

# Fabric Tessellation: Realizing Freeform Surfaces by Smocking

AVIV SEGALL, ETH Zurich, Switzerland

JING REN, ETH Zurich, Switzerland

AMIR VAXMAN, University of Edinburgh, United Kingdom

OLGA SORKINE-HORNUNG, ETH Zurich, Switzerland

We present a novel method for realizing freeform surfaces with pieces of flat fabric, where curvature is created by stitching together points on the fabric using a technique known as *smocking*. Smocking is renowned for producing intricate geometric textures with voluminous pleats. However, it has been mostly used to realize flat shapes or manually designed, limited classes of curved surfaces. Our method combines the computation of directional fields with continuous optimization of a Tangram graph in the plane, which together allow us to realize surfaces of arbitrary topology and curvature with smocking patterns of diverse symmetries. Given a target surface and the desired smocking pattern, our method outputs a corresponding 2D smocking pattern that can be fabricated by sewing specified points together. The resulting textile fabrication approximates the target shape and exhibits visually pleasing pleats. We validate our method through physical fabrication of various smocked examples.

CCS Concepts: • Computing methodologies → Shape modeling.

Additional Key Words and Phrases: Computational fabrication, 3D smocking, optimization, parameterization, surface approximation

ACM Reference Format:

Aviv Segall, Jing Ren, Amir Vaxman, and Olga Sorkine-Hornung. 2024. Fabric Tessellation: Realizing Freeform Surfaces by Smocking. *ACM Trans. Graph.* 43, 4, Article 89 (July 2024), 20 pages. <https://doi.org/10.1145/3658151>

## 1 INTRODUCTION

In computational design and digital fabrication, realizing freeform surfaces has emerged as a pivotal challenge for diverse applications, such as freeform design and deployable structures for medical or architectural purposes. In particular, the exploration of shape approximation through the versatile aesthetics of origami or kirigami tessellations, where sheets of paper are folded along strictly prescribed crease patterns, has attracted significant attention in computer graphics [Choi et al. 2019; Dudte et al. 2016; Jiang et al. 2020; Narumi et al. 2023]. Origami and kirigami are traditionally associated with flat sheets of paper and can potentially be extended to other sheet materials, such as cardboard and silicone rubber [Jin et al. 2020; Martinez et al. 2012]. The rigidity and inextensibility of the used materials pose a unique challenge for the inverse design problem with origami and kirigami: the fabrication constraints of the materials must be taken into account along with the goal of

Authors' addresses: Aviv Segall, ETH Zurich, Switzerland, [aviv.segall@inf.ethz.ch](mailto:aviv.segall@inf.ethz.ch); Jing Ren, ETH Zurich, Switzerland, [jing.ren@inf.ethz.ch](mailto:jing.ren@inf.ethz.ch); Amir Vaxman, University of Edinburgh, United Kingdom, [avaxman@inf.ed.ac.uk](mailto:avaxman@inf.ed.ac.uk); Olga Sorkine-Hornung, ETH Zurich, Switzerland, [sorkine@inf.ethz.ch](mailto:sorkine@inf.ethz.ch).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2024 Copyright held by the owner/author(s).

0730-0301/2024/7-ART89

<https://doi.org/10.1145/3658151>

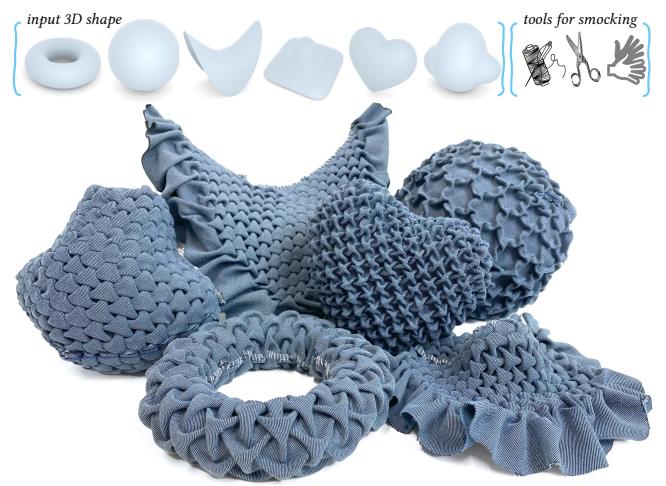


Fig. 1. Photo of fabricated 3D shapes realized by seamless smocking.

approximating a given shape. Typically, validity is measured by how much the result is developable, which becomes the primary constraint in the process.

Fabric is also commonly used for shape approximation in contexts such as designing plush toys [Mori and Igarashi 2007], framework [Zhang et al. 2019] or garments [Pietroni et al. 2022]. The problem is typically approached by computing cuts to flatten the input surface, with special consideration given to the locations and shapes of seams for fabrication purposes. Textile materials are more forgiving than the virtually inextensible paper and cardboard, affording the realization of a broader class of surfaces through the commonly accessible fabrication process of sewing.

In this work, we explore *fabric tessellation* to realize freeform surfaces (see Fig. 1), combining the richness of origami tessellation and the ease of manipulation of fabric materials. The flexibility of cloth makes it much easier to create intricate patterns through simple stitching, in comparison to paper, which demands precisely computed crease lines and careful handling to avoid damaging the

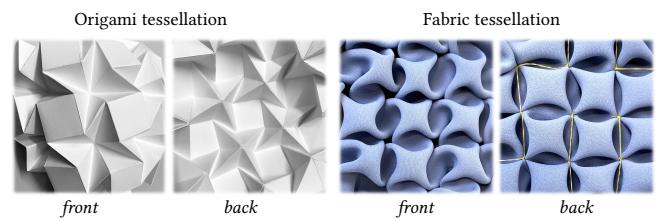


Fig. 2. Origami tessellation (*left*) and fabric tessellation (*right*). We show the front and back view of the fabrication using the WATERBOMB pattern.

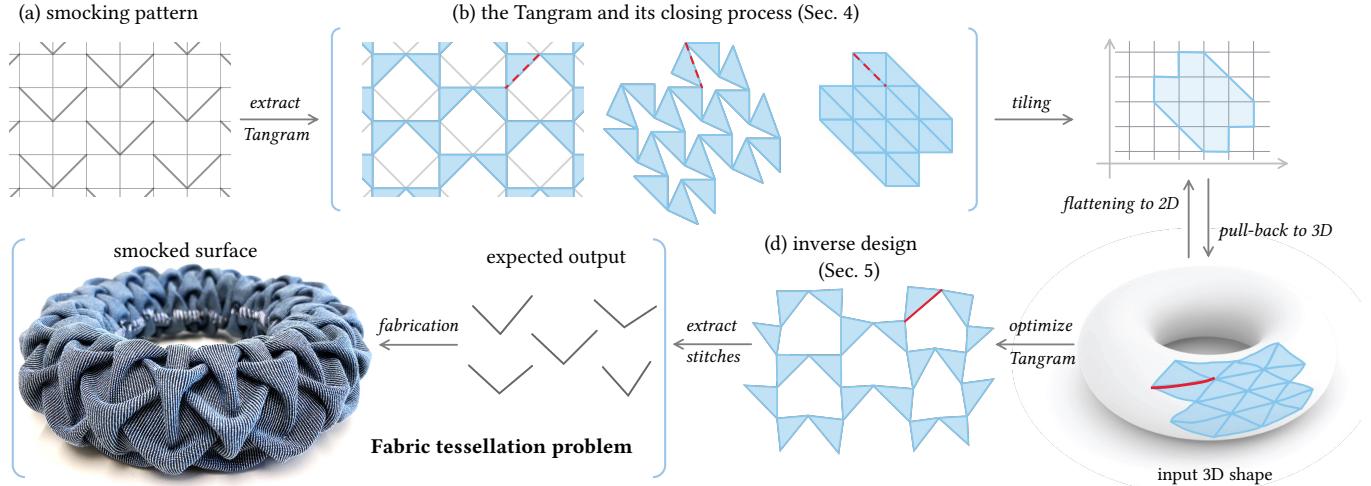


Fig. 3. The inverse design problem for fabric tessellation. We solve for a 2D smocking pattern, such that, after fabrication, the smocked result closely approximates the input 3D shape with regular pleats. Our solution has two important components: (1) a novel formulation of the so-called Tangram graph, which connects the 2D smocking pattern with the 3D smocked result (Sec. 4); (2) an optimization scheme based on the Tangram to achieve the target geometry and realize the input freeform surface (Sec. 5). Red (dashed) lines highlight the combinatorial equivalence across the different configurations of the Tangram.

material (see Fig. 2, left). We employ *smocking* [Durand 1979; Ren et al. 2024], which is a traditional pleating technique, where various points on the textile are stitched together in certain patterns and secured with knots to create geometric texturing (see Fig. 2, right). Conventional smocking results in a pleated, but overall inherently flat surface shape. In our fabric tessellation approach, we generate curvature by computationally modifying the smocking pattern, such that the pushed-out pleats lead to local area deficits or excesses (see Figures 3 and 4). This approach differs from origami tessellation, where patches of paper are folded inwards and are mostly hidden, such that the aesthetics of such patches does not matter. Smocking also significantly departs from the standard garment or plush toy fabrication practice, where *separate* patches are stitched along seams to achieve the desired curvature.

Our goal is to address the inverse design problem of fabric tessellation: given a freeform surface and the user's preferred smocking pleat type, we compute a 2D smocking pattern: a set of instructions on which list of points should be stitched together (see Fig. 4). This pattern, when fabricated, ensures that the overall shape of the final result closely approximates the input shape and exhibits the desired pleated texture. The unique challenge in fabric tessellation is how to approximate the input shape while retaining regular and visually pleasing pleats. Introducing singularities in the tessellation becomes necessary when discretizing surfaces with high Gaussian curvature and non-trivial topology, but traditional smocking does not define singularities and seamless designs of curved and non-disk shapes.

In this work, we investigate the explicit control of pleat shapes to attain a target surface metric. Additionally, we formulate seamless smocking with singularities and show a computational method to create such smocking patterns. Our main contributions are:

- We formulate the inverse design problem of realizing freeform surfaces through fabric tessellation using smocking. Our formulation accommodates various smocking patterns, and is designed to achieve seamless pleating.
- We introduce a formalization for digital smocking, called a Tangram, that connects the 2D smocking pattern and its corresponding 3D smocked result, enabling solving for the inverse design.
- We present an optimization algorithm for the inverse design problem, yielding a 2D smocking pattern that faithfully approximates the input surface after fabrication.

Our method is simple yet effective, as validated by physical fabrication. We also offer a digital visualization tool for previewing the smocked results in 3D. We showcase multiple applications for fabric tessellation, demonstrating its applicability across various domains.

## 2 RELATED WORK

### 2.1 Surface approximation

Surface approximation has many applications ranging from digital fabrication to the design of freeform architecture. Various materials or forms have been explored to approximate an input 3D surface, including paper [Demaine and Tachi 2017; Jiang et al. 2020], fabric [Jourdan et al. 2020; Scherer 2019; Zhang et al. 2019], stripe structures for beams [Liu et al. 2023; Panetta et al. 2019] or elastic ribbons [Ren et al. 2021], shell structures [Brancart et al. 2015; Chen et al. 2023; Ren et al. 2022], tensegrity structures [Pietroni et al. 2017; Shimoda et al. 2023] and auxetic materials [Chen et al. 2021; Konaković et al. 2016, 2018]. In several works [Chen et al. 2021; Panetta et al. 2019; Ren et al. 2022, 2021; Zhang et al. 2019], material properties are taken into account, often through physical simulation, to ensure the feasibility of the proposed solutions, while others are mainly formulated as geometry problems to achieve the prescribed target curvatures and shape [Choi et al. 2019; Dudte et al. 2016;

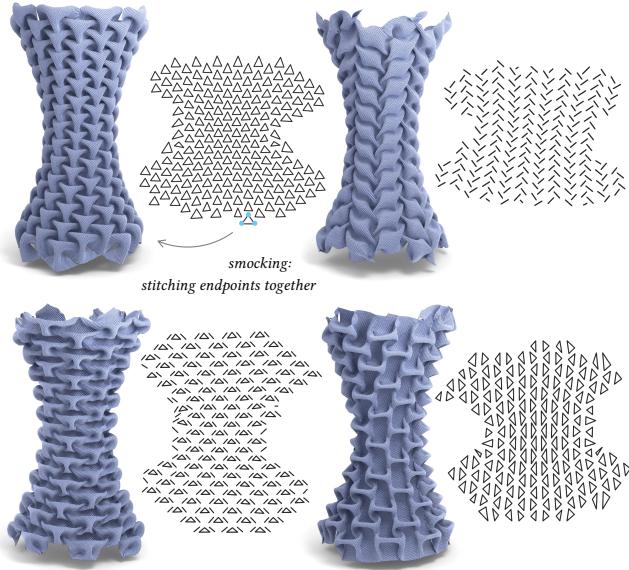


Fig. 4. A hyperbolic surface smocked by different patterns. We show the optimized smocking patterns and a digital preview of the smocked results. The fabrication process comprises the stitching together of line segments, followed by securing them with a knot.

Jiang et al. 2020; Konaković et al. 2018]. In this work, we explore the use of *fabric* materials to approximate a target 3D surface by simply stitching points on the fabric together. Since the resulting shape after smocking is primarily governed by the configuration of the stitches [Ren et al. 2024], we do not consider specific textile material properties in our formulation.

To approximate a *non-developable* surface using *planar* sheets such as paper and fabric, several common strategies exist: “subtracting” local regions through folding [Demaine and Tachi 2017], cutting the material open so that the expanded pattern when pulled, together with the opening holes, approximates the target surface [Choi et al. 2019; Konaković et al. 2018], or employing a hybrid strategy that involves carefully cutting holes out of the paper and folding it to close the holes, ultimately achieving the target surface shape [Jiang et al. 2020]. In this paper, we employ a distinctive strategy that fully utilizes the entire fabric without concealing or cutting out “unneeded” patches. Groups of points are stitched together to create local curvature, whereby small patches of fabric pop out and form pleats. Consequently, the fabricated result after stitching effectively approximates the input surface and possesses regular and aesthetically pleasing pleats. This special fabric manipulation technique is called smocking.

## 2.2 Smocking

Smocking is a surface embroidery technique that serves both decorative and functional purposes, enhancing the stretch and volume of the fabric while exhibiting geometric pleats [Durand 1979; Spufford and Mee 2017; Toplis 2021]. Within the realm of smocking, Canadian smocking stands out as a distinct genre characterized by the creation of volumetric and geometric textures through localized

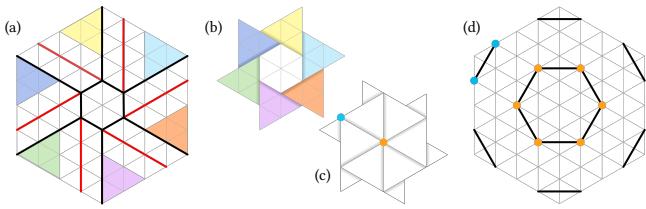
stitches, making it a subject of interest in various scientific fields, including social science [Bauer and Elsey 1992; Elbyaly and Elfeky 2022; Joseph et al. 2011], computational design and fabrication [Efrat et al. 2016; Lind 2019; Scherer 2019], simulation [Kim 2020], and digital preview [Ren et al. 2024; Zhou et al. 2024]. In particular, Ren et al. [2024] propose an efficient automatic method to preview the smocked fabric based on a given smocking pattern. Our study focuses on solving the *inverse problem* of smocking design.

Scherer [2019] proposed the first attempt to address the inverse problem for a specific pattern, namely Resch’s pattern [Resch 1968; Tachi 2013]. Scherer’s approach takes a regular triangle mesh without singularities as input and lays out its triangle faces on a flat hexagonal grid constructed from the 3D mesh dual. Subsequently, the triangles are scaled and rotated to align the appropriate vertices, and stitching lines are extracted to construct the smocking pattern. This approach is tested on a hyperboloid surface, demonstrating its effectiveness. However, it is unclear how to generalize this method to other smocking patterns and how to handle singularities or irregular input meshes. Additionally, the method lacks explicit control over the produced pleats, resulting in highly distorted smocked pleats in the realized surface (see [Scherer 2019, Fig. 10] and cf. Fig. 4). In this work, we aim to provide a general, principled solution capable of handling various smocking patterns and mesh discretizations.

## 2.3 Origami tessellation and shadowfolds

Origami tessellation is a genre of *paper* folding that combines the principles of origami with the concept of tessellation patterns. The Japanese origami artist Shuzo Fujimoto was one of the key figures who popularized tessellation as a distinct branch of origami. In the 1970s and 1980s, Fujimoto introduced new folding techniques that enabled the creation of intricate geometric patterns [Fujimoto 1978; Fujimoto and Nishiwaki 1982]. The art of origami tessellation was further expanded by artists such as Paul Jackson [Jackson 1989], Eric Gjerde [Gjerde 2008] and Chris K. Palmer [Rutzky and Palmer 2011]. The computational aspects of origami tessellation were formalized by Bateman [2002], who built upon the methods developed by Paulo Barreto [Barreto 1997] and Chris K. Palmer [Demaine and Demaine 2002; Rutzky and Palmer 2011]. They presented a computational algorithm to solve for the crease pattern from tiling, providing a formalized approach to analyzing and generating origami tessellations. The feasibility of the computational origami tessellation based on a dual-primal graph was later explored by Lang [2017]. Additionally, origami tessellation has been studied to create complex folded structures in 3D [Dudet et al. 2016; Tachi 2009, 2013].

The shadowfolds technique is a *fabric* folding method inspired by origami tessellation, often used in various forms of fabric art such as quilting, textile design, and fashion [Rutzky and Palmer 2011]. Coined by Palmer in 1995, the term *shadowfolds* refers to the use of pleated cloth, often translucent, which allows light to pass through at varying levels. Rutzky and Palmer [2011], and Wu [2022] note that woven cloth fibers possess greater flexibility than paper fibers, making textiles an ideal medium for creating intricate tessellations with relative ease. Instead of gently folding the fabric along mountain and valley folds (as is done in paper folding), shadowfolds involve sewing multiple points in the fabric and flattening



**Fig. 5. Paper folding vs. fabric stitching.** (a) shows the crease pattern of a hexagon Fujimoto twist for paper folding, with black (red) lines representing the mountain (valley) folds, respectively. (b) and (c) show the front and back views of the folded result. The visible region in the front view (b) is highlighted with the same color scheme as in (a). Multiple points in the folded result coincide, such as the set of orange (blue) points highlighted in (d), which correspond to the orange (blue) point in (c). This allows for the design of a smocking pattern (black stitching lines in (d)) that mimics the origami result when smocked, see [Rutzky and Palmer 2011, page 17].

the resulting pleats and twists, either through sewing or ironing. This characteristic aligns the shadowfolds technique with the art of smocking. In Fig. 5 we show such an example, where the smocked fabric result from pattern (d) looks similar to the folded result from pattern (a). Recent research investigates spatially-varying density for shadowfolds patterns [Wu 2022], and online creators explore applying shadowfolds techniques to fabricate 3D shapes using folded fabric [En Why See 2009]. Our work can be used to realize a general 3D shape using the shadowfolds technique.

## 2.4 Surface tiling and remeshing

Unlike the *Origamizer* [Demaine and Tachi 2017], which aims to find feasible crease patterns for any input polyhedron mesh, many surface realization methods take regularly meshed surfaces as input. These include quad meshes [Jiang et al. 2020], triangle meshes [Konaković et al. 2018; Ren et al. 2022], or other special regular tessellations [Chen et al. 2023]. Our proposed method also involves parameterization-based meshing; we compute a new surface mesh that corresponds to a planar tessellation (after cutting and flattening) and use it to optimize for the smocking pattern. We briefly review recent works on surface tiling and regular remeshing. Surface remeshing can be obtained by computing a parameterization [Floater and Hormann 2005; Hormann et al. 2007], or designing a directional field [Jakob et al. 2015; Vaxman et al. 2016] for the input triangle mesh. For example, Sawhney and Crane [2017] propose a mesh parameterization method that allows better control over the flattening process by careful boundary handling. This method has been successfully used for applications involving conformal remeshing [Konaković et al. 2018; Ren et al. 2022]. Jiang et al. [2015] propose to optimize the planarity of given polyhedral patterns with regularizers to tile a given surface. Peng et al. [2019] present a framework to tile a mesh using a combination of quads and triangles, presenting artistic decorative patterns. Meekes and Vaxman [2021] propose a mesh tiling method using periodic or aperiodic patterns. Drawing a connection to our work, in our case the smocked surface can be seen as a special kind of tessellation, and we therefore make use of regular remeshing methods in our inverse design approach.

## 3 FABRIC TESSELLATION

### 3.1 Problem formulation

Similar to the 2D crease pattern in the origami tessellation, the chosen 2D smocking pattern plays an important role in the fabric tessellation. We represent a *smocking pattern* as a list of 2D stitching lines, where each stitching line is a set of 2D points connected by line segments (see Fig. 3(a)). Smocking, the technique used to realize the smocking pattern on a piece of fabric, involves sewing together the points of each stitching line and securing them with a knot. More detailed descriptions of smocking can be found in [Durand 1979; Ren et al. 2024]. The local stitches fold the fabric into *pleats*, which should be as regular as possible to keep the result uniformly looking. Coincidentally, smocking patterns are usually regular tilings of a so-called *unit smocking pattern*, although non-tiled designs are also possible, as shown in [Ren et al. 2024].

The *forward problem* for fabric tessellation is defined in [Ren et al. 2024]: given a 2D smocking pattern, the objective is to predict the resulting 3D geometry of the fabric after sewing. Our goal is to formulate and solve the *inverse design problem* for fabric tessellation. Given a 3D freeform surface, we aim to compute the 2D smocking pattern such that, when fabricated, the smocked result closely approximates the input shape while exhibiting regular pleats.

*Challenges.* The inverse design problem for fabric tessellation is nontrivial to formulate. Our objective is to compute the 2D positions of an *unknown* number of stitching lines, which are *disconnected*, to ensure that the smocked result *after fabrication* closely approximates the input shape. Formulating such objectives into an optimization problem is challenging, as the variables (the positioning of the stitching lines) and the goal (shape approximation error) are only *indirectly* related to each other through an involved fabrication step. Another challenge is to simultaneously achieve visually appealing smocked pleats while accurately approximating the input shape (see Fig. 15). The regions in the fabric that correspond to the pleats after fabrication become disjoint after sewing, as demonstrated in [Lind 2019] and [Ren et al. 2024, Fig. 8]. As such, an intuitive solution is to partition the fabric into different regions, then force part of the regions to achieve an approximation of the input surface and regularize the remaining regions to achieve well-shaped pleats. However, how to exactly partition the flat fabric into regions with semantically meaningful correspondences to the final smocked result is a nontrivial problem. Additionally, it is difficult to formulate error measures for shape approximation and pleat regularity, since the smocked result is unknown. Finally, a regular tessellation of a closed surface, or an open surface with boundary alignment, must inherently admit singularities, giving rise to the previously unattempted problem of seamlessly incorporating smocking with singularities.

*Potential solutions.* A common strategy for inverse design problems involves adjusting the input based on a physically simulated result [Chen et al. 2021; Panetta et al. 2019; Ren et al. 2022, 2021; Zhang et al. 2019]. One potential approach for the inverse design of fabric tessellation is to modify an initial, regularly tiled 2D smocking pattern according to the shape difference between the simulated smocking result and the input surface. However, as demonstrated in [Kim 2020; Ren et al. 2024], general-purpose cloth simulators

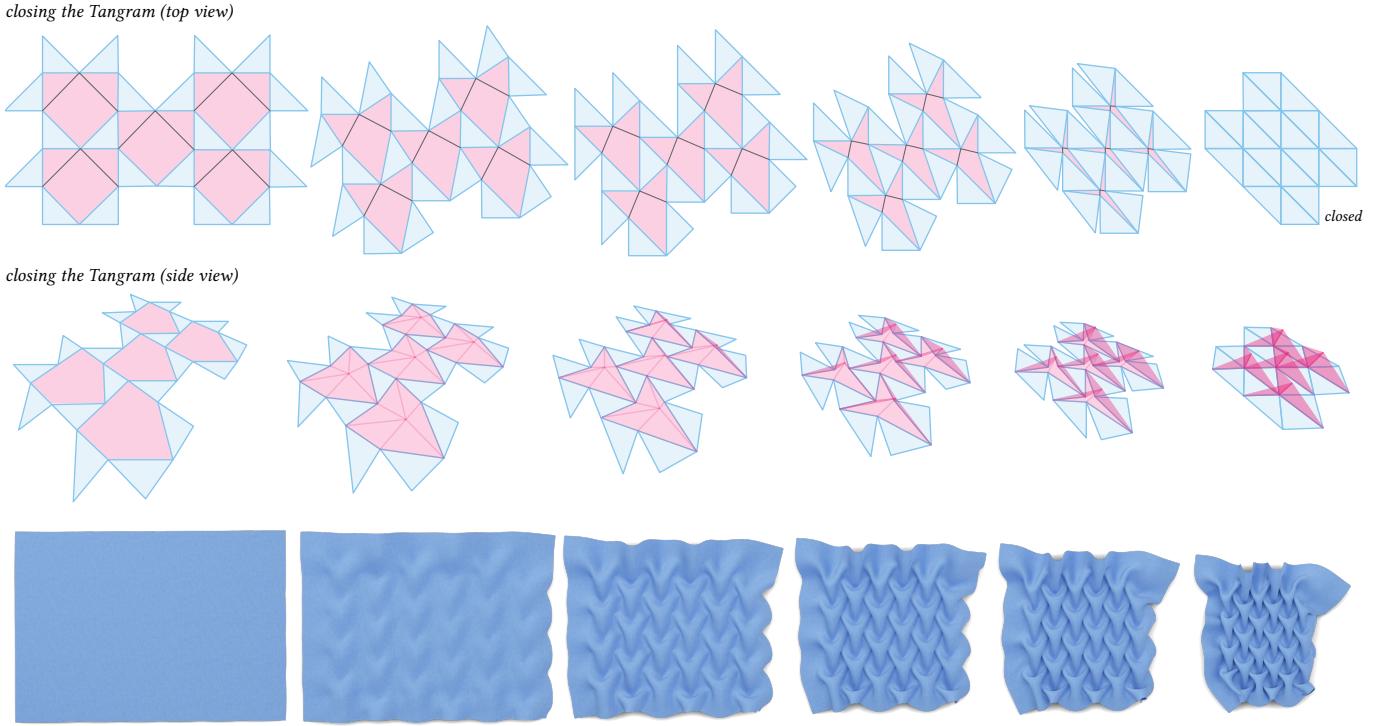


Fig. 6. The Tangram-graph paradigm provides an intuitive representation of how pleats are formed during smocking (bottom row). As the stitching lines (colored gray in the top row) become shorter, they drag the adjacent blue underlay faces and make them rotate, forcing the red pleat faces to shrink in 2D (top row) and pop up to form 3D voluminous pleats (middle row).

struggle with the stitching constraints imposed by smocking. The intricate folds formed during smocking necessitate significant collision handling, making it impractical for inverse design, where the simulation process would be invoked in each iteration. Another potential solution is to adapt the preview method designed for the forward problem by Ren et al. [2024], i.e., utilize their computed smocked result from a regular pattern to tackle the inverse design problem. However, the method by Ren et al. [2024] involves discrete graph operations, and part of the computed smocked result is inherently constrained to lie in 2D. Generalizing this pipeline to 3D surfaces poses challenges. Appendix A provides detailed discussions on why this approach is not suitable for the inverse design.

### 3.2 A Tangram approach to fabric tessellation

Our solution to the inverse design problem addresses the following challenges: (1) how to mathematically describe the smocked result and ensure its closeness to the target shape, (2) how to regularize the pleat shapes, and (3) how to accommodate necessary singularities for curved input surfaces.

A 2D smocking pattern can be decomposed into two parts: underlay faces, which resemble a Tangram pattern (e.g. shown in Fig. 6) and connect to each other through a hinging vertex or edge, and pleat faces as the complementary part. We formalize this observation with a so-called *Tangram graph*. The Tangram graph provides an intuitive interpretation of the fabrication of a smocking pattern: as the length of the stitching lines approaches zero, the underlay faces join

together and form a combinatorial mesh-like structure, while the pleat faces are forced to bend out of the 2D plane to create 3D pleats (see Fig. 6). We refer to the Tangram as *closed* when the smocking constraints are satisfied, i.e., the stitching lines have zero length, indicating that the stitching points have been sewn together. In Fig. 7, we show the interior structure of the smocked torus presented in Fig. 3. The apparent structure aligns with the closed Tangram, resembling a triangle mesh. (Note that we left the top part of the torus unsmocked and open to be able to turn it inside out.)



Fig. 7. Smocked torus interior.

This closing process of the Tangram establishes a connection between the 2D smocking pattern and the 3D structure of the smocked result, offering two crucial benefits: (1) it enables the optimization of the 2D smocking pattern with regularizers designed for the 3D smocked result; (2) as mentioned earlier, once the Tangram is closed, the underlay faces form a mesh-like structure, which can effectively express curvatures or target shapes. Therefore, our goal is to deform the shape of the Tangram graph in a way that its closed configuration approximates the target shape.

To accurately approximate a given surface, we establish a mapping between the 2D domain and the 3D surface shape by remeshing

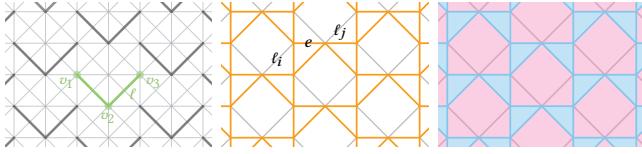


Fig. 8. **Left:** the ARROW smocking pattern, where we highlight one stitching line  $\ell = \{v_1, v_2, v_3\}$  in green. **Middle:** we highlight all underlay edges, connecting different stitching lines, in yellow. For example, the edge  $e$  connects two different stitching lines  $\ell_i$  and  $\ell_j$ . These yellow underlay edges partition the smocking pattern into disjoint regions. **Right:** we color the *pleat faces*, the regions that contain stitching lines, in pink, and the remaining regions, the *underlay faces*, in blue.

the surface with a semi-regular pattern derived from the closed configuration of the Tangram, and tiling the plane with a combinatorially-equivalent mesh where singularities have been cut out, as illustrated in Fig. 3. Some smocking patterns yield closed Tangrams with 3/4/6-symmetry, in which case employing a *seamless* parameterization [Bommes et al. 2009; Meekes and Vaxman 2021] allows us to smock a target shape with the necessary singularities. Finally, we modify the shape of the Tangram in its open configuration (i.e., the original 2D smocking pattern) such that it can realize the remeshed surface upon stitching, effectively approximating the target surface. See Fig. 3 for an overview of our method. Note that our Tangram graph is fundamentally different from the smocked graph proposed by Ren et al. [2024] (see Appendix A for more detailed discussion).

As the optimized Tangram mainly captures the coarse structure of the smocked result, we also provide a preview tool to visualize the precise geometry of the smocked pleats (e.g., Fig. 6, bottom). This is achieved by deforming the higher-resolution fabric guided by closed Tangram using ARAP [Sorkine and Alexa 2007], similar to the approach presented in [Ren et al. 2024]. Appendix B further discusses how the Tangram can be used to determine whether a given smocking pattern is well-constrained, which provides insights for pattern design.

*Paper structure.* In Sec. 4, we introduce notation and define the Tangram graph. In Sec. 5, we propose a formulation and a solution method for the inverse design problem. In Sec. 6 we provide a preview tool to visualize the smocked results after fabrication. We showcase digital results and physical fabrications, and discuss potential applications in Sec. 7, concluding our paper with Sec. 8.

## 4 SMOCKING PATTERNS AND TANGRAMS

We first introduce notations and provide a brief review of smocking design. We then explore an approach to solve both the forward problem and the inverse design of fabric tessellation using a reduction of the smocking pattern to a Tangram tiling.

### 4.1 Smocking patterns

A smocking pattern is delineated on a regular lattice drawn on fabric. We follow [Ren et al. 2024] and represent a *smocking pattern*  $\mathcal{P} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$  by a graph with vertex set  $\mathcal{V}$  and edge set  $\mathcal{E}$  that coarsely represents a piece of fabric, and the *stitching line* annotations  $\mathcal{L}$ . A stitching line  $\ell \in \mathcal{L}$  is a subset of vertices in  $\mathcal{V}$  that are to be stitched

together during fabrication, e.g., the line  $\ell = \{v_1, v_2, v_3\}$  in Fig. 8 (left). The physical smocking process entails stitching all vertices in every stitching line together, so their mutual distance becomes zero. Following [Ren et al. 2024, Def. 4.1 and 4.2], we categorize the vertices and edges in the smocking pattern:

**Definition 4.1.** A vertex  $v \in \mathcal{V}$  is called an **underlay vertex** if it belongs to a stitching line, and a **pleat vertex** otherwise. An edge  $e \in \mathcal{E}$  is called an **underlay edge** if it connects two *different* stitching lines. The set of all underlay edges is denoted as  $\mathcal{E}_u$ .

The vertex set  $\mathcal{V}$  can then be partitioned into two groups: the underlay vertices  $\mathcal{V}_u = \ell_1 \cup \dots \cup \ell_k$ ,  $\forall \ell_i \in \mathcal{L}$ , and the pleat vertices  $\mathcal{V}_p = \mathcal{V} \setminus \mathcal{V}_u$ , where  $k = |\mathcal{L}|$  is the number of stitching lines.

### 4.2 The Tangram graph

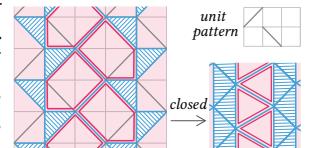
We construct the Tangram graph  $\mathcal{T} = (\mathcal{V}_u, \mathcal{E}_u)$  from a given smocking pattern  $\mathcal{P} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$ , where  $\mathcal{V}_u \subset \mathcal{V}$  is the set of underlay vertices and  $\mathcal{E}_u \subset \mathcal{E}$  is the set of underlay edges. In Fig. 8 (middle) we highlight the underlay edges  $\mathcal{E}_u$  in yellow. The underlay edges partition the fabric plane into regions, or *graph faces*.

**Definition 4.2.** We call a region in the smocking pattern an **underlay face** if it is bounded by underlay edges and does not contain any stitching lines. Similarly, a region is called a **pleat face** if it is bounded by underlay edges and contains some stitching lines.

Fig. 8 (right) highlights the underlay and pleat faces in blue and pink, respectively. We observe that during the smocking process, as the length of stitching lines approaches zero, the underlay faces are drawn and brought closer together. We call it the *closing process* of a Tangram graph.

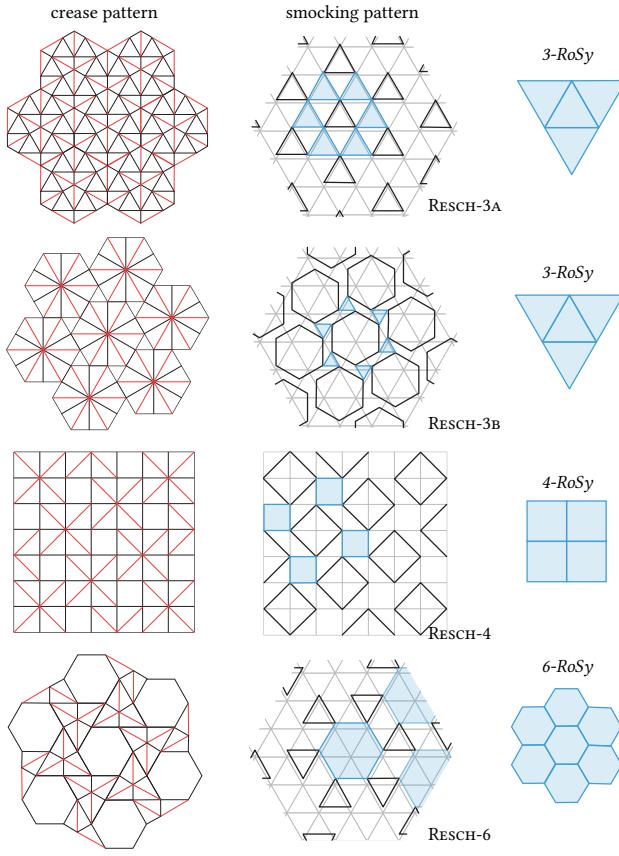
**Definition 4.3.** The Tangram of a smocking pattern is **closed** when the underlay faces and edges are *rividly* rotated to reach a configuration where all stitching lines have zero length. Similarly, we say that the initial Tangram is in the **open** configuration.

Note that the open Tangram is embedded in 2D, as it identifies with the original smocking pattern. The closed Tangram, however, encodes the structure of the 3D smocked result, marking the completion of the sewing process, as the length of all stitching lines is reduced to zero. Fig. 6 shows the intermediate steps of rotating the underlay faces until the Tangram of the ARROW pattern is closed. Fig. 9 shows another example of a Tangram of the BRAID pattern, where in the closed configuration, the pleat faces do *not* completely vanish as in the ARROW example. Fig. 9. Tangram of BRAID pattern. See Appendix B for more examples.



### 4.3 Computing the closed configuration of the Tangram

Deriving the closed Tangram directly from the smocking pattern is often nontrivial. For some smocking patterns, we can construct the closed Tangram using simple Euclidean geometry observations, by rotating the underlay faces according to stitching lengths and the law of cosines. In other cases, the closed Tangram can be effectively computed by an optimization process based on Def. 4.3. For a node



**Fig. 10. Resch's patterns.** Left: we show Resch's origami crease patterns [Resch 1968], where black (red) indicates the mountain (valley) folds (cf. Fig. 5 in [Tachi 2013]). Middle: the corresponding derived smocking patterns, where we highlight some of the underlay regions in blue and show their closed configuration on the right.

$v$  in the Tangram with an initial position  $\mathbf{x} \in \mathbb{R}^2$  in the open configuration, we denote its new position when the Tangram is closed as  $\tilde{\mathbf{x}} \in \mathbb{R}^2$ . Stacking all the node positions of the closed Tangram into a vector of unknowns  $\tilde{\mathbf{X}}$ , we formulate the optimization for closing the Tangram as follows:

$$\min_{\tilde{\mathbf{X}} \in \mathbb{R}^{|\mathcal{V}_u| \times 2}} \alpha_s E_{\text{stitch}}(\tilde{\mathbf{X}}) + \alpha_r E_{\text{rigid}}(\tilde{\mathbf{X}}), \quad (1a)$$

$$\text{where } E_{\text{rigid}}(\tilde{\mathbf{X}}) = \sum_{(i,j) \in \mathcal{E}_r} (\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|_2 - l_{i,j})^2, \quad (1b)$$

$$E_{\text{stitch}}(\tilde{\mathbf{X}}) = \sum_{\ell \in \mathcal{L}} \sum_{(i,j) \in \ell} (\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|_2 - \eta l_{i,j})^2. \quad (1c)$$

Here,  $l_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$  is the original distance between underlay vertices in the open configuration. The set  $\mathcal{E}_r$  in Eq. (1b) is the set of edges we collect that should transform rigidly during the closing process, namely, the underlay faces and edges according to Def. 4.3. To accommodate for non-triangular underlay faces, we triangulate them by adding diagonals to  $\mathcal{E}_r$  (see Fig. 8 (left)), such that rigidity is well-defined.

Eq. (1c) encourages all pairs of underlay vertices within the same stitching line to have a reduced length when closing the Tangram. We introduce the parameter  $\eta \in [0, 1]$  to indicate the progress of the stitching process: when  $\eta = 1$ , the open Tangram is the global optimum and no optimization is needed, while setting  $\eta = 0$  results in the closed configuration of the Tangram, as the expected length for stitching edges approaches zero;  $0 < \eta < 1$  gives the intermediate results, as shown in Fig. 6.

## 5 SMOCKING 3D SURFACES

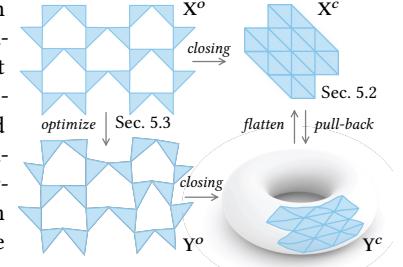
The closed Tangram of a regular smocking pattern is in essence a polygonal plane tiling, as illustrated in Fig. 6, 9, 10 and 16(c). A common approach to realize a 3D shape from plane tilings is through parameterization [Konaković et al. 2018; Meekes and Vaxman 2021]. We next show how we pull back the tiled grid to approximate the target shape, which allows us to optimize for the Tangram in its open configuration to fit the surface upon its closing.

### 5.1 Input & output

As input, we assume that we are provided a freeform surface  $\mathcal{M}$ , represented as a triangle mesh, and a unit smocking pattern. We obtain an initial smocking pattern  $\mathcal{P} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$  by evenly tiling the input unit smocking pattern on a regular grid in the 2D plane. Here we specifically focus on the Tangram graph  $\mathcal{T} = (\mathcal{V}_u, \mathcal{E}_u)$  instead of the entire grid, as the Tangram itself is sufficient for deriving and optimizing the smocking pattern. The vertices in  $\mathcal{V}_u$  have different embeddings in the open and closed Tangram. We use superscript to distinguish the closed Tangram  $\mathcal{T}^c = (\mathbf{X}^c, \mathcal{E}_u)$  from the open Tangram  $\mathcal{T}^o = (\mathbf{X}^o, \mathcal{E}_u)$  with their embedded positions:  $\mathbf{X}^o \in \mathbb{R}^{|\mathcal{V}_u| \times 2}$  are the vertex positions on the planar grid, and  $\mathbf{X}^c \in \mathbb{R}^{|\mathcal{V}_u| \times 2}$  are the positions obtained by solving Eq. (1a), i.e., when the Tangram is closed after stitching.

See Fig. 11 (top) for an illustration of the embeddings  $\mathbf{X}^o$  and  $\mathbf{X}^c$ . Note that  $\mathbf{X}^o$  and  $\mathbf{X}^c$  are in one-to-one correspondence, and represent different embeddings for the underlay vertices  $\mathcal{V}_u$  of the Tangram with known positions. The expected output is the 2D smocking pattern, extracted from the open Tangram  $\mathcal{T}^o = (\mathbf{Y}^o, \mathcal{E}_u)$  with a new embedding  $\mathbf{Y}^o \in \mathbb{R}^{|\mathcal{V}_u| \times 2}$ , such that in its closed configuration  $\mathcal{T}^c = (\mathbf{Y}^c, \mathcal{E}_u)$ , its geometry  $\mathbf{Y}^c \in \mathbb{R}^{|\mathcal{V}_u| \times 3}$  approximates the target surface. Fig. 11 (bottom) illustrates the notation. Here,  $\mathbf{Y}^c$  and  $\mathbf{Y}^o$  are the unknowns to be solved for.

*The algorithm.* The key steps to solving the inverse design are: (1) generating a semi-regular mesh  $\mathbf{Y}^c$  by seamless parameterization, (2) cutting  $\mathbf{Y}^c$  to a combinatorially-regular pattern with seams and embedding it in the plane with geometrically-regular faces as  $\mathbf{X}^c$ , (3) opening  $\mathbf{X}^c$  to obtain  $\mathbf{X}^o$ , and (4) optimizing  $\mathbf{Y}^o$  with  $\mathbf{X}^o$  as the initial solution, so that it reproduces the metric of  $\mathbf{Y}^c$  with



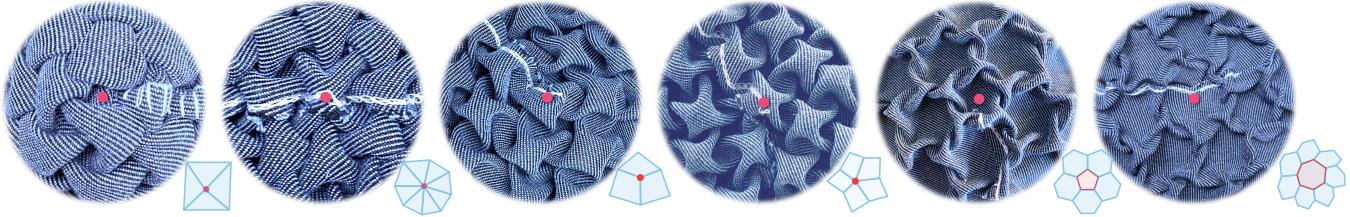


Fig. 12. Fabricated singularities using smocking with indices  $\frac{1}{3}, -\frac{1}{3}, \frac{1}{4}, -\frac{1}{4}, \frac{1}{6}, -\frac{1}{6}$  from left to right respectively. For each case, we show an illustration of the singularity in the closed Tangram. We highlight the singularity vertex ( $N = 3, 4$ ) or singularity face ( $N = 6$ ) in red.

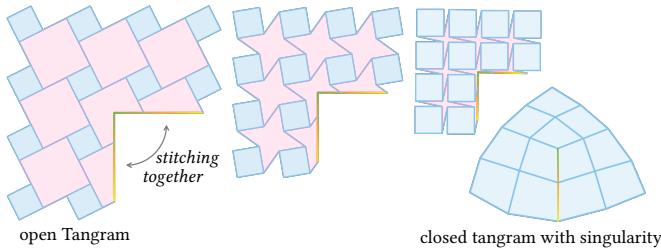


Fig. 13. Seamless smocking with a singularity of index  $\frac{1}{4}$  using WATERBOMB (RESCH-4) pattern. Here we show the closing process of Tangram. When the open Tangram (leftmost) is closed and the corresponding seams are sewn together, the closed Tangram (rightmost) has one singularity.

as-regular-as-possible pleats. Broadly, our algorithm has two components: computing the combinatorics of the closed pattern, which is controlled by the combinatorics of a surface semi-regular mesh with a few singularities, and optimizing for the open pattern such that its closed configuration reproduces the surface.

## 5.2 Seamless smocking patterns

**5.2.1 Symmetry in a closed Tangram.** Some closed Tangrams exhibit translational ( $N = 2$ ) or rotational ( $N = 3/4/6$ ) symmetries. For example, the closed Tangrams of ARROW (Fig. 6), BRAID (Fig. 9), and further patterns in Appendix B show translational symmetry whose structure can be described by 2-fields. Their closed Tangram exhibits an unconventional tiling of the plane, where the unit tile is composed of a mixture of polygons. Smocking patterns can be derived from the famous Resch origami patterns [Resch 1968], whose closed Tangrams exhibit 3-, 4-, or 6-rotational symmetry (see Fig. 10).

**5.2.2 Rotationally-symmetric smocking with singularities.** With rotational symmetries, we have more flexibility in designing smocking patterns for surfaces with arbitrary topologies that require singularities. We introduce *seamless smocking*, where the pleats exhibit identical patterns up to a rotation everywhere except at singularities. See Fig. 12 for some examples of the fabricated singularities with seamless pleats. Seamless smocking can only be achieved for patterns with rotational symmetry. Thus, we focus on Resch's smocking patterns, particularly when singularities are needed due to topology, or desired for better shape approximation. The closed Tangrams of Resch's patterns form a regular triangular, quadrilateral, or hexagonal grid, as shown in Fig. 10, allowing for the straightforward

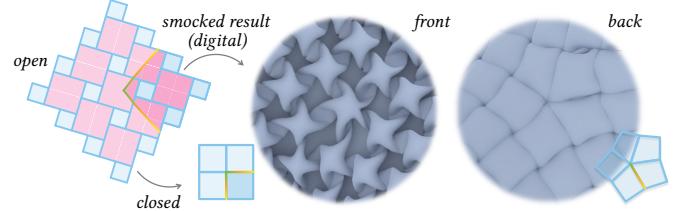


Fig. 14. A seamless smocking pattern that incorporates a singularity to address negative Gauss curvature has overlapping regions in the plane, visualized using additive opacity. The corresponding digitally smocked singularity is in front and back view. In physical fabrication, the overlapping regions are split and annotated on two pieces of fabric and sewn together.

encoding of conventional singularities. We can derive an open Tangram with seams such that, when it is closed and the corresponding seams are sewn together, the resulting closed Tangram encodes the designed singularities. Fig. 13 and 14 show an example of a seamless smocking pattern that leads to a singularity with positive and negative curvature, respectively. Our method enables this computationally by working with seamless parameterizations.

**5.2.3 Computing  $Y^c$  by semi-regular meshing.** The first main challenge is to construct a surface mesh  $Y^c$  that we consider to be a good geometric target that is combinatorially equivalent to the closed planar Tangram. For patterns that exhibit translational symmetry ( $N = 2$ ), we compute the mesh  $Y^c$  by pulling back a grid tiled with the closed Tangram onto the input shape. For patterns with  $N$ -symmetry, we compute  $Y^c$  that respects this symmetry (e.g., triangles for 3-RoSy, quads for 4-RoSy, and hexagons for 6-RoSy) and allows for singularities. Specifically, the mesh  $Y^c$  is generated by computing a seamless parameterization, following the method by [Meekes and Vaxman 2021]; broadly, we optimize a curl-free  $N$ -vector field that is as close as possible to a unit norm singularity-free 2-field or  $N$ -RoSy field with singularities (if necessary), integrate it into  $N$ -functions, and generate a mesh by pulling back a regular grid in the plane. We give details for the algorithm and our modifications of it in Appendix C for completeness.

**5.2.4 Solving  $X^c$ .** Next, we generate  $X^c$  by cutting  $Y^c$  open, with all singularities on the boundary of  $X^c$ , and storing the rotational constraints on the newly-created vertices and edges of the cut seams, for our consequent optimization. We immerse it in the plane by setting all faces as regular (for the given prescribed regularity of the chosen smocking pattern). Note that for a singularity-free mesh

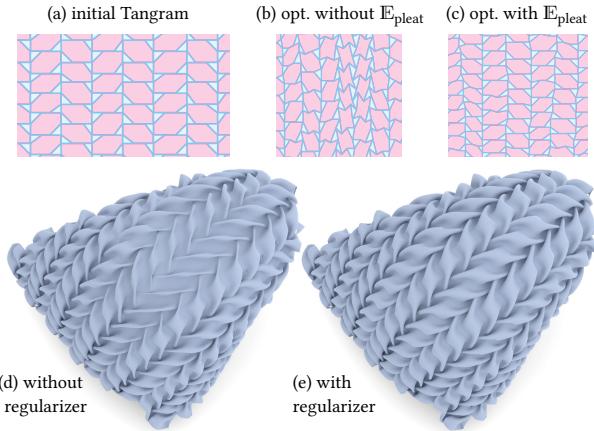


Fig. 15. Explicit control over pleat shapes is essential in solving the inverse design problem. Modifying the 2D smocking pattern to approximate a 3D shape can potentially damage the pleat shape if no regularizer is added, as shown in (d). For comparison, our formulation allows to effectively preserve the pleat shapes, shown in (e).

$\mathbf{Y}^c$ , the mesh  $\mathbf{X}^c$  is nothing but the part of the tiled grid in the plane that was pulled back to generate  $\mathbf{Y}^c$  to begin with. It is important to note that  $\mathbf{X}^c$  is not, in general, an embedding, since faces may be overlapping globally (see example in Fig. 14), but this is immaterial to our algorithm.

We can restore the structure of the closed Tangram  $\mathbf{X}^c$  from the tiled grid by splitting the faces and reestablish their connectivity based on the pattern. The open configuration of the Tangram  $\mathbf{X}^o$  can then be obtained by rotating the Tangram faces and edges until they achieve the structure of the original Tangram of the pattern. In the presence of singularities, cutting the seamless closed Tangram open is straightforward, as shown in Fig. 13 (right). However, computing the seamless  $\mathbf{X}^o$  is then not as trivial. This process involves adding additional pleat faces to the matching seams, as shown in Fig. 13 (left). Specifically, when the matching seams are sewn together, the combined shapes of the newly added pleat faces (except for those at the singularity) match together to pleat faces in the original pattern. For example, in Fig. 13 (left), the pleat faces along the seam are seamlessly combined to form a perfect square when the matching seams are sewn together. The pleat face around the singularity has a missing piece: the angle between the cut seams matches the cone angle of the singularity, e.g.  $\frac{3}{2}\pi$  in the example mentioned above.

### 5.3 Tangram optimization

Since  $\mathbf{Y}^c$  already approximates the geometry of the target surface, the remaining problem is how to modify the embedding of the open Tangram,  $\mathbf{X}^o$ , such that its closed configuration exactly matches  $\mathbf{Y}^c$ , while maintaining the regularity of the pleats. For simplicity of notation, we use the bold font  $\mathbf{e}$  to represent an edge vector and normal font  $e$  to represent its edge length, e.g.,  $\mathbf{e}_{ij} = \mathbf{y}_i - \mathbf{y}_j$  and  $e_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|_2$ . Notation  $\angle(\mathbf{e}_1, \mathbf{e}_2)$  represents the angle between the two vectors  $\mathbf{e}_1$  and  $\mathbf{e}_2$ , measured in the range  $[0, 2\pi]$ .

**5.3.1 Shape energy.** Recall that the underlay edges and faces transform rigidly during the Tangram closing process (see Def. 4.3). Therefore, to achieve the closed embedding  $\mathbf{Y}^c$ , we need to optimize the edge lengths of the open Tangram configuration  $\mathbf{Y}^o$  to match those in  $\mathbf{Y}^c$ . We formulate the following shape energy:

$$\mathbb{E}_{\text{shape}}(\mathbf{Y}) = \sum_{(i,j) \in \mathcal{E}_u} \left( \frac{e_{ij}}{\|\mathbf{y}_i^c - \mathbf{y}_j^c\|_2} - 1 \right)^2. \quad (2)$$

**5.3.2 Pleat Energy.** To regularize the shape of the pleats and ensure that the smocked result is visually pleasing, we use the following energy to preserve the angles between two adjacent boundary edges of the pleat face:

$$\mathbb{E}_{\text{pleat}}(\mathbf{Y}) = \sum_{f \in \mathcal{F}_p} \sum_{(i,j),(j,k) \in f} \frac{1}{2\pi} (\angle(\mathbf{e}_{ij}, \mathbf{e}_{kj}) - \theta_{ijk})^2, \quad (3)$$

where  $\mathcal{F}_p$  is the set of pleat faces and the summation goes over all pairs of adjacent edges in a pleat face;  $\theta_{ijk}$  is the angle between the two edges from the original pattern. Fig. 15 shows the optimized Tangram (*top*) with and without this term  $\mathbb{E}_{\text{pleat}}$ , and its corresponding smocked result (*bottom*). We can see that  $\mathbb{E}_{\text{pleat}}$  can effectively regularize the shape of pleats and lead to improved results.

**5.3.3 Seam energy.** In case there are seams induced from the smocking pattern with singularities, we design the following term to ensure the corresponding cuts can be properly stitched together during fabrication:

$$\mathbb{E}_{\text{seam}}(\mathbf{Y}) = \sum_{(i,j) \in C} \left( \frac{e_{ij}}{e'_{ij}} - \frac{e'_{ij}}{e_{ij}} \right)^2 \quad (4a)$$

$$+ \sum_{(i,j),(j,k) \in C} \frac{1}{2\pi} (\angle(\mathbf{e}_{ij}, \mathbf{e}_{kj}) - \angle(\mathbf{e}'_{ij}, \mathbf{e}'_{kj}))^2 \quad (4b)$$

$$+ \sum_{(i,j) \in C, (j,k) \in C'} \frac{1}{2\pi} (\angle(\mathbf{e}_{ij}, \mathbf{e}'_{kj}) - \theta_j)^2, \quad (4c)$$

where  $e' \in C'$  is the duplicated edge corresponding to the edge  $e \in C$  along the cut. The first term, Eq. (4a), promotes the corresponding edges along the seam to have the same length. This standard equal-length energy for cuts prevents the formation of ruffles during stitching. The second term, Eq. (4b), maintains the angle between two consecutive edges along the cuts. This combined energy encourages any cut  $C$  to preserve its seamless compatibility to its duplicated counterpart  $C'$ . This property facilitates an easier stitching process during fabrication. The third term, Eq. (4c), encourages the angle between each pair of twin seams (at the shared vertex  $y_j \in C \cap C'$ ) to match the cone angle  $\theta_j$  of the corresponding singularity. It is important to note that the cuts always split pleat faces, while the underlay faces remain intact (see Fig. 14). As such, we incorporate (pleat) seamlessness as a soft regularizer instead of imposing hard constraints, recognizing that imperfect seams do not compromise the search space of optimizing underlay faces to reproduce the target shape. However, it might affect the regularity of the pleat shapes along the seams.

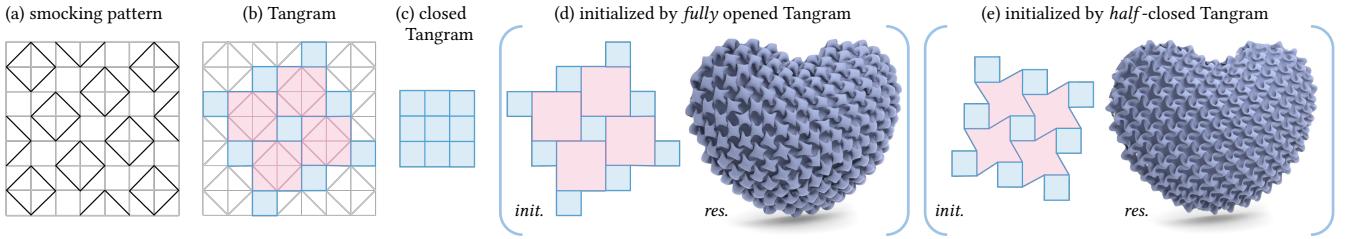


Fig. 16. (a) The WATERBOMB smocking pattern with stitching lines colored in black. (b) The derived Tangram graph, where the underlay (resp. pleat) faces are colored in blue (resp. red). (c) The Tangram in the fully closed configuration, which is used for mapping to the target shape. We then optimize the Tangram to reproduce the expected geodesics starting from two different initial configurations, the fully open Tangram in (d) and a half-closed Tangram in (e). Notably, in (e), the initial pleat regions (colored red) have smaller areas compared to those in (d), resulting in smaller-sized pleats in the final smocked result.

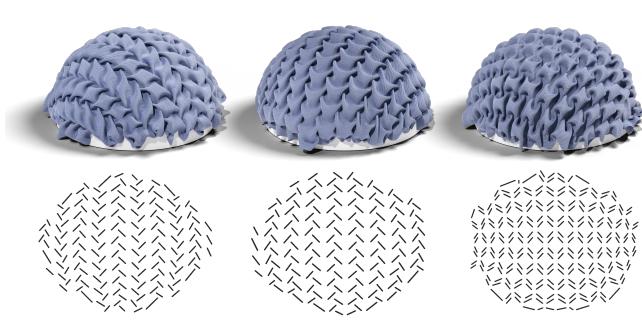


Fig. 17. Top: smocked hemisphere using LEAF, HEART, and Box pattern. Bottom: the corresponding stitching pattern.

**5.3.4 Full optimization.** Starting from the initial *opened* configuration  $\mathbf{Y}^o$ , we optimize the 2D embedding of the Tangram  $\mathbf{Y}$  using the following energy:

$$\mathbf{Y}^o = \arg \min_{\mathbf{Y} \in \mathbb{R}^{|\mathcal{V}_u| \times 2}} w_s \mathbb{E}_{\text{shape}}(\mathbf{Y}) + w_p \mathbb{E}_{\text{pleat}}(\mathbf{Y}) + w_c \mathbb{E}_{\text{seam}}(\mathbf{Y}). \quad (5)$$

The scales of all terms are comparable. We use the same set of parameters for all experiments. During optimization, we give a higher weight ( $w_s = 1$ ) to  $\mathbb{E}_{\text{shape}}$  due to its essential role in realizing the target shape, while  $\mathbb{E}_{\text{seam}}$  is assigned with a lower weight  $w_c = 0.1$  as imperfect seams can be tolerated. We initialize  $w_p = 100$  for  $\mathbb{E}_{\text{pleat}}$  to promote the preservation of the pleat shapes at the beginning of the optimization. During each iteration we minimize the total energy until convergence. At the end of the iteration, we decrease the weight of  $w_p$  by 20% to relax the pleat energy and allow further error reduction in  $\mathbb{E}_{\text{shape}}$  for better shape approximation.

The optimization process is stopped when  $\mathbb{E}_{\text{shape}}$ , the relative edge reproduction error summed over all edges, falls below a pre-defined threshold ( $10^{-4}$ ), or when a specified maximum number of iterations (100) is reached. Our algorithm is efficient and only takes seconds to solve for the smocking pattern. See Sec. 7 for further discussion. Note that we can also start the optimization from a half-closed Tangram configuration and obtain results with different size of pleats, as shown in Fig. 16.

As the stitching lines  $\mathcal{L}$  are defined as lists of underlay vertices, we can extract the modified stitching lines from the optimized embedding  $\mathbf{Y}^o$ , which can be then transported onto fabric for fabrication.

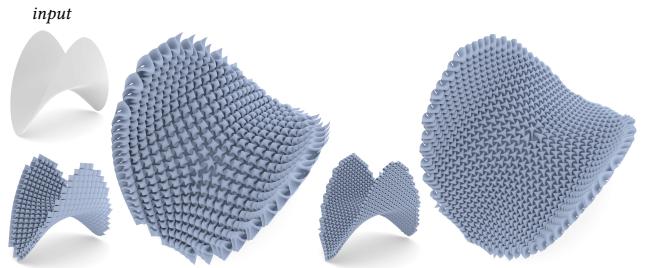


Fig. 18. A smocked “Pringle” shape using RESCH-4 and RESCH-3A pattern; each has one singularity.

## 6 A PREVIEW TOOL FOR SMOCKED RESULTS

Ren et al. [2024] introduced a preview tool for predicting the geometry of the smocked fabric from a regularly tiled pattern, explicitly constraining the underlay vertices to be embedded in 2D. This approach does not apply to our situation, as the underlay vertices (i.e., the closed Tangram) from the optimized pattern are embedded in 3D upon stitching. We create a preview of what our smocking result would approximately look like by emulating the closing process. We upsample and triangulate the smocking pattern to a much finer resolution and apply a seamless version of as-rigid-as-possible ARAP mesh deformation [Sorkine and Alexa 2007], to mimic its deformation from  $\mathbf{Y}^o$  to  $\mathbf{Y}^c$ , while

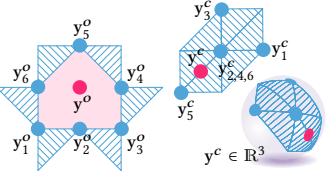
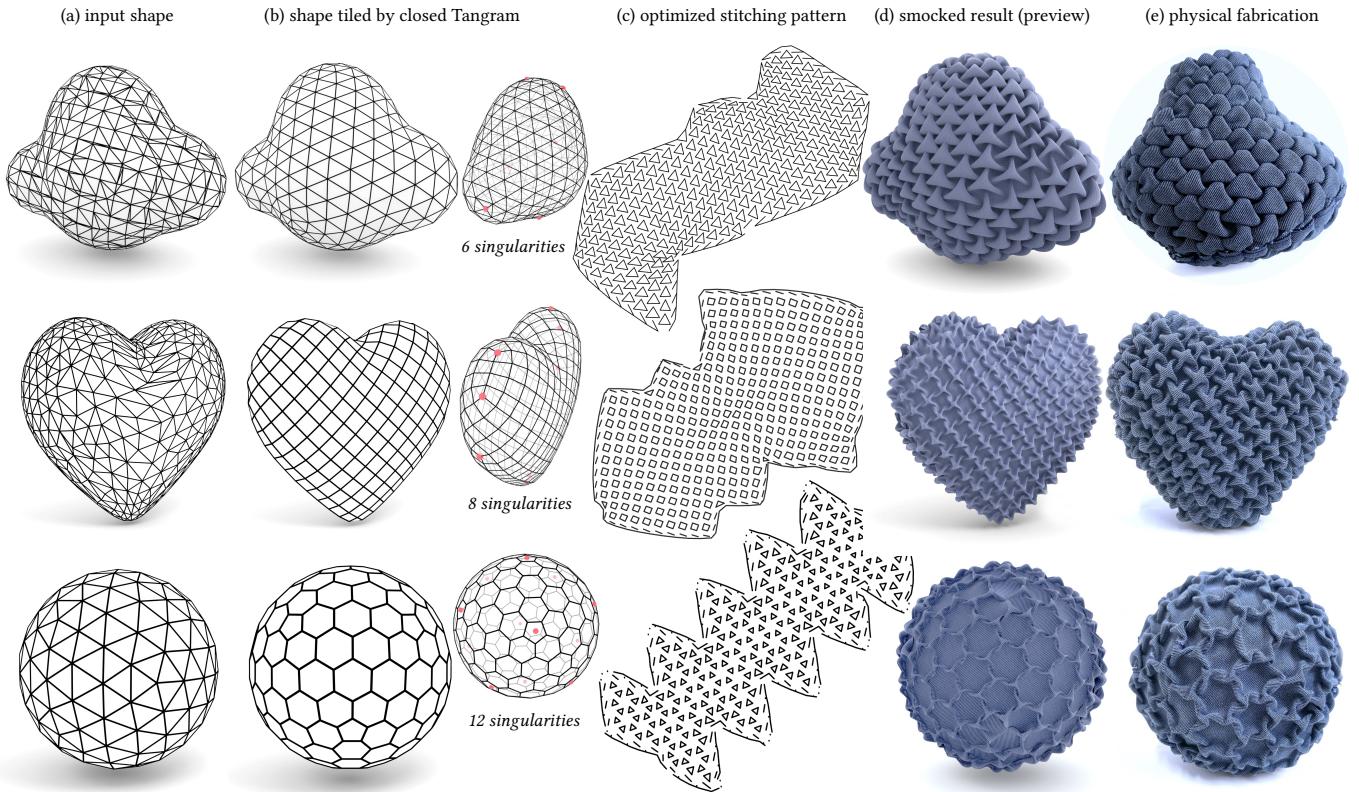


Fig. 20. MVC for the pleat nodes.

constraining the vertices to their prescribed positions. Given that the pleats are equally likely to pop out of the fabric plane in both directions (to the front side and to the back side of the fabric), we introduce additional positional constraints for the pleat faces to resolve this ambiguity: we sample one point in each pleat face in the open Tangram and estimate its position when the Tangram is closed, based on Mean Value Coordinates (MVC) [Floater 2003]. For example, as shown in Fig. 20, we sample a pleat vertex  $y$  inside the red pleat face, which is bounded by six underlay vertices with corresponding MVC  $w_i$  satisfying  $y^o = \sum_{i=1}^6 w_i y_i^o$ . When the Tangram is closed and these underlay vertices are embedded in new positions  $y_i^c$ , we can compute a reasonable estimate for its projected position



**Fig. 19. Seamless 3D smocking.** For a given shape (shown in column (a)), we run our algorithm discussed in Sec. 5 to obtain a seamless remeshing (b), with singularities highlighted in red. Column (c) shows the optimized stitching pattern, including both the stitching lines and the seams. Column (d) shows the smocked results using our preview tool and (e) shows the corresponding physical fabrications. Note that we leave some parts of the fabric unsmocked and open (including the top of the cloud shape, the tip of the heart shape, and the poles of the sphere shape) to allow us to flip the smocked results inside out.

on the 3D surface:  $y^c = \sum_{i=1}^6 w_i y_i^c$ . We then further shift  $y^c$  along the normal direction with height  $h$ ; the same procedure is applied to all pleat faces. Specifically,  $h$  is set to the average target edge length to ensure a reasonable height. The exact value is not critical, as the additional pleat constraints are only used in the first iteration of ARAP to eliminate directional ambiguity and are later discarded.

We implement a seamless version of ARAP to create a faithful preview along the seams when singularities are involved. In addition to the computed stitching lines, the smocking pattern also contains the corresponding seams  $C$  and  $C'$  as lists of edges that need to be stitched together. The seamless version considers the complete 1-ring neighborhoods around seams, leading to improved smoothness across seams in the previewed results. Specifically, since the seams are optimized to be rotationally symmetric (see Eq. (4)), we can find a rotation between corresponding edges on the seams, i.e.,  $e' = R e$  where  $e \in C, e' \in C'$ . These rotations allow us to transport the 1-ring neighborhood of a boundary vertex on seam  $C$  to the 1-ring neighborhood of its corresponding duplicated vertex on  $C'$ , and we merge them to get the full neighborhood information for boundary vertices. The new neighborhood is used both in computing the rotation in the local step and in the global Poisson stitching step, as in standard ARAP. This approach successfully eliminates the

artifacts that may arise from simply stitching the vertices via hard constraints.

## 7 RESULTS AND DISCUSSION

We implemented our algorithm in C++ using the libraries Directional [Vaxman et al. 2019], libigl [Jacobson et al. 2018], and Polyscope [Sharp et al. 2019]. All the experiments are carried out on a MacBook with an Apple M1 Max chip and 32 GB of memory. We set  $\eta = 0$  in Eq. (1c) and solve for the closed Tangram directly. Equations (1a) and (5) are solved using Newton's method with per element projected Hessian (calculated via autodiff). To fabricate the output smocking patterns, we use a laser cutter to engrave the stitching lines extracted from the optimized Tangram onto the fabric. Seam allowance is added to cuts if present. We first sew the corresponding cuts together and then smock the fabric by following the annotated stitching pattern. When fabricating a closed shape, we leave a small region unsmocked (e.g., shown in Fig. 3 and 19) to facilitate turning the smocked fabric inside out, as the stitching is done on the backside of the fabric. Please see the full implementation and the ready-to-smock stitching patterns for the 3D shapes we experimented with here: <https://github.com/segaviv/SmockingTessellations>, and see the accompanying video for various physical and digital fabrications.

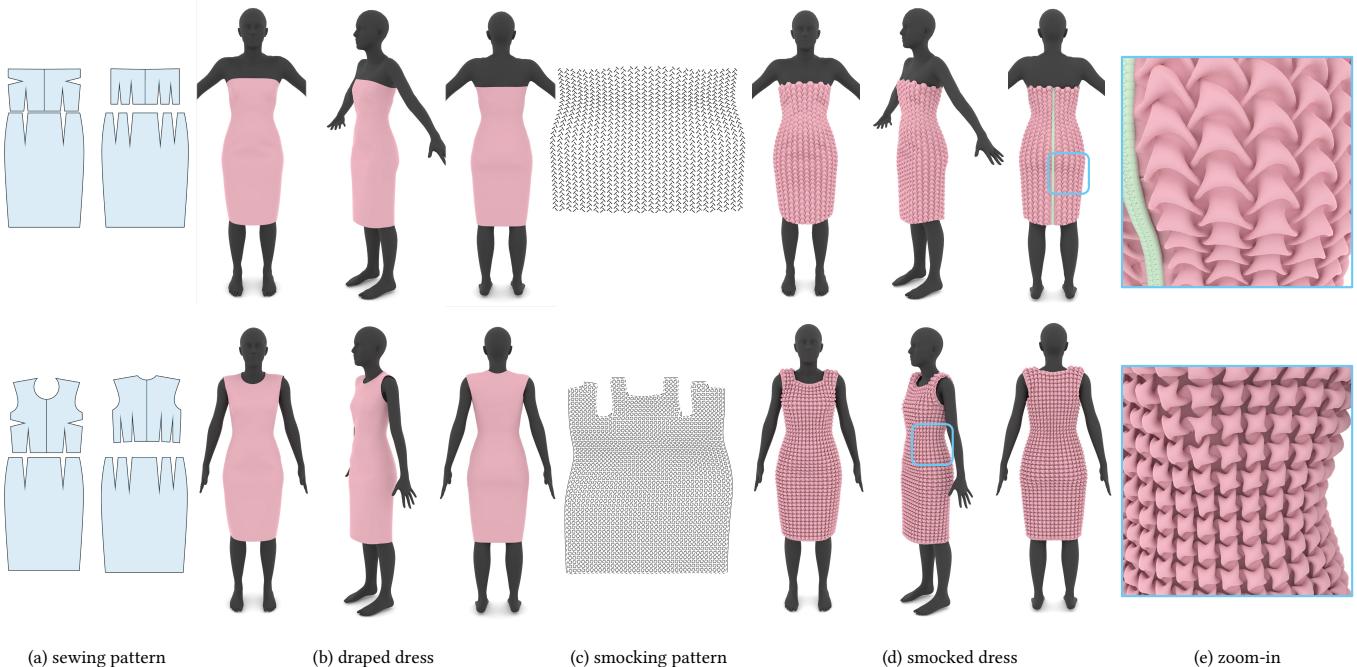


Fig. 21. In (a) we show the sewing patterns for two dresses from [Korosteleva and Sorkine-Hornung 2023] and the corresponding draped dresses in (b). Our algorithm outputs the smocking patterns, shown in (c), to realize the draped garments. The corresponding smocked results are shown in (d) with a zoomed-in view in (e).

## 7.1 Smocked 3D surfaces

We demonstrate that our Tangram formulation is general and can be applied to numerous smocking patterns. Our method is effective for a variety of shapes with both negative and positive Gauss curvatures. For example, in Fig. 4, 17 and 18, we show different smocked results that approximate a hyperboloid, a hemisphere, and a hyperbolic paraboloid (“Pringle”) shape. Note that we do not incorporate boundary alignment into the patterns shown in Fig. 4 and 17. Adding boundary alignment constraints introduces *integral* singularities (of full field rotations), where seamless smocking is undefined. Instead, we opt to crop the closed Tangram falling outside the covered domain, acknowledging that this approach may result in imperfect boundary alignment. For closed shapes and shapes with more prominent curvature, as shown e.g. in Fig. 19, we apply the seamless variant of our method for smocking patterns whose Tangrams have  $N$ -rotational symmetry ( $N = 3, 4, 6$ ).

As demonstrated by the physical fabrications, our method can accurately realize the target shapes, and our preview tool is sufficiently faithful for digital design, despite some self-intersections around the seams. Specifically, for all experiments, the edge reproduction error between the closed Tangram and the target surface summed over all edges, i.e., calculated via Eq. (2), is below  $10^{-4}$ . The smocked “Pringle” in Fig. 1 maintains a tight fit after draping onto the 3D printed input shape shown in Fig. 18. The smocked results of the closed surfaces shown in Fig. 19 cannot be draped onto the 3D-printed shape due to the necessity of sewing seams first, and then smocking across the seams before draping. We stuff

the smocked shape with cotton, resulting in a slight deviation from the input shape due to imperfect stuffing.

Our method for inverse design and result preview is fairly efficient. Table 1 reports the runtime for optimizing the smocking patterns and generating previews of the smocked results: it takes around  $5 \sim 17$  seconds to compute the optimized pattern with  $300 \sim 400$  stitching lines and about  $5 \sim 16$  seconds to generate a preview for an upsampled mesh of  $70 \sim 120K$  vertices.

## 7.2 Potential applications

We believe fabric tessellation has a wide range of potential applications including garment, architecture, acoustics, and auxetic meta-material design.

Table 1. We report the shape complexity (including the number of faces  $|\mathcal{F}_M|$  and vertices  $|\mathcal{V}_M|$  of the remeshed input shape, the number of stitching lines  $|\mathcal{L}|$  that need to be optimized, and the number of vertices  $n$  in upsampled mesh for preview) of the examples shown in Fig. 19. We also report the runtime for optimizing the stitching patterns  $t_{\text{opt}}$  and previewing the smocked results  $t_{\text{preview}}$ .

example	complexity				runtime (sec.)	
	$ \mathcal{F}_M $	$ \mathcal{V}_M $	$ \mathcal{L} $	$n$	$t_{\text{opt}}$	$t_{\text{preview}}$
Fig. 19 (top) cloud	1332	668	351	78,809	16.7	5.0
Fig. 19 (mid.) heart	1126	565	393	120,698	12.5	16.2
Fig. 19 (btm.) sphere	630	320	310	82,953	5.5	10.1

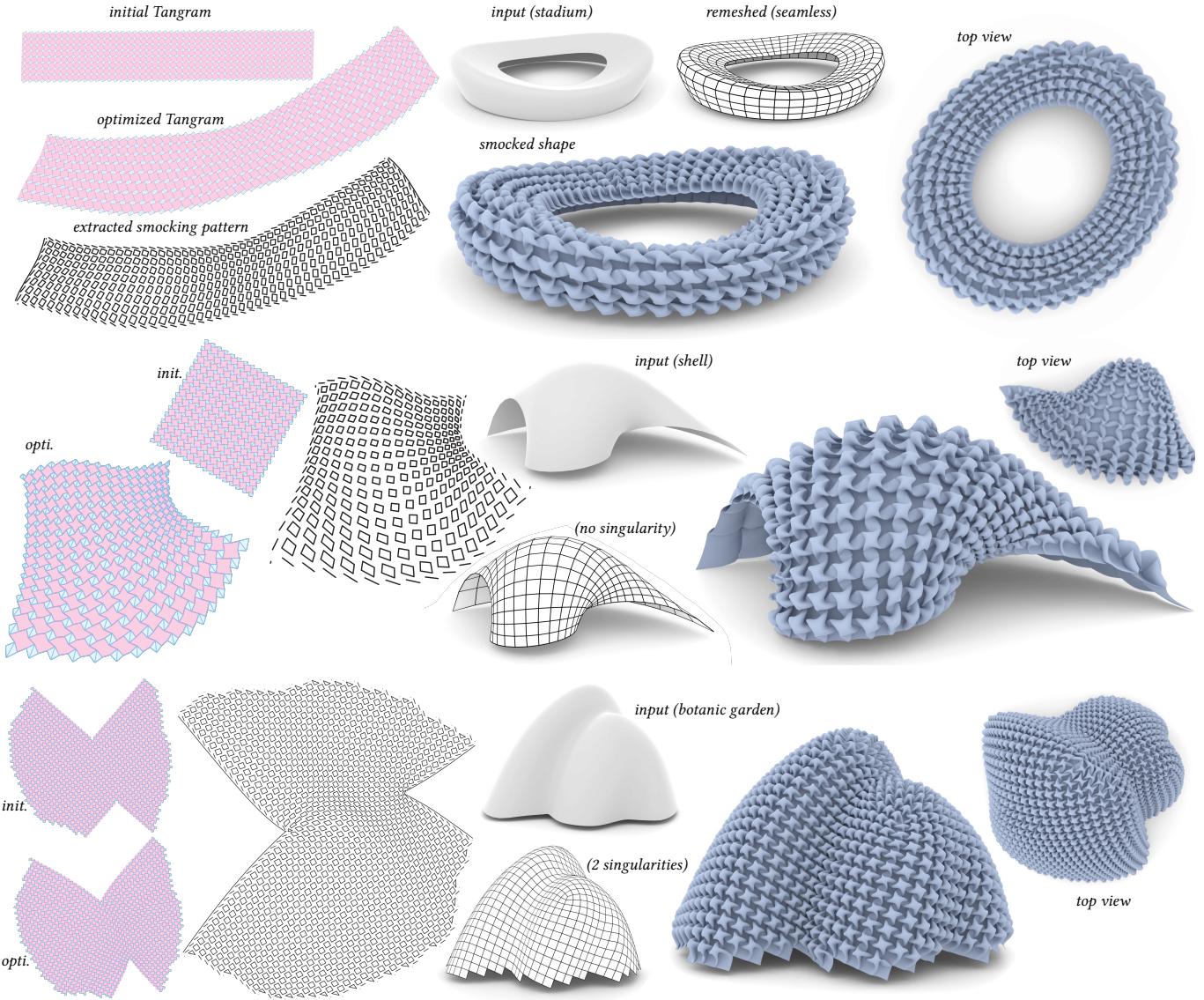


Fig. 22. Smocking for free-form architectural design. For each model, we show the Tangram before and after optimization, where the underlay faces and pleat faces are colored in blue and pink respectively. We also show the smocking pattern extracted from the optimized Tangram, with the digital preview results in different views.

**Garment design.** Intricate pleating and smocking is utilized in couture and creative artisanal fashion design to create texture and extra volume. Normally, especially when making garments out of woven fabric, darts are unavoidable in order to generate the necessary curvature out of a flat sewing pattern and create a close fit, as seen in the carefully designed sewing patterns in Fig. 21 (a). The smocking technique offers an alternative solution for creating local curvature through stitching without cutting. In Fig. 21 we showcase how our method is used to generate an optimized smocking pattern to approximate the 3D shapes of two different dresses. Our algorithm can be used to convert a conventional sewing pattern into a smocked

design, helping designers save time and circumvent painstaking draping and pinning of smocked pleats onto mannequins.

**Architectural design.** In Fig. 22 we show three examples of using the smocking technique to realize 3D architectural surfaces. Our seamless smocking can approximate doubly-curved surfaces by fabric tessellation with singularities while exhibiting regular and seamless pleats. The voluminous structure created by the smocked pleats holds the potential to enhance insulation for buildings. Additionally, as explored by Scherer [2019], smocked fabric can be employed as supporting material for pouring and shaping concrete structures.



Fig. 23. Conceptual illustration of applying fabric tessellation in acoustic design for a musical hall.

**Acoustics design.** Harne and Lynd [2016] combined origami tessellation with the physics of acoustics, and demonstrated its potential in designing tessellated acoustic arrays. This inspires a promising application for fabric tessellation given the suitability of fabric as a good material for sound diffusion [Aksa acoustic Co. 2024; Kamisiński et al. 2012; Stern EWS Co. 2024]. In Fig. 23, we present a conceptual scenario where fabric tessellation contributes to acoustic design in a musical hall [Andersons 2023; Wormald 2022]. Our method enables the accurate realization of a curved surface adorned with visually pleasing tessellated pleats. We leave the exploration of the meta-material properties of tessellated fabric for future work.

**Programmable auxetics.** We notice the similarity between the RESCH-3A pattern shown in Fig. 10 and the auxetic material pattern explored in [Konaković et al. 2018, Fig. 6] (cf. Fig. 24). The two patterns have exactly the same topology of the Tangram, but in Fig. 24, Variant of RESCH-3A pattern different opening (closing) angles, resulting in differently shaped patterns. An intriguing connection exists between the work of [Konaković et al. 2018] and our smocking problem. In [Konaković et al. 2018], the focus is on finding the 2D configuration of the nearly-closed Tangram (or a Kagome lattice, using their terminology) to approximate a given 3D surface when *fully open* (i.e., after prying them apart). Conversely, our goal is to determine the 2D configuration of the Tangram in the opened state, so that it approximates a given 3D surface when *fully closed* (i.e., after stitching). Another perspective is that while Konaković et al. [2018] generates discrete Gauss curvatures via empty spaces (via expansion), our smocking process does this by pushing excess fabric outwards to form pleats.

Inspired by [Konaković et al. 2018], we experiment with our Tangram formulation for designing programmable auxetics. We follow our algorithm discussed in Sec. 5 and Fig. 11 with a small modification: we lift the Tangram in its open configuration onto the target 3D shape through parameterization and then optimize the Tangram in the closed configuration in 2D using the shape approximation energy  $E_{\text{shape}}$  with a simple barrier function to avoid

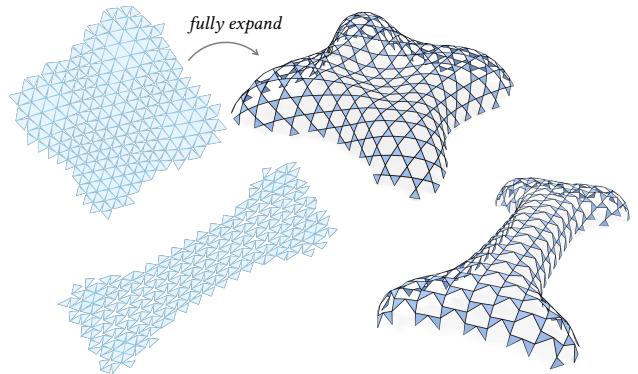


Fig. 25. **Tangram for programmable auxetics.** On the left we show the optimized nearly-closed Tangram. Upon full expansion, it approximates the shape of *Lilium* (top) and *Bone* (bottom) models using the RESCH-3A and ARROW patterns, respectively.



Fig. 26. *Left:* Shadow-folded bell shape using Origamizer [Tachi 2008]. Image from flickr.com by En Why See under CC BY-NC-ND 2.0. *Right:* the exterior and interior of the fabricated smocked bell shape using our method.

collisions between the underlay faces. The promising results shown in Fig. 25 underscore the potential of the Tangram formulation for auxetic material design. While further modifications may be required for more intricate surfaces, these preliminary findings hint at the exciting possibilities in the realm of auxetic materials.

**Shadowfolds for 3D surfaces.** Shadowfolds, a creative technique that combines origami tessellation with translucent cloth material, utilizes patterns derived from Euclidean planar tilings through the shrink-rotate algorithm [Bateman 2002; Lang 2017; Rutzky and Palmer 2011], as detailed in Appendix D. Extending the shadowfolds technique to 3D is challenging, given that planarity is a fundamental assumption in pattern design. An online user [En Why See 2009] experimented with using the Origamizer [Demaine and Tachi 2017; Tachi 2008] to create a 3D bell shape, as shown in Fig. 26 (left). However, since the Origamizer is tailored to origami tessellation and emphasizes feasibility for fabrication rather than regularizing pleat shapes (which are concealed in origami tessellation), using it for shadowfolds offers no guarantee of achieving the desired regular pleats. Our method provides a more suitable design tool for 3D shadowfolds. See Fig. 26 (left) for our fabricated bell shape.

## 8 CONCLUSION, LIMITATIONS AND FUTURE WORK

We have presented a formulation for the fabric tessellation problem, aimed at reproducing target metric to realize a freeform surface. For surfaces where singularities cannot be avoided, such as those with high Gaussian curvature or watertight surfaces, we only consider RESCH patterns, where seamless smocking is relatively easy to derive. For patterns with translational symmetry only, it is not obvious how to integrate singularities into the unconventional tessellation of their closed Tangram, which we leave as future work.

We do not provide theoretical guarantees for the types of 3D closed Tangrams that can be realized by closing a 2D open Tangram. For instance, one can distort the 3D closed Tangram by adjusting the positions of the vertices, making some edge lengths significantly larger than the rest. In such cases, optimizing the 2D smocking pattern to achieve a closed Tangram in such a distorted configuration becomes infeasible. However, due to our choice of a near-isometric parameterization derived from near unit-length directional fields, the lifted 3D Tangrams we experimented with were all feasible. It would be interesting to delve into the theoretical guarantees for realizable 3D Tangrams in future work.

While the geometric features of the pleats appear similar when fabricated with different materials, the exact shapes can vary due to differences in stiffness or bending properties of the materials used. For instance, in Fig. 2 (right), a heavy-weight synthetic crepe fabric is utilized, whereas in Fig. 19 (b), stretchy denim is used to fabricate the WATERBOMB pattern, resulting in different crease patterns. Our preview tool does not take this into consideration. For more realistic, material-aware, and self-intersection-free outcomes, the adoption of advanced physical simulators, such as C-IPC [Li et al. 2021], is recommended.

One interesting direction for future work is to investigate the insulating and mechanical properties of the smocked fabric. For example, during fabrication, we noted that the smocked fabric exhibits elasticity under strong external forces. Moreover, fabric formwork with smocking also holds promising potential in concrete forming, as identified by Scherer [2019]. The predicted geometry of the smocked shape from our algorithm can be effectively integrated with the techniques proposed by Zhang et al. [2019] to design modified smocking patterns, specifically tailored for formwork, with special considerations for how filling materials affect the smocked fabric.

Throughout our research and experiments, we observed intriguing connections between smocking design, origami tessellation, and auxetic material design. Much like the deployment of surfaces using auxetic materials, fabric tessellation can accurately drape over a solid object, forming a tight cover. However, unlike paper or cardboard origami, fabric tessellation may not always maintain its intended shape without support or stuffing, as the material itself is less stiff. Exploring the similarities and differences among these disciplines presents a fascinating avenue for further investigation, potentially inspiring mutual advancements. One intriguing problem is deriving a smocking pattern from an origami pattern and vice versa. Additionally, considering the introduction of cuts into smocking, analogous to the introduction of cuts into origami (i.e., kirigami), could broaden the design space and expand the spectrum of feasible

surfaces that can be well approximated. We also posit that there is untapped theoretical richness in smocking—specifically, understanding how to achieve precise target curvatures through stitching in a general sense, extending beyond the specific patterns discussed in this work. These captivating questions are left for exploration in future studies.

## ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable feedback. This work was supported in part by the ERC Consolidator Grant No. 101003104 (MYCLOTH). Special thanks to Ningfeng Zhou for her assistance in fabricating the heart and cloud shapes, and to all members of IGL for the insightful discussions and kind support.

## REFERENCES

- Aksa acoustic Co. 2024. <https://www.acousticsoundinsulation.com/theater-acoustica-l-insulation-solutions/>
- Andrew Andersons. 2023. Sydney Opera House Concert Hall Renewal by ARM Architecture. <https://architectureu.com/articles/sydney-opera-house-concert-hall-renewal-by-arm-architecture/>
- Paulo Taborda Barreto. 1997. Lines meeting on a surface: The “Mars” paperfolding. In *Proceedings of the 2nd International Meeting of Origami Science and Scientific Origami*. Seian University of Art and Design, Otsu, Shiga, Japan, 323–331.
- Alex Bateman. 2002. Computer tools and algorithms for origami tessellation design. In *Origami3: Proc. the 3rd International Meeting of Origami Mathematics, Science, and Education*. A K Peters, Natick, MA, USA, 121–128.
- Margie Bauer and Barry Elsey. 1992. Smocking: traditional craft as the expression of personal needs and adult community education in Australia. *Australian Journal of Adult and Community Education* 32, 2 (1992), 84–89.
- David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-integer quadrangulation. *ACM Transactions On Graphics (TOG)* 28, 3 (2009), 1–10.
- Stijn Brancart, Aline Vergauwen, Kelvin Roovers, Dimitri Van Den Bremt, Lars De Laet, and Niels De Temmerman. 2015. UNDULATUS: design and fabrication of a self-interlocking modular shell structure based on curved-line folding. In *Proceedings of IASS Annual Symposia*. International Association for Shell and Spatial Structures (IASS), Spain, 1–12.
- Rulin Chen, Pengyu Qiu, Peng Song, Bailin Deng, Ziqi Wang, and Ying He. 2023. Masonry shell structures with discrete equivalence classes. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 115:1–115:12.
- Tian Chen, Julian Panetta, Max Schnaubelt, and Mark Pauly. 2021. Bistable auxetic surface structures. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–9.
- Gary PT Choi, Levi H Dudte, and Lakshminarayanan Mahadevan. 2019. Programming shape using kirigami tessellations. *Nature materials* 18, 9 (2019), 999–1004.
- Erik Demaine and Martin Demaine. 2002. Recent results in computational origami. In *Origami3: Third International Meeting of Origami Science, Mathematics and Education*. A K Peters, Natick, MA, USA, 3–16.
- Erik Demaine and Tomohiro Tachi. 2017. Origamizer: A practical algorithm for folding any polyhedron. In *33rd International Symposium on Computational Geometry (SoCG 2017)*. Dagstuhl Publishing, Saarbrücken/Wadern Germany, 34:1–34:16.
- Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Designing N-PolyVector fields with complex polynomials. *Computer Graphics Forum* 33, 5 (2014), 1–11.
- Levi H Dudte, Etienne Vouga, Tomohiro Tachi, and Lakshminarayanan Mahadevan. 2016. Programming curvature using origami tessellations. *Nature materials* 15, 5 (2016), 583–588.
- Dianne Durand. 1979. *Smocking: Techniques, Projects and Designs*. Dover Publications, NY, USA.
- Tamara Anna Efrat, Moran Mizrahi, and Amit Zoran. 2016. The hybrid bricolage: bridging parametric design with craft through algorithmic modularity. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, NY, USA, 5984–5995. <https://www.tamaraefrat.com/crafted-technology>
- Marwa Yasien Helmy Elbyaly and Abdellah Ibrahim Mohammed Elfeky. 2022. Investigating the effect of podcast to enhance the skills of the Canadian smocking and complex problem solving. *Current Psychology* 41, 11 (2022), 8010–8020.
- En Why See. 2009. <https://www.flickr.com/photos/enwhysee/3703493302/>
- Michael S Floater. 2003. Mean value coordinates. *Computer aided geometric design* 20, 1 (2003), 19–27.
- Michael S Floater and Kai Hormann. 2005. Surface parameterization: a tutorial and survey. *Advances in multiresolution for geometric modelling* (2005), 157–186.

- Shuzo Fujimoto. 1978. *Twist Origami*. <https://archive.org/details/twist-origami-1/>
- Shuzo Fujimoto and Masami Nishiwaki. 1982. *Sojo Suru Origami Asobi eno Shotai (Invitation to creative origami playing)*. Self-published, Japan. <https://archive.org/details/invitation-to-creative-playing-with-origami/>
- Eric Gjerde. 2008. *Origami tessellations: awe-inspiring geometric designs*. CRC Press, NY, USA.
- Ryan L Harne and Danielle T Lynd. 2016. Origami acoustics: using principles of folding structural acoustics for simple and large focusing of sound energy. *Smart Materials and Structures* 25, 8 (2016), 085031.
- Kai Hormann, Bruno Lévy, and Alla Sheffer. 2007. Mesh parameterization: Theory and practice. In *ACM SIGGRAPH 2007 Courses*.
- Paul Jackson. 1989. *Origami: a complete step-by-step guide*. Hamlyn Publisher, London, UK. <https://archive.org/details/origamicomplete000unse>
- Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. <https://libigl.github.io/>.
- Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. 2015. Instant Field-Aligned Meshes. *ACM Transactions on Graphics (TOG)* 34, 6 (Nov. 2015), 189:1–189:15. <https://doi.org/10.1145/2816795.2818078>
- Caigui Jiang, Florian Rist, Helmut Pottmann, and Johannes Wallner. 2020. Freeform quad-based kirigami. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–11.
- Caigui Jiang, Chengcheng Tang, Amir Vaxman, Peter Wonka, and Helmut Pottmann. 2015. Polyhedral patterns. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–12.
- Lishuai Jin, Antonio Elia Forte, Bolei Deng, Ahmad Rafsanjani, and Katia Bertoldi. 2020. Kirigami-inspired inflatables with programmable shapes. *Advanced Materials* 32, 33 (2020), 2001863.
- Ruby Joseph, Kaur Prabhjot, Mehtab Shazia, et al. 2011. Lattice smocking techniques: an innovative approach to smocking. *Asian Journal of Home Science* 6, 1 (2011), 5–11.
- David Jourdan, Mélina Skouras, Etienne Vouga, and Adrien Bousseau. 2020. Printing-on-fabric meta-material for self-shaping architectural models. In *Advances in Architectural Geometry 2020*. Les Presses des ponts, Paris, France, 14:1–14:19.
- Tadeusz Kamisiński, Krzysztof Brawata, Adam Pilch, Jarosław Rubacha, and Marcin Zastawnik. 2012. Sound diffusers with fabric covering. *Archives of Acoustics* 37 (2012), 317–322.
- Minkyung Kim. 2020. A study on reproductions of North American smocking design using a 3D virtual clothing system. *Journal of Fashion Business* 24, 5 (2020), 106–124.
- Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. 2016. Beyond Developable: Computational Design and Fabrication with Auxetic Materials. *ACM Transactions on Graphics (TOG)* 35, 4, Article 89 (2016), 11 pages. <https://doi.org/10.1145/2897824.2925944>
- Mina Konaković, Julian Panetta, Keenan Crane, and Mark Pauly. 2018. Rapid Deployment of Curved Surfaces via Programmable Auxetics. *ACM Transactions on Graphics (TOG)* 37, 4, Article 106 (2018), 13 pages. <https://doi.org/10.1145/3197517.3201373>
- Maria Korosteleva and Olga Sorkine-Hornung. 2023. GarmentCode: Programming Parametric Sewing Patterns. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 16. <https://doi.org/10.1145/3618351>
- Robert J Lang. 2017. *Twists, tilings, and tessellations: Mathematical methods for geometric origami*. CRC Press, NY, USA.
- Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional Incremental Potential Contact. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 170:1–170:24.
- Malin Lind. 2019. Smocked patterns: An exploration of jacquard woven patterns and smocking techniques for a spatial textile design context. Bachelor thesis, University of Borås. <https://api.semanticscholar.org/CorpusID:209055873>
- Daoming Liu, Davide Pellis, Yu-Chou Chiang, Florian Rist, Johannes Wallner, and Helmut Pottmann. 2023. Deployable strip structures. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 103:1–103:16.
- Ramses V Martinez, Carina R Fish, Xin Chen, and George M Whitesides. 2012. Elastomeric origami: programmable paper-elastomer composites as pneumatic actuators. *Advanced functional materials* 22, 7 (2012), 1376–1384.
- Merel Meekes and Amir Vaxman. 2021. Unconventional patterns on surfaces. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–16.
- Yuki Mori and Takeo Igarashi. 2007. Plushie: an interactive design system for plush toys. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 45–es.
- Koya Narumi, Kazuki Koyama, Kai Suto, Yuta Noma, Hiroki Sato, Tomohiro Tachi, Masaaki Sugimoto, Takeo Igarashi, and Yoshihiro Kawahara. 2023. Inkjet 4D print: Self-folding tessellated origami objects by inkjet UV printing. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–13.
- Julian Panetta, Mina Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. 2019. X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.
- Chi-Han Peng, Caigui Jiang, Peter Wonka, and Helmut Pottmann. 2019. Checkerboard Patterns with Black Rectangles. *ACM Transactions on Graphics (TOG)* 38, 6, Article 171 (nov 2019), 13 pages. <https://doi.org/10.1145/3355089.3356514>
- Nico Pietroni, Corentin Dumery, Raphael Falque, Mark Liu, Teresa Vidal-Calleja, and Olga Sorkine-Hornung. 2022. Computational Pattern Making from 3D Garment Models. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 157:1–14.
- Nico Pietroni, Marco Tarini, Amir Vaxman, Daniele Panozzo, and Paolo Cignoni. 2017. Position-based tensegrity design. *ACM Trans. Graph.* 36, 6, Article 172 (nov 2017), 14 pages. <https://doi.org/10.1145/3130800.3130809>
- Jing Ren, Aviv Segall, and Olga Sorkine-Hornung. 2024. Digital 3D Smocking Design. *ACM Transactions on Graphics (TOG)* 43, 2 (2024), 14:1–14:17. Presented at SIGGRAPH ASIA 2023.
- Yingying Ren, Uday Kusupati, Julian Panetta, Florin Isvoranu, Davide Pellis, Tian Chen, and Mark Pauly. 2022. Umbrella Meshes: Elastic Mechanisms for Freeform Shape Deployment. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–15.
- Yingying Ren, Julian Panetta, Tian Chen, Florin Isvoranu, Samuel Poincloux, Christopher Brandt, Alison Martin, and Mark Pauly. 2021. 3D weaving with curved ribbons. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 127:1–127:15.
- Ronald D Resch. 1968. Self-supporting structural unit having a series of repetitive geometrical modules. US Patent 3,407,558.
- Jeffrey Rutzky and Chris K. Palmer. 2011. *Shadowfolds: surprisingly easy-to-make geometric designs in fabric*. Kodansha USA Publishing, NY, USA.
- Rohan Sawhney and Keenan Crane. 2017. Boundary first flattening. *ACM Transactions on Graphics (TOG)* 37, 1 (2017), 1–14.
- Annie Locke Scherer. 2019. Concrete Form [ing] work: Designing and Simulating Parametrically-Patterned Fabric Formwork for Cast Concrete. In *the eCAADe and SIGRADi Conference 2019*, Vol. 7. Blucher Proceedings, Brazil, 759–768.
- Nicholas Sharp et al. 2019. Polyscopic. [www.polyscopic.run](http://www.polyscopic.run).
- Yuta Shimoda, Kai Suto, Sei Hayashi, Tomoyuki Gondo, and Tomohiro Tachi. 2023. Developable Membrane Tensegrity Structures Based on Origami Tessellations. *Advances in Architectural Geometry 2023* (2023), 303.
- Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. *Computer Graphics Forum* 4 (2007), 109–116.
- Margaret Spufford and Susan Mee. 2017. *The Clothing of the Common Sort: 1570–1700*. Oxford University Press, Oxford, UK.
- Stern EWS Co. 2024. Acoustical Panel Fabrics. <https://www.sternandstern.com/acoustical-panel-fabrics/>
- Tomohiro Tachi. 2008. Origamizer. <https://origami.c.u-tokyo.ac.jp/~tachi/software/index.html>
- Tomohiro Tachi. 2009. Generalization of rigid-foldable quadrilateral-mesh origami. *Journal of the International Association for Shell and Spatial Structures* 50, 3 (2009), 173–179.
- Tomohiro Tachi. 2013. Designing Freeform Origami Tessellations by Generalizing Resch's Patterns. *Journal of Mechanical Design* 135, 11 (10 2013), 111006:1–111006:10. <https://doi.org/10.1115/1.4025389>
- Alison Toplis. 2021. *The Hidden History of the Smock Frock*. Bloomsbury Publishing, London, UK.
- Amir Vaxman et al. 2019. Directional: A library for Directional Field Synthesis, Design, and Processing. <https://doi.org/10.5281/zenodo.3338174>
- Amir Vaxman, Marcel Campen, Olga Diamanti, Daniele Panozzo, David Bommes, Klaus Hildebrandt, and Mirela Ben-Chen. 2016. Directional field synthesis, design, and processing. *Computer Graphics Forum* 35, 2 (2016), 545–572.
- Floor Verhoeven, Amir Vaxman, Tim Hoffmann, and Olga Sorkine-Hornung. 2022. Dev2PQ: Planar Quadrilateral Strip Remeshing of Developable Surfaces. *ACM Transactions on Graphics (TOG)* 41, 3 (2022), 29:1–18.
- James Wormald. 2022. The science of acoustic design making auditorium architecture worth listening to. <https://www.architonic.com/en/story/james-wormald-the-science-of-acoustic-design-making-auditorium-architecture-worth-listening-to/20288714>
- Jiangmei Wu. 2022. Grafting Tessellations for Fabric Origami. In *Bridges 2022 Conference Proceedings*. Tessellations Publishing, AZ, USA, 375–378.
- Xiaoting Zhang, Guoxin Fang, Melina Skouras, Gwenda Gieseler, Charlie C.L. Wang, and Emily Whiting. 2019. Computational Design of Fabric Formwork. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 109:1–109:13.
- Ningfeng Zhou, Jing Ren, and Olga Sorkine-Hornung. 2024. Computational Smocking through Fabric-Thread Interaction. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2024)* 43, 2 (2024).

## A CHALLENGES IN ADAPTING [Ren et al. 2024] FOR INVERSE DESIGN

Ren et al. [2024] proposed an effective method to address the forward problem of smocking design: given a 2D smocking pattern, usually obtained by evenly tiling a unit pattern on a regular grid, the method predicts the geometry of the fabric after fabrication. In this section, we describe this method in a bit more details and discuss the challenges in adapting it to address the inverse design problem

In [Ren et al. 2024], the fabric is represented in two resolutions: a lower-resolution grid is used to demonstrate the smocking pattern

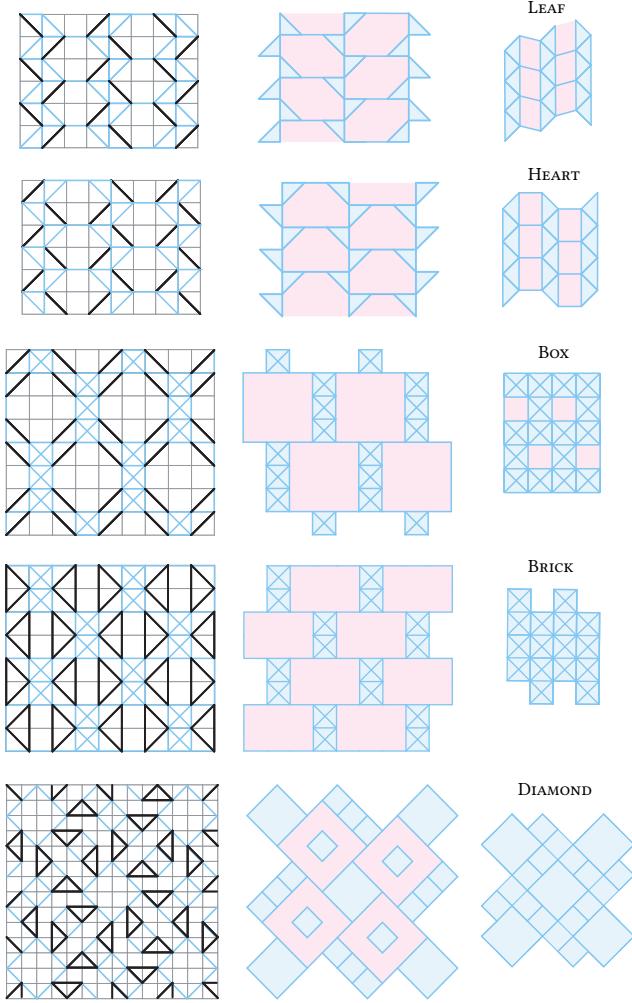


Fig. 27. From left to right, we show the smocking pattern, open Tangram, and the closed Tangram for five different patterns. We highlight the underlay edges in the smocking pattern in blue. The underlay and pleat faces in the open/closed Tangram are colored in blue and pink respectively.

and to extract the smocked graph, while a higher-resolution grid is used to compute the detailed mesh deformation showing the final smocking design. More specifically, given a smocking pattern  $\mathcal{P} = (\mathcal{V}, \mathcal{E}, \mathcal{L})$  with stitching lines  $\mathcal{L}$  defined on the grid graph  $(\mathcal{V}, \mathcal{E})$ , the smocked graph  $\mathcal{S} = (\mathcal{V}_S, \mathcal{E}_S)$  is constructed from the graph  $(\mathcal{V}, \mathcal{E})$  by (1) fusing all underlay vertices sharing the same stitching line into a single vertex, and (2) deleting edges that become degenerated or duplicate as a result of the fusing of underlay vertices. This smocked graph captures the non-manifold structure of the resulting smocking design. The fused vertex from the stitching line  $\ell$  is denoted as  $v_\ell$ .

The smocked graph is then embedded in 3D and used to guide the deformation of the finer-resolution fabric using as-rigid-as-possible (ARAP) deformation [Sorkine and Alexa 2007], obtaining the final smocking design. The key observation to help embed the smocked

graph is that the local geometry is modified after stitching. Specifically, before stitching, the embedded positions of two arbitrary vertices on the fabric are constrained by their geodesic distance on the fabric. After stitching, since multiple underlay vertices are contracted together and pinched to the same position, the embedded positions ( $\mathbf{x}_{\ell_i}, \mathbf{x}_{\ell_j}$ ) of two underlay vertices ( $v_{\ell_i}, v_{\ell_j}$ ) from different stitching lines ( $\ell_i, \ell_j$ ) are constrained by the *shortest* geodesic distances among any pair of vertices in the two stitching lines [Ren et al. 2024, Eq. (3)]:

$$\|\mathbf{x}_{\ell_i} - \mathbf{x}_{\ell_j}\|_2 \leq d_{i,j}, \quad \text{where } d_{i,j} = \min_{v_p \in \ell_i, v_q \in \ell_j} d(v_p, v_q). \quad (6)$$

This so-called *embedding distance constraint* can be generalized and derived for any pair of vertices in  $\mathcal{V}_S$ . These distance constraints are used to embed the smocked graph in 3D while encouraging the vertices to stay away from each other to avoid a trivial solution. After some relaxations, the smocked graph is embedded via:

$$\min_{\mathbf{X} \in \mathbb{R}^{|\mathcal{V}_S| \times 3}} \sum_{(v_i, v_j) \in \mathcal{E}_S} \left( \|\mathbf{x}_i - \mathbf{x}_j\|_2 - d_{i,j} \right)^2. \quad (7)$$

Another important observation from fabrication is that the underlay vertices remain on the same plane after stitching and only the pleat vertices are pushed out of plane. Therefore, Eq. (7) is solved in a two-stage optimization, where the underlay vertices are firstly enforced to be embedded in 2D and then the pleat vertices are embedded in 3D with fixed underlay vertices.

One potential solution to adapt this method for inverse design is to let the distance constraints  $d_{i,j}$  encode the geometry of the target surface, and solve for the 2D positions of the stitching lines such that  $d_{i,j}$  are realized when Eq. (6) is applied. More specifically, we can sample on the target surface to obtain the expected 3D positions for the smocked graph, from which we can compute the distance constraints, e.g., the lower bound of  $d_{i,j}$  in Eq. (6). The next step is to move the stitching lines in the fabric plane such that the induced embedding distance constraints  $d_{i,j}$  from Eq. (6) are consistent with the ones derived from the input 3D surface.

However, this solution is extremely hard to realize since (i) the derivation of the distance constraints  $d_{i,j}$  is *discrete*, making it hard to modify the smocking pattern to achieve the target  $d_{i,j}$ ; (ii)  $d_{i,j}$  provides the upper bound for the embedded distance such that the fabric won't tear up after stitching. There is no *theoretical guarantee* that smocked fabric can be embedded (after optimization in Eq. (7)) at the exact sampled locations to approximate the target surface even with carefully design  $d_{i,j}$ . As a comparison, our Tangram-based approach avoids these pitfalls: the Tangram maintains a topologically one-to-one map between the 2D smocking pattern and the 3D smocked results without any discrete operations. Moreover, the exact metrics are achieved to approximate the target surface thanks to the mesh-like structure of the closed Tangram.

## B SMOCKING PATTERNS AND THEIR TANGRAMS

In Fig. 27 we show the different smocking patterns and their Tangram in open and closed configuration, including LEAF pattern (used in Fig. 4, Fig. 15, and Fig. 17), HEART pattern (used in Fig. 17 and Fig. 21), Box pattern (used in Fig. 17), BRICK pattern (used in Fig. 4),

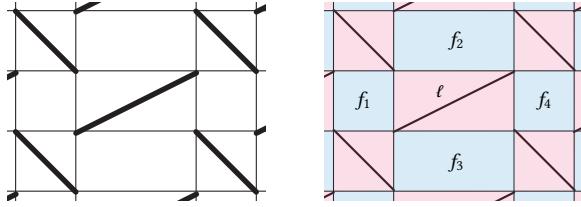


Fig. 28. Left: an under-constrained smocking pattern (cf. Fig. 14 in [Ren et al. 2024]). Right: the corresponding Tangram, where blue (pink) indicates the underlay (pleat) faces.

and DIAMOND pattern. We can see that, similar to the ARROW pattern (shown in Fig. 6), the BRICK and DIAMOND pattern have their pleat faces completely vanish after the Tangram is closed. It is worth noting that the DIAMOND pattern has disconnected underlay faces. Another interesting observation is that the closed Tangram of the LEAF pattern and the HEART pattern have similar structure though their topology are very different in the open configuration. We find that Tangram is a helpful tool to understand the differences or similarities between various patterns, which may not be obvious from the alignment of the stitching lines in the smocking pattern.

We also observe that the extracted Tangram can be used to determine if a smocking pattern is well-constrained, providing valuable insights for designing new patterns. Ren *et al.* [2024, Sec. 5] discuss heuristics for the design of a well-constrained 2D smocking pattern. This ensures that the fabricated result strikes a balance, avoiding being too “loose”, where the underlay vertices have excessive freedom, or too “tight”, which would force the underlay vertices out of plane. However, the provided guideline is somewhat experimental, requiring the execution of the proposed method and subsequent verification if the optimized embedding reaches zero energy. In contrast, our new formulation using Tangram in Sec. 4.2 offers a more principled guideline for regular 2D smocking pattern design:

**Remark B.1.** A smocking pattern is well-constrained if a closed configuration exists for its Tangram.

Specifically, if a closed configuration exists for the Tangram of a smocking pattern, according to Def. 4.3, it implies that after stitching (i.e., when the Tangram reaches a closed state), the result achieves a balanced state, as the underlay faces undergo only rigid transformations without distortion. On the other hand, if such a closed configuration does not exist, it means that after fabrication, when the stitching lines are forced to have zero length, the underlay faces become distorted, leading to undesirable pleats. Consider the under-constrained smocking pattern discussed in [Ren et al. 2024, Fig. 14], where the width of the middle grid cells is doubled (cf. Fig. 28 left). We show its Tangram on the right of Fig. 28, with blue (pink) representing the underlay (pleat) faces. The pleat face containing the stitching line  $\ell$  is surrounded by four underlay faces  $f_1, \dots, f_4$ . For this example, it becomes apparent that it is impossible to rigidly rotate the underlay faces  $f_i$  such that the length of the stitching line  $\ell$  can be reduced to 0, implying that this smocking pattern does not have a closed configuration. In summary, remark B.1 offers a more principled guideline for designing a well-constrained smocking pattern.

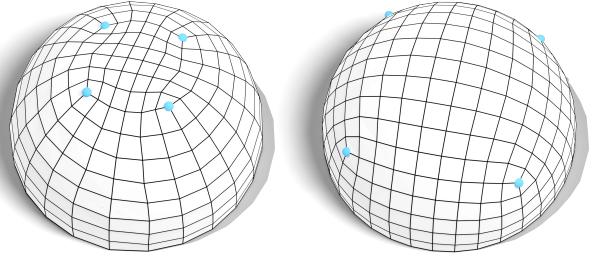


Fig. 29. Compare to the naive remeshing method (left), our algorithm produces a seamless remeshing with a more isometric parameterization (right), which better suited for the inverse design of smocking. We highlight the singularities in blue.

## C DIRECTIONAL FIELDS FOR PARAMETERIZATION

In this section, we discuss the algorithmic details for the parameterization step discussed in Sec. 5.2. We first give a brief introduction to directional fields, then provide the full algorithm to solve for  $N$ -field (for  $N = 3, 4, 6$ ) with singularities for seamless smocking, and our variation for singularity-free fields for smocking patterns that only contain translational symmetry.

### C.1 Background: Directional fields

Consider a triangle mesh  $\mathcal{M} = (\mathcal{V}_\mathcal{M}, \mathcal{E}_\mathcal{M}, \mathcal{F}_\mathcal{M})$ . Our goal is to compute a *curl-free*  $N$ -directional field that can be integrated to achieve a *seamless* parameterization. Let  $\mathcal{F}_c \subset \mathcal{F}_\mathcal{M}$  be the set of faces with input alignment constraints  $\mathbf{z}_f, \forall f \in \mathcal{F}_c$ . These constraints arise from user-specified alignment orientations for selected faces or aligning-to-boundary requirements for boundary faces.

**C.1.1 Raw representation.** The raw representation of an  $N$ -directional field  $\mathbf{Y}$  is an array of  $N$  complex numbers for each face  $f \in \mathcal{F}_\mathcal{M}$ :  $\mathbf{Y}(f) = (y_1(f), \dots, y_N(f))^\top$ , where  $y_i(f) \in \mathbb{C}$  encodes the coordinates of the vector in a local frame, and where they are ordered counterclockwise around the normal to the face.

Given a raw  $N$ -field, we can find the principal matching  $\mathbb{I}_e \in \mathbb{Z}$  for any edge  $e = f \cap g$  (denoting the edge  $e$  shared by face  $f$  and  $g$ ):  $y_n(f)$  is matched to  $y_{n+\mathbb{I}_e}(g)$  (modulo  $N$ ) with the smallest rotation angle. We denote this step as:

$$\mathbb{I} \leftarrow \text{PrincipalMatching}(\mathbf{Y}). \quad (8)$$

Given an order-preserving matching  $\mathbb{I}$ , the field is *curl-free* when

$$\text{Re}\left(y_n(f)\overline{e_f}\right) = \text{Re}\left(y_{n+\mathbb{I}_e}(g)\overline{e_g}\right), \forall n = 1 \dots N. \quad (9)$$

Here,  $\mathbf{e}_f, \mathbf{e}_g \in \mathbb{C}$  are the representation of the shared edge  $e$  in the local frames of faces  $f, g$  respectively. These constraints are linear and therefore can be represented in a compact matrix form  $C_{\mathbb{I}} \mathbf{Y} = 0$  for a given matching  $\mathbb{I} = \{\mathbb{I}_e\}$ .

We can normalize the raw vector fields to have unit length. We call this step  $\mathbf{Y}^* = \text{Normalize}(\mathbf{Y})$  satisfying:

$$\mathbf{y}_n^*(f) \leftarrow \frac{\mathbf{y}_n(f)}{|\mathbf{y}_n(f)|}, \forall f \in \mathcal{F}_\mathcal{M}, \forall n = 1 \dots N. \quad (10)$$

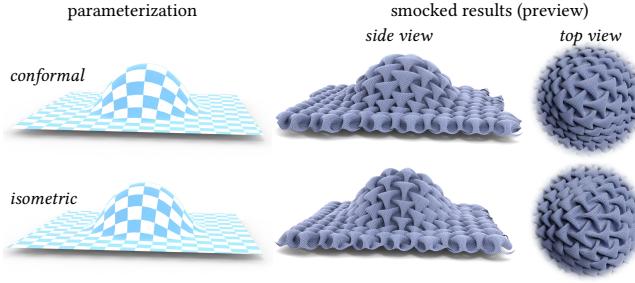


Fig. 30. Compared to using conformal (top) parameterization, isometric (bottom) parameterization suits better for fabric tessellation as it results in more uniformly sized pleats and lower shape approximation error.

**C.1.2 Closeness.** The closeness energy between two  $N$ -field  $\mathbf{Y}$  and  $\mathbf{Y}^*$  in raw representations is defined as follows:

$$\mathbb{E}_{\text{closeness}}(\mathbf{Y}, \mathbf{Y}^*) = \sum_{f \in \mathcal{F}_M} \sum_{n=0}^{N-1} A_f |\mathbf{y}_n(f) - \mathbf{y}_n^*(f)|^2, \quad (11)$$

where  $A_f$  is the area of face  $f$ .

**C.1.3 Curl-free projection with alignment constraints.** We can find a curl-free field closest to a given field  $\mathbf{Y}$  while respecting the input alignment constraints, if any. Specifically, for each constrained face  $f \in \mathcal{F}_c$ , we find the closest vector in  $\mathbf{y}(f)$  that aligns with the constrained direction  $\mathbf{z}_f$  and we denote the corresponding index as  $i_f$ , i.e., we have  $i_f = \arg \min_{j=0, \dots, N-1} |\mathbf{y}_j(f) - \mathbf{z}_f|$ . We then solve for a new field  $\tilde{\mathbf{Y}}$  as follows:

$$\begin{aligned} \min_{\tilde{\mathbf{Y}}} \quad & \lambda_c \mathbb{E}_{\text{closeness}}(\tilde{\mathbf{Y}}, \mathbf{Y}) + \lambda_a \sum_{f \in \mathcal{F}_c} A_f |\tilde{\mathbf{y}}_{i_f}(f) - \mathbf{z}_f|^2, \\ \text{s.t.} \quad & C_{\mathbb{I}} \tilde{\mathbf{Y}} = 0, \end{aligned} \quad (12)$$

where  $C_{\mathbb{I}}$  stores the curl-free constraint per edge from the principal matching  $\mathbb{I}$  in a compact matrix form. The result  $\tilde{\mathbf{Y}}$  is a curl-free field close to the input field  $\mathbf{Y}$  while respecting the alignment constraints. We denote this step as  $\text{ProjCurl}(\mathbf{Y}, \mathbb{I})$ .

**C.1.4 PolyVector representation.** To allow for the singularities in the field to naturally emerge without discrete optimization, we also work with the PolyVector representation [Diamanti et al. 2014], which encodes the raw vector field  $\mathbf{Y}$  as the roots of a monic polynomial with (complex) coefficients  $\Gamma$ :

$$\Gamma = \text{PolyVector}(\mathbf{Y}) = (z - y_1) \cdot \dots \cdot (z - y_N) = z^N + \sum_{n=0}^{N-1} \Gamma_n z^n. \quad (13)$$

We also consider the inverse transformation  $\mathbf{Y} = \text{PolyVector}^{-1}(\Gamma)$  that extracts the roots of the polynomial represented by  $\Gamma$ . The closeness energy Eq. (11) also applies in the PolyVector representation:

$$\mathbb{E}_{\text{closeness}}(\Gamma, \Gamma^*) = \sum_{f \in \mathcal{F}_M} \sum_{n=0}^{N-1} A_f |\Gamma_n(f) - \Gamma_n^*(f)|^2. \quad (14)$$

**C.1.5 PolyVector alignment.** We can use PolyVector field representation to incorporate alignment constraints. Specifically, we follow [Meekes and Vaxman 2021] and project each polynomial

---

#### ALGORITHM 1: Computing $N$ -fields

---

```

Input : Triangle mesh  $M$  and optional alignment
        constraints  $\mathbf{z}_f$  on faces  $f \in \mathcal{F}_c$ 
Output : A curl-free  $N$ -field  $\mathbf{Y}$  with singularities
Parameters:  $\lambda_s = 10$ ,  $\lambda_r = 0.1$ ,  $\lambda_c = 0.1$ ,  $\lambda_a = 100$ 
1 initialize a smooth power field  $\Gamma^{(0)}$ , where  $\Gamma_n^{(0)} = 0, n > 0$ ,
 $\Gamma_0^{(0)} = \arg \min_U \mathbb{E}_{\text{smooth}}(U) \quad \text{s.t. } U(f) = -(\mathbf{z}_f)^N \quad \forall f \in \mathcal{F}_c$ 
2 set  $k \leftarrow 0$ 
3  $\Gamma^{(k+1)} \leftarrow \arg \min_{\Gamma} \mathbb{E}_{\text{pv}}(\Gamma, \Gamma^{(k)})$ 
4  $\mathbf{Y} \leftarrow \text{PolyVector}^{-1}(\Gamma^{(k+1)})$ 
5  $\mathbf{Y} \leftarrow \text{Normalize}(\mathbf{Y})$ 
6  $\mathbb{I} \leftarrow \text{PrincipalMatching}(\mathbf{Y})$ 
7  $\mathbf{Y}^{(k+1)} \leftarrow \text{ProjCurl}(\mathbf{Y}, \mathbb{I})$ 
8  $\Gamma^{(k+1)} \leftarrow \text{PolyVector}(\mathbf{Y}^{(k+1)})$ 
9 set  $\lambda_s \leftarrow 0.8\lambda_s$  and  $k \leftarrow k + 1$ ; then go to step 3

```

---

at a constrained face to the closest PolyVector polynomial that satisfies the alignment constraint at the face. We denote the projected PolyVector as  $\mathbb{P}_{\Gamma}$ , which satisfies that: for a constrained face  $f \in \mathcal{F}_c$ ,  $\mathbf{z}_f$  is one of the roots of  $\mathbb{P}_{\Gamma}(f)$ . The alignment energy for the PolyVector field  $\Gamma$  is then its difference from the projected PolyVector  $\mathbb{P}_{\Gamma}$  at the constrained faces:

$$\mathbb{E}_{\text{align}}(\Gamma) = \sum_{f \in \mathcal{F}_c} \sum_{n=0}^{N-1} A_f |\Gamma_n(f) - (\mathbb{P}_{\Gamma})_n(f)|^2. \quad (15)$$

**C.1.6 Power fields.** An important subspace of PolyVector fields is that of *power fields*, where  $\Gamma_n = 0, \forall n > 0$ , and all vectors in a face are in perfect symmetry (an  $N$ -RoSy). The power field is further of unit-length when  $|\Gamma_0| = 1$  [Meekes and Vaxman 2021; Verhoeven et al. 2022]. We can therefore define the following energy to encourage  $\Gamma$  be a unit-length power field:

$$\mathbb{E}_{\text{RoSy}}(\Gamma) = \sum_{f \in \mathcal{F}_M} A_f \left( \left| \Gamma_0(f) - \frac{\Gamma_0^{(k)}(f)}{|\Gamma_0^{(k)}(f)|} \right|^2 + \sum_{n=1}^{N-1} |\Gamma_n(f)|^2 \right), \quad (16)$$

where the first term enforces the unit-length requirement while the rest enforcing vanishing coefficients. It worth noting that when  $N$  is an even number, the field has a sign symmetry (only  $\frac{N}{2}$  vectors are distinct, and the rest is given as the negation of these vectors). The PolyVector representation will only have non-zero coefficients for the even powers of  $z$ . In this case, the vanishing coefficients energy term only includes the even  $n$ .

**C.1.7 Smooth fields.** We define the *smoothness* of a PolyVector field  $\Gamma$  as follows:

$$\mathbb{E}_{\text{smooth}}(\Gamma) = \sum_{e \in \mathcal{E}_M} \sum_{\substack{n=0 \\ e=f \cap g}}^{N-1} A_e \left| \Gamma_n(f) \overline{\mathbf{e}_f}^{N-n} - \Gamma_n(g) \overline{\mathbf{e}_g}^{N-n} \right|^2. \quad (17)$$

**ALGORITHM 2:** Computing singularity-free Fields

---

**Input** :Triangle mesh  $\mathcal{M}$   
**Output** :Two curl- and singularity free coupled fields  $Y$   
**Parameters**:  $\lambda_s = 10$ ,  $\lambda_c = 0.1$ ,  $\lambda_o = 0.1$

- 1 initialize a smooth raw 2-field  $Y^{(0)} = (U^{(0)}, V^{(0)})$ :
$$U^{(0)} \leftarrow \arg \min_U \mathbb{E}_{\text{smooth}}(U) \text{ and } V^{(0)} \leftarrow iU^{(0)}$$
- 2 set  $k \leftarrow 0$
- 3  $Y \leftarrow \arg \min_Y \mathbb{E}_{\text{raw}}(Y, Y^{(k)})$
- 4  $Y \leftarrow \text{Normalize}(Y)$
- 5  $Y^{(k+1)} \leftarrow \text{ProjCurl}(Y, \mathbb{I})$
- 6 set  $\lambda_s \leftarrow 0.8\lambda_s$  and  $k \leftarrow k + 1$ ; then go to step 3

---

We use the same weight  $A_e$  as in [Verhoeven et al. 2022], i.e., for an edge  $e = f \cap g$  we set  $A_e = \frac{|e|}{|e_{\text{dual}}|} \frac{A_f + A_g}{2}$  with  $A_f$  and  $A_g$  denoting the areas of faces  $f$  and  $g$  respectively, and  $|e_{\text{dual}}|$  is the sum of the lengths of the two edges connecting the centroid of the adjacent faces to the midpoint of the edge  $e$ . Recall  $\mathbf{e}_f, \mathbf{e}_g \in \mathbb{C}$  are the representation of the shared edge  $e$  in the local frames of faces  $f, g$  respectively. The smoothness of a vector field  $y$  is a special case with  $N = 1$ , i.e.,

$$\mathbb{E}_{\text{smooth}}(y) = \sum_{\substack{e \in \mathcal{E}_M \\ e=f \cap g}} A_e |y(f)\overline{\mathbf{e}_f} - y(g)\overline{\mathbf{e}_g}|^2. \quad (18)$$

**C.2 Computing  $N$ -Fields with singularities ( $N = 3, 4, 6$ )**

In Algorithm 1 we show the full details of computing an  $N$ -field. We use the following energy to improve a given PolyVector  $\Gamma^*$ :

$$\begin{aligned} E_{\text{pv}}(\Gamma, \Gamma^*) = & \lambda_s \mathbb{E}_{\text{smooth}}(\Gamma) + \lambda_r \mathbb{E}_{\text{RoSy}}(\Gamma) \cdots \\ & + \lambda_c \mathbb{E}_{\text{closeness}}(\Gamma, \Gamma^*) + \lambda_a \mathbb{E}_{\text{align}}(\Gamma). \end{aligned} \quad (19)$$

We then convert the field to its raw representation and apply a unit-length projection. This step is taken to promote a more isometric parameterization upon integration, see Fig. 29. In smocking design, more isometric parameterization usually leads to more regularly sized pleats and better shape approximation, as shown in Fig. 30. After normalizing the field, we proceed with a curl projection step, as described in Eq. (12), to reduce the integrability error of the field.

**C.3 Computing singularity-free fields**

We modify the algorithm for computing fields with a fixed singularity-free matching, that can be integrated to a parameterization with a low isometric distortion. Instead of working with PolyVector representation, we only use the raw representation, and use two coupled vector fields denoted as  $Y = (U, V)$ . We initialize the first vector field  $U$  by minimizing the smoothness energy defined in Eq. (18), with an initial alignment constraint on an arbitrary face to avoid trivial solutions. The second vector field  $V$  is initialized as  $V = iU$ , i.e., by rotating  $U$  of  $\frac{\pi}{2}$  degree. We then optimize the following energy to improve the smoothness of the fields while optimizing

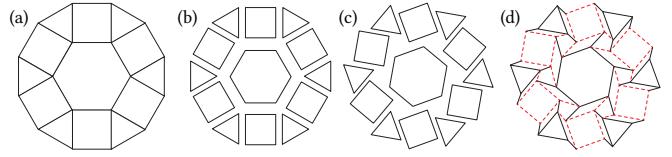


Fig. 31. **Shrink-rotate algorithm** (cf. [Lang 2017, Fig.6.1]). Take the original tiling (a), shrink each polygon by the same factor shown in (b); rotate them by the same angle shown in (c); connect vertices with new creases and assign mountain-valley folds, obtaining the final folding pattern (d).

for orthogonality:

$$\begin{aligned} \mathbb{E}_{\text{raw}}(Y, Y^*) = & \lambda_s \mathbb{E}_{\text{smooth}}(Y) + \lambda_c \mathbb{E}_{\text{closeness}}(Y, Y^*) \\ & + \lambda_o \sum_{f \in \mathcal{F}_M} |V(f) - iU(f)|^2. \end{aligned} \quad (20)$$

Algorithm 2 describes the complete steps of computing these fields.

**D SHRINK-ROTATE FOR ORIGAMI TESSELLATION**

Origami tessellations often exhibit a unique property wherein the symmetry of the tessellation aligns with an established tiling pattern. In addition, each polygon from the original tiling is present in both the crease pattern and the folded form, albeit shrunken and rotated [Bateman 2002; Lang 2017; Rutzky and Palmer 2011]. This property offers a practical methodology for constructing crease patterns in origami tessellation. Taking a given polygonal tiling as an input, we can *precisely* shrink and rotate each polygon. This process allows us to strategically introduce creases, making a valid crease pattern in origami tessellation. Bateman [2002] presented a realization of this concept in Perl, named Tess. We adopt the term “shrink-rotate” as proposed in [Lang 2017] as it provides a more informative and descriptive name for this approach.

Shrink-rotate is a very straightforward algorithm. See Fig. 31 for an illustration. Start with a 2D tiling, such as Euclidean tiling by convex regular polygons, the algorithm proceeds as follows: (1) **Shrinkage**: Each polygon is *uniformly* reduced in size, maintaining a consistent scaling factor. This shrinking operation creates empty spaces within the tessellation to facilitate folding. (2) **Rotation**: Every polygon is rotated by a fixed angle to ensure the feasibility of subsequent folding steps. The angle of rotation remains *consistent* across all polygons within the tiling. (3) **Connection**: The shrunken-and-rotated vertices that originated from the same initial vertex of a polygon are connected with lines, resulting in the final crease pattern. We refer the interested readers to Chapter 6 of [Lang 2017] for more detailed discussions and implementations.

While our Tangram and the shrink-rotate algorithm share a similar principle of allowing certain faces/regions to rotate rigidly to achieve the desired pattern, they serve different purposes. The shrink-rotate algorithm is tailored for the *forward* problem, i.e., constructing a *valid* crease pattern from a given *polygon* tiling. On the other hand, our Tangram addresses the *inverse* problem. More importantly, it is not confined to polygon tiling, as illustrated in Figures 8 and 27.