Computational Design and Automated Fabrication of Kirchhoff-Plateau Surfaces

JESÚS PÉREZ, URJC Madrid
MIGUEL A. OTADUY, URJC Madrid
BERNHARD THOMASZEWSKI, Disney Research

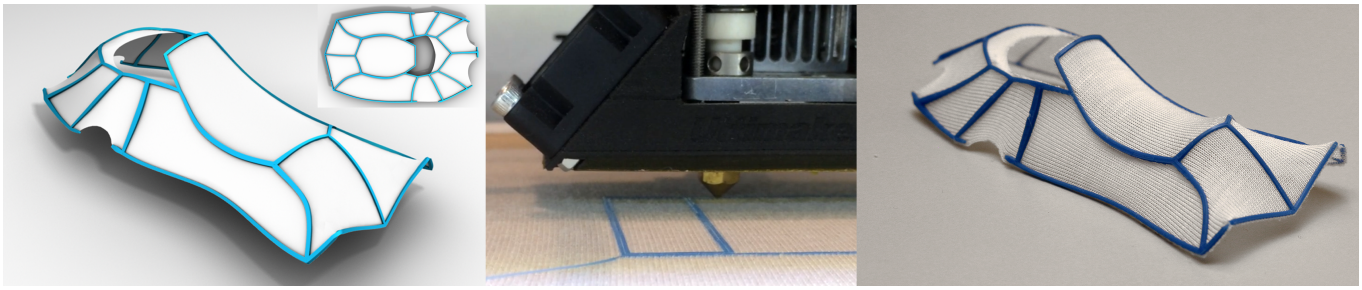

Fig. 1. Our method enables users to quickly create complex Kirchhoff-Plateau Surfaces such as this automotive concept design. *Left*: the 3D shapes designed in this way are equilibrium configurations of pre-stretched fabric membranes structured with planar rod networks (inset figure). *Middle*: we fabricate these designs using a filament-based 3D printer that embeds rods directly into the fabric. *Right*: once cut out, the physical prototype deploys into the desired shape.

We propose a computational tool for designing Kirchhoff-Plateau Surfaces—planar rod networks embedded in pre-stretched fabric that deploy into complex, three-dimensional shapes. While Kirchhoff-Plateau Surfaces offer an intriguing and expressive design space, navigating this space is made difficult by the highly nonlinear nature of the underlying mechanical problem. In order to tackle this challenge, we propose a user-guided but computer-assisted approach that combines an efficient forward simulation model with a dedicated optimization algorithm in order to implement a powerful set of design tools. We demonstrate our method by designing a diverse set of complex-shaped Kirchhoff-Plateau Surfaces, each validated through physically-fabricated prototypes.

CCS Concepts: • **Computer graphics → Computational geometry and object modeling**; *Physically based modeling*;

Additional Key Words and Phrases: Plateau Problem, Fabrication

ACM Reference format:
Jesús Pérez, Miguel A. Otaduy, and Bernhard Thomaszewski. 2017. Computational Design and Automated Fabrication of
Kirchhoff-Plateau Surfaces. *ACM Trans. Graph.* 36, 4, Article 62 (July 2017), 12 pages.
https://doi.org/10.1145/3072959.3073695

1 INTRODUCTION

Minimal surfaces have intrigued scientists and engineers for more than 250 years. Their origins trace back to a problem originally raised by Lagrange that later became known as the *Plateau problem*: finding a surface of minimum area that spans a given rigid boundary curve. From an application point of view, minimal surfaces are appreciated not only for their smooth aesthetic appearance, but also for their inherent material efficiency and structural stability. For these reasons, minimal surfaces are widely used for light-weight and cost-efficient structures, ranging from large-scale roofs, canopies, and shade systems, to acoustic deflectors, light diffusers, and decorative elements for interior design.

One common way of bringing designs of minimal surfaces to practice is by using, e.g., fiberglass or metal rods embedded in stretched fabric, with the added benefit that all components are planar and easy to manufacture. However, the advantages in terms of weight, cost, and fabrication come at the price of a much more difficult design problem—the Generalized Plateau Problem of finding a minimal surface whose tensile forces are in equilibrium with the bend and twist forces of a given elastic rod [Giomi and Mahadevan 2012; Giusteri et al. 2016].

In this work, we explore the design and fabrication of *Kirchhoff-Plateau Surfaces* (KPS), i.e., networks of thin elastic rods embedded in pre-stretched membranes. We focus on surfaces that can be manufactured by 3D-printing planar rods onto stretched fabric—a process that was beautifully demonstrated in recent work by [Guberan and Clopath 2016]. Designing in this space gives rise to several challenges. First, the path to stable surfaces is fraught with perils such as nonlinearities, unstable equilibrium points, and bifurcations. Second, the space of possible designs is restricted by the fact that (i) KPS consist of minimal surface patches that, inherently, can only assume shapes with vanishing mean curvature, and that (ii) the boundaries of the minimal surface patches can only assume shapes corresponding to equilibrium states of planar rods.

In light of these challenges, we cannot expect that there exists a KPS that closely approximates a given target shape in the general

0730-0301/2017/7-ART62 $15.00
https://doi.org/10.1145/3072959.3073695

Fig. 2. The design space of KPS is highly nonlinear and the effects of parameter changes often difficult to predict. The images show a simple square frame (120×120 mm) with different fabric pre-stretch factors s, normal rod width w_n (governing out-of-plane resistance), and binormal rod width w_b (governing in-plane resistance). From left to right: (i) ($s = 1.3$, $w_n = 0.2$ mm, $w_b = 0.4$ mm) base shape; (ii) ($s = 1.6$, $w_n = 0.2$ mm, $w_b = 0.4$ mm) higher stretch increases curvature; (iii) ($s = 1.3$, $w_n = 0.2$ mm, $w_b = 0.2$ mm) lower in-plane resistance leads to sagging and wrinkling; (iv) ($s = 1.3$, $w_n = 0.4$ mm, $w_b = 0.4$ mm) higher out-of-plane resistance leads to a completely different stable configuration.

case. But even when precise shape approximation is not possible, the space of KPS nevertheless provides ample room for shape abstraction and interpretation—qualities found in humans, rather than machines. We therefore turn away from fully-automated solutions in favor of a user-guided but computer-assisted design paradigm.

Overview & Contributions. We propose a fabrication-oriented design tool for KPS—planar rod networks embedded in pre-stretched fabric that deploy into complex, three-dimensional shapes. Similar in spirit to mesh-based modeling tools, the user is responsible for creating the topology of the rod network and for transforming it into the desired shape. During this process, the user can draw from a set of modeling tools that implement simple editing operations directly on the equilibrium state of the surface. Visualization tools that indicate possible changes conforming to the editing goals further assist the user in making informed decisions.

This seemingly simple design approach is enabled by simulation and optimization algorithms that translate editing operations into corresponding parameter updates and analyze the current structure in order to provide feedback on the space of feasible edits. Our simulation and optimization algorithms involve several novel features. We introduce tools for the exploration of the design space through eigen analysis of the sensitivity matrix subject to design goals. Owing to the complex and constrained space of KPS, we furthermore cast inverse design operations as a two-step optimization process: . first, in order to interactively explore first-order feasible target shapes, we combine efficient constraint projection with sensitivity-based linearization of equilibrium constraints; second, given a first-order feasible target shape, we perform nonlinear constrained optimization to compute fully-feasible designs. We demonstrate our method by designing a diverse set of complex-shaped KPS, each validated by physically-fabricated prototypes.

2 RELATED WORK

Surface Design. Our work targets the design of a class of physical surfaces whose shapes are governed by a particular set of equilibrium constraints. Surface design has been in the focus of computer graphics ever since its beginnings. Apart from digital surfaces made for virtual worlds, one important physical application domain is architectural geometry [Pottmann et al. 2015]. Surfaces from this category are often subject to constraints relating, e.g., to the planarity of polygonal faces [Liu et al. 2006; Vaxman 2014] or to compression-only self-supporting structures [Block and Ochsendorf 2007; Vouga et al. 2012]. In addition to enforcing such constraints numerically, exploring the resulting design spaces is a challenging problem as well [Deng et al. 2015; Yang et al. 2011].

As one particular class, developable surfaces have attracted a lot of attention from computer graphics [Decaudin et al. 2006; Kilian et al. 2008; Solomon et al. 2012; Tang et al. 2016], as they arise naturally when creating 3D surfaces from flat, inextensible material such as paper, plastic, or stiff fabric. Based on the principle of auxetic materials, Konakovic et al. [2016] were able to create doubly-curved surfaces by structuring planar sheets of quasi-inextensible material. While developable surfaces are characterized by zero Gaussian curvature, requiring vanishing mean curvature leads to minimal surfaces, which have been intensively studied in mathematics [Dziuk and Hutchinson 1999; Meeks and Pérez 2011] and also computer graphics [Crane et al. 2011; Pan et al. 2012; Tang et al. 2014]. In the context of architecture and engineering, minimal surfaces appear naturally when designing tensile membrane structures; see, e.g., [Koohestani 2014]. One particular line of recent work [Lienhard et al. 2013], [Van Mele et al. 2013] has studied the forward design of membrane structures coupled with bending-active elements. However, the inverse problem of automatically determining parameters such that the resulting equilibrium shape approximates given design goals has, to the best of our knowledge, not been investigated so far.

Beyond minimal surfaces and the related Plateau problem, there has been an increasing interest in the generalized Plateau problem which, instead of assuming a rigid boundary, considers the case of Euler elastica [Giomi and Mahadevan 2012] and Kirchhoff rods [Giusteri et al. 2017]. While these works focus primarily on questions of existence and uniqueness, we investigate the problem of modeling such surfaces for the purpose of fabrication.

Inverse Design of Physical Surfaces. The shape of elastic structures is governed by equilibrium conditions, requiring balance between internal and external forces such as self-weight and applied loads. The problem of designing structures that assume desired equilibrium shapes under gravity has been studied extensively in computer graphics, e.g., in the context of hair [Derouet-Jourdan et al. 2010] and cloth [Twigg and Kačić-Alesić 2011] animation, as well as for real-world settings such as self-supporting surfaces [de Goes et al. 2013; Liu et al. 2013; Panozzo et al. 2013; Vouga et al. 2012; Whiting et al. 2009], custom-shaped elastic solids [Chen et al. 2014], or more generalized shape optimization problems [Musialski et al. 2016]. Our work shares some aspects of these inverse design problems but differs in the sense that the driving force is membrane stretch which, unlike gravity, depends on the state of the system.

Several works have considered the problem of designing physical surfaces from flat panels with applications to plush [Mori and Igarashi 2007], clothing [Bartle et al. 2016; Umetani et al. 2011], inflatable structures [Skouras et al. 2014], thermo-formed models [Schüller et al. 2016], and surfaces made from interlocking elements [Skouras et al. 2015]. Rather than automatically generating a seam network for the input surface, Skouras et al. [2014] leave the topology problem to the user and instead optimize performance such as to enable fast design iterations. Our approach follows the same spirit, but instead of optimizing for a fixed target shape, our method allows the user to explore the space of feasible designs using a set of editing tools that leverage simulation and optimization. This same paradigm of computer-assisted but user-guided design is implemented by several previous approaches, including the work by Umetani et al. [2011], who use first-order sensitivity analysis in order to quickly predict the impact of parameter changes on the equilibrium shape of clothing. As one difference, our method extends this forward design approach by inverse modeling tools that, for user-specified editing objectives, automatically compute first-order optimal directions in parameter-space. Allowing the user to directly edit the 3D equilibrium state was also the driving motivation for the work of Bartle et al. [2016]. However, while their method uses a heuristics-based, gradient-free approach specifically tailored to the problem of pattern optimization for garment modeling, we capitalize on derivative information in order to implement fast forward and inverse design tools.

Another line of recent work has explored the use of elastic rods for physical surface design [Miguel et al. 2016; Pérez et al. 2015; Zehnder et al. 2016]. The method by Pérez et al. [2015] automatically computes optimal parameters for networks of elastic rods such as to approximate a set of target shapes under given external loads. Our method leverages the same set of parameters, i.e., the centerline geometry and cross-section shape of the rods, but instead of designing 3D networks that balance applied loads, our method computes a 2D layout optimized with respect to membrane forces. The approach by Miguel et al. [2016] targets physical surface representations using bent wires that securely interlock by virtue of deformation. Our approach shares the two-dimensional nature of their design space, but the underlying mechanics are very different. While all of the above methods, including ours, renounce automatic topology generation, Zimmer et al. [2014] address exactly this discrete problem, albeit in a context that does not involve equilibrium constraints. Finally, wire meshes can be considered a special case of rod networks, but rather than physics-based optimization, their particular structures motivate a geometric approach as shown by Garg et al [2014].

Filament-Based 3D Printing. Inspired by the work of Guberan and Clopath [2016], we use filament-based 3D printers in order to structure stretched fabric with embedded elastic rods. Although we are primarily interested in understanding and navigating the corresponding design space, this particular manufacturing process is also a very promising way of fast and inexpensive digital fabrication. With a similar motivation, several previous works have investigated the combination of filament-based printing with other media [Mueller et al. 2014] or by printing onto existing objects [Chen et al. 2015]. The fabrication process that we pursue in this work can also benefit from intelligent software for print path generation [Zhao et al. 2016] and planning [Hergel and Lefebvre 2014] in order to alleviate hardware limitations of current filament-based printers.

3 KIRCHHOFF-PLATEAU SURFACES

Kirchhoff-Plateau Surfaces are flexible structures made from networks of elastic rods embedded in pre-stretched textile membranes. Although fabricated in a planar state, KPS can deploy into complex three-dimensional shapes that are governed by the balance between membrane and rod forces. These equilibrium shapes are influenced by a number of factors pertaining to the rod network and the membrane. In addition to the material of the rods, their resistance to bending and twisting is determined by their cross-sectional geometry. By varying this geometry, it is possible to control the ratio between in-plane and out-of-plane bending stiffness [Pérez et al. 2015]. Membrane stretch, on the other hand, induces compressions in the rods, leading to unstable, planar equilibrium configurations that resolve into bending and twisting upon slight perturbations. As shown in Figure 2, the ratio between membrane stretch and out-of-plane bending resistance is, effectively, a means of controlling the amount of curvature in the equilibrium shape. However, a sufficient amount of membrane stretch is also required to ensure that the surface remains tense and free of sagging or wrinkling; see Figure 2-iii.

From a designer's perspective, the mechanics of KPS alone provide no direct insight into the space of shapes that can be achieved. However, even though fabric membranes are not strictly area minimizing, the intuition about KPS can be strengthened by considering them as *piece-wise minimal surfaces*: the rod network induces a decomposition of the surface into membrane patches, each bounded by a closed loop of rods. As a minimal surface, the area gradient vanishes everywhere inside each patch, which is equivalent to vanishing mean curvature. Since planar KPS configurations are generally unstable, principal curvatures have the same nonzero magnitude but opposite sign, leading to strictly negative Gaussian curvature everywhere inside a patch. Although the constraint on strict equality of principal curvatures is somewhat mitigated by real-world textile membranes, the sign constraint still applies: membrane patches can only assume so called *anticlastic* shapes with negative Gaussian curvature in every point. Moreover, the fabrication constraint that KPS must have a planar rest state implies that the 3D rod network must be embeddable in 2D without compressions, further restricting the space of possible surfaces. Nevertheless, while a single KPS patch is necessarily anticlastic, rods introduce discontinuities in the surface normals of adjacent membrane patches. Thanks to this property, it is possible to approximate surfaces with overall positive Gaussian curvature by connecting anticlastic patches. Figure 3 shows examples of diverse shapes that can be obtained by varying the internal mesh topology of a disk.

In summary, the space of KPS offers interesting and complex 3D shapes that can be created using a simple and cost-efficient 2D manufacturing process. However, the restrictions on shape and the complex mapping between parameters and shape make navigating

Fig. 3. Disks of the same size and pre-stretch produce diverse shapes depending on their internal topology. From left to right: (i) empty interior; (ii) split rod on the boundary; (iii) two crossing rods in the interior; (iv) an inner disk connected to the outer boundary.

this space a challenging task without assistance. In seeking a computational tool to help with the design process, a central question is the balance between control and automation. At one extreme of the spectrum, a fully-automated solution where the user provides a target shape and the KPS is determined through simulation and optimization requires the least amount of user intervention. However, the restricted shape of KPS will often require compromising between pure approximation quality and aesthetic considerations, which are difficult to quantify and automate. At the other end of the spectrum, manual exploration of the parameter space affords a maximum degree of artistic freedom. However, the nonlinear and unintuitive relation between parameters and shape can make manual design a tedious and frustrating process. In seeking a middle ground between those extremes, we opt for a primarily user-guided, but computer-assisted approach to shape exploration. In particular, the user is in charge of creating the structure and shape of the KPS, but can draw from a number of editing and visualization tools that simplify the design task. While this forward design approach is seemingly simple, it relies heavily on simulation and optimization to implement inverse modeling and visualization tools. The computational basis for our approach is described next.

4 COMPUTATIONAL MODEL

In order to enable computer-assisted design of KPS with desired shapes, we require a computational model for predicting (i) the equilibrium configuration for given parameter values, and (ii) the effect that parameter changes have on equilibrium shape. We start by introducing our parameterization for KPS.

4.1 Parameterization of the Design Space

KPS are characterized by three sets of variables: design parameters, rest-state configuration, and deformed configuration. The design parameters uniquely define the rest-state configuration, from which the deformed configuration results through force-equilibrium constraints. Due to our planar fabrication process, we parameterize both membrane and embedded rods in two-dimensional space. This fabrication process allows us to control the global stretch of the membrane, the topology and geometry of the rod mesh, as well as the width and thickness of the individual rods. Formally, we parameterize each of these four properties as follows:

- For ease of fabrication, we assume uniform isotropic *membrane stretch* $s \in \mathbb{R}$.
- We define *rod mesh topology* by a set of rods $\mathcal{R}$ and a set of junctions $\mathcal{J}$ between end-points of the rods.
- We define *rod geometry* by the positions $\mathbf{q}_i \in \mathbb{R}^2$ of a set of Q rod control points, which determine the positions of rod junctions as well as the shapes of individual rods.
- *Rod cross-sections* are characterized by width $w_{b,i}$ and thickness $w_{n,i}$ values at rod control points, corresponding to in-plane and out-of-plane directions, respectively.

In Section 5 we describe several tools for exploring the design space of KPS as defined by the above parameterization. For the formal description that follows, we conveniently group rod control points and cross-section widths in a vector of design parameters $\mathbf{p} = (\mathbf{q}, \mathbf{w}_n, \mathbf{w}_b)^T$, with $\mathbf{q} = (\mathbf{q}_1, \ldots, \mathbf{q}_Q)^T$, $\mathbf{w}_n = (w_{n,1}, \ldots, w_{n,Q})^T$, and $\mathbf{w}_b = (w_{b,1}, \ldots, w_{b,Q})^T$.

4.2 Discrete Rods

For convenience in editing and optimization, we represent rods as Catmull-Rom splines that interpolate the rod control points $\mathbf{q}_i$. For the purpose of simulation, however, we ultimately resort to the discrete elastic rod model by Bergou et al. [2010; 2008], whose explicit centerline representation and compact parameterization are attractive for our application—but other models [Bertails et al. 2006; Spillmann and Teschner 2007] could, in principle, be used as well. We evaluate each rod spline at a number of equidistant locations in spline parameter space, yielding a set of R rod vertices that define piece-wise linear curves for the discrete rod model. We denote as $\bar{\mathbf{r}}_i \in \mathbb{R}^2$ the rest positions of rod vertices, and group them in a vector $\bar{\mathbf{r}} = (\bar{\mathbf{r}}_1, \ldots, \bar{\mathbf{r}}_R)^T$. For a given set of rod control points, the rest positions of rod vertices are linear functions of the positions of rod control points, i.e., $\bar{\mathbf{r}} = \mathbf{H}\,\mathbf{q}$, with $\mathbf{H}$ a constant matrix of Hermite basis function evaluations. We use the same Hermite interpolation to define smooth cross-section widths at rod vertices.

The deformation of the rod mesh is characterized by three sets of degrees of freedom: the positions of rod vertices $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_R)^T$, $\mathbf{r}_i \in \mathbb{R}^3$; twist angles $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_E)^T$, $\omega_j \in \mathbb{R}$, between reference and material frames defined at E rod edges; and rotations of J junctions $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_J)^T$ represented using Euler angles $\boldsymbol{\theta}_k \in \mathbb{R}^3$ [Zehnder et al. 2016].

4.3 Discrete Membrane

We model the membrane based on nonlinear continuum mechanics discretized using linear triangle finite elements. More concretely, we use a St. Venant-Kirchhoff (StVK) material model to compute the internal forces of the deformed fabric. We found this simple nonlinear model sufficiently accurate for our purposes since, for the kind of material and stretch factors we aim for ($\approx 50\%$), the stress-strain

relation is barely nonlinear (see, e.g., Miguel et al. [2012]). Although the material that we used is almost isotropic, our implementation uses an orthotropic StVK model [Li and Barbič 2014] in order to accommodate other types of fabrics, if desired. Since the pre-stretch of the fabric is significant, internal forces are strongly dominated by membrane forces and bending forces are negligible. However, to avoid numerical problems when fabric sags, we add weak bending forces to our membrane model based on discrete shells [Grinspun et al. 2003].

While the membrane model itself is standard, we additionally have to account for coupling between the membrane and the rods embedded in it. Our approach is conceptually similar to the ones described in [Umetani et al. 2011] and [Skouras et al. 2014], but our specific problem and choice of discretization lead to several differences that we describe below. For the sake of computational efficiency and in order to achieve good shape approximation even for coarse meshes, we opt for an implicit coupling approach through collocation, i.e., by sharing degrees of freedom between rod and membrane vertices. To this end, we ask that the triangle mesh representing the membrane be conforming to the embedded rods such that, for each rod vertex, there is a collocated membrane vertex.

During design operations, the change of rod vertex positions effectively induces a change in the rest configuration of the membrane mesh. To ensure sufficient mesh quality for simulation, we define the rest positions for the membrane vertices implicitly as a function of the rest positions of the rod vertices. To this end, we introduce rest positions for the M membrane vertices that are not coincident with rod vertices as $\frac{1}{s}\,\bar{\mathbf{m}} = \frac{1}{s}\,(\bar{\mathbf{m}}_1, \ldots, \bar{\mathbf{m}}_M)^T$, where $\bar{\mathbf{m}}_i \in \mathbb{R}^2$ are the positions of membrane vertices after the pre-stretch s is applied.

Given an initial membrane mesh and a set of rod vertices that define its boundary, we compute smoothly distributed rest positions $\bar{\mathbf{m}}$ using harmonic interpolation. The interpolation weights are computed by requiring vanishing Laplacian coordinates [Sorkine 2005], which yields a linear system $\mathbf{L}_m\,\bar{\mathbf{m}} + \mathbf{L}_r\,\bar{\mathbf{r}} = 0 \Rightarrow \bar{\mathbf{m}} = -\mathbf{L}_m^{-1}\,\mathbf{L}_r\,\bar{\mathbf{r}}$, with $(\mathbf{L}_m, \mathbf{L}_r)$ the Laplacian matrix. In practice, we compute cotangent weights for the Laplacian once at initialization, leaving them constant until remeshing is necessary. These constant Laplacian weights also translate into constant harmonic interpolation weights. Whenever the quality of any membrane triangle falls below a given threshold, we remesh using CGAL's Delaunay triangulation algorithm.

4.4 Static Equilibrium and Sensitivity

The shape of a KPS is governed by force-equilibrium constraints,

$$\mathbf{f}(\mathbf{p}, \bar{\mathbf{x}}(\mathbf{p}), \mathbf{x}) = -\nabla_{\mathbf{x}} E = 0\,, \tag{1}$$

that require the gradient of internal energy E to vanish with respect to all degrees of freedom of the deformed configuration $\mathbf{x} = (\mathbf{r}, \boldsymbol{\omega}, \boldsymbol{\theta}, \mathbf{m})^T$. In the above equation, we use E as a shorthand for the total internal energy and refer to the original works for concrete expressions for membrane and rod energies. It is also worth noting that the internal forces depend on the design parameters $\mathbf{p}$ both explicitly, e.g., through the cross-sectional geometry of the rods, and implicitly through the rest state geometry $\bar{\mathbf{x}} = (\bar{\mathbf{r}}, \frac{1}{s}\,\bar{\mathbf{m}})^T$. Given values for all design parameters, we compute the corresponding equilibrium state of the KPS by solving (1) using a standard Newton method with line search and adaptive regularization for increased robustness.

Our computational design tools, described in Section 5, require the evaluation of shape changes as a function of design parameter changes. In order to establish this connection, we observe that any admissible change in parameters must leave the force constraints satisfied. Formally, we have

$$\frac{d\mathbf{f}}{d\mathbf{p}} = \frac{\partial \mathbf{f}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}}{\partial \bar{\mathbf{x}}}\frac{\partial \bar{\mathbf{x}}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{p}} = \mathbf{0}\,, \tag{2}$$

from which we obtain the so called *sensitivity* matrix

$$\mathbf{S} = \frac{\partial \mathbf{x}}{\partial \mathbf{p}} = -\frac{\partial \mathbf{f}}{\partial \mathbf{x}}^{-1}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}}{\partial \bar{\mathbf{x}}}\frac{\partial \bar{\mathbf{x}}}{\partial \mathbf{p}}\right)\,. \tag{3}$$

In order for the above expression to be valid, the force Jacobian $\nabla_{\mathbf{x}}\mathbf{f}$ has to be invertible. We eliminate null-spaces due to rigid transformations in both $\nabla_{\mathbf{x}}\mathbf{f}$ and $\nabla_{\bar{\mathbf{x}}}\mathbf{f}$ by constraining a small set of (user-selected) vertices in the deformed and rest configurations, respectively. If the system remains singular or indefinite, we add a diagonal regularizer and iteratively increase its weight until the linear solver succeeds.

It is worth mentioning that our choice of discretization yields a linear relationship between rest-state positions $\bar{\mathbf{x}}$ and design parameters $\mathbf{p}$ and, consequently, the Jacobian $\frac{\partial \bar{\mathbf{x}}}{\partial \mathbf{p}}$ is constant. More specifically, we have

$$\frac{\partial \bar{\mathbf{r}}}{\partial \mathbf{q}} = \mathbf{H} \quad \text{and} \quad \frac{\partial \bar{\mathbf{m}}}{\partial \mathbf{q}} = -\mathbf{L}_m^{-1}\,\mathbf{L}_r\,\mathbf{H}\,. \tag{4}$$

5 COMPUTATIONAL DESIGN

Building on the computational model described in the previous section, we introduce several design tools that allow users to explore the expressive space of Kirchhoff Plateau Surfaces, and to edit their shape and structure according to aesthetic considerations. In particular, we describe forward design tools for topology and base shape editing, a sensitivity analysis tool for design space exploration, and an inverse design tool for direct shape editing. These design tools make use of three major mathematical elements: fast evaluation of equilibrium KPS shapes resulting from parametric edits; sensitivity analysis for fast approximation of the mapping from parametric edits to their corresponding shape modifications; and a two-step optimization solver that computes parametric changes that realize desired shape edits.

5.1 Forward Design

During forward design, the user directly modifies the parameters listed in Section 4.1 and the resulting KPS is computed automatically using the equilibrium solver discussed in Section 4.4. This design approach proves most useful during the initial stage of the design, when the user defines the overall desired shape, or when topology or fabric pre-stretch need to be modified to counteract structural limitations of an intermediate design. While forward design poses virtually no restrictions on the shapes that can be achieved, it provides only limited assistance to the user. Nevertheless, we propose several guidelines based on geometric information as well as other heuristics that have proven successful in our design experience.

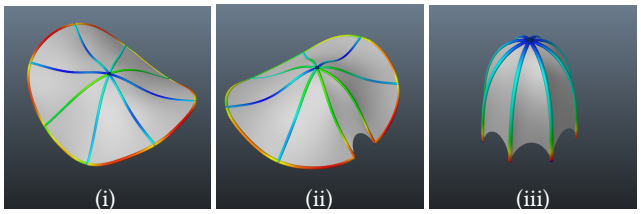

Fig. 4. **Rod Removal**. Rods are colored according to their strain with blue/red denoting highest compression/stretching. (i) Initially, radial rods are highly compressed and prevent the design from further deforming. (ii) Removal of a boundary rod relaxes compression in incident radial rods and increases the ability to bend. (iii) Removal of all boundary leads to non-hyperbolic shapes with high curvatures.

Alignment of rods with directions of principal curvature. Since the mean curvature of the membrane vanishes everywhere, rods are the only elements whose curvature can be directly controlled to approximate the desired shape. Consequently, we generally place rods in such a way that they follow lines of principal curvatures or align with specific features.

Anisotropy of patches aligned with principal curvatures. Isotropic patches with equal width in two orthogonal directions complicate the design as they favor membrane curvatures that are not aligned with rods (see Figure 5-ii). Anisotropic patches, on the other hand, favor curvature in the rod directions, and their shorter sides should be aligned with the direction of major curvature. Figure 5-iii shows a cylindrical patch subdivided along the curved direction.

Anisotropy of rod cross-sections. Anisotropic rods with smaller normal width w_n than binormal width w_b favor out-of-plane over in-plane deformations; compare Figure 2-i and Figure 2-iv. Consequently, they favor the alignment of principal curvatures with rods.

Rod removal to relax compression. The developability of a target design is largely determined by the rod mesh, and lack of developability can be detected by monitoring compression in the rods. When highly compressed rods are incident to the boundary of the design, removal of adjacent boundary rods relaxes compression and enables the exploration of more—or differently—curved shapes as

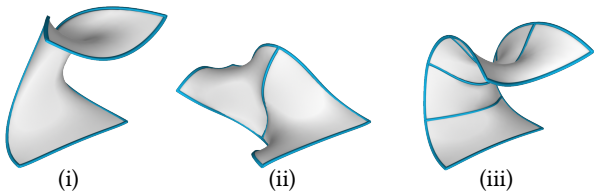

Fig. 5. **Patch Anisotropy**. The degree of patch anisotropy affects how well a rectangular piece of fabric produces a cylindrical shape. (i) With no subdivisions, the patch curves most along the short side and the cylindrical patch appears laterally compressed. (ii) With subdivision into square patches, curvatures do not align with rods and the shape twists. (iii) With multiple subdivisions along the curved direction, the desired shape is achieved, the short rods curve as desired, and the long rods resist compression.

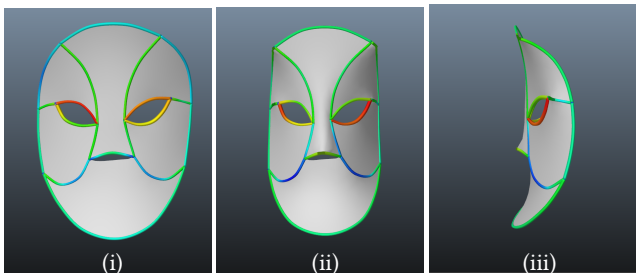

Fig. 6. **Rod Splitting**. Split rods enable high-curvature features. (i) Boundary rods at the top corners exhibit high compression and prevent further bending of the mask. (ii), (iii) Splitting rods at the top corners and the nose leads to overall increased curvature and enables sharp features.

shown in Figure 4. We implement the removal of boundary rods by transforming them into *ghost rods*, thus avoiding actual topology changes. Ghost rods are not simulated, but they are necessary for defining the control points that determine the rest positions of the membrane.

Rod splitting at high-curvature features. When a rod is already bent but does not reach the desired curvature, the curvature of the membrane can be increased by splitting the rod in two. The membrane forces will produce a kink at the location of the cut, which can also be used for aesthetic purposes as shown in Figure 6.

Stretch increase to avoid sagging. If the pre-stretch factor is not sufficiently high, the equilibrium state of the membrane can exhibit sagging and wrinkling as shown in Figure 2-iii. We therefore monitor and visualize the principal stretches throughout the fabric and trigger an increase in pre-stretch whenever the minimum value of any triangle element falls below a given threshold. In order to minimize deviations from the previous equilibrium shape, an increase in pre-stretch must be compensated by a corresponding increase in rod widths to withstand the larger membrane forces.

5.2 Sensitivity Exploration

Due to the complexity of the space of KPS, design goals often fall outside the feasible space. In order to enable shape exploration while remaining within the feasible space, we visualize the design space around a given equilibrium shape . The motivating insight for this approach is that a singular value decomposition (SVD) of the sensitivity matrix reveals valuable information about the local structure of the design space. In particular, the shape changes corresponding to the *dominant*, i.e., largest singular values convey in a concise way the major changes that a given equilibrium shape can undergo, thus offering inspiration for design changes; see Figure 7 for an example.

In practice, analyzing the sensitivity matrix $\mathbf{S}$ alone is not sufficient as position constraints will significantly impact the resulting modes. We therefore extend the analysis as follows.

Hard Constraints. During the design process, it is often convenient to prevent parts of the model from moving or deforming. We use hard constraints to implement design goals such as fixing parts of the surface to a support structure, or for attaching disconnected parts of the surface (see Figure 7). Let $\mathbf{c}(\mathbf{x}) = 0$ denote the set of

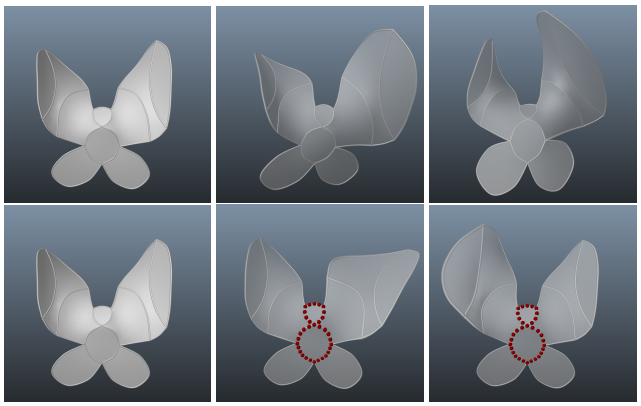

Fig. 7. **Sensitivity Exploration**. A butterfly model is deformed using the two dominant modes of the sensitivity matrix (middle and right columns). In the bottom row, hard constraints are imposed on the body of the butterfly (simulation vertices, not control points), leading to different dominant modes.

all such positions constraints. If the current configuration already fulfills the constraints, we require that

$$\mathbf{J}\Delta\mathbf{p} = \mathbf{0}\,, \quad \text{with} \quad \mathbf{J} = \frac{\partial \mathbf{c}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{p}}\,. \tag{5}$$

To enforce hard constraints to first order during editing operations, it suffices to post-multiply the sensitivity matrix $\mathbf{S}$ by a projection matrix $\mathbf{P}$,

$$\mathbf{P} = \mathbf{I} - \mathbf{J}^T\left(\mathbf{J}\mathbf{J}^T\right)^{-1}\mathbf{J}\,, \tag{6}$$

to map arbitrary parameter increments $\Delta\mathbf{p}$ to deformed state increments $\Delta\mathbf{x}$ in the null-space of the constraints as $\Delta\mathbf{x} = \mathbf{S}\,\mathbf{P}\,\Delta\mathbf{p}$. This constraint projection scheme assumes that $\mathbf{J}\mathbf{J}^T$ is invertible, which is the case if $\mathbf{S}$ is full rank and if there are no redundant constraints. It should be pointed out that this approach to enforcing hard constraints cannot remove drift and constraint stabilization is necessary (see Section 5.3).

Sensitivity Decomposition. We incorporate the effect of hard constraints by computing the SVD of $\mathbf{S}\,\mathbf{P}$. While we always perform the full decompostion, we generally only visualize the dominant modes corresponding to the largest singular values. For each of them, we take the corresponding right singular vector and multiply it by the sensitivity to obtain the corresponding change in shape which is then displayed to the user (see Figure 7).

Using the regular sensitivity $\mathbf{S}$ in the decomposition may be misleading since the parameter vector $\mathbf{p}$ combines control point positions and cross-sectional widths, i.e., parameters with very different scales. To avoid bias, we first compute the average column norm of the sensitivity matrix for both control point and width parameters. We then normalize the columns in order to exhibit identical average norms before applying the SVD.

5.3 Inverse Design

In addition to forward exploration of the design space, it is often convenient to directly specify desired changes to the equilibrium shape.

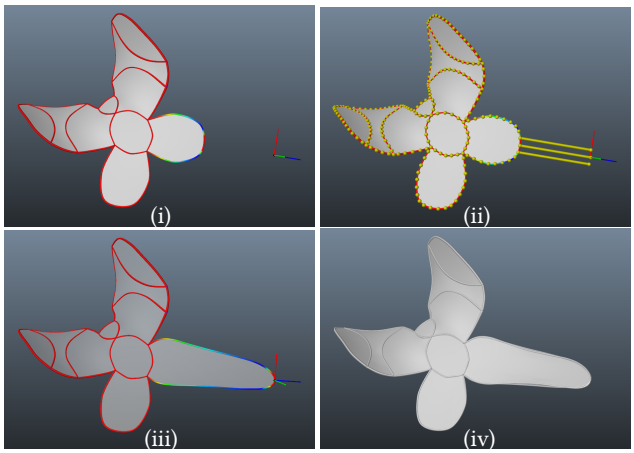

Fig. 8. **Two-Step Optimization for User-Guided Inverse Design**. Given a base shape (i), the user defines an edit (ii) by translating a subset of the rod vertices. We interactively optimize for a target shape (iii) subject to linearized equilibrium constraints. Once the target is accepted by the user, we compute a full nonlinear optimization to obtain the final shape (iv).

A standard approach to implementing such *inverse design* operations is to have the user define an edit relative to a given equilibrium configuration, and to perform nonlinear optimization in order to find a feasible shape that best approximates the user-provided target. In our setting, however, this approach is impractical: due to the restricted space of KPS, there is no guarantee that the target shape specified by the user will be close to a feasible configuration—and computing the closest feasible shape requires time-consuming nonlinear optimization. The resulting delay would interrupt the design process and the computed shape is not guaranteed to meet the expectations of the user.

Two-Step Optimization. Due to the complex nature of the design problem, nonlinear optimization cannot be entirely avoided. In order to still accelerate the design process, we propose a two-step optimization approach whose goal is to increase the likelihood of the user-specified target shape to be close to feasible.

In the first step, the user specifies an edited shape $\tilde{\mathbf{x}}$ starting from a given equilibrium configuration $\mathbf{x}_0$. We subsequently compute a target shape $\mathbf{x}^*$ by minimizing the distance to the user-provided edit, subject to linearized equilibrium and position constraints. While the resulting target shape will generally not satisfy the user input exactly, this optimization step is very fast allowing for interactive and progressive exploration. Once a satisfying target $\mathbf{x}^*$ shape is found, the second step performs a fully-nonlinear optimization to compute the equilibrium shape $\mathbf{x}$ that best matches the target. In this way, the time-consuming nonlinear optimization is executed only for target designs that are likely feasible.

Target Optimization. In order to specify desired changes to the equilibrium shape, the user can select sets of vertices in the deformed configuration and apply rotations, translations, or scaling transformations to them (Figure 8-ii). We also provide a soft selection tool, allowing the user to specify per-vertex weights $w \in [0, 1]$ that determine to what extent a given vertex is influenced by the

edited shape during optimization. The edited shape $\tilde{\mathbf{x}}$ defined in this way is generally infeasible, i.e., there is no choice of admissible parameter values such that the corresponding equilibrium configuration coincides exactly with the user-specified target. However, performing a nonlinear optimization to compute the closest feasible target shape would be too time-consuming for interactive exploration. We therefore use sensitivity analysis in order to compute an approximately feasible target shape that satisfies force and position constraints to first order.

Given a desired change in equilibrium shape $\Delta\tilde{\mathbf{x}} = \tilde{\mathbf{x}} - \mathbf{x}_0$, we seek to compute a change in parameters $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}_0$ that minimizes the distance measure $\|\mathbf{W}(\Delta\tilde{\mathbf{x}} - \mathbf{SP}\Delta\mathbf{p})\|^2$, where $\mathbf{W}$ is a diagonal matrix of soft selection weights whose entries indicate whether—or to what extent—the position of a given vertex is defined by the editing transformation. Directly minimizing this distance measure with respect to the unknown $\Delta\mathbf{p}$ is problematic since the projected sensitivity matrix $\mathbf{SP}$ is rank-deficient. We therefore add a regularizer $\|\mathbf{J}\Delta\mathbf{p}\|^2$ that only penalizes parameter updates in directions that affect the position constraints.

It should be noted that, while the change in shape $\mathbf{SP}\Delta\mathbf{p}$ satisfies both equilibrium and position constraints to first order, we also have to satisfy fabrication-related constraints on the design parameters, i.e., maximum and minimum rod widths. One option to enforce the corresponding bound constraints is by constrained quadratic programming, but we found even commercial solvers to be too slow for the interactive rates required by our application. Instead of enforcing them explicitly, we therefore choose to eliminate bound constraints through parameter transformation. In essence, we introduce mapped parameters $\hat{\mathbf{p}} = \phi(\mathbf{p})$ that are asymptotically clamped to their bounds via a trigonometric transfer function; see [Musialski et al. 2016] for details. Consequently, the sensitivity matrix is transformed to the new parameter space as $\hat{\mathbf{S}} = \mathbf{S}\nabla_p\phi$, which also affects the dependent matrices $\hat{\mathbf{J}}$ and $\hat{\mathbf{P}}$ in (6).

Combining the above components, the target optimization problem is finally formulated as

$$\min_{\Delta\hat{\mathbf{p}}} \frac{1}{2} \left(\|\mathbf{W} \left(\Delta\tilde{\mathbf{x}} - \hat{\mathbf{S}}\hat{\mathbf{P}}\Delta\hat{\mathbf{p}} \right) \|^2 + \|\hat{\mathbf{J}}\Delta\hat{\mathbf{p}}\|^2 \right), \tag{7}$$

where $\Delta\hat{\mathbf{p}} = (\hat{\mathbf{p}} - \hat{\mathbf{p}}_0)$. By minimizing (7), we obtain the optimal parameter change $\Delta\hat{\mathbf{p}}^*$ for a user-specified edit, subject to linearized equilibrium and position constraints, and compute the first-order feasible target shape as $\mathbf{x}^* = \mathbf{x}_0 + \hat{\mathbf{S}}\hat{\mathbf{P}}\Delta\hat{\mathbf{p}}^*$. Moreover, since $\hat{\mathbf{S}}$ and $\hat{\mathbf{J}}$ are kept constant, (7) is a quadratic minimization problem that can be solved fast enough for the user to interactively explore first-order feasible edits: the user can modify the equilibrium shape until the target shape is satisfactory or the norm of the nonlinear forces exceeds a given threshold, indicating that the linear approximation is becoming invalid. In either case, the full nonlinear problem is solved with the current parameter values, and the matrices $\hat{\mathbf{S}}$, $\hat{\mathbf{J}}$, and $\hat{\mathbf{P}}$ are recomputed.

Result Optimization. When the first-order feasible target $\mathbf{x}^*$ is satisfactory or the residual forces are excessive, we solve the full nonlinear constrained optimization problem. The objective function is the quadratic distance from the equilibrium configuration to the target. To eliminate drift due to constraint projection, we augment the objective function with a quadratic term that penalizes the violation of the hard constraints $\mathbf{c}(\mathbf{x})$. We set a larger weight k for the hard constraints than for the target positions, but in practice, the two terms hardly interfere since the target is already first-order feasible. In addition to these objective terms, we add nonlinear force equilibrium constraints as well as bound constraints, leading to the following optimization problem:

$$\begin{aligned} \mathbf{p} = \arg\min \frac{1}{2} \| \left(\mathbf{x}^* - \mathbf{x}(\mathbf{p}) \right) \|^2 + \frac{1}{2} k \, \|\mathbf{c}(\mathbf{x})\|^2 \\ \text{s.t. } \mathbf{f}(\mathbf{p}, \bar{\mathbf{x}}(\mathbf{p}), \mathbf{x}) = 0 \text{ and } \mathbf{p}_{\min} \le \mathbf{p} \le \mathbf{p}_{\max} . \end{aligned} \tag{8}$$

We solve this nonlinear optimization problem using a variant of Sequential Quadratic Programming (SQP). To this end, we initialize the problem with the equilibrium shape $\mathbf{x}_0$ and corresponding parameter vector $\mathbf{p}_0$, then alternate between optimization steps and static equilibrium solves. In each SQP iteration, we solve the corresponding bound-constrained Quadratic Program (QP) using the QuickQP algorithm from the ALGLIB library. In order to efficiently compute the objective Hessian, we combine an exact Gauss-Newton-type expression for the first-order part with a BFGS-like approximation for the second-order part as described in [Nocedal and Wright 2006]. Given a parameter update returned by the QP solver, we compute the corresponding updated shape by solving for static equilibrium.

Figure 8 shows an example of target and result optimization during inverse design. Our two-step optimization approach is key for achieving an interactive user experience while, at the same time, providing the accuracy demanded by the nonlinearity of the underlying design problem.

6 RESULTS

We have used our method to design and fabricate a set of Kirchhoff Plateau Surfaces that, taken together, provide an indication of the diversity of shapes that can be achieved.

6.1 Examples

In a typical design session, the user starts with a minimum structure and then progressively modifies the topology and geometry of the rod mesh through various editing operations. Whenever needed, the user can visualize the design space using our sensitivity exploration tool and make targeted edits using inverse design operations. Below we describe each of these examples in detail and report on the experiences that we made with our design tools.

Butterfly. (Figure 10) The overall shape of this example was designed through forward editing of the rod topology and geometry, but the aesthetic aspects required a combination of advanced forward and inverse design tools, as shown also in the accompanying video. In particular, singular value decomposition of the sensitivity matrix revealed interesting wing deformations that were used as a basis for the final design. Inverse design operations were used for adjusting the shape and curvature of the wings.

Automotive Concept Design. (Figure 1) As one of the most challenging examples, this design brings together complex geometric shapes, fine detail, and unintuitive topology. Most importantly, the overall shape of the target object is convex and, consequently, many of the rods were highly compressed initially. Some of the boundary

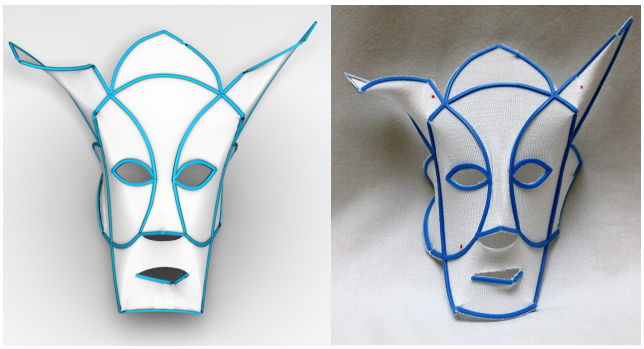

Fig. 9. **Mask**. Simulated design (*left*) and physical prototype (*right*). In this example, facial features lead to high structural stiffness, preventing the surface from reaching the desired curvature; see also Figure 6-i. Rod splitting and inverse design proved essential to achieve both overall shape and details.

rods were removed to release compression and to allow the planar rod mesh to deploy into an approximately convex shape. Inverse design operations were used to flesh out important features such as the windshield, and to adjust the size and volume of the car once the overall shape was defined. Rod splitting was also used to add detail to the front of the car, indicating the headlights.

Mask. (Figure 9) This example combines complex internal topology with the need for large curvature in order to approximate the shape of a face. These two targets compete, and the complexity in topology leads to high structural stiffness which, in turn, prevents the surfaces from achieving the desired curvature. Splitting of rods at multiple points allowed the mask to curve as desired and to emphasize salient facial features such as nose and mouth. Both the overall shape and the details were edited using inverse design.

Warrior's Armor. (Figure 11) This composite example showcases several small parts that are geometrically and topologically simple, but the high curvature needed within a small area leads to challenging fabrication constraints. Some parts such as the arms or the helmet benefited strongly from singular value decomposition of the sensitivity matrix in order to control their curvature. In addition,

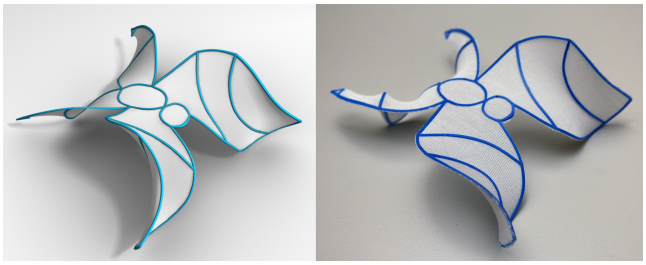

Fig. 10. **Butterfly**. Simulated design (*left*) and physical prototype (*right*). In this example, interactive deformation feedback was key for the topology and geometry design of the overall shape. As shown in the accompanying video, all our modeling tools were used for creating the final result.

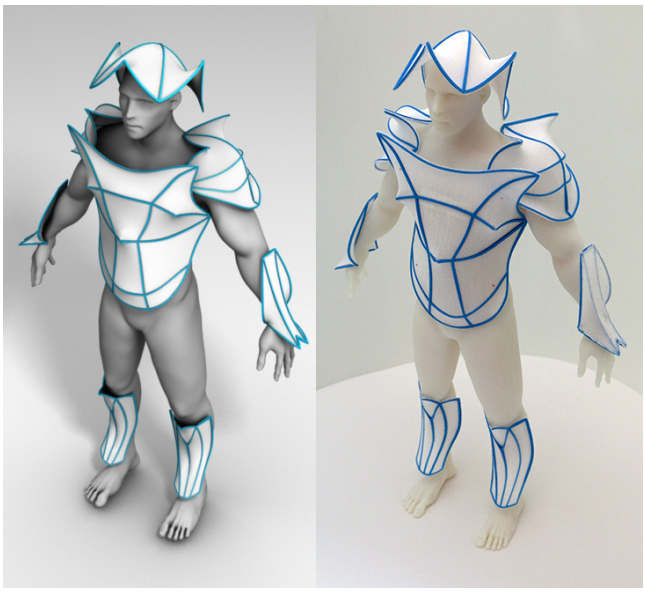

Fig. 11. **Warrior**. Simulated designs (*left*) and physical prototypes (*right*). For this example, sensitivity exploration and inverse design was used heavily in order to adjust the size and curvature of the individual parts such as to conform to the scale of the body.

inverse design was used heavily to adjust the size of the different pieces to a common scale.

Flower. (Figure 12) Another composite example, made of three parts, that illustrates the possibility to create diverse and appealing designs through a combination of forward and inverse design operations. As can be seen from the side-by-side view in Figure 12, this example also indicates very good correspondence between our simulation and the physically fabricated prototypes.

Design and simulation complexity for all the examples is summarized in Table 1. All designs were computed on a desktop machine with a 3.10GHz Intel i7 3770S processor with 16GB RAM. Note that target optimization was interactive for all examples. The static equilibrium solver required under one second for all examples, whereas the final result optimization took a few seconds on average.

6.2 Fabrication

For the fabrication of our examples, we used an off-the-shelf Ultimaker 2 FDM printer. The printed material is a standard PLA filament, and we used the material parameters provided by the manufacturer, i.e, Young's modulus $E = 3.31 \times 10^3 KPa$, Poisson ratio $\nu = 0.36$, density $\rho = 1240 Kg/m^3$. The membrane fabric that we used is a highly elastic, finely knitted, elastane-cotton blend. In order to obtain the material parameters for this fabric, we performed an experimental estimation of the stretch elastic moduli in both course and wales directions. We first performed several static deformation tests on a square piece of fabric, then optimized for the material parameters of a simulated counterpart. The fabric parameters obtained in this way are Young's modulus $E = 4.72 \times 10^{-2} KPa$, Poisson ratio $\nu = 0.30$, density $\rho = 122 Kg/m^3$.

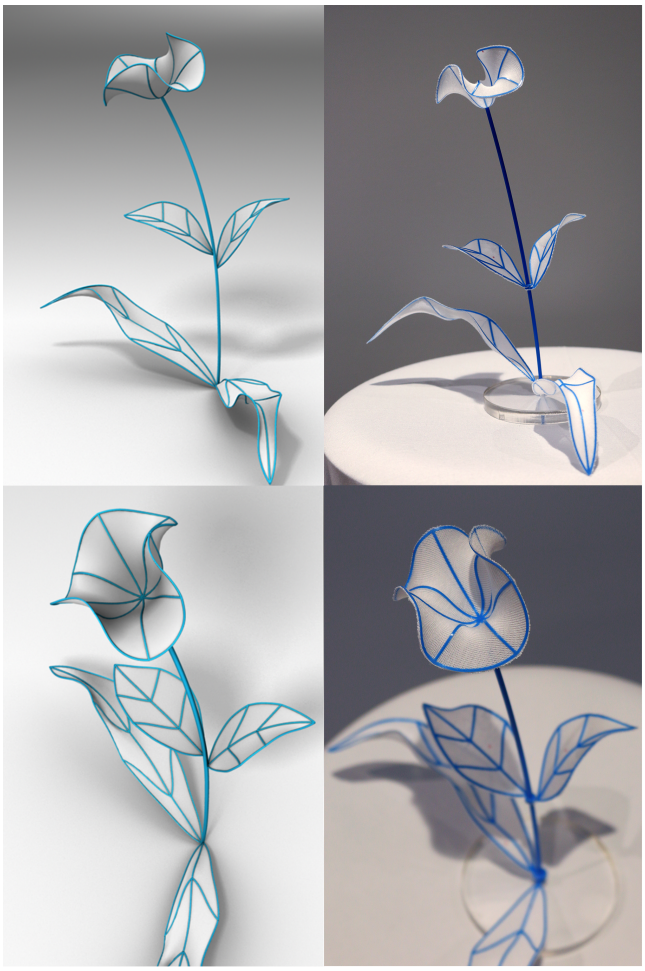

Fig. 12. **Flower**. This design is one example of the very good agreement that we observe between the equilibrium shapes predicted by simulation (*left*) and the actual fabricated prototypes (*right*).

Model	Control points	Rod nodes	Membr. nodes	Statics (ms)	Target (ms)	Result (s)
Car	81	3292	4803	746	140	7.28
Mask	72	2417	3223	602	146	8.10
Flower	52	1127	2172	418	46	4.32
Butterfly	50	2351	2505	226	42	6.22
Helmet	38	1280	3522	575	33	4.13
Shoulder	37	1373	3672	333	14	5.32
Arm	28	1266	3087	311	5	3.52
Shin	46	1540	3219	510	22	3.31
Chest	88	2147	3789	545	92	6.21

Table 1. Summary of design complexity (number of control points), simulation complexity (number of rod and membrane nodes), and solver performance (equilibrium solver, target optimization, and result optimization) for all examples.

Before printing, the membrane is manually stretched over a wooden frame, then clamped and inserted into the printing tray. We draw a calibration square on top of the membrane as a guideline to ensure the resulting stretch is as uniformly distributed and isotropic as possible. For each design, the amount of pre-stretch must be chosen such as to balance fabrication constraints (e.g., minimum radii) and aesthetic considerations (e.g., absence of wrinkles). We started with a moderately low value of 40% for all examples and let the user manually increase pre-stretch if necessary.

Considering the tray size of the printer, we are restricted to small rod radii. Below some threshold, rods do not bond well to the fabric. To avoid this problem, we set a lower bound of $0.5mm$ for the cross-section. Regarding the printer settings, we set head and bed temperatures to $210°C$ and $60°C$ respectively. Finally, we used 100% flow rate and $30mm/s$ printing speed for PLA filament with $1.8mm$ diameter.

7 CONCLUSIONS

We have presented a method for user-guided but computer-assisted design of Kirchhoff-Plateau Surfaces. To circumvent the modeling challenges of KPS, we have designed simulation and optimization methods that provide the user with powerful, interactive tools that allow for intuitive exploration of the design space.

Our work is not free of limitations. First, the materials and fabrication process used set important limitations on the range of results that can be achieved. As one particular example, we are currently limited by the tray size of the printer. In order to create larger models we can, in principle, combine several KPS. This, however, raises the question of how to best decompose a desired shape into pieces that can be well-approximated with KPS. The segmentation of arbitrary input models into sets of minimal surface patches is an interesting direction for future work. Other limitations are related to the modeling accuracy, both in terms of the rod and membrane models, as well as the material properties. Higher modeling accuracy would likely produce higher accuracy in the results. Finally, our method could be complemented with automatic topology optimization in order to increase the range of edits that can be achieved with inverse design operations.

ACKNOWLEDGEMENTS

The authors wish to thank Christian Schumacher for his help with results preparation and Javier Tapia for his work on parameter acquisition. We also thank Jonas Zehnder, Melina Skouras and Peter Kaufmann for their code contribution, as well as the other team members at URJC and DRZ. This work was supported in part by grants from the Spanish Ministry of Economy (TIN2015-70799-R) and the European Research Council (ERC Starting Grant no. 280135 Animetrics).

REFERENCES

Aric Bartle, Alla Sheffer, Vladimir G. Kim, Danny M. Kaufman, Nicholas Vining, and Floraine Berthouzoz. 2016. Physics-driven Pattern Adjustment for Direct 3D Garment Editing. *ACM Trans. Graph.* 35, 4, Article 50 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925896

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete Viscous Threads. *ACM Trans. Graph. (Proc. SIGGRAPH)* 29, 4, Article 116 (2010).

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete Elastic Rods. *ACM Trans. Graph. (Proc. SIGGRAPH)* 27, 3, Article 63 (2008).

Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. 2006. Super-helices for Predicting the Dynamics of Natural Hair. In *Proc. of ACM SIGGRAPH '06*.

Philippe Block and John Ochsendorf. 2007. Thrust Network Analysis: A New Methodology for Three-Dimensional Equilibrium. *Journal of the International Association for Shell and Spatial Structures* 48, 3, Article 155 (2007).

Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. 2014. An Asymptotic Numerical Method for Inverse Elastic Shape Design. *ACM Trans. Graph.* 33, 4, Article 95 (July 2014), 11 pages. https://doi.org/10.1145/2601097.2601189

Xiang 'Anthony' Chen, Stelian Coros, Jennifer Mankoff, and Scott E. Hudson. 2015. Encore: 3D Printed Augmentation of Everyday Objects with Printed-over, Affixed and Interlocked Attachments. In *ACM SIGGRAPH 2015 Posters (SIGGRAPH '15)*. ACM, New York, NY, USA, Article 3, 1 pages. https://doi.org/10.1145/2787626.2787650

Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2011. Spin Transformations of Discrete Surfaces. *ACM Trans. Graph.* 30, 4, Article 104 (July 2011), 10 pages. https://doi.org/10.1145/2010324.1964999

Fernando de Goes, Pierre Alliez, Houman Owhadi, and Mathieu Desbrun. 2013. On the Equilibrium of Simplicial Masonry Structures. *ACM Trans. Graph.* 32, 4 (2013).

Philippe Decaudin, Dan Julius, Jamie Wither, Laurence Boissieux, Alla Sheffer, and Marie-Paule Cani. 2006. Virtual Garments: A Fully Geometric Approach for Clothing Design. *Computer Graphics Forum* (2006). https://doi.org/10.1111/j.1467-8659.2006.00982.x

Bailin Deng, Sofien Bouaziz, Mario Deuss, Alexandre Kaspar, Yuliy Schwartzburg, and Mark Pauly. 2015. Interactive Design Exploration for Constrained Meshes. *Computer-Aided Design* 61 (2015), 13 – 23.

Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, and Joëlle Thollot. 2010. Stable Inverse Dynamic Curves. *ACM Trans. Graph.* 29, 6, Article 137 (Dec. 2010), 10 pages. https://doi.org/10.1145/1882261.1866159

Gerhard Dziuk and John E. Hutchinson. 1999. The Discrete Plateau Problem: Algorithm and Numerics. *Math. Comput.* 68, 225 (Jan. 1999), 1–23. https://doi.org/10.1090/S0025-5718-99-01025-X

Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. 2014. Wire Mesh Design. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4 (2014).

L. Giomi and L. Mahadevan. 2012. Minimal surfaces bounded by elastic lines. *Proceedings: Mathematical, Physical and Engineering Sciences* 468, 2143 (2012), 1851–1864.

Giulio G. Giusteri, Paolo Franceschini, and Eliot Fried. 2016. Instability Paths in the Kirchhoff–Plateau Problem. *Journal of Nonlinear Science* 26, 4 (2016), 1097–1132.

Giulio G. Giusteri, Luca Lussardi, and Eliot Fried. 2017. Solution of the Kirchhoff–Plateau Problem. *Journal of Nonlinear Science* (2017), 1–21. https://doi.org/10.1007/s00332-017-9359-4

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 (SCA '03)*. 62–67.

Christophe Guberan and Carlo Clopath. 2016. Active Shoes. http://www.selfassemblylab.net/ActiveShoes.php. (2016). MIT Self Assembly Lab.

Jean Hergel and Sylvain Lefebvre. 2014. Clean Color: Improving Multi-filament 3D Prints. *Computer Graphics Forum* (2014). https://doi.org/10.1111/cgf.12318

Martin Kilian, Simon Flöry, Zhonggui Chen, Niloy J. Mitra, Alla Sheffer, and Helmut Pottmann. 2008. Curved Folding. *ACM Trans. Graph.* 27, 3, Article 75 (Aug. 2008), 9 pages. https://doi.org/10.1145/1360612.1360674

Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. 2016. Beyond Developable: Computational Design and Fabrication with Auxetic Materials. *ACM Trans. Graph.* 35, 4, Article 89 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925944

K. Koohestani. 2014. Nonlinear force density method for the form-finding of minimal surface membrane structures. *Communications in Nonlinear Science and Numerical Simulation* 19, 6 (2014), 2071 – 2087. https://doi.org/10.1016/j.cnsns.2013.10.023

Yijing Li and Jernej Barbič. 2014. Stable Orthotropic Materials. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '14)*. 41–46.

Julian Lienhard, Holger Alpermann, Christoph Gengnagel, and Jan Knippers. 2013. Active Bending: a Review on Structures where Bending is Used as a Self-Formation Process. *International Journal of Space Structures* 28, 3-4 (2013), 187–196.

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. https://doi.org/10.1145/2461912.2461927

Yang Liu, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang. 2006. Geometric Modeling with Conical Meshes and Developable Surfaces. *ACM Trans. Graph.* 25, 3 (July 2006), 681–689. https://doi.org/10.1145/1141911.1141941

William H. III Meeks and Joaquín Pérez. 2011. The classical theory of minimal surfaces. *Bull. Amer. Math. Soc.* 48 (2011), 325–407.

Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A. Otaduy, and Steve Marschner. 2012. Data-Driven Estimation of Cloth Simulation Models. *Comput. Graph. Forum* 31, 2 (2012), 519–528. https://doi.org/10.1111/j.1467-8659.2012.03031.x

Eder Miguel, Mathias Lepoutre, and Bernd Bickel. 2016. Computational Design of Stable Planar-rod Structures. *ACM Trans. Graph.* 35, 4, Article 86 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925978

Yuki Mori and Takeo Igarashi. 2007. Plushie: An Interactive Design System for Plush Toys. *ACM Trans. Graph. (Proc. SIGGRAPH)* (2007).

Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofen, and Patrick Baudisch. 2014. faBrickation: Fast 3D Printing of Functional Objects by Integrating Construction Kit Building Blocks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3827–3834. https://doi.org/10.1145/2556288.2557005

Przemyslaw Musialski, Christian Hafner, Florian Rist, Michael Birsak, Michael Wimmer, and Leif Kobbelt. 2016. Non-linear Shape Optimization Using Local Subspace Projections. *ACM Trans. Graph.* 35, 4, Article 87 (July 2016), 13 pages. https://doi.org/10.1145/2897824.2925886

J. Nocedal and S. J. Wright. 2006. *Numerical Optimization* (2nd ed.). Springer, New York.

Hao Pan, Yi-King Choi, Yang Liu, Wenchao Hu, Qiang Du, Konrad Polthier, Caiming Zhang, and Wenping Wang. 2012. Robust Modeling of Constant Mean Curvature Surfaces. *ACM Trans. Graph.* 31, 4, Article 85 (July 2012), 11 pages. https://doi.org/10.1145/2185520.2185581

Daniele Panozzo, Philippe Block, and Olga Sorkine-Hornung. 2013. Designing Unreinforced Masonry Models. *ACM Trans. Graph.* 32, 4, Article 91 (July 2013), 12 pages. https://doi.org/10.1145/2461912.2461958

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. (Proc. SIGGRAPH)* 34, 4, Article 138 (2015).

Helmut Pottmann, Michael Eigensatz, Amir Vaxman, and Johannes Wallner. 2015. Architectural Geometry. *Comput. Graph.* 47, C (April 2015), 145–164. https://doi.org/10.1016/j.cag.2014.11.002

Christian Schüller, Daniele Panozzo, Anselm Grundhöfer, Henning Zimmer, Evgeni Sorkine, and Olga Sorkine-Hornung. 2016. Computational Thermoforming. *ACM Trans. Graph.* 35, 4, Article 43 (July 2016), 9 pages. https://doi.org/10.1145/2897824.2925914

Mélina Skouras, Stelian Coros, Eitan Grinspun, and Bernhard Thomaszewski. 2015. Interactive Surface Design with Interlocking Elements. *ACM Trans. Graph.* 34, 6, Article 224 (Oct. 2015), 7 pages.

Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. 2014. Designing Inflatable Structures. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4 (2014).

Justin Solomon, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2012. Flexible Developable Surfaces. *Comput. Graphics Forum* 31, 5 (2012).

Olga Sorkine. 2005. Laplacian Mesh Processing. In *Eurographics 2005 - State of the Art Reports*, Yiorgos Chrysanthou and Marcus Magnor (Eds.). The Eurographics Association.

J. Spillmann and M. Teschner. 2007. CoRdE: Cosserat Rod Elements for the Dynamic Simulation of One-dimensional Elastic Objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '07)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 63–72. http://dl.acm.org/citation.cfm?id=1272690.1272700

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. https://doi.org/10.1145/2832906

Chengcheng Tang, Xiang Sun, Alexandra Gomes, Johannes Wallner, and Helmut Pottmann. 2014. Form-finding with Polyhedral Meshes Made Simple. *ACM Trans. Graph.* 33, 4, Article 70 (July 2014), 9 pages. https://doi.org/10.1145/2601097.2601213

Christopher D. Twigg and Zoran Kačić-Alesić. 2011. Optimization for Sag-free Simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '11)*. ACM, New York, NY, USA, 225–236. https://doi.org/10.1145/2019406.2019437

Nobuyuki Umetani, Danny M. Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive Couture for Interactive Garment Modeling and Editing. *ACM Trans. Graph.* 30, 4, Article 90 (July 2011), 12 pages. https://doi.org/10.1145/2010324.1964985

T. Van Mele, L. De Laet, D. Veenendaal, M. Mollaert, and P. Block. 2013. Shaping tension structures by actively bent linear elements. *International Journal of Space Structures* 28, 3 (2013), 127–135. https://doi.org/10.1260/0266-3511.28.3-4.127

Amir Vaxman. 2014. A Projective Framework for Polyhedral Mesh Modelling. *Comput. Graph. Forum* 33, 8 (Dec. 2014), 121–131. https://doi.org/10.1111/cgf.12405

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. https://doi.org/10.1145/2185520.2185583

Emily Whiting, John Ochsendorf, and Frédo Durand. 2009. Procedural Modeling of Structurally-sound Masonry Buildings. *ACM Trans. Graph.* 28, 5, Article 112 (Dec.

2009), 9 pages. https://doi.org/10.1145/1618452.1618458

Yong-Liang Yang, Yi-Jun Yang, Helmut Pottmann, and Niloy J. Mitra. 2011. Shape Space Exploration of Constrained Meshes. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 30, 6 (2011).

Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski. 2016. Designing Structurally-sound Ornamental Curve Networks. *ACM Trans. Graph.* 35, 4, Article 99 (2016), 99:1–99:10 pages.

Haisen Zhao, Fanglin Gu, Qi-Xing Huang, Jorge Garcia, Yong Chen, Changhe Tu, Bedrich Benes, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. 2016. Connected Fermat Spirals for Layered Fabrication. *ACM Trans. Graph.* 35, 4, Article 100 (July 2016), 10 pages. https://doi.org/10.1145/2897824.2925958

Henrik Zimmer and Leif Kobbelt. 2014. Zometool rationalization of freeform surfaces. *IEEE Trans. on Visualization and Computer Graphics* 20, 10 (2014).