerica 4* (1995), 1–51. `doi:10.1017/S0962492900002518`. 13

[BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Proc. SIGGRAPH* (1998), pp. 43–54. 2

[CCK*21] CHEN Z., CHEN H.-Y., KAUFMAN D. M., SKOURAS M., VOUGA E.: Fine wrinkling on coarsely meshed thin shells. *ACM Transactions on Graphics 40*, 5 (2021), 1–32. 2

[CK02] CHOI K.-J., KO H.-S.: Stable but responsive cloth. *ACM Transactions on Graphics 21*, 3 (July 2002), 604–611. 2

[CKSV23] CHEN Z., KAUFMAN D., SKOURAS M., VOUGA E.: Complex wrinkle field evolution. *ACM Transactions on Graphics 42*, 4 (2023). 2

[CLMMO14] CIRIO G., LOPEZ-MORENO J., MIRAUT D., OTADUY M. A.: Yarn-level simulation of woven cloth. *ACM Transactions on Graphics 33*, 6 (2014), 207:1–207:11. 3

[CLO23] CLO: clo3d.com. https://www.clo3d.com, Jan. 2023. 2

[DIY17] DIYSTITCHING: Easy DIY ideas you need to try | beaded smocking by DIY stitching, 2017. URL: https://www.youtube.com/watch?v=1cfKC-UvMt4. 2

[DIY18] DIYSTITCHING: Simple DIY ideas for clothes: Smocking tutorial by DIY stitching, 2018. URL: https://www.youtube.com/watch?v=bQWsZSTiJcU. 2

[EMZ16] EFRAT T. A., MIZRAHI M., ZORAN A.: The hybrid bricolage: bridging parametric design with craft through algorithmic modularity. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), pp. 5984–5995. URL: https://www.tamaraefrat.com/crafted-technology. 1

[Fym21] FYMS S.: Everyone should like to know this easy technique of smocking, 2021. URL: https://www.youtube.com/watch?v=3BiU3ARD9mw. 8

[Han22] HANDIWORKS: Smocking design for dress making projects, 2022. URL: https://www.youtube.com/watch?v=oxOfZTKnVaI. 2

[JHR*15] JUNG A., HAHMANN S., ROHMER D., BEGAULT A., BOISSIEUX L., CANI M.-P.: Sketching folds: Developable surfaces from non-planar silhouettes. *ACM Transactions on Graphics 34*, 5 (2015), 1–12. 2

[Kim20] KIM M.: A study on reproductions of North American smocking design using a 3D virtual clothing system. *Journal of Fashion Business 24*, 5 (2020), 106–124. 2

[KJM10] KALDOR J. M., JAMES D. L., MARSCHNER S.: Efficient yarn-based cloth with adaptive contact linearization. *ACM Transactions on Graphics 29*, 4 (2010), 105:1–105:10. 3

[Kra88] KRAFT D.: A software package for sequential quadratic programming. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt* (1988). 5, 13

[LBOK13] LIU T., BARGTEIL A. W., O'BRIEN J. F., KAVAN L.: Fast simulation of mass-spring systems. *ACM Transactions on Graphics 32*, 6 (Nov. 2013), 209:1–7. 2

[LFS*20] LI M., FERGUSON Z., SCHNEIDER T., LANGLOIS T. R., ZORIN D., PANOZZO D., JIANG C., KAUFMAN D. M.: Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Transactions on Graphics 39*, 4 (2020), 49. 3

[LKJ21] LI M., KAUFMAN D. M., JIANG C.: Codimensional incremental potential contact. *ACM Transactions on Graphics 40*, 4 (2021). 1, 2, 3, 5, 8, 9, 10

[LSGV18] LI M., SHEFFER A., GRINSPUN E., VINING N.: FoldSketch: enriching garments with physically reproducible folds. *ACM Transactions on Graphics 37*, 4 (2018), 133:1–133:13. 2

[MS22] MA C., SUN Z.: Multilayered stitch generating for random-needle embroidery. *The Visual Computer 38*, 11 (2022), 3667–3679. 1

[MSYA21] MOUHOU A. A., SAAIDI A., YAKHLEF M. B., ABBAD K.: Wrinkle synthesis for cloth mesh with Hermite radial basis functions. *Multimedia Tools and Applications 80* (2021), 1583–1610. 2

[NSO12] NARAIN R., SAMII A., O'BRIEN J. F.: Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics 31*, 6 (2012). 2

[PZB*09] POPA T., ZHOU Q., BRADLEY D., KRAEVOY V., FU H., SHEFFER A., HEIDRICH W.: Wrinkling captured garments using space-time data-driven deformation. *Computer Graphics Forum 28*, 2 (2009), 427–435. 2

[RSSH23] REN J., SEGALL A., SORKINE-HORNUNG O.: Digital 3D smocking design. *ACM Transaction on Graphics* (2023). Accepted for publication, presented at SIGGRAPH ASIA 2023. URL: https://igl.ethz.ch/projects/smockingdesign/. 1, 2, 3, 4, 9

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing* (2007), pp. 109–116. 11

[Shi13] SHIMADA N.: *Naoko Shimada Craft Technique*. Bunka Publisher, 2013. 2

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. In *Proc. SIGGRAPH* (1987), pp. 205–214. 2

[VGO*20] VIRTANEN P., GOMMERS R., OLIPHANT T. E., HABERLAND M., REDDY T., COURNAPEAU D., BUROVSKI E., PETERSON P., WECKESSER W., BRIGHT J., VAN DER WALT S. J., BRETT M., WILSON J., MILLMAN K. J., MAYOROV N., NELSON A. R. J., JONES E., KERN R., LARSON E., CAREY C. J., POLAT İ., FENG Y., MOORE E. W., VANDERPLAS J., LAXALDE D., PERKTOLD J., CIMRMAN R., HENRIKSEN I., QUINTERO E. A., HARRIS C. R., ARCHIBALD A. M., RIBEIRO A. H., PEDREGOSA F., VAN MULBREGT P., SCIPY 1.0 CONTRIBUTORS: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods 17* (2020), 261–272. doi:10.1038/s41592-019-0686-2. 13

[Wol96] WOLFF C.: *The art of manipulating fabric*. Penguin Publisher, 1996. 1, 2

[ZDF*22] ZHANG J. E., DUMAS J., FEI Y. R., JACOBSON A., JAMES D. L., KAUFMAN D. M.: Progressive simulation for cloth quasistatics. *ACM Transactions on Graphics 41*, 6 (2022). 3

[ZPH*23] ZHENYUAN L., PIOVARČI M., HAFNER C., CHARRONDIÈRE R., BICKEL B.: Directionality-aware design of embroidery patterns. *Computer Graphics Forum 42*, 2 (2023), 397–409. 1

[ZSS20] ZHENG Y., SHI H., SAITO S.: Creating reference image of realistic cloth folded surface using sketch-based interactive modeling. *Computers & Graphics 91* (2020), 108–117. 2

Appendix A: Algorithm convergence behavior

Here we analyze the convergence of Algorithm 1, designed to address the constrained problem discussed in Sec. 4.1. Specifically,

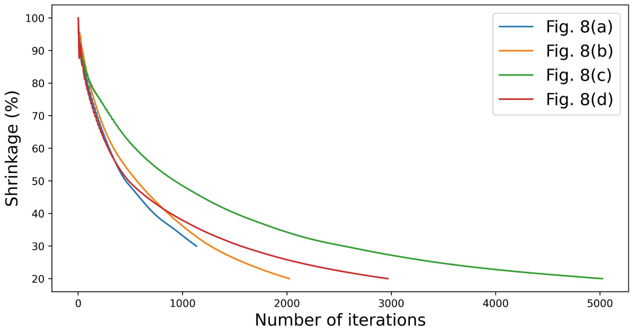

Figure 17: *Convergence analysis of Alg. 1 for the patterns in Fig. 8.*

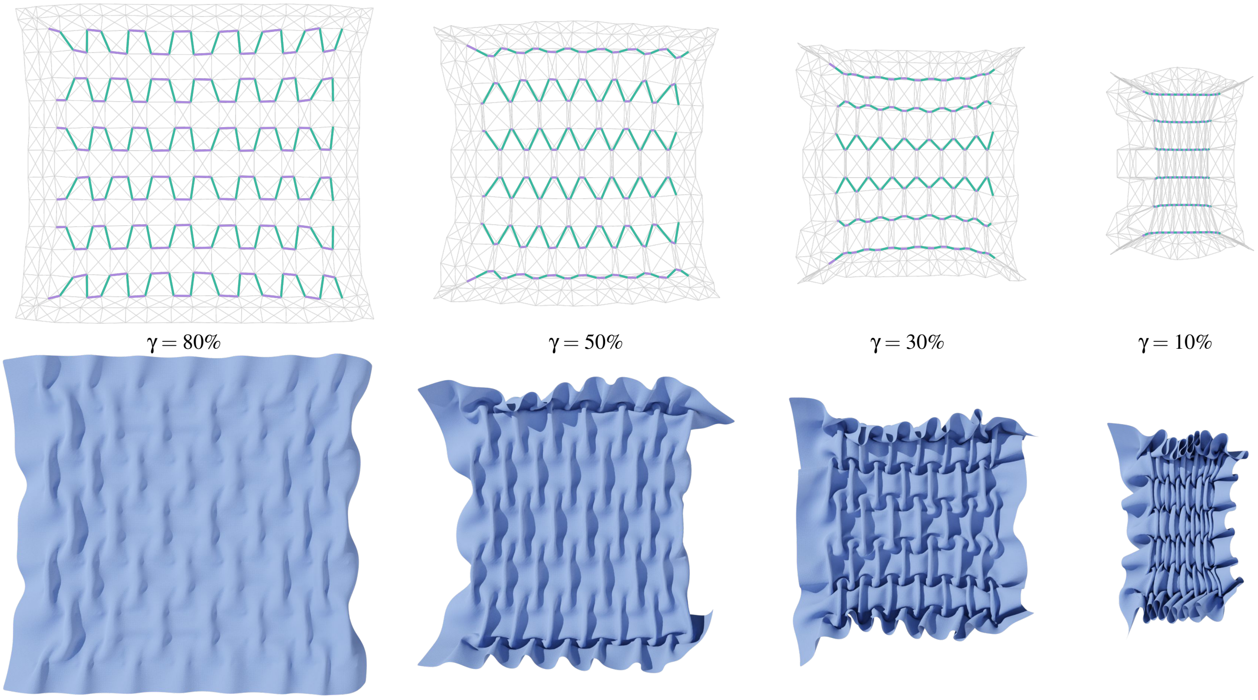

Figure 18: *The 2D simulation (*top*) and the corresponding 3D results (*bottom*) with different shrinkages γ for the pattern shown in Fig. 8 (a).*

the goal is to simulate the 2D mass-spring system with dynamic target spring lengths, adhering to the constraints detailed in Eq. (1b) and Eq. (1c). Algorithm 1, by design, successfully satisfies all the constraints except for the shrinkage requirement over iterations. Therefore, we consider the algorithm converged when the current shrinkage ratio aligns with the user-specified target shrinkage.

In Fig. 17 we report the convergence results for the four patterns showed in Fig. 8. The convergence behavior exhibits a similar logarithmic decrease across different patterns. Notably, the green line in Fig. 17, corresponding to the pattern in Fig. 8 (c), shows a slower convergence. This is attributed to the vertical stitching lines, which tend to shrink less rapidly compared to stitching lines in other orientations when the threads are pulled horizontally. This behavior aligns with real fabrications and is explicitly considered in Eq. (5).

Fig. 18 shows a qualitative example showcasing the algorithm's progression at different shrinkage values and the corresponding 3D results derived from the solved 2D configurations. The intermediate steps can still provide valid positional constraints to achieve realistic smocked results in 3D before the target shrinkage is reached. This feature benefits simulation of the shrinking process under different shrinkages in a single pass, as shown in Fig. 10 and Fig. 18.

Appendix B: SQP for 2D simulation

An alternative solution to the problem in Eq. (1) is using off-the-shelf solvers, such as sequential quadratic programming (SQP) [BT95], designed for constrained nonlinear optimization. Here we compare Scipy's SLSQP solver [Kra88, VGO*20] to our Algorithm 1 in solving Eq. (1) on two different patterns: the CURVE pattern shown in Fig. 1 and the pattern in Fig. 8 (a).

Table 4: *We report the input complexity for the examples in Fig. 17, including #stitching vertices $|\mathcal{V}_s|$, #grid vertices $|\mathcal{V}|$, #constraints N_c in Alg. 1, and shrinkage γ, and the runtime for 2D simulation.*

smocking pattern	*complexity*				*2D sim. runtime (sec)*	
	$|\mathcal{V}_s|$	$|\mathcal{V}|$	N_c	γ	SLSQP	**ours**
Fig. 8 (a)	174	304	2562	30%	161.5	2.2
Fig. 1	237	437	3716	10%	3140.1	68.8

We report the problem complexity and runtime comparison in Table 4, and show the corresponding qualitative comparison in Fig. 19. It is worth noting that solving Eq. (1) is quite challenging, primarily due to the significantly greater number of constraints (N_c) compared to the number of variables ($|\mathcal{V}|$). The SLSQP solver manages to find feasible solutions even when the initialization (planar fabric) does not fully satisfy all constraints. However, the solver still struggles with generating regular pleats given the complicated search space from the large number of constraints. This also justifies our choice of designing a specialized solver, leading to more accurate and visually pleasing results at a significantly reduced computational cost.

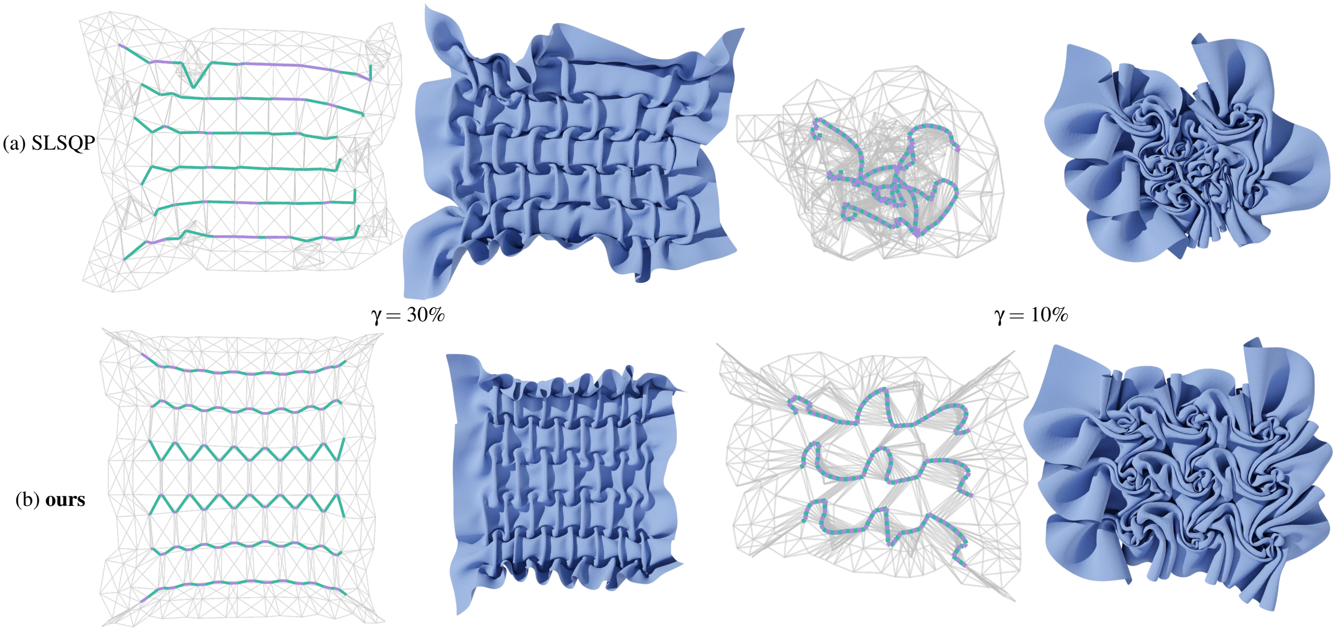

Figure 19: *We compare our simulated results to SQP-optimized results on two patterns shown in Fig. 8 (a) (left) and Fig. 1 (right).*

Appendix C: Rod-fabric interaction modeling for smocking

An alternative approach to simulating smocking involves treating the long threads as thin rods and exploring the interaction between the rods and fabric. Specifically, we thread thin rods through the pre-drilled tiny holes in the fabric. The initial positioning of the rods aligns with the input smocking pattern, with additional height adjustments relative to the front and back stitches. One end of the rods is secured with knots, while the other end is left free to simulate the pulling process in real fabrication, as shown in Fig. 20 (b).

We encountered several non-trivial challenges when modeling the fabrication process: (1) The real fabrication process includes both pulling the threads and pushing the fabric in the opposite direction. It is unclear how to translate the pushing action of the fabric into the simulation process. (2) Since the fabric undergoes shrinkage in both vertical and horizontal directions during the process, setting up accurate fixed-boundary conditions for simulation involves unknown positions. (3) Since multiple local minima exist, it is necessary to include a regularizer to find more regularly shaped pleats, which is challenging to formulate.

In all configurations we have explored, the final results are cluttered with irregular pleats, worse than the baselines we considered (i.e., Blender and C-IPC without any priors). See Fig. 20 (c) and Fig. 20 (d) for two representative failure cases encountered. Both experiments involve directing the free ends of the rods to move in a constant rightward direction. We model the pushing action of the fabric as a body force to the left applied across the entire fabric (setting (c)). In setting (d) we additionally fix the left side of the fabric. The simulated results in these naive configurations prove unsatisfactory. In setting (c), despite applying a sufficiently strong body force, the fabric slides along the rods before the pleats form when the rods are pulled – an occurrence that is unlikely in reality.

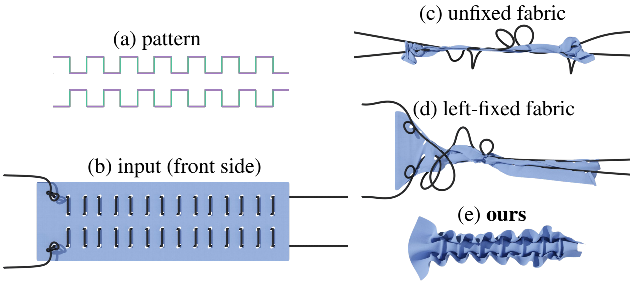

Figure 20: *We compare the results of formulating smocking via rod-fabric interaction to our thread-fabric interaction.*

To counter such sliding, the left side of the fabric is fixed in setting (d), resulting in only marginally improved results. Besides, both settings struggle to achieve a regular distribution of fabric shrinkage, showcasing the complexity of modeling the pulling action. Additionally, the overly cluttered regions induce complex (and mostly unnecessary) collision handling that requires significantly higher computational resources. These challenges emphasize the benefits of integrating geometric priors into the simulation, which leads to aesthetic pleats at significantly lower computational costs.