

Designing with Tension: Nearly-Developable Patch Layouts

ANNA MARIA EGGLER, ETH Zurich, Switzerland

NICO PIETRONI, University of Technology Sydney, Australia

PENGBIN TANG, ETH Zurich, Switzerland

MICHAL PIOVARCI, ETH Zurich, Switzerland

BERND BICKEL, ETH Zurich, Switzerland



Fig. 1. We present a novel method for approximating input models (left) with tensile-like surfaces. Our method automatically decomposes the target mesh into near-developable patches, identifies a sparse set of supports, and generates a sparse set of tension strings (middle). The output of our method captures the high visual appeal of tensile structures and is ready for fabrication (right).

We propose a novel method to automatically approximate a free-form surface using a set of near developable patches that form a tensile-like structure when anchored at a sparse set of points. These structures are appealing for their ability to span large areas with low material cost and structural weight, while also offering strong aesthetic potential. Our algorithm strikes a balance between approximation accuracy, patch simplicity, and visual quality, while ensuring manufacturability and structural feasibility. The layout is guided by a curvature field and refined through a combinatorial process that incrementally adds patches until performance and fabrication constraints are met. Redundant elements are then removed to improve clarity and elegance.

We demonstrate the effectiveness of our method on several architectural surfaces, supported by fabricated prototypes that showcase the interplay between geometric design, structural behavior, and visual appeal.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; **Mesh geometry models**; **Parametric curve and surface models**.

Authors' Contact Information: Anna Maria Egger, annamaria.egger@inf.ethz.ch, ETH Zurich, Zurich, Switzerland; Nico Pietroni, University of Technology Sydney, Sydney, Australia, nico.pietroni@uts.edu.au; Pengbin Tang, ETH Zurich, Zurich, Switzerland, petang@ethz.ch; Michal Piovarci, ETH Zurich, Zurich, Switzerland, piovarci@arch.ethz.ch; Bernd Bickel, ETH Zurich, Zurich, Switzerland, bickelb@ethz.ch.



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

SA Conference Papers '25, Hong Kong, Hong Kong

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2137-3/2025/12

<https://doi.org/10.1145/3757377.3763941>

Additional Key Words and Phrases: Architectural Geometry, Tensile Structures

ACM Reference Format:

Anna Maria Egger, Nico Pietroni, Pengbin Tang, Michal Piovarci, and Bernd Bickel. 2025. Designing with Tension: Nearly-Developable Patch Layouts. In *SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25)*, December 15–18, 2025, Hong Kong, Hong Kong. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3757377.3763941>

1 Introduction

Tensile structures are lightweight surfaces shaped by tension and anchored at discrete support points, combining material efficiency with expressive geometric form. These structures offer elegant solutions for efficiently covering large spans. Recently, tensile structures have garnered significant interest in both architecture and industry, thanks to their low cost and rapid deployability compared to other structural systems. A key advantage of these structures is their ability to conform to complex geometries without the need for strict structural constraints, such as funicularity, that are typically required in systems like gridshells.

Tensile structures consist of large, lightweight patches whose self-weight is typically negligible for structural evaluation. These patches achieve equilibrium when tensioned and anchored at a sparse set of points. Despite their practical relevance, most of the geometry processing literature has surprisingly focused on the optimization and design of gridshells or masonry-inspired structures [Pottmann et al. 2015]. In contrast, few methods focus on the computational design

of tensile structures, with most of the work addressing tensegrity systems [Gauge et al. 2015; Pietroni et al. 2017]. Designing tensile structures computationally involves negotiating a complex trade-off between geometric accuracy, fabrication constraints, and aesthetic appeal [Gale and Lewis 2016; Kamal 2020; Wagner 2005]. Two key challenges lie in deriving a minimal set of visually coherent patches that accurately approximate a target surface, and ensuring that the structure remains in tension when anchored at a sparse set of fixed points. Moreover, to ensure fabricability using standard materials, each patch must exhibit high developability, permitting only minimal stretching when brought into tension.

In this work, we introduce the first method for automatically generating nearly developable tensile-like structures from arbitrary input surfaces. Given an initial patch decomposition, our method minimizes the number of fixed anchor points needed to approximate the input within a prescribed geometric error. This yields forms that are structurally efficient and visually coherent. At the core of our framework is a global optimization process that selects the fixed vertices from a patch layout such that the remaining surface can relax into a smooth tensile equilibrium. To ensure fabricability using minimally stretchable materials, we map the resulting surface to a set of nearly planar patches and enforce developability via iterative optimization. These geometric procedures are embedded within a framework that incrementally refines the patch decomposition, minimizing complexity while satisfying structural and approximation constraints. A key feature of our approach is the integration of aesthetic criteria: aligning seams with principal curvature lines produces visually clean, harmonious structures that reflect the surface’s intrinsic geometry while remaining physically feasible.

2 Related Work

In recent years, the field of architectural geometry [Pottmann et al. 2015] has extensively leveraged principles of computational design to create structures that are both functional and aesthetically pleasing. Significant progress has been made in the context of optimal design of architectural models [Whiting et al. 2009, 2012], masonry structures [de Goes et al. 2013; Liu et al. 2013; Panozzo et al. 2013; Vouga et al. 2012], structures composed by interlocking planar pieces [Cignoni et al. 2014; Hildebrand et al. 2012; Schwartzburg and Pauly 2013] or structurally sound planar tessellations [Bouaziz et al. 2012; Pietroni et al. 2015; Tonelli et al. 2016]. In computational design, the user focuses primarily on the aesthetic aspects of the form, while practical considerations such as structural stability, cost, and assemblability are delegated to an underlying algorithm. Following this paradigm, a wide range of algorithms have been developed for architectural computational design.

While some work has been done in the context of architectural geometry, the research most closely related to ours focuses on the design and optimization of tensegrity structures. Additionally, our problem shares similarities with methods for defining developable surfaces and with form-finding approaches for tensile and membrane structures. Finally, we provide an overview of techniques for patch decomposition, which are central to our framework.

2.1 Tensile membrane structure

Despite a rich body of literature on tensile simulation, fabric behavior, and construction methodologies, most computational design methods for tensile structures focus on optimizing predefined configurations. Some approaches [Gale and Lewis 2016] are limited to optimizing planar panel arrangements using discrete modeling and advanced flattening techniques. Others [Wagner 2005] require an initial patch layout that is flattened and then slightly adjusted to ensure fabricability. However, there still remains a research gap in the problem of automatically approximating a target shape with physically realizable tensile structures [Kamal 2020]. Our work addresses this gap by reversing the traditional workflow, starting with the target shape and working backward to derive an appropriate tensile structure.

A related line of work arises in inflatable structure design, where the “patterning problem” couples flat panel cutting with non-linear physical response. Seminal approaches [Pérez et al. 2015; Siéfert et al. 2019; Skouras et al. 2012, 2014; Zhang 2023] highlight this tight interplay between geometry and physics. Unlike these workflows, which involve a physically different setup and rely on manually specified constraints, our method advances the process by introducing a fully automatic anchor insertion strategy.

2.2 Tensegrities

Tensegrities are structures typically composed of a network of cables and struts, where equilibrium is achieved through a balance of compressive forces (carried by the struts) and tensile forces (carried by the cables). Recent computational methods have addressed the automatic design of tensegrity systems from input surfaces [Gauge et al. 2015; Pietroni et al. 2017]. At the same time, the mechanical engineering and architectural communities have devoted significant attention to the topic [Micheletti and Podio-Guidugli 2022], driven by the structural advantages of tensegrities, such as lightness, flexibility, and material efficiency. Like our setting, tensegrity design is often framed as an optimization problem, where equilibrium constraints are enforced iteratively. For example, the approach proposed by Pietroni and colleagues [2017] combines geometric and static constraints to ensure the structure is both stable and resistant to external forces. However, the computational design of tensile structures introduces a fundamentally different set of challenges. In tensegrities, the structure is defined by a network of linear elements whose connectivity and force distribution are explicitly modeled. In contrast, tensile structures are typically composed of continuous surface patches whose shape and equilibrium emerge from their boundary conditions and tension state. This requires different strategy: instead of balancing discrete forces along struts and cables, we should model global equilibrium of a deformable surface anchored at a (possibly minimal) sparse set of points. Moreover, ensuring the developability of each patch introduces additional geometric constraints not present in tensegrity design.

2.3 Form-finding

Form-finding refers to the process of discovering a shape that is in static equilibrium under specific loading and boundary conditions, such that it experiences only in-plane internal forces, typically axial

forces, as in grid shells. Several core numerical methods have been developed for this purpose [Veenendaal and Block 2012], including the Force Density Method [Schek 1974] and the Thrust Network Analysis [Block 2009]. Despite some fundamental similarities, tensile structures present a significantly different setup compared to classical form-finding. Traditional approaches often aim to produce catenary-like surfaces with clearly defined boundary conditions. In contrast, in our case, the identification of boundary constraint, i.e., the location and number of anchor points, is itself part of the problem to be solved.

2.4 Developability

Developability has been extensively studied in geometry processing, with numerous generative algorithms proposed for different applications [Bo and Wang 2007; Rabinovich et al. 2018; Tang et al. 2016]. In the context of fabrication using inextensible materials, the problem is typically formulated as finding the closest perfectly developable approximation of a given discrete input mesh. Approaches in this category include convex optimization under developability constraints [Sellán et al. 2020], wrapping objects with developable sheets [Ion et al. 2020], segmenting surfaces into developable regions [Zhao et al. 2022], and more recent methods based on genetic algorithms [Zhao et al. 2023] or curvature fields [Baharami et al. 2025]. However, most of these techniques focus only on deforming the input surface to achieve developability. In contrast, the design of tensile structures imposes additional requirements: patch layouts must remain in tension and be supported only at a sparse set of anchor points. This fundamentally changes the structure of the problem and introduces new constraints that are not addressed by standard developable surface methods. The proposed framework generates patch layouts which are not only nearly developable and fabrication-friendly, but also remain in equilibrium under tension when anchored at a minimal set of points—thereby aligning geometric, structural, and aesthetic objectives within a unified computational design process.

2.5 Patch Decomposition

Flattening arbitrary 3D surfaces often leads to significant deformation [Yoshizawa et al. 2004] and can even cause self-intersections in the 2D domain [Rabinovich et al. 2017]. A common strategy to mitigate these issues is to cut the surface into smaller patches before flattening [Li et al. 2018; Sharp and Crane 2018]; for a detailed overview, we refer the reader to [Campen 2017]. State-of-the-art techniques—including ours—generate cuts by tracing tangent vector fields over the surface, often aligned with principal curvature directions [Nuvoli et al. 2019; Pietroni et al. 2022, 2021, 2016; Razafindrazaka et al. 2015]. Several patch decomposition techniques have already been employed in computational design problems, such as in fashion design [Pietroni et al. 2022] and inflatable structures [Skouras et al. 2014].

Our method extends the robust cut construction strategy introduced in [Pietroni et al. 2021], which promotes cut straightness and compactness. In our case, patch decomposition is not only used to facilitate flattening but serves as a core mechanism for deriving an optimized patch layout that meets the physical constraints required to realize tensile structures in the real world.

3 Overview

Our primary goal is to approximate a given input surface (Figure 2.a) with a tensile structure composed of an interconnected network of welded patches (Figure 2.b). Each patch is held in tension in its deployed configuration, connected to adjacent patches and anchored at a sparse set of anchor points (Figure 2.c). These anchors may be attached to supporting structures, fixed to the ground, or suspended via tensioned cables connected to rigid poles (Figure 2.d). The resulting structure is physically sound—it satisfies force equilibrium at every node and exhibits approximately uniform tension across all patches. To ensure the surface is fabricable, each patch is designed to be nearly developable, allowing minimal extensibility to meet tension constraints while remaining manufacturable. Further details of the workflow are provided in the accompanying pseudocode.

3.1 Design Goals

To be a valid and fabricable tensile structure, our solution must satisfy several key constraints:

Tension and equilibrium The assembled structure must be in static equilibrium, with all patches held under tension. Each node must satisfy force balance conditions, ensuring mechanical soundness.

Bijectivity Each patch must be realizable by cutting from a textile sheet. This requires that every patch admits a bijective, overlap-free mapping to the 2D plane.

Disk-like topology All patches must be homeomorphic to a disk, with a single boundary component. In addition, patches must not be self-glued, as this would complicate tensioning and require an impractical number of anchors to maintain stability.

Aesthetic quality To facilitate fabrication and ensure practical architectural applicability, we aim for a limited number of patches, each with a simple and visually pleasing shape.

Geometric fidelity The tensile structure must approximate the target surface within a specified tolerance.

Since patches must be mapped to 2D, a certain degree of developability is inherently required. To encourage this property, it is advantageous for the seams between patches to align with the principal curvature directions of the surface.

Our pipeline takes as input a target surface and a cross field, a smooth 4-rosy tangent vector field aligned with the principal curvature directions of the input surface [Vaxman et al. 2017] (see Figure 2.a).

The algorithm proceeds by iteratively inserting a set of uniformly distributed paths over the surface. These paths subdivide the surface into a collection of patches (see Section 5). Each path is aligned with one of the directions defined by the cross field and may either form closed loops or terminate at the boundary of the input mesh. Paths are allowed to intersect orthogonally but not tangentially. The resulting patches are bounded by a small number of corners (typically between 3 and 6), to keep their geometry simple and suitable for fabrication.

At each iteration, after inserting a new path, we evaluate the quality of the current patch layout by simulating the corresponding tensile structure. This evaluation is performed every time the layout

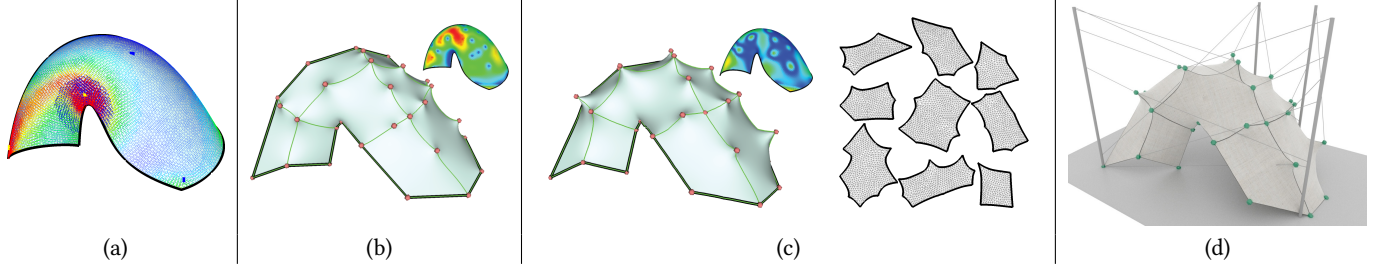


Fig. 2. Processing pipeline overview: (a) Starting from a triangle mesh equipped with a smooth curvature-aligned cross field; (b) we compute an optimal patch layout that accounts for the physical simulation of the resulting tensile structure; (c) we refine the anchor point positions through a final optimization step to minimize the deviation from the target surface and enhance developability, the approximation error is visualized, with red indicating 5% of the bounding box diagonal; (d) optionally, we compute an optimized cable configuration to ensure structural equilibrium.

is updated through the addition or removal of a path. The first step verifies that all patches satisfy the previously defined local topological constraints: each patch must be homeomorphic to a disk, possess a single boundary component, and must not self-intersect or be self-connected.

If all patches pass the topological validation, we proceed to compute the optimal tensile structure corresponding to the current layout (see Sections 4.1 and 4.2). We then mark any patches that fail to satisfy our geometric constraints: the approximation error with respect to the target surface must remain below a predefined threshold; each patch must admit a bijective mapping to 2D; and the distortion of this mapping must stay within a specified limit to ensure the patch is nearly developable. The algorithm continues by selectively refining the layout, inserting new paths within problematic patches until all topological and geometric constraints are met. Once the patch insertion process is complete, we perform a cleanup phase where redundant paths are removed. Each candidate path is removed only if the resulting layout still satisfies all constraints required for a valid tensile structure (see Figure 2.b).

Finally, we perform a global optimization of the anchor point positions to minimize the distance between the resulting tensile surface and the original target surface (see Figure 2.c). In this step, we also derive the final 2D mapping of the patches ready for fabrication. In a subsequent optional step, we compute the layout of cables and supports required to maintain the structure in a physically stable, tensioned equilibrium (see Section 6.1) (see Figure 2.d). The finalized patches are then flattened, producing a 2D layout ready for cutting and assembly into the final tensile structure.

4 Patch Validation and Tensile Structure Derivation

To guide the patch decomposition process, we must continuously assess whether the current patch layout is valid—that is, whether it satisfies the required *topological*, *geometric*, and *physically-based* constraints. As outlined in Section 3.1, patches must first fulfill a set of local topological properties. In addition, geometric validity is evaluated by simulating the corresponding tensile structure and verifying both its approximation quality and its physical fabricability.

4.1 Tensile Surface as Membrane

A key observation in characterizing tensile structures is that a surface under tension behaves as a *membrane*—a configuration in which internal forces are purely in-plane, and bending stiffness is neglected. In such configurations, the geometry of the surface is dictated by static equilibrium conditions at every vertex. Each interior vertex can be expressed as a convex linear combination of its neighbors, reflecting the smooth and energy-minimizing nature of membrane surfaces. This behavior is captured numerically via a Laplacian-like smoothing process [Botsch and Sorkine 2008] that converges to an equilibrium state determined by a set of boundary constraints (modelled as fixed anchor vertices).

Given a triangle mesh, defined by a set of vertices V and triangular faces F if we design a subset $A \subset V$ as the set of *anchor* vertices, then the remaining set $M = V \setminus A$ will consist of *membrane* vertices. To define the geometric configuration of the membrane, we impose the following constraint on each membrane vertex $m_i \in M$ in a least-squares sense,

$$\mathbf{v}_i = \sum_{j \in \mathcal{N}(i)} w_{ij} \mathbf{v}_j \quad (1)$$

Here, $\mathcal{N}(i)$ denotes the 1-ring neighborhood of vertex m_i , and w_{ij} are weights. In our implementation we use cotangent weights which remain fixed during the optimization.

Surface Approximation. In addition to enforcing the geometric condition of a membrane, we also aim at minimizing the deviation from the target surface. Let $\mathbf{v}_i^{\text{target}}$ denote the projection of vertex $v_i \in V$ onto the target surface. We define the approximation energy as

$$E_{\text{approx}} = \sum_{v_i \in V} \left\| \mathbf{v}_i - \mathbf{v}_i^{\text{target}} \right\|^2. \quad (2)$$

As computing the exact projection of a vertex onto the target surface cannot be efficiently performed within the optimization loop, we instead consider the position of each vertex in the target shape. While some anchor vertices are constrained (e.g., those fixed to the ground), we allow others to move freely during optimization to improve the overall approximation quality. This flexibility often

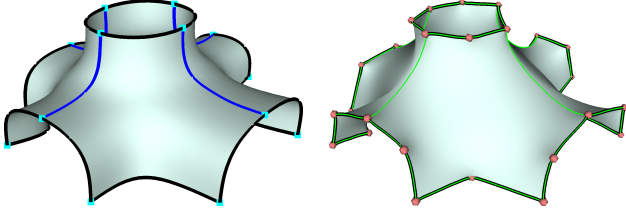


Fig. 3. The effect of tensile structure computation (right) on a given patch decomposition (left).

results in a better fit between the simulated membrane surface and the target geometry (see Figure 2.c).

Anchor Selection. To allow adaptive anchor placement, we introduce binary variables $b_i \in \{0, 1\}$, where $b_i = 1$ indicates that v_i is selected as an anchor. The anchor set A is thus defined by the subset of vertices with $b_i = 1$. To encourage sparsity in anchor usage, we include a regularization term

$$E_{\text{anchor}} = w_{\text{anchor}} \sum_{v_i \in V} b_i, \quad (3)$$

where w_{anchor} is a weight that penalizes the use of multiple anchor points, promoting a sparser selection. For the examples used in this paper, we set w_{anchor} to 10. To avoid degenerate solutions in which no anchors are chosen, we enforce a maximum allowed deviation from the target surface for each vertex $v_i \in V$ as

$$C_i = \left\| \mathbf{v}_i - \mathbf{v}_i^{\text{target}} \right\| - \delta_{\max} \leq 0. \quad (4)$$

Optimization Loop. We find the positions $\mathbf{V} = (\mathbf{v}_0, \dots, \mathbf{v}_n)$ subject to the constraints of 1 and anchor points $\mathbf{B} = (b_0, \dots, b_n)$ of a tensile surface by solving the following optimization problem

$$\arg \min_{\mathbf{V}, \mathbf{B}} E_{\text{approx}} + E_{\text{anchor}} \quad \text{s.t.} \quad C_i \leq 0, \forall v_i \in V. \quad (5)$$

This optimization is re-evaluated every time the patch layout is modified during decomposition. To improve efficiency, we restrict the anchor candidate set to a uniformly sampled subset of vertices between patch corners, including the corners themselves as potential anchors. The effect of this tensile structure derivation is shown in figure 3.

4.2 Developability Optimization

Once we have verified that the patch layout satisfies the topological conditions and derived an initial solution, we must also ensure that each patch can be mapped onto the 2D plane with minimal distortion. This is necessary to allow the patches to be fabricated from flat sheets of fabric and assembled as prescribed by the verified design. The process consists of two alternating steps:

Parameterization For each patch, we compute a locally injective 2D parameterization using As Rigid As Possible (ARAP) parameterization [Sorkine et al. 2004], which produces a planar embedding that approximates a developable unfolding. This embedding defines the rest lengths for the mesh edges within each patch.

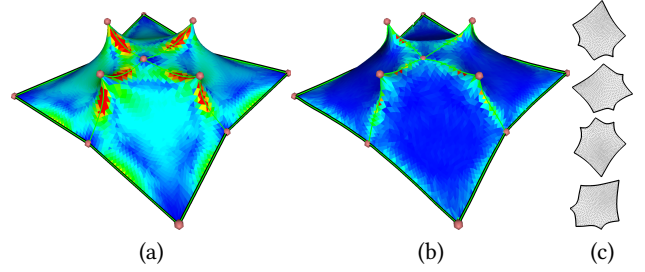


Fig. 4. The effect of the two-step flattening and simulation process: (a) The initial mesh after the tensile membrane computation, color shows the initial distortion of the derived 2D mapping (red corresponds to 1% of stretching); (b) The result after the developability optimization step; (c) The final patch layout.

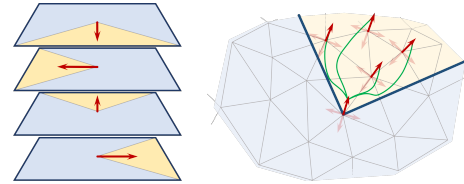


Fig. 5. We formulate the problem of tracing field-aligned paths as finding minimal paths over a graph that assigns four nodes to each mesh vertex, one for each cross-field direction (left). Edges in the graph connect nearby nodes with matching tangent directions (right).

Simulation Next, we simulate the physical behavior of the surface using Position-Based Dynamics [Müller et al. 2007], treating the flattened patches as inextensible materials under tension (using the bending constraint and distance constraint as explained in the original paper). Each edge’s target length corresponds to the rest length resulting from parameterization step. Although the resulting configuration is not perfectly developable, the simulation converges to a shape with minimal in-plane stretching.

These two steps are repeated until the distortion in the parameterization (measured as edge stretch) falls below a predefined threshold. At the end of the process, we consider a mapping valid if it is bijective and the distortion remains within the prescribed limits. Patches that do not satisfy these criteria are marked as unsolved and will be further refined with additional paths.

The effect of this refinement step is illustrated in Figure 4. As shown, it typically results in only minor deformations in 3D while significantly improving developability.

5 Patch Layout Generation

We begin by computing a cross field aligned with the principal curvature directions of the input surface, using the method proposed in [Diamanti et al. 2014]. To trace the paths that define the patch layout, we adopt the approach introduced in [Nuvoli et al. 2019; Pietroni et al. 2021]. Specifically, following the intuition of [Campen et al. 2012], we construct a graph with four nodes for each vertex of the input mesh, where each node corresponds to one of the

four directions of the cross field. Edges are added between adjacent vertex-nodes according to the field's parallel transport (see Figure 5 for illustration). In this setup, tracing paths reduces to computing shortest paths in the graph.

Path Insertion. We first sample a set of seed nodes on the surface to define a collection of candidate paths. Following a greedy strategy similar to [Livesu et al. 2020], we iteratively insert the path that is furthest from previously inserted paths and from the boundary of the mesh. Following this insertion order, and as outlined in Section 3.1, we only add the paths that are necessary to satisfy our topological, geometric, and physically based constraints. Only orthogonal intersections with existing paths are permitted; tangential intersections are not allowed. As proposed in [Pietroni et al. 2016], tangential intersections between paths are efficiently detected by checking whether two paths pass through the same vertex in non-orthogonal directions, which are encoded in the nodes of the graph. The distance between a candidate path and previously inserted ones is computed by averaging the distances between their respective nodes. Since distances are measured in the graph domain, paths that are far apart can still intersect orthogonally, while paths that run in parallel tend to be considered close. This strategy promotes the formation of well-shaped, evenly distributed patches that conform to the topology of the underlying cross field. Intuitively, patches with valence 3 or 5 form around singularities of corresponding index, while regular regions of the mesh tend to form 4-valence patches.

Path Smoothing. Since paths are traced over mesh vertices, they can exhibit staircase-like artifacts due to discretization. To alleviate this issue, we apply a smoothing step after each path is inserted. This operation is also extended to neighboring paths. Smoothing is performed in the tangent space along each path, and the smoothed positions are then projected back onto the original surface.

Layout Updating. After each insertion, we update the patch layout and verify whether the current configuration satisfies the topological constraints described in Section 3.1. If the layout is valid, we proceed to compute the corresponding tensile structure and its embedding in the 2D plane. We then evaluate which patches satisfy the geometric constraints. In the next insertion step, only candidate paths that lie on unresolved patches are considered. An example sequence of path insertions is shown in Figure 6 (left).

Final Path Layout Simplification. After all paths have been inserted, we simplify the layout by testing each path in reverse order of insertion (to preserve even distribution). During this step, we remove only those paths whose deletion does not violate any of the topological or geometric constraints. The result of this simplification is illustrated in Figure 6 (right).

6 Tensile Structure Finalization

Once the final decomposition is obtained, we perform a concluding membrane optimization step by iteratively reducing the maximum distance threshold $\delta_{\max}$ until the problem described in Section 4.1 no longer admits a valid solution. This process pushes the final configuration to adhere as closely as possible to the target surface.

As a final step, we flatten each patch and pack the resulting 2D patterns, making them ready for fabrication.

6.1 Boundary Constraints

At the end of the previous step, we obtain a tensile structure composed of nearly developable patches, which remains in equilibrium when tensioned and anchored at a set of fixed points. These anchor points can be physically implemented using supports that hold the structure in place. In the context of architectural surfaces, this solution is particularly convenient for anchor points located on the mesh boundary—especially those on the ground, which can be easily secured. However, anchoring interior points requires the construction of internal support structures to hold each anchor in position. While feasible this approach may reduce interior volume of the structure. In fact, many practical tensile structures employ external cables to maintain tension and equilibrium without occupying interior volume.

Automatic rope derivation. To adopt this solution, we introduce a system of virtual *ropes* connected to a small set of *support poles* P , where the size of the set can be user defined $|P| \leq P_{\max}$, selected from boundary vertices. Each rope connects an anchor point a_i to a pole $p_j \in P$ at a variable attachment height h_{ij} . However, not all heights are feasible: to prevent intersections with the tensile surface, we precompute for each pole-anchor pair (a_i, p_j) a feasible height interval $(h_{ij}^{\min}, h_{ij}^{\max})$, within which the rope remains clear of the mesh. For simplicity, we will use a single bounding interval in the following, although multiple intervals may exist. We also associate with each rope a non-negative variable $t_{ij} \in \mathbb{R}_{\geq 0}$ representing the tension in the rope. The goal is to compute a minimal set of such ropes and their corresponding tensions and heights to balance the internal forces of the tensile structure.

The optimization objective is twofold:

- (1) Ensure that the sum of rope tensions acting on each interior anchor closely counterbalances the net force induced by the mesh's elastic energy at that point.
- (2) Promote sparsity in the rope network by minimizing the number of active ropes.

The resulting optimization problem can be formulated as

$$\begin{aligned} \min_{t_{ij}, h_{ij}} \quad & \sum_i \left\| \sum_j t_{ij} \mathbf{d}_{ij} - \mathbf{f}_i \right\|^2 + \lambda \sum_{i,j} \|t_{ij}\|_0 \\ \text{s.t.} \quad & t_{ij} \geq 0 \quad \text{and} \quad h_{ij}^{\min} \leq h_{ij} \leq \min(h_{ij}^{\max}, h_{\max}). \end{aligned} \quad (6)$$

Here, $\mathbf{f}_i$ is the net force acting on anchor point i , derived from the mesh's energy, and $\mathbf{d}_{ij}$ is the unit direction vector from anchor a_i to pole p_j at height h_{ij} . The ℓ_0 -norm encourages sparsity by penalizing the number of non-zero rope tensions. The constant λ is a blending factor that controls the trade-off between minimizing the number of ropes and ensuring force equilibrium (for the examples in this paper we set it to 0.5). In practice, this term is approximated using an ℓ_1 relaxation or solved via a mixed-integer programming formulation (an example or derivation of rope is shown in Figure 8). To prevent excessively long ropes, particularly when the structure's energy

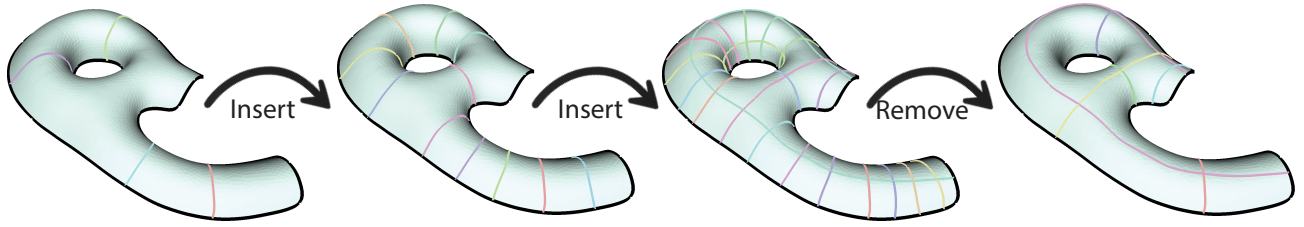


Fig. 6. The result of a few path insertion steps followed by the final layout simplification step.

acts primarily in the vertical direction, we impose a global upper bound h_{max} on all attachment heights and include the option to add a pole in the center.

Note that this formulation does not guarantee force equilibrium at every anchor point, as enforcing it strictly would likely require an excessive number of supporting poles. To address this, we introduce a threshold to control the residual force. If the resulting force at an anchor exceeds this threshold, we add supporting poles. If the number of added poles surpasses a predefined limit, we instead fix the remaining anchors directly to the ground (see Figures 8.b and 2.d).

7 Results

We implemented our system on a Mac with an Apple M3 Pro, using Eigen [Guennebaud et al. 2010] and libigl [Jacobson et al. 2018] as the core geometry processing framework. We used Gurobi [Gurobi Optimization, LLC 2024] to solve the mixed-integer and boolean-constrained optimization problems described in Sections 4.1 and 4.2. The total processing time for the models shown in the paper ranged from 1 to 20 minutes. The main parameter controlling the final output is the maximum allowed approximation error δ_{max} . An example of different outcomes resulting varying this parameter is shown in Figure 7.

Figure 11 show a variety of input architectural models processed by our system. In the examples of Figure 12, we also enforced symmetry in the final shape, a feature often desirable in architectural applications. Figure 8 illustrates several examples where the tensile surface is supported using ropes, poles, or direct ground anchoring. Table 1 summarizes statistics, including patch layout structure, approximation error, and both maximum and average distortion during patch flattening. Distortion is computed as the edge stretch ratio per triangle.

Figure 9 illustrates the advantage of using a curvature-aligned partitioning. This approach yields a simpler and more compact patch decomposition, which typically leads to a better approximation of the target surface.

Fabricated Examples. To fabricate our prototypes, we cut the flattened patches from a slightly stretchable fabric and we 3D-printed the supports and external poles when ropes were used. The fabric pieces were sewn together along each shared edge, and the resulting tensile structure was then placed on the supports and tensioned. The full assembly process took between 120 and 200 minutes for the examples shown in this paper. Figures 1 and 13 present the final fabricated structures alongside the corresponding digital models

generated by our method. Figure 10 shows the approximation error between our tensile structure and the 3D-scanned physical replica presented in Figure 1. The average error amounts to 0.73% of the bounding box diagonal. To further validate our method, we use a discrete thin shell model [Grinspun et al. 2003] to simulate the structure under gravity and report the maximum strain for each triangle at static equilibrium state as in Figure 10 (right). From the result, we can see a relatively large maximum strain, but this only happens for triangles around the anchor points. The overall in-plane strain is still very low.

8 Conclusion

We introduced a novel method to generate a tensile structure from a given freeform input surface. Our tensile structure is composed of a set of compact patches that remain in equilibrium when anchored at a limited number of points. The resulting form is both compact and visually clean, making it feasible for fabrication and suitable for practical applications. Our approach does not impose any constraints on the input surface’s topology or geometry. We demonstrated its versatility across a range of architectural examples.

To the best of our knowledge, this is the first method that presents a complete pipeline for the automatic derivation of this type of structure, which has seen increasing practical adoption in recent years. We successfully fabricated several physical prototypes to validate our approach.

Limitations and Future Work. One current limitation the proposed framework is that, due to its computational complexity, our method does not support real-time interaction. While the system could be adapted to allow the user to manually fix anchor points or insert seams, the results cannot yet be visualized instantaneously.

The initial membrane approximation of the target geometry enforces a tensile structure. However, during mapping of the fabrication patches to 2D we do not explicitly enforce the tension requirement. As can be seen on the fabricated samples this omission has relatively small impact on the overall appearance. However, for functional reasons it can be still interesting to enforce tension also for the 2D patches. We leave this as an opportunity for future work.

In our experiments we focused on open meshes common in architectural context. While our algorithm supports watertight models we find this class of meshes less relevant for the desired application of inexpensive, easy to assemble shelters.

Our simulation assume isotropic material properties. Such an assumption is suitable for stretchable fabrics since up to a certain degree of stretch they behave as isotropic materials. However, after a

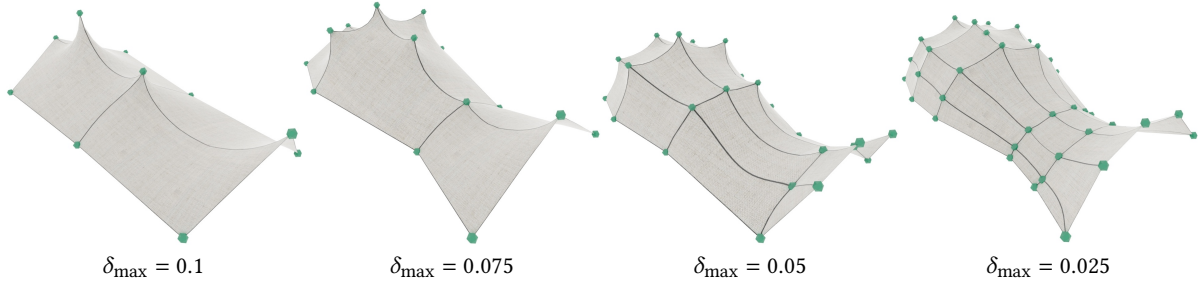
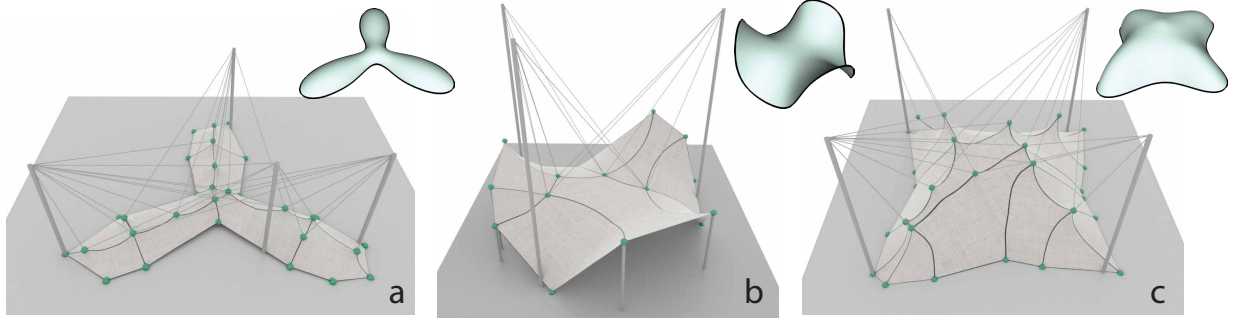
Fig. 7. Influence of the approximation error threshold $\delta_{\max}$ on the final layout.

Fig. 8. Example of tensile structures that combines cables with ground anchors to secure some of the remaining points.

Table 1. Summary statistics for each tensile structure example. The table reports the number of patches (#P), the number of anchor points (#A), the average and maximum geometric error (Avg E and Max E), measured as the Hausdorff distance from the target mesh and expressed as a percentage of the bounding box diagonal. It also includes the average (Avg D) and maximum (Max D) distortion of the 2D mapping, computed as the percentage of edge stretch.

Fig	# P	# A	Avg E	Max E	Avg D
8.a	13	43	1.56	3.77	0.69
8.b	10	24	1.17	4.03	0.69
8.c	19	30	3.41	8.44	0.36
11.a	10	26	3.73	11.90	0.82
11.b	8	52	1.37	5.51	1.09
11.c	4	30	1.24	5.28	1.94
11.d	2	14	1.82	5.83	2.64
12.a	4	15	2.24	5.34	2.14
12.b	6	18	2.79	9.69	0.78
12.c	4	25	0.69	4.98	1.9
12.d	10	68	2.09	10.47	0.78
13 top	12	25	0.23	0.50	0.47
13 bottom	8	15	0.03	0.07	0.32

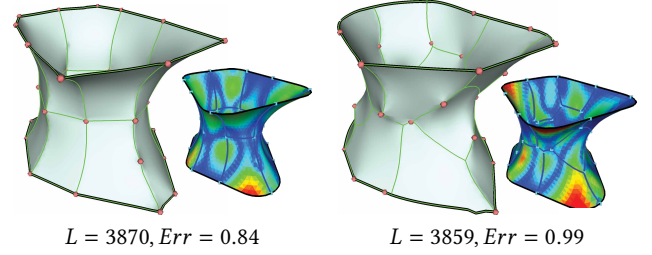


Fig. 9. Comparison between different partitioning strategies used to derive the tensile structure: our field-based optimization (left) versus a Voronoi-based partitioning (right), both processed using our membrane derivation method. Each experiment reports the total seam length and the average Hausdorff distance from the target surface.

certain threshold the exact behavior of a fabric can only be predicted when considering the fiber orientation. Our numerical simulator can be adapted to support fiber orientation similarly to [Eggler et al. 2024; Pietroni et al. 2022]. Such a change would introduce another degree of freedom into the optimization, namely the orientation of the patch. However, in practice we did not find this inclusion critical as the stretch required by our optimized patches is well below the natural stretch of common fabrics. An additional direction for future research involves analyzing and optimizing the structural aspects of the tensile structures derived from our method, with a focus on the physical properties of the employed materials.



Fig. 10. Visualization of the Hausdorff distance between our tensile structure (left) and the physical replica (center). Red indicates a deviation of 2% relative to the bounding box diagonal. Right: Thin shell simulation of a thick fabric ($t = 1.2$ mm, $E = 120$ MPa, $\nu = 0.35$, $\rho = 250$ kg/m³), yielding a maximum strain of 10.6% and an average strain of 0.18%.

Our algorithm exhibits greedy behavior in the layout refinement procedure. We provide a more detailed discussion in the supplementary material including the parameters. Future work could explore improvements for this.

Acknowledgments

The authors thank the anonymous reviewers for their feedback and gratefully appreciate the assistance and valuable expertise of Helga Bargehr and Brigitte Egger-Bargehr in sewing during sample fabrication. The authors also thank Fabio Gramazio, Matthias Kohler, and Sarah Schneider for their initial discussions.

References

- Hassan Baharami, Michal Piovarci, Marco Tarini, Bernd Bickel, and Nico Pietroni. 2025. Fabricable Discretized Ruled Surfaces. *ACM Trans. Graph.* 44, 3, Article 30 (June 2025), 15 pages. doi:10.1145/3734519
- Philippe Block. 2009. *Thrust Network Analysis: Exploring Three-Dimensional Equilibrium*. Ph.D. Thesis. Massachusetts Institute of Technology, Cambridge, MA. <https://dspace.mit.edu/handle/1721.1/49539> Department of Architecture.
- Pengbo Bo and Wenping Wang. 2007. Geodesic-Controlled Developable Surfaces for Modeling Paper Bending. *Comput. Graph. Forum* (2007). doi:10.1111/j.1467-8659.2007.01059.x
- Mario Botsch and Olga Sorkine. 2008. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics* 14, 1 (2008), 213–230.
- Sofien Bouaziz, Mario Deuss, Yuliy Schwartzburg, Thibaut Weise, and Mark Pauly. 2012. Shape-Up: Shaping Discrete Geometry with Projections. *Comput. Graph. Forum* 31, 5 (Aug. 2012), 1657–1667. doi:10.1111/j.1467-8659.2012.03171.x
- M. Campen. 2017. Partitioning Surfaces Into Quadrilateral Patches: A Survey. *Comput. Graph. Forum* 36, 8 (2017), 567–588. doi:10.1111/cgf.13153 arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13153>
- Marcel Campen, David Bommes, and Leif Kobbelt. 2012. Dual loops meshing: quality quad layouts on manifolds. *ACM Trans. Graph.* 31, 4, Article 110 (July 2012), 11 pages. doi:10.1145/2185520.2185606
- Paolo Cignoni, Nico Pietroni, Luigi Malomo, and Roberto Scopigno. 2014. Field-aligned mesh joinery. *ACM Trans. Graph.* 33, 1, Article 11 (Feb. 2014), 12 pages. doi:10.1145/2537852
- Fernando de Goes, Pierre Alliez, Houman Owhadi, and Mathieu Desbrun. 2013. On the equilibrium of simplicial masonry structures. *ACM Trans. Graph.* 32, 4, Article 93 (July 2013), 10 pages. doi:10.1145/2461912.2461932
- Olga Diamanti, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Designing N -PolyVector Fields with Complex Polynomials. *Comput. Graph. Forum (proceedings of EUROGRAPHICS Symposium on Geometry Processing)* 33, 5 (2014), 1–11.
- Anna Maria Egger, Raphael Falque, Mark Liu, Teresa Vidal-Calleja, Olga Sorkine-Hornung, and Nico Pietroni. 2024. Digital Garment Alteration. *Comput. Graph. Forum (proceedings of Pacific Graphics 2024)* 43, 7 (2024).
- Stuart Gale and Wanda J. Lewis. 2016. Patterning of tensile fabric structures with a discrete element model using dynamic relaxation. *Computers & Structures* 169 (2016), 112–121. doi:10.1016/j.compstruc.2016.03.005
- Damien Gauge, Stelian Coros, Sandro Mani, and Bernhard Thomaszewski. 2015. Interactive design of modular tensegrity characters. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Copenhagen, Denmark) (SCA '14). Eurographics Association, Goslar, DEU, 131–138.
- Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) (SCA '03). Eurographics Association, Goslar, DEU, 62–67.
- Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. <http://eigen.tuxfamily.org>.
- Gurobi Optimization, LLC. 2024. Gurobi Optimizer Reference Manual. <https://www.gurobi.com>
- Kristian Hildebrand, Bernd Bickel, and Marc Alexa. 2012. crdbd: Shape Fabrication by Sliding Planar Slices. *Comput. Graph. Forum* 31, 2pt3 (May 2012), 583–592. doi:10.1111/j.1467-8659.2012.03037.x
- Alexandra Ion, Michael Rabinovich, Philipp Herholz, and Olga Sorkine-Hornung. 2020. Shape Approximation by Developable Wrapping. *ACM Trans. Graph. (proceedings of SIGGRAPH ASIA)* 39, 6 (2020). doi:10.1145/3414685.3417835
- Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. <https://libigl.github.io/>.
- Mohammad Arif Kamal. 2020. An Investigation into Tensile Structure System: Construction Morphology and Architectural Interventions. 7 (12 2020), 236–254. doi:10.5281/zenodo.4308968
- Minchen Li, Danny M. Kaufman, Vladimir G. Kim, Justin Solomon, and Alla Sheffer. 2018. OptCuts: joint optimization of surface cuts and parameterization. *ACM Trans. Graph.* 37, 6, Article 247 (Dec. 2018), 13 pages. doi:10.1145/3272127.3275042
- Yang Liu, Hao Pan, John Snyder, Wenping Wang, and Baining Guo. 2013. Computing self-supporting surfaces by regular triangulation. *ACM Trans. Graph.* 32, 4, Article 92 (July 2013), 10 pages. doi:10.1145/2461912.2461927
- Marco Livesu, Nico Pietroni, Enrico Puppo, Alla Sheffer, and Paolo Cignoni. 2020. LoopyCuts: practical feature-preserving block decomposition for strongly hex-dominant meshing. *ACM Trans. Graph.* 39, 4, Article 121 (Aug. 2020), 17 pages. doi:10.1145/3386569.3392472
- Andrea Micheletti and Paolo Podio-Guidugli. 2022. Seventy years of tensegrities (and counting). *Archive of Applied Mechanics* 92, 9 (2022), 2525–2548. doi:10.1007/s00419-022-02192-4
- Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118. doi:10.1016/j.jvcir.2007.01.005
- Stefano Nuvoli, Alex Hernandez, Claudio Esperança, Riccardo Scateni, Paolo Cignoni, and Nico Pietroni. 2019. QuadMixer: layout preserving blending of quadrilateral meshes. *ACM Trans. Graph.* 38, 6, Article 180 (Nov. 2019), 13 pages. doi:10.1145/3355089.3356542
- Daniele Panozzo, Philippe Block, and Olga Sorkine-Hornung. 2013. Designing Unreinforced Masonry Models. *ACM Trans. Graph. (proceedings of ACM SIGGRAPH)* 32, 4 (2013), 91:1–91:12.
- Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy. 2015. Design and fabrication of flexible rod meshes. *ACM Trans. Graph.* 34, 4, Article 138 (July 2015), 12 pages. doi:10.1145/2766998
- Nico Pietroni, Corentin Dumery, Raphael Falque, Mark Liu, Teresa Vidal-Calleja, and Olga Sorkine-Hornung. 2022. Computational pattern making from 3D garment models. *ACM Trans. Graph.* 41, 4, Article 157 (2022), 14 pages. doi:10.1145/3528223.3530145
- Nico Pietroni, Stefano Nuvoli, Thomas Alderighi, Paolo Cignoni, and Marco Tarini. 2021. Reliable feature-line driven quad-remeshing. *ACM Trans. Graph.* 40, 4, Article 155 (July 2021), 17 pages. doi:10.1145/3450626.3459941
- Nico Pietroni, Enrico Puppo, Giorgio Marcias, Roberto Roberto, and Paolo Cignoni. 2016. Tracing Field-Coherent Quad Layouts. *Comput. Graph. Forum* 35, 7 (Oct. 2016), 485–496.
- 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. doi:10.1145/3130800.3130809
- Nico Pietroni, Davide Tonelli, Enrico Puppo, Maurizio Froli, Roberto Scopigno, and Paolo Cignoni. 2015. Statics Aware Grid Shells. *Comput. Graph. Forum (Special issue of EUROGRAPHICS 2015)* 34, 2 (2015), 627–641. <http://vcg.isti.cnr.it/Publications/2015/PTPFSC15>
- Helmut Pottmann, Michael Eigensatz, Amir Vaxman, and Johannes Wallner. 2015. Architectural geometry. *Comput. Graph.* 47 (2015), 145–164. <http://dblp.uni-trier.de/db/journals/cg/cg47.html#PottmannEVW15>
- Michael Rabinovich, Tim Hoffmann, and Olga Sorkine-Hornung. 2018. Discrete Geodesic Nets for Modeling Developable Surfaces. *ACM Trans. Graph.* 37, 2 (2018).
- Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable Locally Injective Mappings. *ACM Trans. Graph.* 36, 2, Article 16 (April 2017), 16 pages. doi:10.1145/2983621
- Faniyir H. Razafindrazaka, Ulrich Reitebuch, and Konrad Polthier. 2015. Perfect matching quad layouts for manifold meshes. In *Proceedings of the Eurographics Symposium*

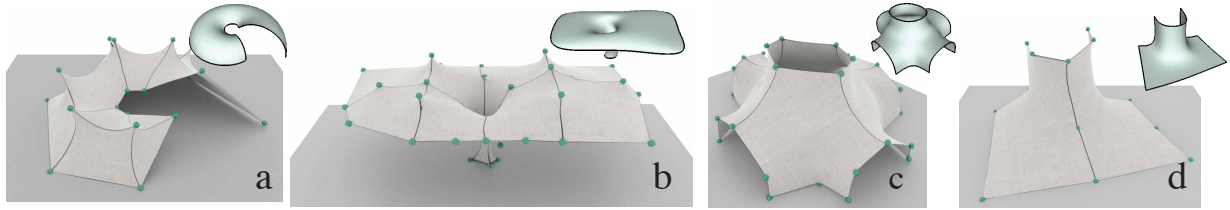


Fig. 11. A gallery of examples and their target mesh.

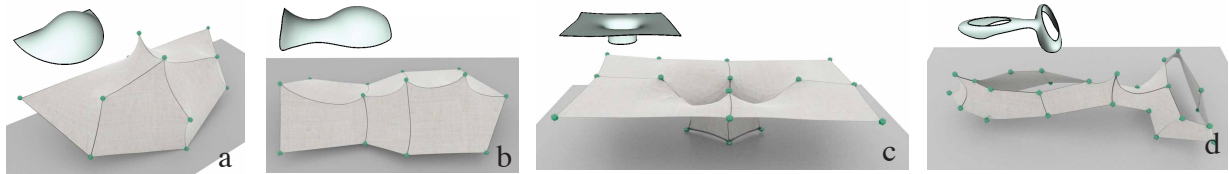


Fig. 12. A gallery of symmetric examples and their target mesh.



Fig. 13. Tensile structures fabricated using our method based on architectural designs. The structures are supported only at the optimized locations. One option is to construct pillars from the ground that hold the cloth in place (top row). For challenging designs, or to improve internal space we propose to use tension strings (bottom row). Our method automatically identifies a sparse set of rods and strings that can be used to support the structure. The final structures are close to the input target surfaces and possess high visual appeal associated with tensile structures.

- on *Geometry Processing* (Graz, Austria) (SGP '15). Eurographics Association, Goslar, DEU, 219–228. doi:10.1111/cgf.12710
- H. Schek. 1974. The force density method for form finding and computation of general networks. *Computer Methods in Applied Mechanics and Engineering* 3, 1 (Jan. 1974), 115–134. doi:10.1016/0045-7825(74)90045-0
- Yuliy Schwartzburg and Mark Pauly. 2013. Fabrication-aware Design with Intersecting Planar Pieces. *Comput. Graph. Forum* 32, 2pt3 (2013), 317–326. doi:10.1111/cgf.12051
- Silvia Sellán, Noam Aigerman, and Alec Jacobson. 2020. Developability of heightfields via rank minimization. *ACM Trans. Graph.* 39, 4, Article 109 (Aug. 2020), 15 pages. doi:10.1145/3386569.3392419
- Nicholas Sharp and Keenan Crane. 2018. Variational Surface Cutting. *ACM Trans. Graph.* 37, 4 (2018).
- Emmanuel Siéfert, Etienne Reyssat, José Bico, and Benoît Roman. 2019. Programming curvilinear paths of flat inflatables. *Proceedings of the National Academy of Sciences* 116, 34 (2019), 16692–16696. doi:10.1073/pnas.1904544116 arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.1904544116
- Mélina Skouras, Bernhard Thomaszewski, Bernd Bickel, and Markus Gross. 2012. Computational Design of Rubber Balloons. *Comput. Graph. Forum* 31, 2pt4 (May 2012), 835–844. doi:10.1111/j.1467-8659.2012.03064.x
- Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. 2014. Designing inflatable structures. *ACM Trans. Graph.* 33, 4, Article 63 (July 2014), 10 pages. doi:10.1145/2601097.2601166
- O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (Nice, France) (SGP '04). Association for Computing Machinery, New York, NY, USA, 175–184. doi:10.1145/1057432.1057456
- Chengcheng Tang, Pengbo Bo, Johannes Wallner, and Helmut Pottmann. 2016. Interactive Design of Developable Surfaces. *ACM Trans. Graph.* 35, 2, Article 12 (Jan. 2016), 12 pages. doi:10.1145/2832906
- D. Tonelli, N. Pietroni, E. Puppo, M. Froli, P. Cignoni, G. Amendola, and R. Scopigno. 2016. Stability of Statics Aware Voronoi Grid-Shells. *Engineering Structures* 116 (2016), 70–82. doi:10.1016/j.engstruct.2016.02.049
- Amir Vaxman, Marcel Campen, Olga Diamanti, David Bommes, Klaus Hildebrandt, Mirela Ben-Chen Technion, and Daniele Panozzo. 2017. Directional field synthesis, design, and processing. In *ACM SIGGRAPH 2017 Courses* (Los Angeles, California) (SIGGRAPH '17). Association for Computing Machinery, New York, NY, USA, Article 12, 30 pages. doi:10.1145/3084873.3084921
- D. Veenendaal and P. Block. 2012. An overview and comparison of structural form finding methods for general networks. *International Journal of Solids and Structures* 49, 26 (2012), 3741–3753. doi:10.1016/j.ijsolstr.2012.08.008
- Etienne Vouga, Mathias Höbinger, Johannes Wallner, and Helmut Pottmann. 2012. Design of self-supporting surfaces. *ACM Trans. Graph.* 31, 4, Article 87 (July 2012), 11 pages. doi:10.1145/2185520.2185583
- Rosemarie Wagner. 2005. *On the Design Process of Tensile Structures*. Springer Netherlands, Dordrecht, 1–16. doi:10.1007/1-4020-3317-6_1
- Emily Whiting, John Ochsendorf, and Frédo Durand. 2009. Procedural modeling of structurally-sound masonry buildings. *ACM Trans. Graph.* 28, 5 (Dec. 2009), 1–9. doi:10.1145/1618452.1618458
- Emily Whiting, Hijung Shin, Robert Wang, John Ochsendorf, and Frédo Durand. 2012. Structural optimization of 3D masonry buildings. *ACM Trans. Graph.* 31, 6, Article 159 (Nov. 2012), 11 pages. doi:10.1145/2366145.2366178
- Shin Yoshizawa, Alexander Belyaev, and Hans-Peter Seidel. 2004. A Fast and Simple Stretch-Minimizing Mesh Parameterization. In *Proceedings of the Shape Modeling International 2004 (SMI '04)*. IEEE Computer Society, USA, 200–208.
- Kai Zhang. 2023. *Inflatable Patternner*. Accessed: 2025-08-13.
- Zheng-Yu Zhao, Qing Fang, Wenqing Ouyang, Zheng Zhang, Ligang Liu, and Xiao-Ming Fu. 2022. Developability-driven piecewise approximations for triangular meshes. *ACM Trans. Graph.* 41, 4, Article 43 (July 2022), 13 pages. doi:10.1145/3528223.3530117
- Zheng-Yu Zhao, Mo Li, Zheng Zhang, Qing Fang, Ligang Liu, and Xiao-Ming Fu. 2023. Evolutionary Piecewise Developable Approximations. *ACM Trans. Graph.* 42, 4, Article 120 (July 2023), 14 pages. doi:10.1145/3592140