Intuitive Design Exploration of Constrained Meshes

Xin Zhao **Cheng-Cheng Tang** **Yong-Liang Yang** **Helmut Pottmann**

KAUST

Niloy J. Mitra

University College London

Abstract. *Based on the recent work of Yang et al. [2011], we propose several intuitive tools to quickly create new architectural freeform shapes starting from a single input design, while conforming to a set of prescribed constraints and optimizing for specified quality measures. We allow the user to control the final shape by prescribing desirable curves on the final shape, access the desirable regions of the constrained mesh manifold using smartly selected 2D mappings, and computationally generate multiple design alternatives that satisfy the user hints. These tools allow the designer to intuitively navigate the constrained mesh manifold and pick desirable shapes using a design gallery interface. We demonstrate the efficiency of the proposed tools using various case studies.*

1 Introduction

Designing architectural freeform surfaces remains challenging. The design process broadly consists of two phases: a *creative phase* where the designer comes up with a well-conceived meaningful 3D shape, and a *rationalization phase* where the initial geometric design is optimized to satisfy material-specific fabrication and environment-related global constraints. These two phases require complementary skills: the creative phase requires a mix of artistic talents, conceptual design, awareness of the project theme; while, the rationalization process, which optimizes the initial design such that the geometric form can be actually constructed, requires understanding of various engineering and fabrication constraints, e.g., material used for the surface panels, layout of the support network, etc.

Although such a two-stage approach is common, the separation results in several complications. The key difficulty is that any post-rationalization that optimizes the geometry to conform to non-linear constraints (e.g., element faces should be flat, floor-curves should be planar, etc.) changes the initial design in ways that are difficult for the designer to foresee. In a typical scenario, the designer starts with an initial 3D geometry that is rationalized by modifying the geometry, thus possibly disrupting the original design intent. Hence, the designer specifies further modifications, which in turn undergo further changes due to rationalization, and the

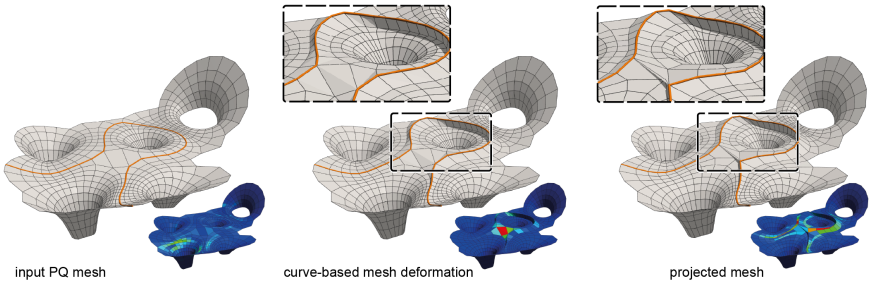

Figure 1: One option to enable curve-based constrained mesh deformation is to directly deform the mesh using curve-based mesh deformation tools and then use an appropriate *projection* procedure to restore the original constraints. Such a workflow does not facilitate intuitive exploration of design possibilities as the user has to manually account for non-linear projections often resulting in artifacts and undesirable fold-overs. We show a typical example using PQ meshes, where each mesh face is constrained to be planar. Insets show color-coded deviation from desirable planarity measure with blue denoting near planar regions.

process is iterated until the designer is satisfied. Such an iterative process is slow, expensive, hinders creative form-finding, and often encourages conservative rule-based designs. For example, in the case of PQ (planar quad) meshes, i.e., meshes with planar quadrilateral faces, optimization based rationalization methods are popular [Liu et al. 2006; Liu et al. 2011; Zadravec et al. 2010]. Effectively, such an algorithm *projects* the input geometric shape to a PQ mesh. Although the rationalization method works well when the initial design is close to a PQ mesh, in other cases the projection can produce significant and unpredictable changes, especially due to the non-linearity of the face constraints (see Figure 1). Such changes can be counter-intuitive and difficult to manually account for. Similar problems arise in presence of other constraints, e.g., floor curves should be planar, characteristic curves to be preserved, etc. Effectively, in absence of appropriate computational support, the designer is left guessing what are the achievable shapes that respect the specified constraints. This makes form-finding difficult.

Recently, Yang et al. [2011] introduced *constrained mesh manifolds* consisting of meshes that are characterized by the associated constraints involving the mesh vertices (connectivity is kept fixed). Each variant of the input mesh is simply represented as a point in a high-dimensional manifold (see Figure 2). Among all such possible shapes only those that conform to certain mesh qualities such as fairness are interesting. The main goal of the work is to characterize the useful design space consisting only of such desirable meshes and then efficiently explore them.

We use the above framework to present novel design tools allowing the users to control target designs via curve-level design handles. We generate a family of design possibilities, where each member satisfies all the input set of constraints. Such a guided exploration interface enables design space navigation that is currently very difficult using any existing approach. We evaluate the effectiveness and useability of the proposed computational design tools using various case studies.

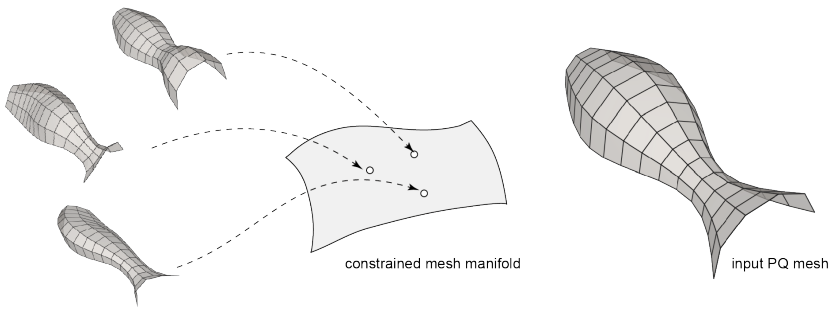

Figure 2: Given a single input mesh along with a set of constraints that should be preserved (e.g., each mesh face should be planar as in this example), we construct a *constrained mesh manifold* to capture all meshes having the same connectivity as the input mesh while conforming to all the input constraints within allowable error margin. We develop intuitive tools to restrict exploration to only certain desirable parts of such constrained mesh manifolds.

2 Overview

Given a single architectural surface mesh with a set of non-linear constraints, we provide the user with several convenient tools to quickly create new design variations directly from the input. The new meshes are generated such that they fulfill the shape preference of the user and also the prescribed constraints (see Figure 3).

In summary, we develop

- an intuitive curve editing metaphor to interactively generate new constrained meshes, and
- a fast and automatic sampling algorithm to generate many variant designs while satisfying the prescribed constraints.

3 Background on the Design Space of Constrained Meshes

Let us briefly recall some essentials of the mathematical framework developed by Yang et al. [2011] for the exploration and navigation of design spaces of constrained meshes. Given a single input mesh that meets the constraints, the useful design space contains all meshes that share the combinatorics and constraints of the input mesh, while having desirable mesh qualities that are mostly related to mesh fairness and other soft constraints.

Starting from a single input mesh, the family of meshes that share the same mesh connectivity is simply represented by their varying vertex positions, i.e., a point $\mathbf{x} = (v_1, \ldots, v_n) \in \mathbb{R}^D$, where D is 3 times the number n of deformable vertices v_i. Then any vector $\mathbf{d} \in \mathbb{R}^D$ defines a deformation field on the mesh producing the new mesh $(\mathbf{x} + \mathbf{d})$. For large steps in the tangent space, we rely on the availability of a projection operator to return to the constrained mesh manifold (see Figure 5),

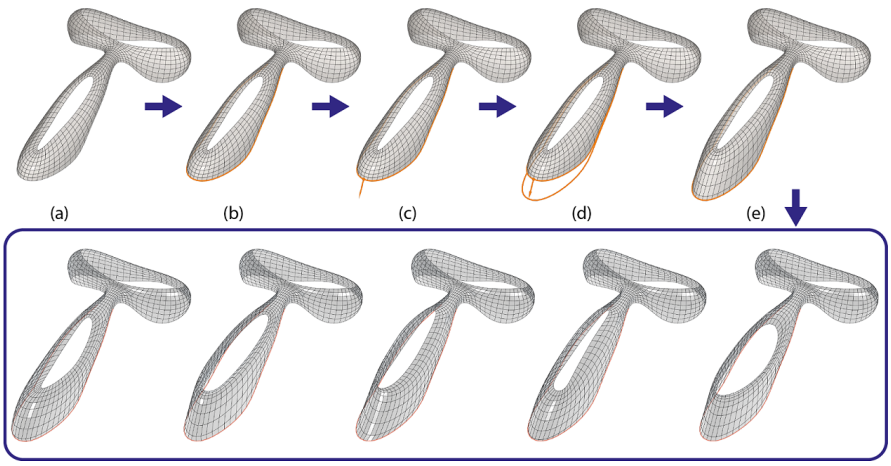

Figure 3: Typical workflow using our curve deformation based shape sampling. Starting from a given mesh (a), the user selects a control curve (b), drags a control point (c) to deform the curve (d). We first generate a new mesh (e) constrained by the deformed control curve. Further, we present a design-gallery styled exploration interface using a shape space sampling strategy that generates more possible designs (bottom). Each such design contains the user-prescribed curve while still satisfying the original set of constraints. These tools make form-finding more efficient since the designer only explores the useful part of the mesh manifold and does not have to manually account for non-linear effects in post-rationalization steps.

e.g., for PQ meshes we use a rationalization as proposed by Liu et al. [2006]. Our framework heavily relies on those deformations that keep the constraints in first differentiation order; for a second order analysis we refer to Yang et al. [2011].

Let $E_i(\mathbf{x}) = 0,\ i = 1,\ldots,m$, denote the constraints imposed on a mesh $\mathbf{x}$, where $|E_i(\mathbf{x})|$ are practically meaningful deviation measures. We assume to have m mostly non-linear constraints. From a mathematical perspective, the only requirement on the constraint functions $E_i(\mathbf{x})$ is that gradients ∇E_i are well defined. The corresponding *constrained mesh manifold* $\mathcal{M}$ is then formed by those meshes (or points in $\mathbb{R}^D$) which satisfy all constraints, and thus it is the *intersection* of the m surfaces $\Gamma_i = \{\mathbf{x} \in \mathbb{R}^D \mid E_i(\mathbf{x}) = 0, i = 1,\ldots,m\}$. Hence, $\mathcal{M}$ is in general of dimension $D-m$.

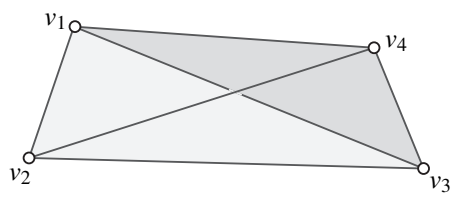

Figure 4: In the context of PQ meshes, we measure deviation of planarity for each quad face as the signed distance between its two diagonals. In our examples, for glass panels of dimensions 2m×2m, we assume a diagonal deviation margin of 10mm as acceptable.

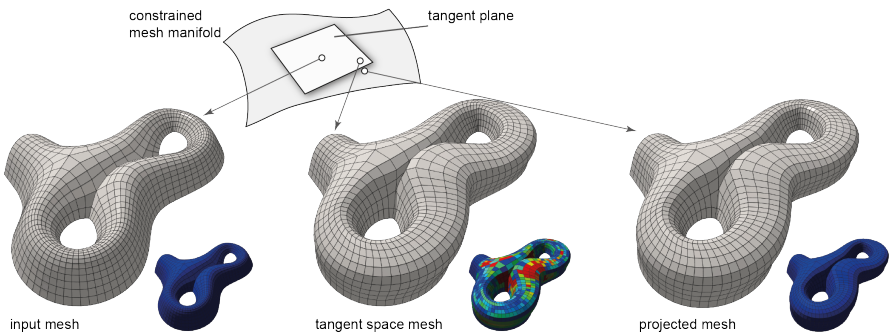

Figure 5: In our framework, we restrict navigation to local tangent planes of the underlying constrained mesh manifold. While this leads to interactive performance, approximation errors can appear along long traversals on the tangent space as the meshes leave the constrained mesh manifold. We use appropriate projection operators to return back to the mesh manifold with little visible geometric difference, which is important in preserving the desired design. In this example we use PQ meshes to illustrate this behavior.

We illustrate our framework on the specific example of planar quad (PQ) meshes, where the non-linear constraints are the (deviation from) planarity measure associated with each face. As this measure $E_i(\mathbf{x})$, we use the signed distance between the face diagonals for any quad face $f_i := \{v_1, v_2, v_3, v_4\}$ of mesh $\mathbf{x}$ (see Figure 4),

$$E_i(\mathbf{x}) = \left(\frac{v_1+v_3}{2} - \frac{v_2+v_4}{2}\right) \frac{(v_1-v_3)\times(v_2-v_4)}{\|(v_1-v_3)\times(v_2-v_4)\|}. \tag{1}$$

The definition directly correlates to tolerances typically allowed by various fabrication technologies.

3.1 Tangent space of the constrained mesh manifold

A given mesh corresponds to a point $\mathbf{x}_0 \in \mathcal{M}$. The tangent space of $\mathcal{M}$ at $\mathbf{x}_0$ is the intersection of the m tangent hyperplanes to the surfaces Γ_i. The normals of these tangent hyperplanes are given by the gradients $\nabla E_i(\mathbf{x}_0)$ and therefore the the *normal space* of $\mathcal{M}$ at a point $\mathbf{x}_0$ is spanned by the gradients $\nabla E_i(\mathbf{x}_0), i = 1, \ldots, m$. At any *regular* point of $\mathcal{M}$, i.e., where the gradients are linearly independent, we have a normal space of dimension m and a tangent space of dimension $D-m$. The tangent space $\mathrm{T}_{\mathcal{M}}(\mathbf{x}_0)$ to the constrained mesh manifold $\mathcal{M}$ is attached to the point $\mathbf{x}_0$ and spanned by vectors $\mathbf{t}$ orthogonal to each of $\nabla E_i(\mathbf{x}_0)$,

$$\mathrm{T}_{\mathcal{M}}(\mathbf{x}_0) := \{\mathbf{x}_0 + \mathbf{t} \mid \nabla E_i^T(\mathbf{x}_0) \cdot \mathbf{t} = 0 \,\forall\, i = 1, \ldots, m\}. \tag{2}$$

Suppose the basis of the normal space at the current point $\mathbf{x}_0$ is $\{\mathbf{n}_1, \mathbf{n}_2, ..., \mathbf{n}_m\}$ and the basis of the tangent space is $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_{D-m}\}$. Then any tangent vector can be expressed in the form $\mathbf{t} = \sum_j u_j \mathbf{e}_j$ where $\mathbb{u} = [u_1\, u_2 \ldots u_{D-m}]^T \in \mathbb{R}^{D-m}$ parameterizes the tangent space. Note that $\mathbf{t}$ represents a mesh deformation field preserving constraints up to first order.

3.2 Accessing the design space

Unfortunately, a significant portion of the constrained mesh manifold $\mathcal{M}$ will be useless for our application, since the corresponding meshes do not meet the expectations on other mesh qualities such as fairness. The part of $\mathcal{M}$ which meets these additional requirements is the actual *design space*. We now show how to reach this design space and how to navigate in it. Here, we assume that the input mesh which has been used to define $\mathcal{M}$ lies in the design space.

We describe mesh aesthetics and other soft mesh properties with help of functions $F(\mathbf{x})$ defined on the embedding space $\mathbb{R}^D$ (not just restricted to the mesh manifold). We will require a second order analysis, and thus have to look at the second order Taylor expansion of $F(\mathbf{x})$ at mesh $\mathbf{x}_0$,

$$\begin{aligned} F(\mathbf{x}) &= F(\mathbf{x}_0) + \nabla F^T \cdot (\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T \cdot H_F \cdot (\mathbf{x}-\mathbf{x}_0) \\ &\quad + o(\|\mathbf{x}-\mathbf{x}_0\|^2), \end{aligned} \tag{3}$$

Here H_F denotes the Hessian matrix of the function F at $\mathbf{x}_0$. If we restrict navigation of the mesh manifold to the tangent space $\mathrm{T}_{\mathcal{M}}(\mathbf{x}_0)$, we are restricted to points of the form $\mathbf{x} = \mathbf{x}_0 + \sum_{i=1}^{D-m} u_i \mathbf{e}_i$. The function $F(\mathbf{x})$ can then be expressed in terms of the parameter vector $\mathbb{u}$ as

$$F(\mathbf{x}_0) + \sum_{i=1}^{D-m} (\nabla F^T \cdot \mathbf{e}_i) u_i + \frac{1}{2} \mathbb{u}^T \cdot H_F^r \cdot \mathbb{u} + o(\mathbb{u}^2). \tag{4}$$

Here H_F^r is the so-called reduced Hessian of H_F with respect to the tangent space. It is of the form $H_F^r = [\mathbf{e}_1\, \mathbf{e}_2 \ldots \mathbf{e}_{D-m}]^T H_F [\mathbf{e}_1\, \mathbf{e}_2 \ldots \mathbf{e}_{D-m}]$.

As quality functions F we use fairness energies F_{fair} for the families of polylines an aesthetically pleasing mesh can be decomposed into (away from extraordinary vertices). Moreover, if our exploration should keep the mesh close to a reference surface, we use an appropriate function F_{close}. In order to see significant shape changes, it is a good idea to look at local deformation fields that are as orthogonal as possible to the reference geometry, which again can be described with an appropriate function F_{ortho}. For details on these functions, we refer the reader to Yang et al. [2011].

Finally, the functions F_i one wants to use in an exploration are combined with desired weights λ_i into a single energy $F = \sum_i \lambda_i F_i$. Its reduced Hessian is then simply

$$H^r(F) = \textstyle\sum_i \lambda_i H^r(F_i). \tag{5}$$

The reduced Hessian (even better, the intrinsic Hessian of [Yang et al. 2011]) is useful to identify deformation vector fields for exploration which do not change the mesh quality too much. Recall that the eigenvalues of the intrinsic Hessian H_F^r are the locally extremal second directional derivatives of the quality function F. Assuming that the input mesh has high quality, all first directional derivatives of F at $\mathbf{x}_0$ will be small, and thus a tangent vector of the mesh manifold which also

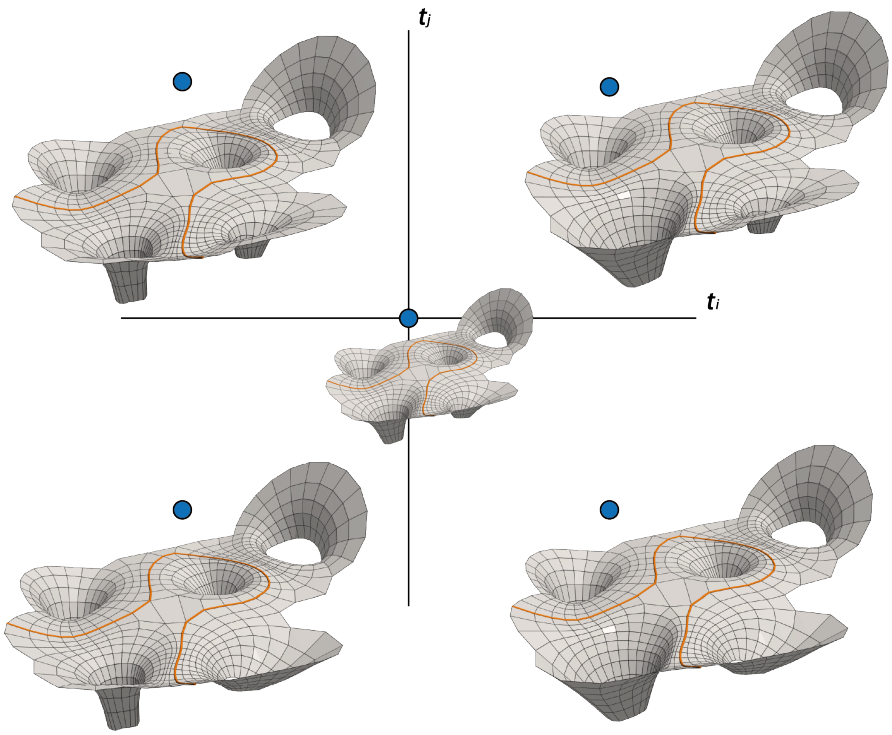

Figure 6: Spectral analysis of the reduced Hessian matrices extracted from the input constrained mesh provides rich exploration possibilities. A pair of selected eigen-directions $(\mathbf{t}_i, \mathbf{t}_j)$ of the reduced Hessian exposes a 2D section of space of desirable constrained meshes. During exploration, the user directly specifies a point on the spectral slice of the design space while the system interactively displays the corresponding constrained mesh. The highlighted curve denotes the user-prescribed curve that is guaranteed to lie on all the designed meshes; further, each mesh face is constrained to be planar within allowable margin of deviation.

has a small second derivative of F will indicate a good exploration direction. We therefore perform an eigen-analysis of H_F^r (see Figure 6). We restrict navigation to the subspace of the tangent space $\mathrm{T}_{\mathcal{M}}(\mathbf{x}_0)$ that is spanned by the eigenvectors to the lowest few eigenvalues of the reduced Hessian H_F^r (lowest 5% in our examples).

We now describe the tools we introduce to facilitate intuitive handling of architectural freeform shapes and subsequent sampling and exploration possibilities.

4 Designing in Constrained Mesh Manifolds

4.1 Curve as control handles

In freeform architectural design, a few curves may dominate the aesthetic characteristic of the shapes (see e.g., [Mehra et al. 2009]). Hence, designers often take great

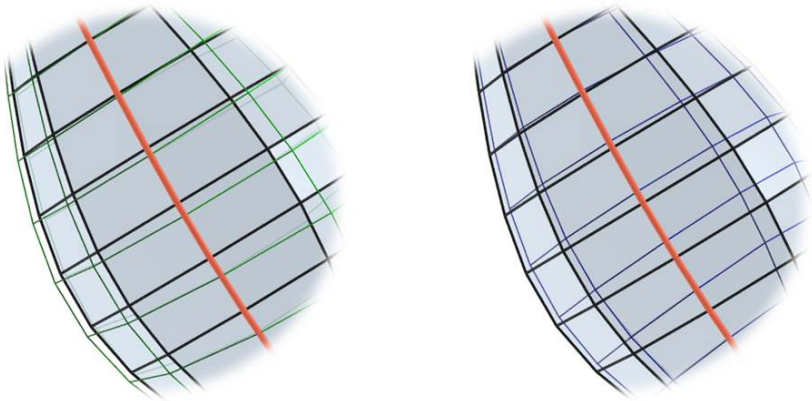

Figure 7: Curve constraint: on a given mesh (black), when a curve (thick red) is specified as a constraint by the user, vertices on the curve can only slide along the tangential directions. This ensures that the resultant meshes (thin green on the left, and thin blue on the right) also contain the desired curve.

care to prescribe such curves and expect the rest of the shape to blend in. Further, in such a curve-based manipulation framework, the exact vertex positions on the prescribed curves are less important, as long as the final shape carries the curves (see Figure 7). Thus, a desirable design tool should (i) support curve-based control handles for shape manipulation and (ii) allow the vertices to move along the prescribed curves to enhance the aesthetic appeal of the final shape.

Yang et al. [2011] suggest vertex-based manipulation where the designer specifies the final locations of a few vertices, say $v_j \to v'_j$ (including fixed vertices), while the optimization moves the free vertices to restrict the final mesh to the constrained mesh manifold and maximize the prescribed quality function. For small displace-

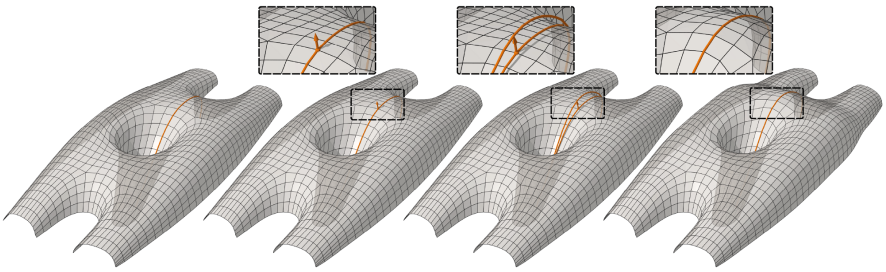

Figure 8: A curve editing example: starting from an input PQ mesh (left), the user selects and deforms a curve (middle-left, middle-right) to produce a final mesh containing the desired curve while still satisfying all the input constraints (e.g., each mesh face remains planar) using the corresponding constrained mesh manifold.

ments, this amounts to solving for a tangent vector $\mathbf{t} \in \mathrm{T}_{\mathcal{M}}(\mathbf{x}_0)$ as,

$$\begin{aligned} &\min_{\mathbf{t}} F(\mathbf{x}_0 + \mathbf{t}) \quad \text{such that,} \\ &\nabla E_i^T \cdot \mathbf{t} = 0,\ \forall\, i = 1, \ldots, m; \quad t_j = v'_j - v_j, \end{aligned} \tag{6}$$

where t_j denotes the j-th 'coordinate' of $\mathbf{t}$ (actually 3 coordinates, associated with vertex v_j).

In order to support curve-based manipulations, we allow the user to select and deform the curve handles as follows (see Figure 8). The user first selects a sequence of mesh edges as a discrete curve (i.e., a polyline) on the surface and then relocates one or more points on such curves to guide the curve deformation. We then deform the selected curve so that the point constraint(s) are satisfied using a differential coordinate-based curve deformation approach proposed by Nealen et al. [2007]. Thus the user can prescribe a set of curve deformations of the form $\mathcal{C}_j \to \mathcal{C}'_j$ (see also Bouaziz et al. [2012] for a handle-based constrained mesh deformation framework for shape exploration).

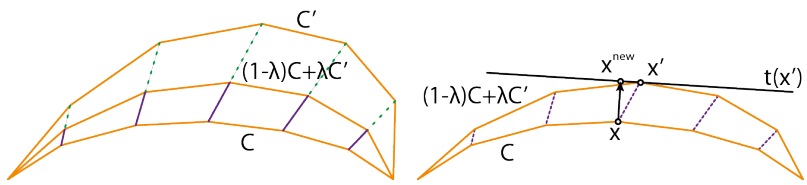

Figure 9: For large deformations, we move curve $\mathcal{C}$ to $\mathcal{C}'$ in multiple steps. In each step, we create a temporary curve linearly interpolating the current and the goal curve with interpolation parameter λ. Note that the curve points are allowed to move on the tangent to the interpolated curve at the corresponding new points.

One way to support curve-based interaction handles is to apply the vertex-based deformation as proposed in Equation 6 to all vertices of the deformed curve. Such an approach, however, can produce artifacts as the curve vertices are unnecessarily fixed. Instead, for each vertex $x \in \mathcal{C}$ and a small displacement, we restrict the new position x^{new} to an estimated tangent of $\mathcal{C}'$ at x' via the constraint $(x^{new} - x') \times t(x') = 0$. Here, $t(x')$ denotes a unit direction vector of the tangent (see Figure 9). Thus, for

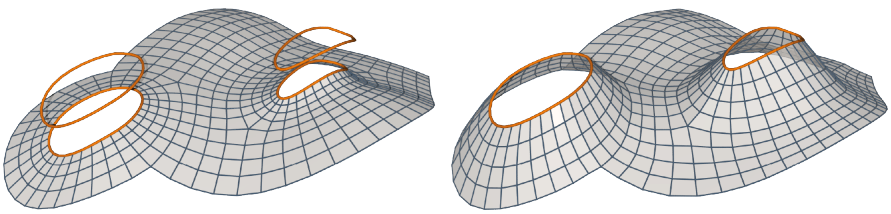

Figure 10: Another curve based deformation of a PQ mesh. The user prescribed boundary curves as control handles in this example.

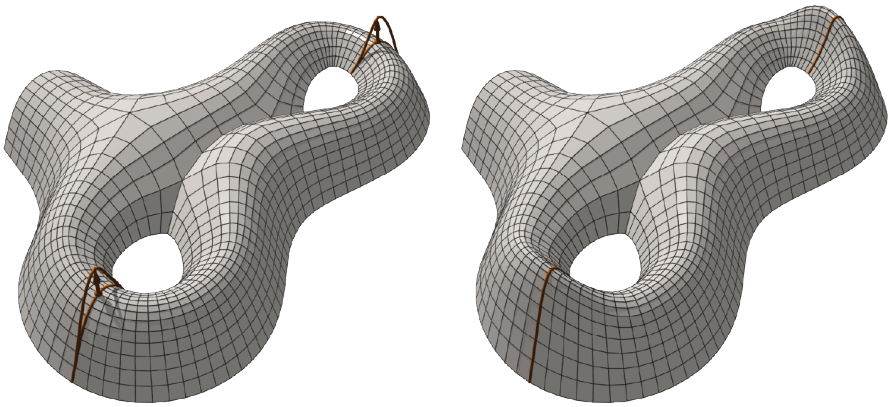

Figure 11: Curve editing with multiple control curves.

small steps, the vertices freely slide along the curves to produce a mesh with higher quality measure.

Larger displacements of curves are handled in several steps. Instead of aiming directly at $\mathcal{C}'$, we move towards a linearly blended polygon $(1-\lambda)\mathcal{C}+\lambda\mathcal{C}'$, which is made sufficiently close by a suitable choice of $\lambda < 1$. We proceed in this way (with increasing λ) until $\mathcal{C}'$ can be reached in a single step. The same framework also handles multiple curves-handles (see Figures 11 and 10).

Note that since we only use first order information, for large deformations the constraints may deviate beyond the allowable tolerance level, i.e., the final mesh may be too far off the constrained mesh manifold. We then use a projection operation to bring the mesh back onto the manifold. For planarity constraints, we employ the projection algorithms proposed in Liu et al. [2006].

4.2 Exploring the design space

Starting from the new mesh containing the deformed curve, we support exploration of the neighboring design space within the constrained mesh manifold as in Yang et al. [2011]. Note that in addition to the prescribed non-linear constraints such as planarity, we also want to preserve the deformed curve as marked by the user (see Figure 12). The simplest way is to just fix all the points on the curve and restrict exploration only to a constrained mesh created using the free part of the mesh. Such an approach, however, unnecessarily restricts the solution space. Instead, we integrate the curve constraints directly into the constrained mesh manifold.

Instead of fixing the whole curve, we allow the curve points to move along the curve. Specifically, we constrain curve points to locally move along the tangent direction. Suppose a vertex on a curve is $v \in \mathcal{C}$, its current position is v_0 and the tangent is $t(v_0)$. Sliding along the curve then amounts to $(v-v_0)\times t(v_0) = 0$. For each such curve point, the zero cross product with its tangent can be simply satisfied by adding two linear independently constraints.

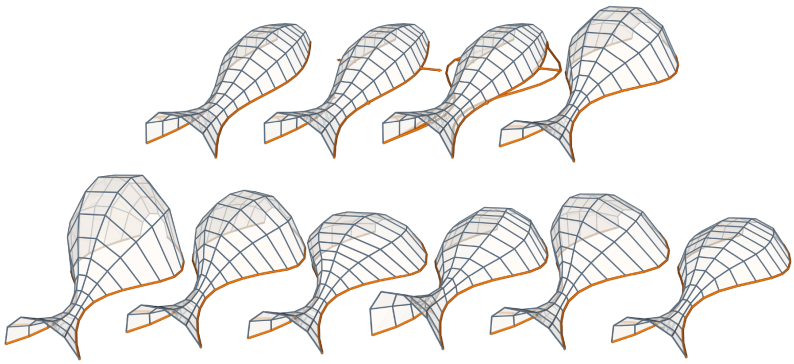

Figure 12: (Top) Starting from a constrained mesh (each mesh face is planar; also each of brown curves are planar), user prescribes curve edits to create a new constrained mesh using our framework. The bottom row shows design space sampling results with the input set of constraints along with the user prescribed curve constraint.

We then construct the reduced Hessian of the new constrained mesh manifold as described in Section 3.2. A practical way of exploration uses only a pair of eigen-directions. Such a pair captures a planar slice through the tangent space in which the user can directly explore only meshes with good aesthetic behavior (see Figure 6).

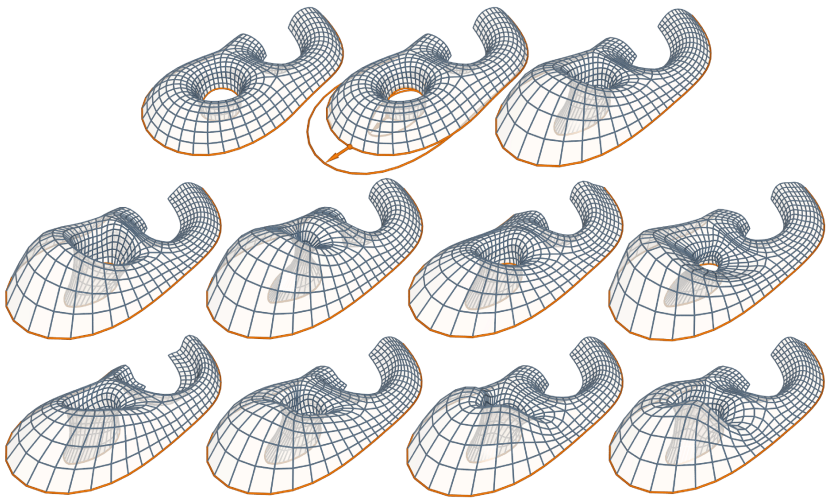

Figure 13: Starting from an input PQ mesh (top-left), the user edits the constrained mesh using two curve handles, and then we present multiple variations satisfying the input set of constraints and the curve specifications using our design space exploration tools.

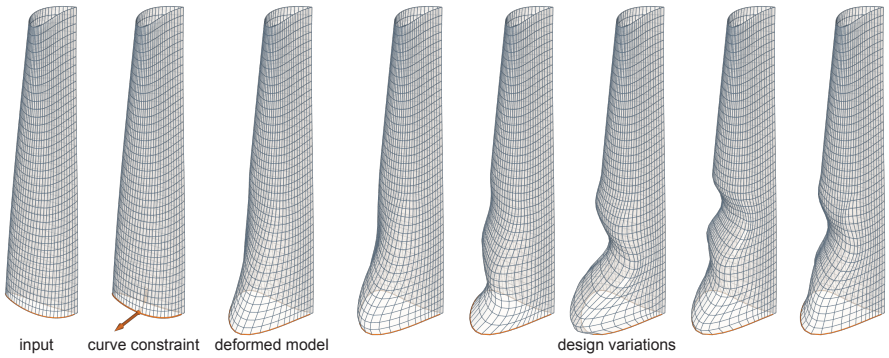

Figure 14: Starting from an input PQ model, the designer manipulates a curve-handle producing a deformed model. The prescribed curve handle, which can be distinctive in this case, can then be retained during subsequent design exploration thus simplifying the exploratory design phase. (Input model courtesy of Alexander Schiftner.)

4.3 Sampling the design space

In addition to allowing direct exploration of the design space using eigen-vectors of the corresponding reduced Hessian matrices, we also sample the parameter space to provide a preview of the design possibilities. Specifically, given a constrained mesh, we first deform the mesh using the user-prescribed curve constraints as demonstrated before. Let such a mesh be denoted by $\mathbf{x}_0$. We create a constrained mesh manifold (along with the corresponding linear constraints due to specified curve constraints), compute its reduced Hessian to identify a desirable subspace to explore (we retain only the top 5 directions in our tests). Let the desirable subspace be $T'_{\mathcal{M}}(\mathbf{x}_0)$. We start a set of sampled meshes with $\mathcal{M} = \{\mathbf{x}_0\}$. Then, we randomly sample parameters in the extracted desirable subspace to extract meshes $\mathbf{x}_i \in T'_{\mathcal{M}}(\mathbf{x}_0)$ and add $\mathbf{x}_i$ to $\mathcal{M}$, i.e., $\mathcal{M} \leftarrow \mathcal{M} \cup \mathbf{x}_i$ only if the current sample is not too close to an existing sample, i.e., $d(\mathbf{x}_i, \mathbf{x}_j) > \Delta$ for all $\mathbf{x}_j \in \mathcal{M}$, where Δ is a user prescribed density threshold.

We stop when the size $|\mathcal{M}|$ exceeds a predefined threshold (5-10 in our tests). Such meshes can be seen as samples of the design domain and offer a quick preview of design possibilities via a design-gallery interface (see also [Marks et al. 1997]). Figures 12, 13, and 14 provide typical examples. Note that the user can select any of the sampled designs and then directly refine the parameters on the tangent space $T'_{\mathcal{M}}(\mathbf{x}_0)$. If needed, the solutions are projected to the constrained mesh using the corresponding projection operator, but since the deviations from the constrained mesh manifold are small, such projections have negligible visual effects (see Figure 15).

In all the examples, curve deformation runs in real-time, curve-handle based constrained mesh deformation takes about 1-2 seconds, while sampling the constrained mesh manifold takes about 50 meshes per second. However, precomputation of the tangent space and spectral decomposition of the reduced Hessian takes 2-3 seconds for meshes with 300 faces as measured on a laptop.

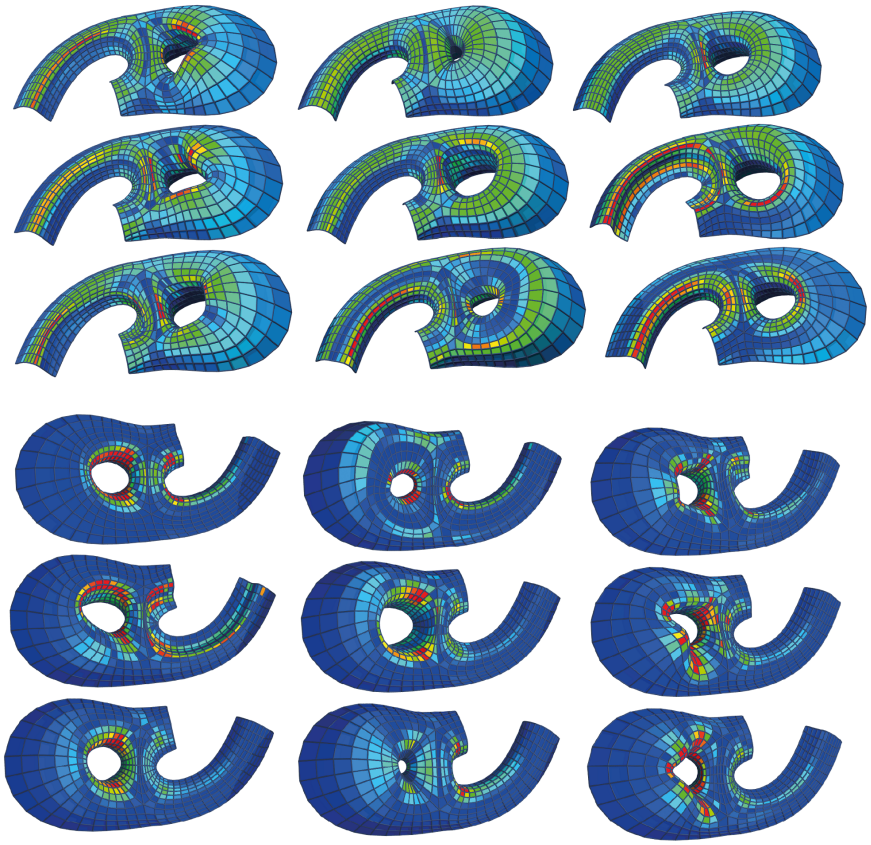

Figure 15: Visualization of the mean curvature (top) and Gaussian curvature (bottom) confirms the variety of design sampling results in the constrained manifold.

5 Conclusion

We presented a set of intuitive tools for specifying, designing, and exploring constrained meshes and associated design possibilities. Specifically, starting from a single constrained mesh, the user can directly control the final shape by prescribing desirable curves on the final shape while we computationally enable navigation of the desirable regions of the constrained mesh manifold. We also provide smartly selected 2D mappings and generate multiple design alternatives that satisfy the user hints. We believe that our proposed tools facilitate intuitive form-finding by allowing the designers to only focus on the design space, formed by the valid and desirable regions of the constrained mesh manifold. We demonstrated the efficiency of the tools using various case studies.

Acknowledgments

We thank Michael Eigensatz, Mathias Hobinger, Alexander Schiftner, and Johannes Wallner for their help with test models and also for their valuable comments, and the anonymous reviewers for their feedback. The work has been partially supported by Austrian Science Fund (FWF) grant P23735-N13, Austrian Science Promotion Agency (FFG) grant 813391, and Marie Curie Career Integration Grant 303541.

References

BOUAZIZ, S., DEUSS, M., SCHWARTZBURG, Y., WEISE, T., AND PAULY, M. 2012. Shaping discrete geometry with projections. *Symp. on Geometry Processing*.

LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph. 25*, 3, 681–689.

LIU, Y., XU, W., WANG, J., ZHU, L., GUO, B., CHEN, F., AND WANG, G. 2011. General planar quadrilateral mesh design using conjugate direction field. *ACM Trans. Graph. (SIGGRAPH Asia) 30*, 6, 140:1–140:10.

MARKS, J., ANDALMAN, B., BEARDSLEY, P. A., FREEMAN, W. T., GIBSON, S., HODGINS, J. K., KANG, T., MIRTICH, B., PFISTER, H., RUML, W., RYALL, K., SEIMS, J., AND SHIEBER, S. M. 1997. Design galleries: a general approach to setting parameters for computer graphics and animation. In *SIGGRAPH*, 389–400.

MEHRA, R., ZHOU, Q., LONG, J., SHEFFER, A., GOOCH, A., AND MITRA, N. J. 2009. Abstraction of man-made shapes. *ACM Trans. Graph. (SIGGRAPH Asia) 28*, 5, #137, 1–10.

NEALEN, A., IGARASHI, T., SORKINE, O., AND ALEXA, M. 2007. Fibermesh: designing freeform surfaces with 3d curves. *ACM Trans. Graph. 26*, 3.

YANG, Y.-L., YANG, Y.-J., POTTMANN, H., AND MITRA, N. J. 2011. Shape space exploration of constrained meshes. *ACM Trans. Graph. (SIGGRAPH Asia) 30*, 6, 124:1–124:12.

ZADRAVEC, M., SCHIFTNER, A., AND WALLNER, J. 2010. Designing quad-dominant meshes with planar faces. *Computer Graphics Forum 29*, 5, 1671–1679.