Designing Inflatable Structures

Mélina Skouras[1,2] Bernhard Thomaszewski[2] Peter Kaufmann[2] Akash Garg[3] Bernd Bickel [2]
Eitan Grinspun [3] Markus Gross[1,2]

[1]ETH Zurich [2]Disney Research Zurich [3]Columbia University

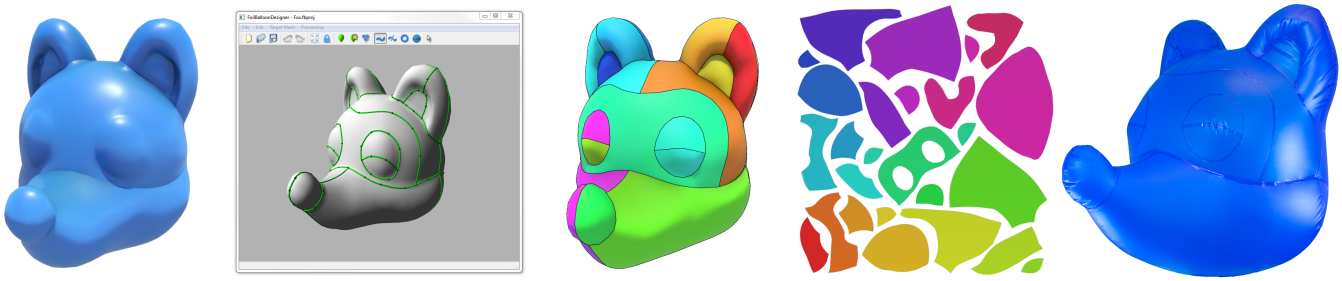


Figure 1: *An overview of our design system: the user provides a target shape (left) and sketches seams to indicate desired segment boundaries (2^{nd} from left). Our system automatically computes flat panels such that the inflated structure (middle) is as close as possible to the target. The generated panels (2^{nd} from right) can be used to fabricate a physical prototype (right).*

Abstract

We propose an interactive, optimization-in-the-loop tool for designing inflatable structures. Given a target shape, the user draws a network of seams defining desired segment boundaries in 3D. Our method computes optimally-shaped flat panels for the segments, such that the inflated structure is as close as possible to the target while satisfying the desired seam positions. Our approach is underpinned by physics-based pattern optimization, accurate coarse-scale simulation using tension field theory, and a specialized constraint-optimization method. Our system is fast enough to warrant interactive exploration of different seam layouts, including internal connections, and their effects on the inflated shape. We demonstrate the resulting design process on a varied set of simulation examples, some of which we have fabricated, demonstrating excellent agreement with the design intent.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

Keywords: computational design, physics-based modeling, pattern optimization, seam design

Links: DL PDF

1 Introduction

Inflatables are structures made of flat membrane elements that assume complex curved shapes when pressurized. Thanks to their lightweight nature, rapid deployment, and cost efficiency, they enjoy widespread popularity in entertainment, advertisement, engineering, and architecture. From foil balloons to parade floats, inflatable furniture to portable architectural structures—myriad applications abound.

Designing inflatable structures requires solving a complicated *patterning* problem: what is the shape of the flat panels that we must cut, and how must we interconnect the panels, such that the final assembly inflates to the desired curved shape? This task is extremely challenging since the designer must anticipate, and invert, the effects of pressure on the shape of the structure, while simultaneously taking into account the aesthetics of seams. The combination of functional and aesthetic requirements make patterning the most difficult aspect of current manual design processes.

We propose a computational approach for the interactive design of inflatable structures. We assume that the designer already has a certain *target shape* at hand, perhaps acquired from real world data, designed via modeling software, or provided as a specification by a client. Our goal is to help the designer to make plans for an inflatable structure that corresponds to the given target.

Our approach ensures that the designer retains full control over aesthetic considerations; to do so, we lay aside fully automated approaches in favor of an interactive, optimization-in-the-loop methodology. As the designer sketches the proposed placement of seams, the underlying optimizer alleviates the iterative and error-prone tasks of reverse-engineering the physics of inflation, proposing a set of panels that best accommodate the desired seams and target shape.

The simplicity of this user experience requires some complex machinery under the hood. We develop a fast physics-based model for inflatable membranes motivated by tension field theory, and we employ a dedicated optimization method for computing the shape of the 2D patterns. We demonstrate the resulting design process on a set of inflatable structures with complex shapes and elaborate seam layouts. Finally, we validate the feasibility of our designs on three physical prototypes.

2 Related Work

Fabrication-Oriented Computational Design seeks to develop software, typically based on a mix of interaction- and optimization-

based approaches, that facilitates the design of artifacts, taking into account not only considerations of functionality and aesthetics, but also manufacturing and assembly. Recently, several approaches have been presented for translating functional goals such as appearance [Hullin et al. 2013], articulation [Bächer et al. 2012; Calì et al. 2012], deformation behavior [Bickel et al. 2012; Skouras et al. 2013], kinematic motion [Zhu et al. 2012; Coros et al. 2013; Ceylan et al. 2013], or a combination of these [Chen et al. 2013] into manufacturable designs. While many design problems admit fully automatic approaches, many others require that the user remain in the loop. In such cases, interactive systems can help their users to explore complex design spaces, for instance to create physically valid furniture [Umetani et al. 2012; Lau et al. 2011].

The design of inflatable balloons was recently investigated by Skouras et al. [2012]. However, they focused on rubber balloons which (*a*) stretch significantly during inflation, (*b*) have 3D rest shapes fabricated by dipping a mold into liquid rubber, and (*c*) are typically pliant even when maximally inflated. By contrast, our inflatable structures (*a*) have great membrane stiffness and negligible stretching strain, yield more durable artifacts; (*b*) are manufactured by assembly of piecewise *flat* rest shapes, facilitating maintenance and repair, allowing for larger physical dimensions, and permitting decoration with paint, ink, or other appliques using standard printing methods; and (*c*) can be quite stiff when inflated, expanding the potential functionality beyond the realm of purely decorative. Finally, the space of designs we consider is richer than that spanned by rubber balloons, including complex shapes with sharp creases. The wrinkling behaviour of inflated membranes was studied by Baginski et al. [2008], who used forward simulations on simple patterns to determine the shape of high altitude scientific balloons. Inspired by their work, we also employ tension field theory.

Tools to computationally design objects made of flat patterns have already been proposed in the past. Umetani et al. [2011] presented *Sensitive Couture*, an interactive tool for garment design; Mori and Igarshi proposed *Plushie* to create custom plush toys [2007], which was extended to non-stretchy materials by Futura et al. [2010]. However, some fundamental differences distinguish our tool from *Plushie*. Whereas *Plushie* projects begin with a blank canvas, our projects begin with a given target shape. *Plushie* helps the user to focus on the modeling task, whereas our work helps the user to focus on the control of seam placement and panel shape, essential for obtaining compelling designs for myriad organic forms, and for geometric shapes such as simple sphere (see Fig. 6). In that sense, the workflow of our system draws inspiration from *Pillow*'s [Igarashi and Igarashi 2008]. However, unlike our method, *Pillow* employs flattening that does not include an inflation-based metric, nor warrants satisfaction of fabrication constraints. This target-driven approach also shares some similarities with the one employed by Wang and Tang [2010] in the context of compression garment design. However, the underlying goals and problems differ substantially: their system aims at producing clothing with prescribed strains and normal pressures, whereas our method focuses on matching an inflated shape while adhering to user-provided seam constraints.

Automatic Segmentation and Parametrization techniques seek to create mappings with low distortion [Hormann et al. 2007]. For example, *DCharts* [Julius et al. 2005] segments models into almost developable patches, which can then be flattened using *ABF++* [Sheffer et al. 2005] to create approximately conformal parametrizations with small stretch. Flattening multiple panels independently generally produces incompatible seam lengths, requiring alterations such as pleats or cuts that complicate fabrication and impose a very particular visual appearance. To address this point, Wang [2008] considered flattening subject to both minimal stretch and seam length compatibility. We draw inspiration from Wang's approach to compute initial guesses for pattern optimization.

An alternative approach is to approximate a 3D shape by a set of ruled surfaces such as cones and planes [Shatz et al. 2006], generalized cylinders [Massarwi et al. 2007]), or paper strips [Mitani and Suzuki 2004]. Also, there are a number of tools for modeling developable surfaces. Kilian et al. [2008] use curved creases to design complex-shaped surfaces from planar sheets. Solomon et al. [2012] describe an interactive design system that exactly satisfies discrete developability conditions at all times. While these methods work well for designing physically-realizable surfaces made, e.g., out of paper, developability does not guarantee that the object's surface is in equilibrium under pressure. We directly optimize for developable patches that, when joined together and inflated, approximate a desired 3D shape at equilibrium. However, our method does not necessarily lead to an inflated shapes with zero Gaussian curvature and therefore does not guarantee a piecewise developable surface.

Cloth and Shell Simulation Similar to inflatables, cloth buckles at the onset of compression. This behavior leads to the characteristic folding patterns that define the typical appearance of real textiles, but it is inherently difficult to treat numerically: compression gives rise to negative eigenvalues in the force Jacobian and thus slows or even breaks most linear solvers. Choi and Ko [2002] proposed a mass-spring model that turns off force and Jacobian contribution from compressed springs and replaces them with custom-tailored *buckling springs*. Another approach to combat indefiniteness was presented by Teran et al. [2005], who clamp negative eigenvalues of elemental stiffness matrices. Instead of trying to avoid indefiniteness, the method of Rohmer et al. [2010] exploits the compression field extracted from elemental deformation tensors in order to add detailed wrinkles to a coarse simulation. Inspired by tension field theory [Pipkin 1986; Steigmann 1990], we propose a fast physics-based model that addresses the difficulty of wrinkling analysis with a relaxed energy formulation that fades to zero before compressive stresses can occur.

3 System Overview

The goal of our system is a workflow that makes the design of inflatable structures intuitive and efficient for the user. We start the description by making precise the notion of inflatable structures.

3.1 Anatomy of Inflatable Structures

Inflatable structures are made from flat *panels*, i.e., thin layers of metallic foil, vinyl, or textile. While our approach does not exclude stretchy materials such as rubber per se, we target structures that show little stretch but large bending deformations. We therefore focus on quasi-inextensible materials that exhibit a high resistance to stretching, but are compliant to bending.

A connection between two panels is called a *seam*. Depending on the type of material, seams are created through gluing, heat sealing, or stitching. As an important design constraint, the segments forming the seam should have the same length on both panels; otherwise, the design must be altered during manufacturing using cuts or pleats.

The shape of inflatable structures is governed by the requirement that they have to be *stable under pressure*: the pressure forces must be balanced by membrane forces in every point on the surface. This equilibrium constraint puts limits on what kind of shapes can be obtained with a structure that consists of a single closed surface. However, the space of possible designs can be significantly enlarged by

Figure 2: *Internal connections (left) allow the designer to realize shapes with sharp creases and concave features (middle) that would be poorly approximated without internal structures (right).*

allowing for *internal connections* (see Fig. 2). Such internal connections can be used to attach parts of the surface to each other that would otherwise be pushed apart by the pressure forces. They can also be used to generate creased feature curves.

3.2 Design Loop

Interface The design interface consists of three views: the inflated view, target view, and pattern view (see Fig. 3). Each new design session starts by loading a closed triangle mesh that represents the 3D target shape. The user incrementally builds a seam layout that partitions the target mesh into a set of segments. Once a seam layout or edit is committed, our system flattens the corresponding segments. The inflated view immediately shows a preliminary shape for the resulting structure, obtained by inflating the model with the current patterns in simulation (Sec. 4). Meanwhile, the system continuously optimizes the pattern (Sec. 5). Both views are continuously updated while the optimization proceeds in the background.

Seam Design The user draws seams directly on the target model using a spline tool that implements a *geodesics* metaphor, i.e., connects seam points by taking an approximately shortest path on the surface. We represent seam curves using cubic Hermite splines that are defined through a coarse set of 3D control points. For segmentation, we simply project the spline curve onto the surface mesh. The seam tool supports snap-on functionality in order to link new seams to existing ones. In addition, the user can also edit existing seams by simply dragging control points.

Generally, some of the seams will have aesthetic purposes or requirements, while others simply subdivide a larger region in order

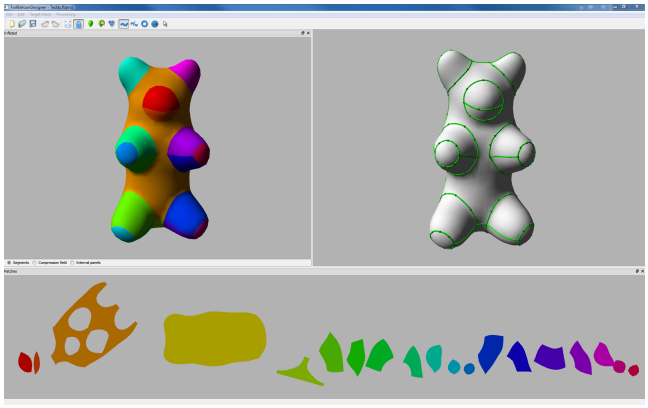


Figure 3: *Our design interface, showing the inflated view (top left), target view (top right), and pattern view (bottom).*

to increase the shape approximation quality. For each of the seams, the user can therefore specify a weight that indicates how important it is for the seam to remain in its original location with respect to the target shape. These conditions are then enforced through corresponding objectives during pattern optimization.

Internal Connections As a necessary condition for a given shape to be a feasible balloon, it has to be stable under pressure. Clearly, this requirement limits the space of shapes that can be realized as balloons. However, the design space can be significantly enlarged by allowing for internal connections, i.e., panels that are not visible from the outside and serve a purely functional purpose. Internal connections are created by connecting existing seams on the surface as indicated by the user. Technically, internal connections are no different from the other, visible patches. However, paired with our optimization, they provide a powerful tool for creating complex shapes with distinct features (see Fig. 2).

Pattern Optimization tightly integrates with simulation in order to compute flat panels that allow for an optimal approximation of the target shape. Whenever a seam layout or edit is committed by the user, the involved segments are first flattened in order to obtain an initial guess for the pattern shape (Sec. 5.2). Afterwards, the shapes of the flat panels are computed by optimizing various objectives, including shape approximation and seam quality (Sec. 5.1). In order to deal with the overall nonlinear nature of the problem and the large changes in pattern shapes, our system uses an iterative optimization algorithm based on Sequential Quadratic Programming (SQP) with integrated remeshing (Sec. 5.3). Although this method can take some time to fully converge, the results are typically visually stable after just a few seconds. The user can thus quickly explore different seam layouts without a disruptive delay.

4 Simulation

As a core component of our design system, we must be able to rapidly compute the deformed shape of inflatable structures. Like many other thin structures, inflatables exhibit a strong resistance to stretching but will wrinkle at the onset of compression. This behavior poses challenges that our simulation must confront efficiently.

4.1 Origins of Compression

As a didactic example, consider the simple foil balloon depicted in Fig. 4, assembled from two disc-shaped panels. When inflating the balloon, we expect that (*1*) the distance between the centers of the two panels increases due to *pressure*; (*2*) the seam remains on its original plane due to *symmetry*; (*3*) each radial line, extending from the center to the seam, will remain unstretched due to *inextensibility*. To meet these requirements the diameter *must* decrease during inflation. Correspondingly, the circumference must shrink, implying a compressive deformation on the seam that is resolved through the typical wrinkles observed in foil balloons (Fig. 4, *right*). Wrinkling is a characteristic trait of thin surface structures, but a major struggle for simulation codes. First, compressions give rise to negative eigenvalues in the energy Hessian, thus breaking the fundamental assumption of most fast linear solvers, i.e., a positive-definite matrix. Second, since the location of wrinkles can usually not be predicted, a simulation mesh with uniformly high resolution is required. Clearly, both these properties are highly detrimental to efficiency.

The problem of compressions in thin surfaces is not new to graphics. For example, Choi and Ko [2002] proposed a modified mass-spring system that allows for stable animations of buckling cloth.

Both the original work and its extension to triangle meshes [Choi and Ko 2003] handle compressions along buckling springs that connect pairs of particles that are at topological distance two. However, in order to provide accurate results even for coarse simulation meshes, we would like a model that is able to handle compressions along arbitrary directions, irrespective of mesh structure.

4.2 Tension Field Theory

The wrinkling of membranes has been intensively studied in mathematical and physical sciences [Pipkin 1986; Steigmann 1990]. The difficulty of wrinkling analysis stems from the fact that the elastic energy density is not convex in the presence of compressions, jeopardizing the uniqueness and existence of solutions. Tension field theory offers a solution to this problem by postulating a *relaxed* energy density that reflects the average energy value in wrinkled regions and fades to zero before compressive stresses can occur. The underlying reasoning is that while wrinkled regions can carry longitudinal loads, they do not exhibit resistance to transversal deformations. This formulation enables a macroscopic treatment of wrinkling that accurately captures the deformation behavior on the coarse level but abstracts away geometric detail. There are two important advantages of this approach, both of which translate directly into computational efficiency: it requires fewer elements and it removes the problems due to indefiniteness. It is furthermore worth noting that, although the tension field approach does not directly provide geometric information on the wrinkles, the compression field does give strong indications on the locations and the directions of the expected wrinkles. We found these regions to be in very good correspondence with both the locations predicted by a high-res simulation of the full model and our actual, fabricated prototypes.

Relaxed Energy Density The deformation around a given point on the surface is described by the deformation gradient $\mathbf{F}$. With a view to the relaxed strain energy density, we introduce the 2×2 right Cauchy Green tensor

$$\mathbf{C} = \mathbf{F}^T\mathbf{F} = \lambda_1\mathbf{N}_1\mathbf{N}_1^t + \lambda_2\mathbf{N}_2\mathbf{N}_2^t \,, \tag{1}$$

in terms of its principal stretches λ_1, λ_2 and corresponding eigenvectors $\mathbf{N}_1$, $\mathbf{N}_2$. Without loss of generality, we assume that $\lambda_1 \geq \lambda_2$. Assuming that the material is incompressible and does not exhibit transverse shearing, we can expand $\mathbf{C}$ to the 3×3 tensor

$$\hat{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & 0 \\ 0 & J^{-1} \end{bmatrix} \,, \tag{2}$$

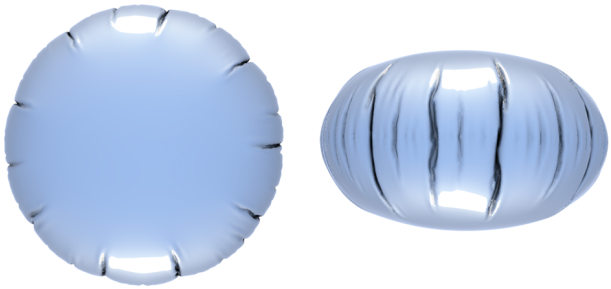


Figure 4: *Compression-induced wrinkling in a simple foil balloon.*

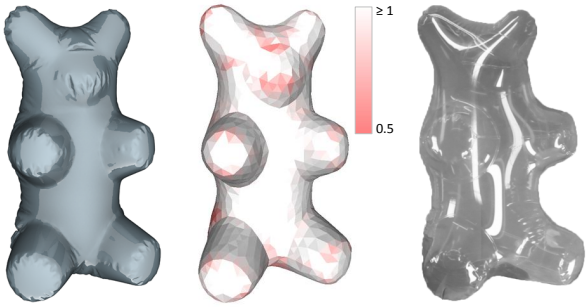


Figure 5: *Wrinkling analysis using the full model (*left*), compressive deformations from the tension field model (*middle*), real-world prototype (*right*). Colors indicate compressed elements with deformations ranging from $\lambda_2 = 0.5$ to $\lambda_2 = 1$.*

where $J = \lambda_1\lambda_2$ is the determinant of $\mathbf{C}$. This deformation measure is amenable to standard material models and we opt for a Neo-Hookean material, whose strain energy density is defined as

$$\psi = \kappa\left(\mathrm{tr}(\hat{\mathbf{C}}) - 3\right) = \kappa\left(\lambda_1 + \lambda_2 + \frac{1}{\lambda_1\lambda_2} - 3\right) \,, \tag{3}$$

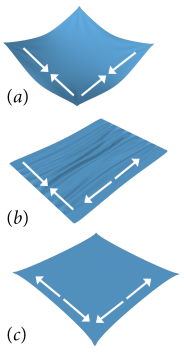
where κ is the stiffness coefficient. As described above, the essence of tension field theory can be condensed into a relaxed strain energy density

$$\tilde{\psi}(\lambda_1,\lambda_2) = \begin{cases} 0 & \lambda_1 < 1, \lambda_2 < 1 \\ \psi(\lambda_1, \tilde{\lambda}_2(\lambda_1)) & \lambda_1 \geq 1, \lambda_2 < \tilde{\lambda}_2(\lambda_1) \\ \psi(\lambda_1,\lambda_2) & \lambda_1 \geq 1, \lambda_2 \geq \tilde{\lambda}_2(\lambda_1) \end{cases} \,, \tag{4}$$

where $\tilde{\lambda}_2$ is the energetic minimum of λ_2,

$$\tilde{\lambda}_2(\lambda_1) = \underset{\lambda_2}{\mathrm{argmin}}\ \psi(\lambda_1,\lambda_2) = \frac{1}{\sqrt{\lambda_1}} \,. \tag{5}$$

The three cases listed in (4) are illustrated in the inset figure. For the first case (*a*), the surface is assumed to be *slack*, i.e., both stretches are negative and the energy is set to zero. The second case (*b*) corresponds to wrinkling and the original model is applied with the compressive stretch replaced by its energetically optimal value. The third case (*c*) corresponds to a *taught* surface with both stretches positive for which the original model can be applied without modifications. Since the first and second cases are energetically optimal with respect to compressive stretch, the material model will never give rise to compressive stresses.



It is worth noting that the modified energy density $\tilde{\psi}$ is convex but only C^1 continuous. Although we did not notice any adverse effects when solving for equilibrium states, the discontinuous force derivatives pose a significant problem for optimization. We therefore smooth the transitions between the different regimes using quadratic interpolation as described in the supplemental material.

4.3 Discretization

Notation Let m denote the number of external and internal panels of the inflatable structure. The geometry of each panel in its

undeformed state is described by a triangle mesh $\mathcal{P}_i$. The deformed mesh, comprising all panels, is denoted by $\mathcal{M}$. Furthermore, we let $\mathbf{x} \in \mathbb{R}^{3n}$ denote the vector of deformed positions for the n nodes of the surface. Likewise, $\mathbf{X} \in \mathbb{R}^{2N}$ holds the positions of the N undeformed panel vertices in their two-dimensional domain. Note that $N > n$ since, for each deformed vertex on a panel boundary in $\mathcal{M}$, there are at least two corresponding undeformed vertices from boundaries of different panels.

Forces We use a standard finite element approach based on Constant Strain Triangles for discretizing (4) and its derivatives. The internal forces $\mathbf{f}_i^{\text{int}}$ at each node are obtained as

$$\mathbf{f}_i^{\text{int}} = -\sum_{e \in \mathcal{F}_i} \frac{\partial \tilde{\psi}^e}{\partial \mathbf{x}_i} V_e = -\sum_{e \in \mathcal{F}_i} \left(\frac{\partial \tilde{\psi}^e}{\partial \lambda_1^e} \frac{\partial \lambda_i^e}{\partial \mathbf{x}_i} + \frac{\partial \tilde{\psi}^e}{\partial \lambda_2^e} \frac{\partial \lambda_2^e}{\partial \mathbf{x}_i} \right) h A_e \,,$$

where $\mathcal{F}_i$ denotes the set of triangle elements incident to the vertex i, A_e is the initial area of element e, h is the thickness of the panels and $V_e = hA_e$ is the volume of e. It is evident from this expression that the gradient and Hessian of (4) require the first and second derivatives of the principal stretches. We provide the corresponding derivations and other details in the supplemental document.

The pressure forces are defined directly in the discrete setting as

$$\mathbf{f}_i^{\text{p}} = p \frac{\partial V}{\partial \mathbf{x}_i} = \sum_{e \in \mathcal{F}_i} \frac{1}{3} p A_e \mathbf{n}_e \,, \tag{6}$$

where p is the pressure value and $\mathbf{n}_e$ and A_e are the outward normal and the area of element e. The shape of the inflated structure can then be computed by solving the static equilibrium problem

$$\mathbf{f}_i^{\text{int}}(\mathbf{x}, \mathbf{X}) + \mathbf{f}_i^{\text{p}}(\mathbf{x}) = 0 \,, \quad 1 \leq i \leq n \,. \tag{7}$$

5 Automatic Pattern Generation

Our system combines user-guided seam design with automatic pattern generation. During seam design, the user will repeatedly invoke the pattern optimization scheme in order to explore the effect of a given layout or edit on the inflated structure.

Formally, given a target mesh $\mathcal{T}$ and a set of seam lines segmenting the mesh into m parts, we seek to find optimal panel shapes $\mathbf{X}$ for each part such that the distance between the inflated mesh $\mathcal{M}$ and the target mesh $\mathcal{T}$ is as small as possible. We cast this goal into the form of a constrained minimization problem,

$$\min_{\mathbf{x}, \mathbf{X}} E(\mathbf{x}, \mathbf{X}) \quad \text{s.t. } \mathbf{f}(\mathbf{x}, \mathbf{X}) = \mathbf{0} \,, \tag{8}$$

where the constraints $\mathbf{f}(\mathbf{x}, \mathbf{X}) = \mathbf{0}$ require force equilibrium in every node and $E(\mathbf{x}, \mathbf{X})$ summarizes various objective terms. We solve this optimization problem using the Sequential Quadratic Programming (SQP) method described by Byrd et al. [2010], which guarantees progress even in non-convex regions.

5.1 Objectives

Distance to Target In order to quantify the distance between the inflated mesh $\mathcal{M}$ and the target mesh $\mathcal{T}$, we construct a distance field on $\mathcal{T}$ using implicit moving least squares [Öztireli et al. 2009]. The distance penalty is defined as

$$E_{\text{target}}(\mathbf{x}) = \sum_i \frac{\sum_k \mathbf{n}_k \cdot (\mathbf{x}_i - \mathbf{c}_k) \phi_k(\mathbf{x})}{\sum_k \phi_k(\mathbf{x})} \,, \tag{9}$$

where $\phi_k(\mathbf{x}) = \left(1 - \frac{||\mathbf{x} - \mathbf{X}_k||_2^2}{h^2}\right)^4$ are locally-supported kernel functions that vanish beyond their support radius h, that we set to twice the average length of the target mesh edges, while $\mathbf{c}_k$ and $\mathbf{n}_k$ denote the vertex positions and normals of $\mathcal{T}$, respectively. This measure allows the vertices of $\mathcal{M}$ to slide freely over $\mathcal{T}$, whereas a simpler pair-wise vertex distance would lead to bias and thus unnecessarily restrict approximation quality.

Seam Locations Seams are critical to the aesthetics of inflatable structures. Our interface provides tools that allow the user to rapidly create seam layouts on the target surface. Some of these seams simply split larger areas into smaller parts, in which case their exact location is not critical. Our optimization can leverage such freedom for better shape approximation. Others, however, serve an important aesthetic role that the final inflatable structure has to respect by adhering to the shape and location of the seams. We therefore let the user specify the importance of a given seam $\mathcal{S}_i$ by assigning a weight σ_i to it that determines how strongly the corresponding seam vertices on $\mathcal{M}$ are attracted to their target locations on $\mathcal{S}_i$. We define the penalty function based on vertex-wise L_2-distance as

$$E_{\text{seam}}^i(\mathbf{x}) = \sigma_i \sum_{j \in \mathcal{S}_j} ||\mathbf{x}_j - \mathbf{s}_j||_2^2 \,, \tag{10}$$

where $\mathbf{s}_j$ denotes the target position for $\mathbf{x}_j$ on the seam. It is worth noting that the seam vertices are not restricted to be a subset of the vertices of $\mathcal{M}$—seams can run freely across the target surface.

Fabrication Constraints In order for two panels $\mathcal{P}_j$ and $\mathcal{P}_k$ to join in a seam, the corresponding boundary segments must have the same length on both panels. Otherwise, discrepancies have to be corrected *a posteriori* using cuts or pleats, which increases fabrication time and degrades the visual quality of the product. We enforce this equal-length requirement per seam as

$$E_{\text{length}}(\mathbf{X}) = \sum_{i=1}^{s_e} (L_j^i(\mathbf{X}) - L_k^i(\mathbf{X}))^2 \,, \tag{11}$$

where s_e is the number of seam edges in $\mathcal{M}$ and L_j^i and L_k^i are the lengths of corresponding edge vectors on the boundaries of $\mathcal{P}_j$ and $\mathcal{P}_k$. While the lengths of the boundary segments have to be the same, they can exhibit different curvatures. Nevertheless, boundaries should still remain smooth, which we encourage with a corresponding penalty term,

$$E_{\text{smooth}}(\mathbf{X}) = \sum_{i=1}^{m} \sum_{j=1}^{b_i} ||\mathbf{Q}_l^i - 2\mathbf{Q}_j^i + \mathbf{Q}_r^i||_2^2 \,, \tag{12}$$

where b_i is the number of non-corner boundary vertices $\mathbf{Q}^i$ of panel $\mathcal{P}_i$, whereas $\mathbf{Q}_l^i$ and $\mathbf{Q}_r^i$ denote the left and right neighbors of boundary vertex $\mathbf{Q}_j^i$.

Regularization The distance energy (9) allows the vertices $\mathbf{x}$ to slide on the target surface, but this freedom comes at the price of a nullspace: for any displacement of a given internal panel vertex $\mathbf{X}_i$, there is a corresponding world-space displacement $\mathbf{x}_i$ such that neither objectives nor constraints change. In order to obtain a well-posed problem, we use a Laplacian regularizer that asks for a smooth distribution of internal panel vertices as

$$E_{\text{laplace}}(\mathbf{X}) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} \mathcal{L}(\mathbf{X}_j) \,, \tag{13}$$

where $\mathcal{L}(\mathbf{X}_j)$ is the Tutte Laplacian [Tutte 1963]. As a desirable side effect, this regularizer also promotes well-shaped elements.

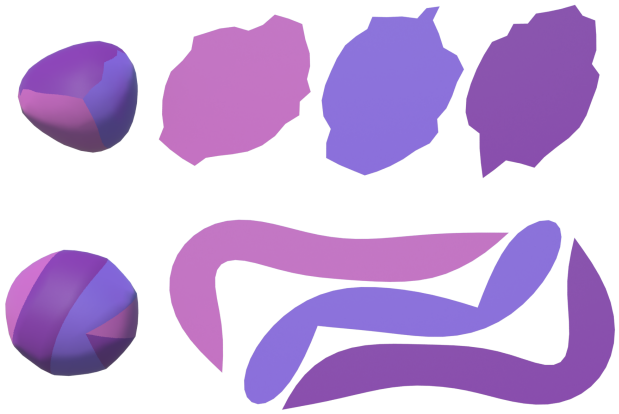


Figure 6: *Pattern optimization for a spherical balloon. The balloon (left) and its patterns (right) are shown for the initial guess (top), and after optimization (bottom).*

5.2 Initial Flattening

The final shape of the patterns is obtained by solving the constrained minimization problem (8). However, a good initial guess is crucial for rapid convergence. In principle, any mesh parametrization method can be used to create an initial guess. A particular aspect of our setting is, however, that the shapes of the panels are entirely defined by their boundary vertices—the shape of interior elements is, to a large extent, an afterthought. Among the many existing methods, we therefore took inspiration in one that preserves the lengths of the segment boundaries [Wang 2008].

We start by converting the user-provided seams, represented by smooth spline curves, into sets of edge vectors, defining a partitioning of the target mesh $\mathcal{T}$. For each partition, we first flatten its boundary $\mathbf{q}^i$ by minimizing an objective function that penalizes squared differences in edge lengths and internal angles as

$$E_{\text{fl}}(\mathbf{Q}^i) = \sum_{j=1}^{b_i} \tfrac{1}{l_{ij}}(L_{ij}(\mathbf{Q}^i) - l_{ij})^2 + l_j(\Theta_j(\mathbf{Q}^i) - \theta_j)^2 \ , \quad (14)$$

where L_{ij} are the lengths of the boundary edges of panel $\mathcal{P}_i$ and l_{ij} are the corresponding lengths in $\mathcal{T}$. Moreover, Θ_j is the sum of internal angles around $\mathbf{Q}^i_j$, θ_j is the corresponding quantity on $\mathcal{T}$, and l_j is the average length of the two edges incident to $\mathbf{q}^i_j$. The resulting nonlinear problem is solved with a few iterations of Newton's method. Keeping the boundary vertices fixed, the positions of the internal vertices are then computed by minimizing a Laplacian energy analogous to (13), which amounts to a single linear solve.

We note that, although a good initial guess helps speed convergence, our optimization scheme is not very sensitive to this choice. The example shown in Fig. 6 puts this robustness to a test, using *D-Charts* and *ABF++* (see [Julius et al. 2005]) as initial guess for the three panels of a spherical balloon. Here, the lengths of the panel boundaries are very different from the corresponding lengths on the target shape. Nevertheless, our method is able to find panel shapes that allow for a close approximation of the target. It is also worth noting that, since the seams were not restricted to stay in place, the final patterns are very different from the initial guess, revealing a surprisingly symmetric and elegant solution.

5.3 Remeshing

Fig. 6 exemplifies the potential difference between initial and final patterns. In order to robustly handle such extreme changes in size and shape, we integrate the optimization with a remeshing method that maintains well-shaped elements at all times. A number of works in graphics have explored the integration of remeshing and simulation, either globally [O'Brien and Hodgins 1999; Bargteil et al. 2007; Wojtan and Turk 2008] or in a locally adaptive manner [Wicke et al. 2010; Narain et al. 2012]. Since our mesh sizes are comparatively small, we opt for a global remeshing scheme that builds on *Triangle* [Shewchuk 1996]. Remeshing is invoked whenever the aspect ratio of an element falls below a given threshold. We first resample the boundaries of the patches to satisfy a minimum- and maximum-length criterion on the edges, maintaining correspondence between adjacent patches. We then invoke *Triangle* to remesh the interior of each patch and carbon-copy all changes to the inflated mesh as well as the target mesh.

6 Results

We used our design system to create a diverse set of inflatable structures, seven of which we present and discuss in this section. For validation, we also created physical prototypes for three of these examples. The design interface and the results are demonstrated in detail in the accompanying video and in Fig. 8 and Fig. 9. The fabricated models are made out of PVC plastic sheets. The optimized patches generated by our system were cut with a computer-controlled cutting machine and then manually hot-sealed. In the following, we validate some of our design decisions and discuss our results in more detail.

Simulation We compared our relaxed energy density based on tension field theory to a full simulation using the strain energy density of an incompressible Neo-Hookean material as described by 3. The full simulation for the Teddy model shown in Fig. 7 with ≈59k elements took more than two hours, whereas our relaxed energy density is computationally more efficient because it requires fewer elements and avoids problems due to indefiniteness. We tested our simulation approach with several different resolutions, ranging from ≈3k elements to ≈59k elements, and report the approximation error in Fig. 7. The computation for the three meshes of ≈3k, ≈15k and ≈59k elements took 4.5, 77 and 157 seconds, respectively. In practice, we observed that already with a coarse mesh we are able to obtain satisfactory accuracy. Although the relaxed energy formulation inhibits wrinkles in the simulated geometry, the regions in which wrinkling occurs can be inferred from the

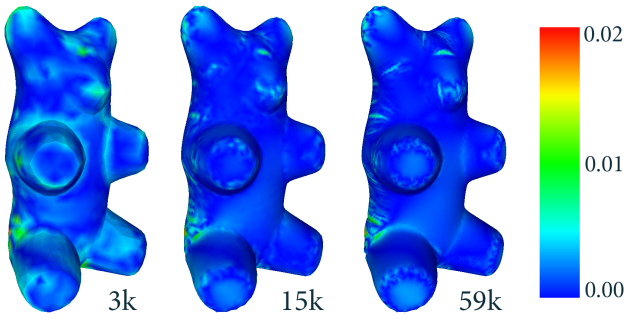


Figure 7: *Comparison of simulations with relaxed energy model on meshes with resolutions of 3k, 15k, and 59k elements to a reference simulation using the original energy on the 59k mesh. Colors indicate per-vertex difference for a unit size model.*

compression field. By displaying this information as a color-coded overlay, we can provide the user with feedback on wrinkle location and amplitude during seam design.

Performance of Optimization To evaluate the performance of our optimization, we start from a given seam layout and measure the required time for convergence when no later edit is performed. This comprises the initial flattening, initialization of all data structures, remeshing operations and subsequent data structure updates, as well as optimization. The optimization is the most expensive step of the process but the evaluation of the different involved quantities largely dominates the linear solves. All computations were done on a standard desktop computer with 3.20 GHz and 12 cores. Our prototype implementation is written in C++ and, except for the linear solver, not parallelized, leaving ample room for performance improvements. As a stopping criterion, we require the infinity-norm of the gradient of the Lagrangian corresponding to (8) to be smaller than $\text{tol} = 10^{-3}$, whereas a tighter tolerance of $\text{tol} = 10^{-5}$ is used for the forces. We use the same thresholds for all models presented in this paper. Table 1 lists detailed performance numbers for all examples, including the number of elements, the number of required remeshing steps, and the number of SQP iterations. In practice, we observed that already after a few iterations the result is close to the final solution. We therefore opted to visualize the incremental steps of the optimization to the user, providing fast visual feedback and intuition about the quality of the seam placement. As shown in Table 1, after less than a minute, the relative error measured between the starting point of the optimization and the final converged result is below 5%, providing sufficient accuracy for pre-visualization.

Seam Placement The approximation quality, i.e., how well a given inflatable structure matches its target shape, depends on the number of patches as well as the seam layout. In our experiments we observed that there is a tradeoff between shape approximation and aesthetic requirements. As illustrated in Fig. 6, seams can slide significantly during the optimization on the target surface. We allow the user to intuitively control the admissible amount of sliding by adjusting the weight of the corresponding penalty term (10). If seams are assigned a small weight, they can move such as to optimize the overall shape approximation. If seams are assigned a high weight, they stay close to their target location in 3D. As the location of the seams is generally very important for the aesthetics of the inflatable structure, we used relatively high weights for all models, except for the "Sphere". This approach proved particularly important and effective for modeling the "Fox" (Fig. 8), for which characteristic features such as the eyes and eyelids were delineated by corresponding seams.

Our interface allows non-expert users to efficiently add, edit, and replace seams and explore the impact of these operations on the inflated shape in an interactive manner. A demonstration of the design process can be found in the accompanying video. On average, designing a foil balloon took between 8 minutes for simple models ("Teddy") and less than half an hour for sophisticated models with internal connections ("Fox").

Internal Connections Several of our models ("Fox", "Elephant", "Flower", "Twisty") rely on internal connections, which are created by connecting existing seams on the surface as indicated by the user. As shown in Fig. 2, internal connections can be used, for example, to create sharp concave creases—a salient feature for many models. Fig. 8 and Fig. 9 visualize the internal patches generated by our method and demonstrate that the resulting inflated shapes are in very good agreement with the desired behavior.

7 Limitations and Future Work

We have presented a design system for creating inflatable structures made from flat panels. The enabling technology of our system is an automatic physics-based pattern generation method, combining fast simulation based on tension field theory and robust constraint optimization. Combined with an intuitive user interface, even non-expert users are able to design and explore intricate structures by simply drawing and editing seams on an input model. As demonstrated by our results, our system also supports internal connections, thus significantly broadening the range of shapes that can be designed. Nevertheless, our system has some limitations and many exciting opportunities for future work remain. In particular, input meshes that flatten to exceedingly thin panels pose challenges for subsequent remeshing and optimization. It would be helpful to automatically update the segments connectivity during the optimization and merge thin panels to adjacent panels automatically. Also, we make no attempt at inferring the location of internal connections in an automatic way nor automatically consider geometric properties of our input model such as ridges, symmetries or curvature. Future work could include higher-level tools that exploit this information for supporting the seam placement or for auto-completion of partially drawn seams. Although we have not observed collisions or self-intersecting panels during optimization, we currently do not explicitly prevent these cases. Finally, even with internal connections, there is a limit on what kind of shapes can be obtained with an inflatable structure. For example, planar regions and sharp convex edges (as shown in the "Twisty" example, Fig. 9) are inherently difficult to reproduce. For future work, it would be interesting to indicate at the beginning of the design process infeasible regions and limits on achievable approximations.

Acknowledgements

We would like to thank the reviewers for their insightful comments and Robert Kohn for introducing us to tension field theory. We also greatly appreciated the help of Urs Meier Aegler, Christian Schumacher and Maurizio Nitti. This work was partly funded by the NCCR Co-Me of the Swiss NSF.

References

BÄCHER, M., BICKEL, B., JAMES, D. L., AND PFISTER, H. 2012. Fabricating articulated characters from skinned meshes. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4.

BAGINSKI, F., BARG, M., AND COLLIER, W. 2008. Existence theorems for tendon-reinforced thin wrinkled membranes subjected to a hydrostatic pressure load. *Math. Mech. Solids 13*, 6.

BARGTEIL, A. W., WOJTAN, C., HODGINS, J. K., AND TURK, G. 2007. A finite element method for animating large viscoplastic flow. *ACM Trans. Graph. (Proc. SIGGRAPH) 26*, 3.

BICKEL, B., KAUFMANN, P., SKOURAS, M., THOMASZEWSKI, B., BRADLEY, D., BEELER, T., JACKSON, P., MARSCHNER, S., MATUSIK, W., AND GROSS, M. 2012. Physical face cloning. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4.

BYRD, R. H., CURTIS, F. E., AND NOCEDAL, J. 2010. An inexact newton method for nonconvex equality constrained optimization. *Mathematical Programming 122*, 2.

CALÌ, J., CALIAN, D. A., AMATI, C., KLEINBERGER, R., STEED, A., KAUTZ, J., AND WEYRICH, T. 2012. 3d-printing of non-assembly, articulated models. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 31*, 6.

Model	#Elements	#Panels total	#Panels internal	#Remeshing steps	#Iterations / Time [s] preview		#Iterations / Time [s] full convergence	
Sphere	692	3	0	5	8	4	128	21
Teddy	2974	17	0	2	23	17	193	258
Fox	7544	22	3	4	11	35	398	420
Twisty	6084	34	3	2	15	33	247	430
Flower (bottom)	6069	12	1	3	51	60	298	546
Flower (top)	7888	23	0	3	47	69	197	506
Tentacle	4458	21	0	2	9	23	175	131
Elephant	14436	40	2	3	30	137	401	1768

Table 1: *Statistics and processing time for our results. Timings are given for preview quality (relative error between displayed mesh and final result $< 5\%$ of the size of the model) and full convergence (gradient of objective $< 1.e^{-3}$ and forces $< 1.e^{-5}$)*

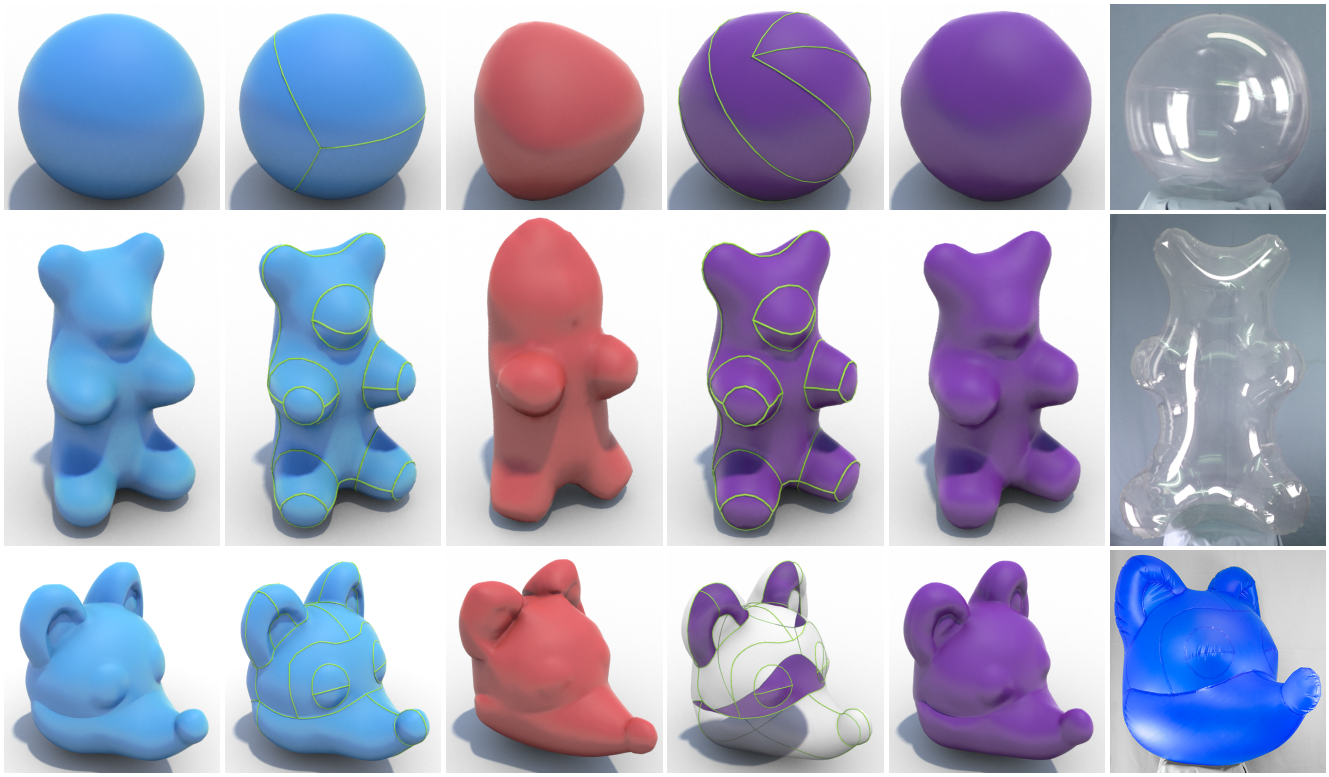


Figure 8: *Overview of our results (from left to right): input model, input model with seam layout, nonoptimized inflated shape, optimized result with seams, optimized result, fabricated prototype. The rows show (from top to bottom) the Sphere, Teddy, and Fox examples.*

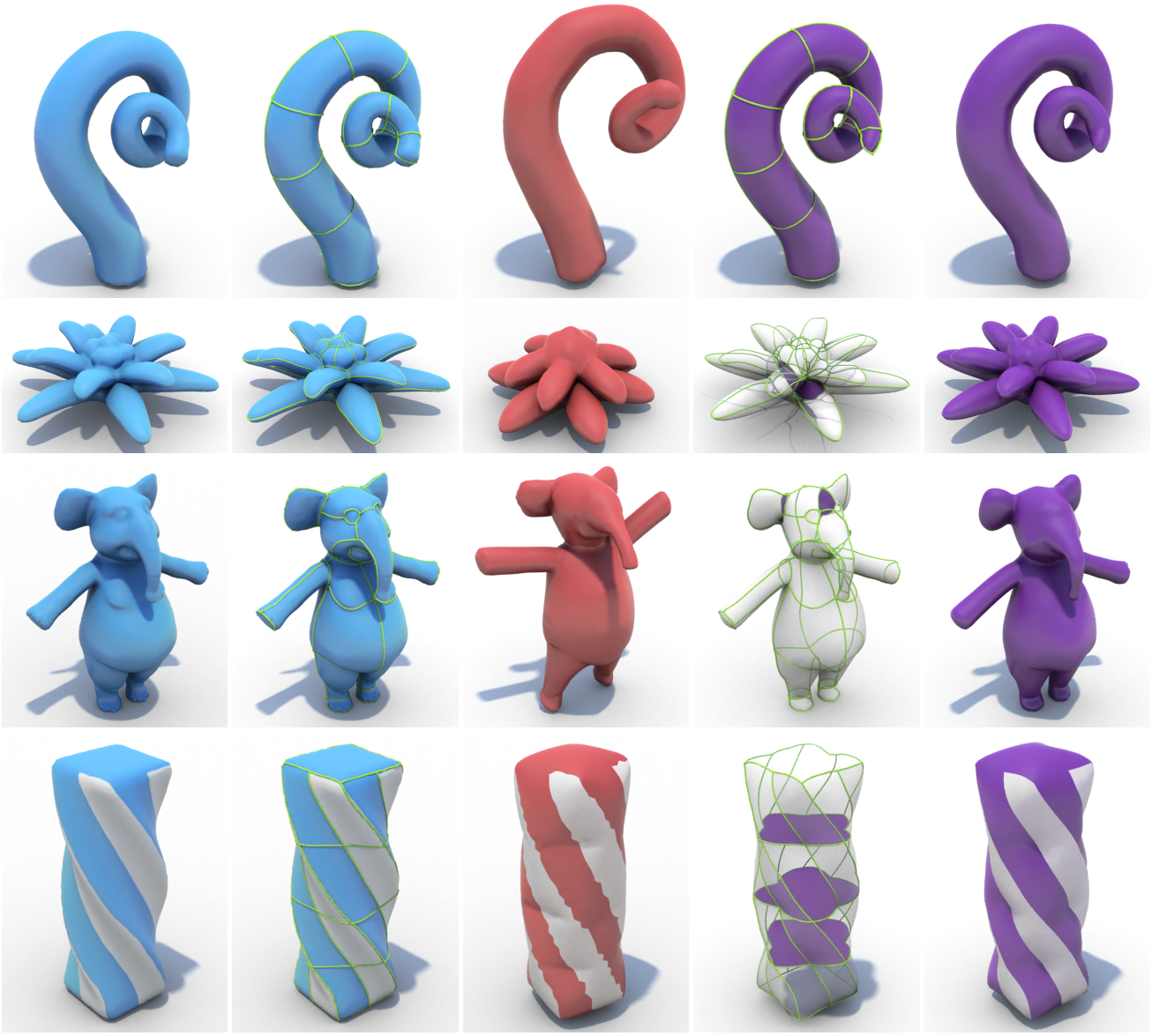


Figure 9: *Overview of results (from left to right): input model, input model with seam layout, nonoptimized inflated shape, optimized result with seams, optimized result without seams. The rows show (from top to bottom) the Tentacle, Flower, Elephant, and Twisty examples.*

CEYLAN, D., LI, W., MITRA, N. J., AGRAWALA, M., AND PAULY, M. 2013. Designing and fabricating mechanical automata from mocap sequences. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 31*, 6.

CHEN, D., LEVIN, D. I. W., DIDYK, P., SITTHI-AMORN, P., AND MATUSIK, W. 2013. Spec2Fab: A reducer-tuner model for translating specifications to 3D prints. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4.

CHOI, K.-J., AND KO, H.-S. 2002. Stable but responsive cloth. *ACM Trans. Graph. (Proc. SIGGRAPH) 21*, 3.

CHOI, K.-J., AND KO, H.-S. 2003. Extending the immediate buckling model to triangular meshes for simulating complex clothes. *Eurographics 2003 Short Presentations*, 187–191.

COROS, S., THOMASZEWSKI, B., NORIS, G., SUEDA, S., FORBERG, M., SUMNER, R. W., MATUSIK, W., AND BICKEL, B. 2013. Computational design of mechanical characters. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4.

FURUTA, Y., UMETANI, N., MITANI, J., IGARASHI, T., AND FUKUI, Y. 2010. A film balloon design system integrated with shell element simulation. *Eurographics - Short papers*, 33–36.

HORMANN, K., LÉVY, B., AND SHEFFER, A. 2007. Mesh parameterization: Theory and practice. In *ACM SIGGRAPH Courses*.

HULLIN, M. B., IHRKE, I., HEIDRICH, W., WEYRICH, T., DAMBERG, G., AND FUCHS, M. 2013. Computational fabrication and display of material appearance. In *Eurographics STARs*.

IGARASHI, Y., AND IGARASHI, T. 2008. Pillow: Interactive flattening of a 3d model for plush toy design. In *Proc. of the 9th International Symposium on Smart Graphics, SG 2008*, 1–7.

JULIUS, D., KRAEVOY, V., AND SHEFFER, A. 2005. D-charts: Quasi-developablemesh segmentation. *Comput. Graph. Forum 24*, 3, 581–590.

KILIAN, M., FLÖRY, S., CHEN, Z., MITRA, N. J., SHEFFER, A., AND POTTMANN, H. 2008. Curved folding. *ACM Trans. Graph. (Proc. SIGGRAPH) 27*, 3.

LAU, M., OHGAWARA, A., MITANI, J., AND IGARASHI, T. 2011. Converting 3d furniture models to fabricatable parts and connectors. *ACM Trans. Graph. 30*, 4.

MASSARWI, F., GOTSMAN, C., AND ELBER, G. 2007. Papercraft models using generalized cylinders. In *Proc. of Pacific Graphics '07*, IEEE.

MITANI, J., AND SUZUKI, H. 2004. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graph. (Proc. SIGGRAPH) 23*, 3.

MORI, Y., AND IGARASHI, T. 2007. Plushie: An interactive design system for plush toys. *ACM Trans. Graph. (Proc. SIGGRAPH) 26*, 3.

NARAIN, R., SAMII, A., AND O'BRIEN, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 31*, 6.

O'BRIEN, J. F., AND HODGINS, J. K. 1999. Graphical modeling and animation of brittle fracture. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, 137–146.

ÖZTIRELI, C., GUENNEBAUD, G., AND GROSS, M. 2009. Feature preserving point set surfaces based on non-linear kernel regression. *Comput. Graphics Forum (Proc. Eurographics) 28*, 2.

PIPKIN, A. C. 1986. The relaxed energy density for isotropic elastic membranes. *IMA Journal of Applied Mathematics 36*.

ROHMER, D., POPA, T., CANI, M.-P., HAHMANN, S., AND SHEFFER, A. 2010. Animation wrinkling: Augmenting coarse cloth simulations with realistic-looking wrinkles. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 29*, 6.

SHATZ, I., TAL, A., AND LEIFMAN, G. 2006. Paper craft models from meshes. *The Visual Computer 22*, 9.

SHEFFER, A., LÉVY, B., MOGILNITSKY, M., AND BOGOMYAKOV, A. 2005. Abf++: Fast and robust angle based flattening. *ACM Trans. Graph. 24*, 2 (Apr.).

SHEWCHUK, J. R. 1996. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Applied Computational Geometry: Towards Geometric Engineering*, vol. 1148.

SKOURAS, M., THOMASZEWSKI, B., BICKEL, B., AND GROSS, M. 2012. Computational design of rubber balloons. *Comput. Graphics Forum (Proc. Eurographics) 31*, 2.

SKOURAS, M., THOMASZEWSKI, B., COROS, S., BICKEL, B., AND GROSS, M. 2013. Computational design of actuated deformable characters. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4.

SOLOMON, J., VOUGA, E., WARDETZKY, M., AND GRINSPUN, E. 2012. Flexible developable surfaces. *Comput. Graphics Forum 31*, 5.

STEIGMANN, D. J. 1990. Tension-field theory. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 429*, 1876.

TERAN, J., SIFAKIS, E., IRVING, G., AND FEDKIW, R. 2005. Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05.

TUTTE, W. T. 1963. How to draw a graph. *Proc. London Math. Soc. 3*, 13, 743767.

UMETANI, N., KAUFMAN, D. M., IGARASHI, T., AND GRINSPUN, E. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph. (Proc. SIGGRAPH) 30,* 4.

UMETANI, N., IGARASHI, T., AND MITRA, N. J. 2012. Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4.

WANG, C. C., AND TANG, K. 2010. Pattern computation for compression garment by a physical/geometric approach. *Computer-Aided Design 42*, 2.

WANG, C. 2008. Computing length-preserved free boundary for quasi-developable mesh segmentation. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (Jan.).

WICKE, M., RITCHIE, D., KLINGNER, B. M., BURKE, S., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2010. Dynamic local remeshing for elastoplastic simulation. *ACM Trans. Graph. (Proc. SIGGRAPH) 29*, 4.

WOJTAN, C., AND TURK, G. 2008. Fast viscoelastic behavior with thin features. *ACM Trans. Graph. (Proc. SIGGRAPH) 27*, 3.

ZHU, L., XU, W., SNYDER, J., LIU, Y., WANG, G., AND GUO, B. 2012. Motion-guided mechanical toy modeling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 31*, 6.