Shape Space Exploration of Constrained Meshes

Yong-Liang Yang
KAUST

Yi-Jun Yang
KAUST

Helmut Pottmann
KAUST/TU Vienna

Niloy J. Mitra
KAUST/TU Vienna

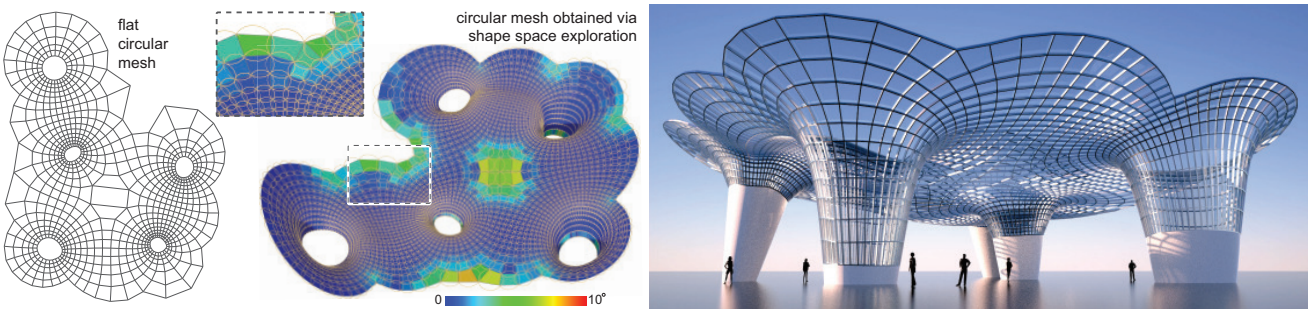


Figure 1: *Starting from a single input mesh along with a set of non-linear constraints, our geometric framework allows local characterization, navigation, and exploration of the corresponding shape space. The figure shows a sample design (right) created using our method, starting from a flat circular mesh (left).*

Abstract

We present a general computational framework to locally characterize any shape space of meshes implicitly prescribed by a collection of non-linear constraints. We computationally access such manifolds, typically of high dimension and co-dimension, through first and second order approximants, namely tangent spaces and quadratically parameterized osculant surfaces. Exploration and navigation of desirable subspaces of the shape space with regard to application specific quality measures are enabled using approximants that are intrinsic to the underlying manifold and directly computable in the parameter space of the osculant surface. We demonstrate our framework on shape spaces of planar quad (PQ) meshes, where each mesh face is constrained to be (nearly) planar, and circular meshes, where each face has a circumcircle. We evaluate our framework for navigation and design exploration on a variety of inputs, while keeping context specific properties such as fairness, proximity to a reference surface, etc.

Keywords: shape space, manifold navigation, design exploration, computational differential geometry, constrained mesh

1 Introduction

In geometry processing, meshes are often specified by a collection of non-linear constraints, typically associated with mesh faces or edges. Exploring and navigating the corresponding shape space, i.e., the possible meshes sharing the same combinatorics as the input mesh while satisfying the constraints, are widely believed to be challenging. Even seemingly simple handle-driven deformations restricted to such shape spaces turn out to be challenging, and remain an active topic of research (see [Botsch et al. 2006; Kilian et al. 2007; Botsch and Sorkine 2008; Gal et al. 2009]).

In this paper, we propose a mathematical framework for the design and manipulation of non-linearly constrained meshes. Our approach is based on the exploration of an appropriate shape space as follows: Geometric models are mapped to points in a high-dimensional space $\mathbb{R}^D$, where the models that satisfy the constraints form a certain manifold $\mathcal{M} \subset \mathbb{R}^D$ (shape space). Modeling proceeds by navigating in the practically useful parts of the manifold $\mathcal{M}$, as prescribed by application specific quality measures. Such a manifold typically has high dimension and co-dimension, making it difficult to directly employ standard differential geometry concepts such as curvatures, especially in an efficient and computationally feasible manner. We locally approximate the manifold using tangent spaces and quadratically parameterized osculant surfaces, and propose how to computationally estimate the local curvature of the manifold to decide between the two representations.

We demonstrate the utility of our framework for two concrete example scenarios: (i) planar quad (PQ) meshes, i.e., meshes with each quad face being planar, and (ii) circular meshes, i.e., meshes with each quad face having a circumcircle. These meshes are attractive geometry representations for architectural freeform structures. Although various computational techniques have been proposed for creating such meshes, effective exploration of the associated design spaces remains largely unexplored (see [Ceccato et al. 2010] and references therein). Starting from a single PQ/circular mesh, we build the corresponding mesh manifold. Moving on the manifold allows us to discover *neighboring* PQ/circular meshes, while retaining aesthetic quality measures of the input model (see Figure 1). Here, we already point to the fact that planarity or circularity of faces is in practice subject to user-specified manufacturing tolerances. Our framework is capable of staying strictly within a given tolerance band.

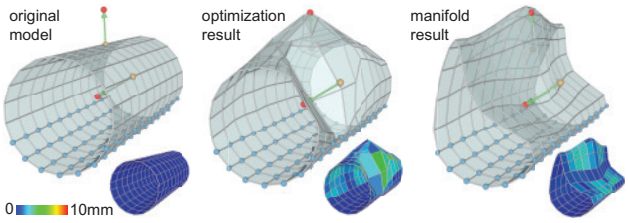


Figure 2: *Decoupling deformation and planarization is undesirable for shape design. Given a PQ mesh (left), the user prescribes a deformation using vertex handles, and the deformed mesh is planarized using an optimization approach [Liu et al. 2006]. The result can be unsatisfactory (middle). In contrast, our PQ mesh manifold exploration characterizes the (non-linearly constrained) design space, allowing direct design (right).*

In the context of mesh deformation, a naïve possibility is to manipulate a constrained mesh using a standard mesh deformation tool, and then re-optimize to restore the prescribed constraints. Unfortunately, due to the non-linearity of the constraints, a large deformation followed by subsequent optimization can significantly change the deformed model, thus making it challenging to warp the shape into desired forms (see Figure 2). An alternate solution is to take small deformation steps, interleaved with optimization, but the process is slow, cumbersome, difficult to control, and hampers the designer's work flow. More importantly, such an approach neither provides a good interface to restrict navigation to the implicitly prescribed shape space, nor does it enable exploration of good deformation directions while optimizing for desired quality measures.

1.1 Overview and contributions. Given a single input mesh along with a set of non-linear constraints (in terms of the mesh vertices), our goal is to explore other meshes with the same connectivity while respecting the prescribed constraints. We model the problem by mapping the meshes to points $\mathbf{x} \in \mathbb{R}^D$, where D is 3 times the number of deformable vertices. Each constraint defines a hypersurface in $\mathbb{R}^D$, and the intersection of all these hypersurfaces is our corresponding shape space, or mesh manifold, $\mathcal{M}$. For example, face planarity leads to the PQ mesh manifold. We locally navigate in $\mathcal{M}$ (in fact – due to tolerances – close to $\mathcal{M}$) with help of local approximations of $\mathcal{M}$. These are tangent spaces and quadratically parameterized surfaces having second order contact with $\mathcal{M}$ (Section 2). Further, we report effective theoretical and computational tools for estimating curvatures of such non-trivial spaces (i.e., spaces with high dim. normal spaces), and understand the trade-offs between tangent and osculant space navigation.

While any point of the shape space $\mathcal{M}$ represents a valid constrained mesh, only certain parts of $\mathcal{M}$ are desirable according to application specific quality measures, e.g., fairness of selected mesh polygons. For design exploration, it is important to be able to efficiently identify such useful parts of $\mathcal{M}$, and restrict navigation to such desirable regions. We enable this with the help of appropriate energy functions and their second order approximations that are *intrinsic* to $\mathcal{M}$ (Section 3). Eigen-analysis of the intrinsic Hessians of energy functions turns out to be a highly effective tool for the identification of the good parts of the shape space for subsequent exploration (see Figures 1, 9, 14, 16, 17, and 19). In Section 4, using the example of PQ meshes, we demonstrate our proposed framework for design exploration, optimization, and handle driven deformation with boundary conditions, while conforming to prescribed constraints. Interestingly, the local approximants also provide a natural way to access the relative difficulty of deforming various parts of the input model (see Figure 11). In Section 5, we present exploration results for circular meshes and explorative design examples starting from flat meshes (see Figures 1 and 18).

Our main contribution, in the context of geometry processing, is a computationally feasible yet mathematically precise formulation that allows navigation and exploration of non-linearly constrained shape spaces, which are typically of high dimension and co-dimension. Our focus is on the access to the variety of feasible designs meeting the specified constraints and *not* on the solution of a single constrained optimization problem. In the context of architectural geometry, we unify two traditionally separate phases in freeform design, namely, (i) shape design and (ii) rationalization in view of the actual fabrication.

1.2 Related work. Although there is little prior work in the area of design and shape exploration of nonlinearly constrained geometric models, we briefly present relevant research efforts.

PQ meshes appeared first in discrete differential geometry (cf. [Bobenko and Suris 2008]) as discrete counterparts of so-called conjugate curve networks, in particular of the network

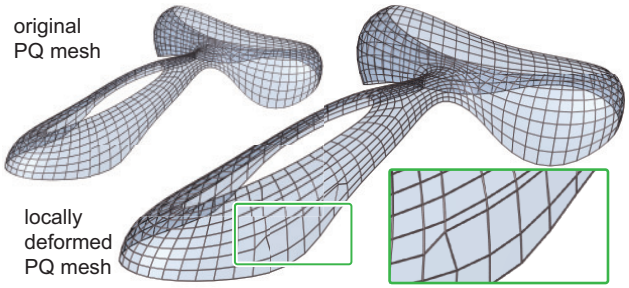


Figure 3: *Very local deformations of PQ meshes are possible while preserving face planarity (see also Hoffmann [2011]). Such an approach, however, quickly destroys the aesthetics of the mesh, which being visible in the beam layout forms an integral design element.*

of principal curvature lines. The importance of PQ meshes for freeform architecture led to the development of computational tools which are based on nonlinear optimization [Liu et al. 2006; Pottmann et al. 2007], since theoretically existent direct constructions turned out to be either unstable or impractical. The difficulty in designing a PQ mesh is rooted in the fact that such a mesh is strongly guided by the curvature behavior of an underlying smooth surface. The layout of a PQ mesh on a given design surface S basically amounts to the design of a conjugate curve network on S [Zadravec et al. 2010]. So far, direct modeling of PQ meshes and related structures such as developable strip models has been limited to the interleaved subdivision and optimization approach [Liu et al. 2006; Pottmann et al. 2008]. Only simple modification tools have been presented such as tranformations based on natural invariance properties (e.g., projective, Möbius, Laguerre and Lie transformations) or on mesh parallelism [Pottmann et al. 2007]. The latter already indicated the difficulties in generating useful designs from existing ones. This has further been confirmed by recent work on very local PQ mesh deformations that are directly performed within a projective geometric framework [Hoffmann 2011], but in our experience are not suitable as a basis of a design modification tool (see Figure 3).

Shape deformation. Significant research efforts have been devoted towards manipulation of triangle meshes using various linear and non-linear formulations (see survey [Botsch and Sorkine 2008]), or in presence of interrelations across feature curves [Gal et al. 2009]. In case of triangles, however, planarity is trivially satisfied. Other approaches include isometric, as-rigid-as possible or conformal deformations (see [Gu and Yau 2008; Lipman et al. 2008] and the references therein). Unlike such methods, in addition to deformations, we also want to support optimization and explorations restricted to the implicitly prescribed shape space.

Morphable models. In the context of character animation and modeling, researchers have employed statistical tools to learn principal modes of model variation. Starting from a representative template and a collection of aligned models, morphable model learning techniques have been effectively used for faces [Blanz and Vetter 1999], human bodies, and animation poses characterized as deformation gradients [Sumner et al. 2005]. In the context of shape analysis, Huang et al. [2009] use eigen-modes of surface Hessians to learn useful shape segmentations. Kilian et al. [2007] propose Riemannian metrics for construction of useful shape spaces for design and modeling of geometric shapes. None of these methods, however, can be simply extended to characterize and explore the space of nonlinearly constrained geometric models, which is the goal of our work. Note that we explore the shape space as defined by a *single* non-linearly constrained mesh, rather than a collection of meshes. Thus our research is fundamentally different from work in machine learning, where shape manifolds are computed from input poses.

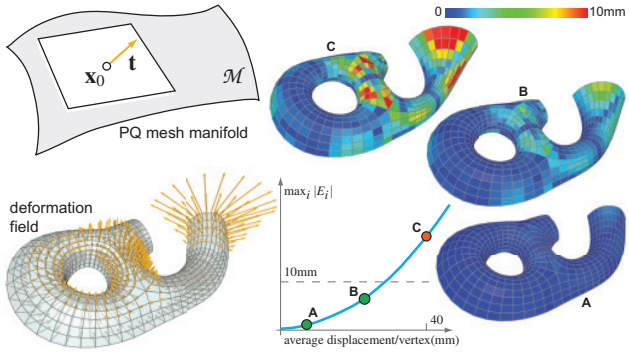


Figure 4: *A given PQ mesh $\mathbf{x}_0$ is a point on the PQ mesh manifold $\mathcal{M}$ of all PQ meshes which share the same connectivity. Vectors $\mathbf{t}$ lying in the tangent space at $\mathbf{x}_0$ represent deformation fields (bottom-left) on the original mesh that preserve face planarity up to first order. Typically, we can take non-trivial steps in such tangent directions before the deviation from planarity $\max_i |E_i(\mathbf{x})|$ exceeds fabrication limits, e.g., 10mm/m for glass panels (average panel length is 1m in all examples). In this example, mesh A and mesh B meet fabrication bounds, but mesh C does not (all illustrations show computed results).*

2 Constrained Mesh Manifolds

Given a single constrained mesh, our goal is to characterize, navigate, and explore the space of meshes sharing the same connectivity, while maintaining the prescribed constraints, within a tolerance margin. In this section, we formalize the notion of such *shape spaces* (mesh manifolds), and derive their local tangent and osculant approximations that subsequently form the basis of various exploration metaphors.

Starting from an input mesh (in $\mathbb{R}^3$), 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 $\mathbf{d} \in \mathbb{R}^D$ is a deformation field on the mesh producing the new mesh $(\mathbf{x} + \mathbf{d})$. A useful distance measure between any two meshes $\mathbf{x}_1, \mathbf{x}_2$ is defined as $d(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|$, i.e., as the Euclidean distance of the corresponding points in $\mathbb{R}^D$, which can be interpreted in $\mathbb{R}^3$ via $d^2(\mathbf{x}_1, \mathbf{x}_2) = \sum_i (v_{i,1} - v_{i,2})^2$.

Let $E_i(\mathbf{x}) = 0$ denote the i-th constraint imposed on a mesh $\mathbf{x}$, where $|E_i(\mathbf{x})|$ shall be a practically meaningful deviation measure. We assume to have m constraints, which will mostly be non-linear. It is possible to use any constraint function $E_i(\mathbf{x})$ with our formulation as long as gradients and Hessians are well defined. The corresponding shape space $\mathcal{M}$ 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 hypersurfaces $\Gamma_i = \{\mathbf{x} \in \mathbb{R}^D \mid E_i(\mathbf{x}) = 0\}$, $i = 1, \ldots, m$. Hence, $\mathcal{M}$ is in general of dimension $D - m$ and codimension m (e.g., 600-1000 dimensions and 300-500 codimensions in our examples on PQ meshes).

We illustrate our framework on the specific example of planar quad meshes, where the non-linear constraints are the (deviation from) planarity measure associated with each face (in Section 5, we also investigate circular mesh manifolds). Specifically, in this work, we use the signed distance between the face diagonals as the planarity measure $E_i(\mathbf{x})$ for any quad face f_i. The definition directly correlates to approximation margins typically allowed by various fabrication technologies. For example, for glass panels of dimensions 2m×2m, a diagonal deviation margin up to $10 - 20$mm is considered allowable. In practice, such near-planar panels are obtained by cold bending of the panels, and do not require custom molds.

Tangent space. 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 hypersurfaces Γ_i.

The normal of any Γ_i at $\mathbf{x}_0$ is along the gradient $\nabla E_i(\mathbf{x}_0)$, and thus 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 $D - m$. In practice we remove any dependencies by computing a normal space basis using SVD. The tangent space to the constrained mesh manifold $\mathcal{M}$ is a linear space attached to the point $\mathbf{x}_0$ containing tangent vectors $\mathbf{t}$ orthogonal to each of $\nabla E_i(\mathbf{x}_0)$ and is characterized as,

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

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 that satisfies prescribed constraints up to first order (see Figure 4).

Osculant. Due to the non-linearity of constraints, tangent space navigation may allow only small steps before one of the deviation measures $|E_i(\mathbf{x})|$ exceeds the prescribed tolerance. Hence we seek a better approximation. A simple option to obtain a 2nd order approximation is to first compute the osculating paraboloid (2nd order Taylor approximation) for each of the constraint hypersurfaces Γ_i, and then compute their intersection. Unfortunately, the algebraic computation of the intersection surface is cumbersome since the intersection surface can be of order 2^m. Instead, we derive a better approximation in the form of a locally approximating surface sharing second-order contact with the mesh manifold $\mathcal{M}$. Let this approximating surface, henceforth simply referred to as the *osculant*, be parameterized over the tangent space $\mathrm{T}_{\mathcal{M}}(\mathbf{x}_0)$ as,

$$\mathbf{S}(\mathbb{u}) = \mathbf{x}_0 + \sum_{i=1}^{D-m} u_i \mathbf{e}_i + \frac{1}{2} \sum_{j=1}^{m} (\mathbb{u}^T \cdot A_j \cdot \mathbb{u}) \mathbf{n}_j. \quad (2)$$

The parameterization of our surface exhibits quadratic forms with symmetric $(D - m) \times (D - m)$ matrices $A_j, j = 1, \ldots, m$ for each of the m coordinates in the normal space. Unlike (smooth) surface points in $\mathbb{R}^3$ with unique surface normals, any point $\mathbf{x}_0$ on manifold $\mathcal{M}$ has a normal space of dimension m (see Figure 5).

In order for $\mathbf{S}(\mathbb{u})$ to be an osculant to $\mathcal{M}$, it should have second order contact at $\mathbf{x}_0$ with *each* of the hypersurfaces Γ_i : $E_i(\mathbf{x}) = 0$. The second order Taylor expansion of E_i at $\mathbf{x}_0$ is

$$\begin{aligned} E_i(\mathbf{x}) \;&=\; E_i(\mathbf{x}_0) + \nabla E_i^T \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_0)^T \cdot H_i \cdot (\mathbf{x} - \mathbf{x}_0) \\ &\quad + o(\|\mathbf{x} - \mathbf{x}_0\|^2), \end{aligned}$$

where H_i denotes the Hessian of E_i evaluated at $\mathbf{x}_0$. Substituting $\mathbf{S}(\mathbb{u})$ into the above form, we have

$$\begin{aligned} E_i(\mathbb{u}) \;&=\; E_i(\mathbf{x}_0) + \frac{1}{2} \sum_{j=1}^{m} (\nabla E_i^T \cdot \mathbf{n}_j)(\mathbb{u}^T \cdot A_j \cdot \mathbb{u}) \\ &\quad + \frac{1}{2} \sum_{p=1}^{D-m} \sum_{q=1}^{D-m} (\mathbf{e}_p^T \cdot H_i \cdot \mathbf{e}_q) u_p u_q + o(\|\mathbb{u}\|^2), \end{aligned} \quad (3)$$

where we used the orthogonality of gradient vectors ∇E_i^T and tangent basis vectors $\mathbf{e}_j$. For the surface approximant $\mathbf{S}(\mathbb{u})$ to have second order contact with $E_i(\mathbf{x}) = 0$, each second order term $u_p u_q$ should vanish, i.e.,

$$\sum_{j=1}^{m} (\nabla E_i^T \cdot \mathbf{n}_j) A_j^{p,q} + \mathbf{e}_p^T \cdot H_i \cdot \mathbf{e}_q = 0, \;\forall p, q = 1, ..., D - m. \quad (4)$$

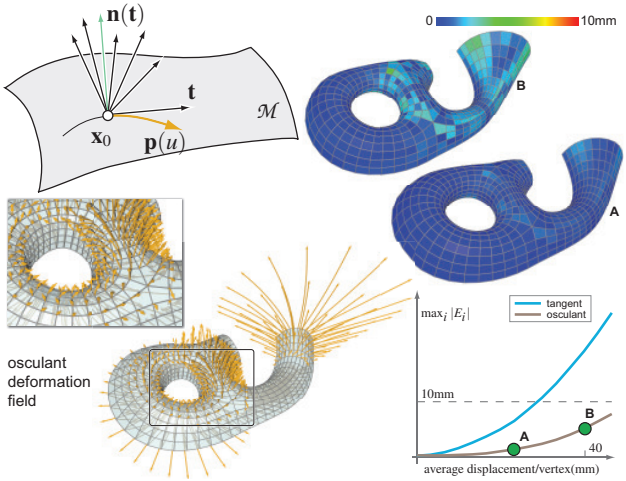


Figure 5: *At any point $\mathbf{x}_0$ of a PQ mesh manifold $\mathcal{M}$ the normal space is spanned by the face planarity gradient vectors. For any tangent line $u\mathbf{t}$, the local* osculant $\mathbf{S}$ *to the manifold $\mathcal{M}$ defines a unique vector $\mathbf{n}(\mathbf{t})$, shown in green, in the normal space which is the axis of an osculating parabola $\mathbf{p}(u)$. Moving along $\mathbf{p}(u)$ amounts to vertices tracing curved paths (parabolae), resulting in better planarity preservation (compare with Figure 4).*

where $A_j^{p,q}$ is the matrix element of A_j at the p-th row and the q-th column. Considering constraints E_i of all m faces, we arrive at the following linear system for $A_j^{p,q}$ for $j = 1, ..., m$

$$\begin{bmatrix} \nabla E_1^T \mathbf{n}_1 & \nabla E_1^T \mathbf{n}_2 & \dots & \nabla E_1^T \mathbf{n}_m \\ \nabla E_2^T \mathbf{n}_1 & \nabla E_2^T \mathbf{n}_2 & \dots & \nabla E_2^T \mathbf{n}_m \\ \vdots & \vdots & \ddots & \vdots \\ \nabla E_m^T \mathbf{n}_1 & \dots & \dots & \nabla E_m^T \mathbf{n}_m \end{bmatrix} \begin{bmatrix} A_1^{p,q} \\ A_2^{p,q} \\ \vdots \\ A_m^{p,q} \end{bmatrix} = - \begin{bmatrix} \mathbf{e}_p^T H_1 \mathbf{e}_q \\ \mathbf{e}_p^T H_2 \mathbf{e}_q \\ \vdots \\ \mathbf{e}_p^T H_m \mathbf{e}_q \end{bmatrix}. \tag{5}$$

Note that the left matrix is independent of p and q, and hence can be reused across p, q. By solving the above linear system, we can compute the matrix element of the p-th row and q-th column for each A_j, and thus obtain the osculant $\mathbf{S}(\mathbb{u})$.

Approximate constrained meshes. Typically, a constrained mesh is obtained via an optimization approach, e.g., a PQ mesh can be created using the algorithm proposed by Liu et al. [2006]. Such a mesh, however, is only approximate as the face constraints are satisfied within a tolerance margin. Thus, the corresponding point $\mathbf{x}_0$ is not exactly on the constrained mesh manifold $\mathcal{M}$, but close to it. The osculant surface then is an osculant to a slightly shifted version of the manifold $\mathcal{M}$ in controlled distance to $\mathcal{M}$. This is not an offset in the usual sense. Consider a curve $\mathcal{M}$ in 3-space defined as intersection of two surfaces, e.g., $E_1(\mathbf{x}) = 0$, $E_2(\mathbf{x}) = 0$. In our approach, we would work on a nearly parallel curve, defined as intersection of two surfaces $E_i(\mathbf{x}) = \varepsilon_i$ (for very small ε_i). In contrast, an offset would be a pipe surface around $\mathcal{M}$ (see Figure 6).

Alternate formulation. A much simpler mathematical formulation of constrained meshes would be to combine all the constraint scores into a single deviation measure $\tilde{E}(\mathbf{x}) := \sum E_i^2(\mathbf{x})$ (see [Liu et al. 2006] for PQ meshes). The corresponding level sets $\tilde{E}(\mathbf{x}) = \varepsilon$ for small ε are pipes forming boundaries of tubular neighborhoods (see Figure 6). This approach has a number of disadvantages: (i) it is hard to guarantee maximum tolerances for each of the constraints; (ii) for such a measure $\tilde{E}$ the gradient $\nabla\tilde{E}(\mathbf{x})$ vanishes for meshes on the exact constrained mesh manifold; (iii) walking on any level set of $\tilde{E}(\mathbf{x})$ has the disadvantage that there is an $(m-1)$-dimensional subspace of directions along which we make insignificant progress. Note that our osculant surface is a

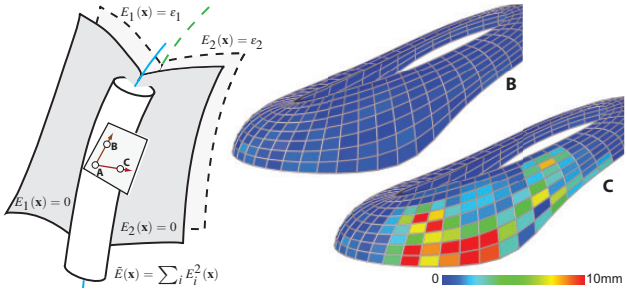


Figure 6: *In practice, instead of exploring the exact PQ mesh manifold (blue curve), we work with the intersection (dotted green curve) of hypersurfaces which are level sets to very small values ε_i of the face planarity measures E_i. Alternately, one can use the ε-level set (pipe-surface) of a combined energy $\tilde{E} = \sum E_i^2$. However, at any point (A) on such a level set, there can be an $(m-1)$-dimensional space of undesirable directions (e.g., red vector) along which little is gained in terms of navigating the (approx.) PQ mesh manifold.*

better approximant than can be obtained using a second order analysis of $\tilde{E}$: Our osculant is in at least *second order contact* with the corresponding level set $\tilde{\Gamma}$ of $\tilde{E}$, and it captures only those directions that lead to significant progress when walking on $\tilde{\Gamma}$.

Osculating parabolae. Any straight line $u\mathbf{t}$ through the origin of the parameter domain is mapped via our parameterization (Equation 2) of the osculant to an *osculating parabola* $\mathbf{p}(u)$, i.e., a parabola with vertex at $\mathbf{x}_0$ sharing a second order contact with $\mathcal{M}$ at $\mathbf{x}_0$. The surface $\mathbf{S}(\mathbb{u})$ is formed by all such osculating parabolae. This is a generalization of the familiar osculating paraboloid of a hypersurface. However, in the latter case all osculating parabolae have the same axis (the unique surface normal), while in our case the axis directions are varying. Each tangent direction $\mathbf{t}$ determines the normal which is suitable as the axis of an osculating parabola (see Figure 5). The plane of the osculating parabola contains a manifold normal and hence it has second order contact with a *geodesic* in $\mathcal{M}$ passing through $\mathbf{x}_0$ with tangent $\mathbf{t}$. Also note that moving along an osculating parabola $\mathbf{p}(u)$, we obtain a (constrained) mesh deformation where all vertices move along parabolae (described by those 3 coordinates of $\mathbf{p}$ which represent the corresponding vertex).

Curvatures and generalized Dupin indicatrix. The study of curvatures for manifolds with high codimension (m in our case) is considered to be complicated and cumbersome (see [Schouten and Struik 1931] pp. 92 onwards), and we are unaware of any previous work that effectively maps to a computational framework. In

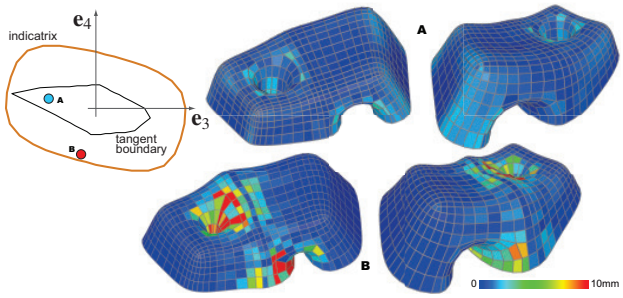


Figure 7: *Generalized Dupin indicatrix (in brown) in a 2D subspace spanned by third and fourth eigen-directions of the intrinsic Hessian on combined fairness and orthogonality energies. The 10mm-tangent boundary (in black) has similar behavior as the indicatrix since the PQ mesh manifold is less curved along the longer indicatrix direction. Hence, although meshes A and B are equidistant from the origin, mesh A has better planarity behavior.*

fact, our osculants provide this framework, which shall be briefly addressed below. The osculant $\mathbf{S}(\mathfrak{u})$ is formed by the osculating parabolae to all directions $\mathbf{t}$ in the tangent space $\mathrm{T}_{\mathcal{M}}(\mathbf{x}_0)$. Each such parabola has $\mathbf{x}_0$ as its vertex, and in a local (x,y)-frame with the tangent $\mathbf{t}$ as the x-axis and the y-axis lying in the normal space of $\mathcal{M}$, the parabola can be expressed as $y = (\kappa/2)x^2$. Hence, $|\kappa|$ is a normal curvature $\kappa_n(\mathbf{t})$ of $\mathcal{M}$ in the respective tangent direction $\mathbf{t}$. Since the manifold has a codimension m (≥ 2), assigning sign to a normal curvature is meaningless. The points of any such parabola at distance $1/2$ from the x-axis have x-coordinates $\pm\sqrt{1/|\kappa|}$, providing a measure of how quickly the manifold pulls off the tangent plane in direction $\mathbf{t}$.

This leads to a *generalized Dupin indicatrix* as a characterization of the local shape of the underlying manifold. We collect all the points of the osculant $\mathbf{S}(\mathfrak{u})$ at a distance $1/2$ from the tangent space $\mathrm{T}_{\mathcal{M}}(\mathbf{x}_0)$ and project them down orthogonally to $\mathrm{T}_{\mathcal{M}}(\mathbf{x}_0)$, which is also the parameter space of the osculant. We obtain a radial diagram for $\sqrt{1/\kappa_n(\mathbf{t})}$, a natural extension of the familiar Dupin indicatrix, which is used for surfaces with codimension one [do Carmo 1976]. The generalized indicatrix is characterized as

$$\Psi := \{\mathfrak{u} \;\mid\; \sum_{j=1}^{m} (\mathfrak{u}^T \cdot A_j \cdot \mathfrak{u})^2 = 1\}. \tag{6}$$

This is a centrally symmetric algebraic hypersurface Ψ of order 4 in the tangent space (for a 2D slice of an indicatrix, see Figure 7). Extremal curvatures belong to those points on Ψ, whose normals pass through the origin, i.e., are characterized by $\nabla G(\mathfrak{u}) = \lambda\mathfrak{u}$, where $G(\mathfrak{u}) = \sum_j (\mathfrak{u}^T \cdot A_j \cdot \mathfrak{u})^2$.

3 Functions on Constrained Mesh Manifolds

The osculant locally approximates the mesh manifold $\mathcal{M}$, and enables a parameterized navigation in close proximity to the input mesh $\mathbf{x}_0$. Often such constrained mesh families come with associated quality measures based on application specific functions. As a result, large parts of $\mathcal{M}$ do not represent desirable meshes, and hence are not useful for design and exploration. For example, in the context of a PQ mesh manifold, for a 10×10 neighborhood on a real model with about a 200-dimensional tangent space, we found only about 15-20 dimensions being useful once we restrict our inspection to mesh polygons with good fairness, a desirable aesthetic property.

In this section, we abstract mesh aesthetics and other mesh properties in the form of functions $F(\mathbf{x})$ defined on the embedding space $\mathbb{R}^D$ (not just restricted to the mesh manifold) and derive corresponding intrinsic Hessian approximations to study the behavior of $F(\mathbf{x})$ restricted to the osculant. Subsequently, this allows us to restrict navigation to only the *good* regions of the mesh manifold as (implicitly) prescribed by the chosen functions.

Second order intrinsic derivative of $F(\mathbf{x})$

The second order Taylor expansion of $F(\mathbf{x})$ at mesh $\mathbf{x}_0$ is

$$\begin{aligned} F(\mathbf{x}) \;&=\; F(\mathbf{x}_0) + \nabla F^T \cdot (\mathbf{x}-\mathbf{x}_0) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_0)^T \cdot H_F \cdot (\mathbf{x}-\mathbf{x}_0) \\ &\quad + o(\|\mathbf{x}-\mathbf{x}_0\|^2), \end{aligned} \tag{7}$$

where H_F denotes the Hessian 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 be expressed in terms of parameter vector $\mathfrak{u}$ as

$$F(\mathbf{x}_0) + \sum_{i=1}^{D-m} (\nabla F^T \cdot \mathbf{e}_i) u_i + \frac{1}{2}\mathfrak{u}^T \cdot H_F^r \cdot \mathfrak{u} + o(\mathfrak{u}^2), \tag{8}$$

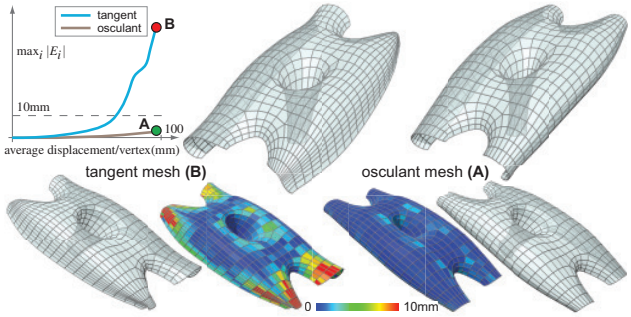


Figure 8: *Tangent space navigation with fairness assessed using the reduced Hessian (Equation 8) vs. navigation on the osculant with fairness assessed using the intrinsic Hessian (Equation 10).*

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

Restriction to the tangent space, however, is quite misleading since we ignore the curvature of the underlying manifold $\mathcal{M}$. The local osculant $\mathbf{S}(\mathfrak{u})$ allows us to correctly analyze the function behavior directly on the manifold. We define the function $f(\mathfrak{u}) = F(\mathbf{S}(\mathfrak{u}))$ on the tangent space (parameter domain of the osculant) and obtain its second order approximation at $\mathbf{x}_0$ ($\mathfrak{u} = 0$) by inserting the parameterization (Equation 2) of the osculant into the Taylor approximation (Equation 7) of F; we get

$$f(\mathfrak{u}) = F(\mathbf{x}_0) + \sum_{i=1}^{D-m} (\nabla F^T \cdot \mathbf{e}_i) u_i + \frac{1}{2}\mathfrak{u}^T \cdot H_F^I \cdot \mathfrak{u} + o(\mathfrak{u}^2). \tag{9}$$

Here, the quadratic terms are described by the *intrinsic* Hessian at $\mathbf{x}_0$ given by

$$H_F^I = H_F^r + \sum_{j=1}^{m} (\nabla F^T \cdot \mathbf{n}_j) A_j. \tag{10}$$

Recall that any tangent $\mathbf{t}$ gets mapped onto the osculant $\mathbf{S}(\mathfrak{u})$ as an osculating parabola having second order contact with the geodesic in $\mathcal{M}$ passing through $\mathbf{x}_0$ in direction $\mathbf{t}$. Thus, our second directional derivatives obtained with the Hessian H_F^I can also be seen as second directional derivatives of $f(\mathfrak{u})$ along geodesics. This is a completely intrinsic formulation, and hence we refer to H_F^I as the intrinsic Hessian. Figure 8 demonstrates the importance of analyzing the function directly on the osculant manifold as opposed to a tangent space approximation using the reduced Hessian, especially in regions of high curvature (see discussion of Dupin indicatrix).

Remark: The intrinsic Hessian equals the reduced Hessian in two very special cases: (i) the gradient of F is tangential to $\mathcal{M}$, i.e., all the directional derivatives of F in directions orthogonal to $\mathcal{M}$ vanish; in particular, ∇F may be zero meaning that $\mathbf{x}_0$ is a stationary point of F or, (ii) $\mathcal{M}$ has a *flat* point at $\mathbf{x}_0$, i.e., all the matrices A_j vanish.

We now describe various aesthetic and practical mesh quality measures and then demonstrate their typical applications to design exploration (Section 5).

Fairness energies. Away from extraordinary vertices, a mesh is naturally decomposed into families of polylines. These are three families for a triangle mesh and two for a quad mesh. In architecture, even for a hex mesh one would select sequences of vertices which should form aesthetically pleasing polylines. Thus, we now assume that we have selected polylines P_k, $k = 1, \ldots, K$, (with vertices $v_{k,j}$) about whose fairness we care.

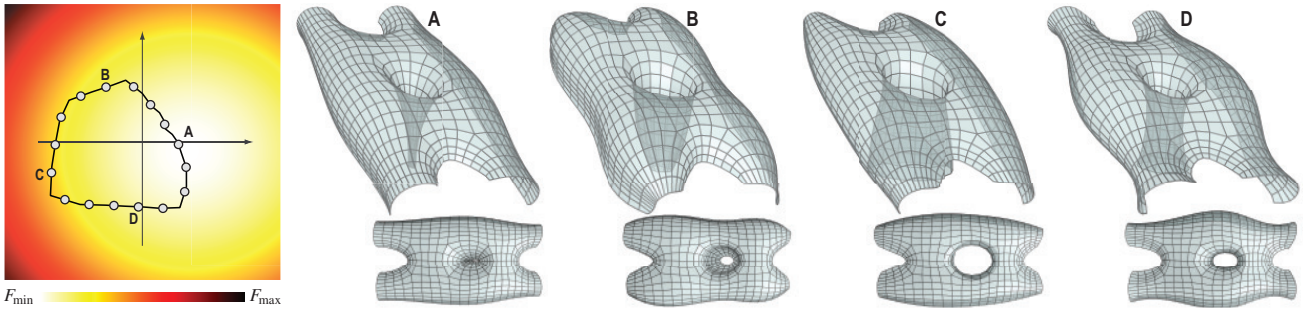


Figure 9: *Spectral analysis of the intrinsic Hessian of energy functions helps to identify good meshes in the neighborhood of an input PQ mesh. The user can interactively navigate a parameter plane T^2, while we show the corresponding meshes (points on the osculant). The boundary of* $\max_i |E_i| = 10mm$ *is shown on T^2 for guidance. Regular sampling of the boundary polygon provides a quick overview of the exposed design space (see accompanying demo for examples in $d = 3$). See Figure 11-second row for the starting model.*

We use two different types of fairness energies. The first one is based on second order differences (see also [Liu et al. 2006]), and for a single polyline P_k, reads

$$f^{(2)}_{k,fair}(\mathbf{x}) := \sum_j [\Delta^2(v_{k,j*}) - \alpha\Delta^2(v^0_{k,j*})]^2. \tag{11}$$

Here, $\Delta^2(v_{k,j*}) = v_{k,j-1} - 2v_{k,j} + v_{k,j+1}$, and $\Delta^2(v^0_{k,j*})$ is similarly defined for the original mesh $\mathbf{x}_0$. Parameter $\alpha = 0$ denotes *absolute* fairness, and $\alpha = 1$ denotes fairness *relative* to the input mesh $\mathbf{x}_0$.

The second type of fairness is based on the third order differences,

$$f^{(3)}_{k,fair}(\mathbf{x}) := \sum_j [\Delta^3(v_{k,j*}) - \alpha\Delta^3(v^0_{k,j*})]^2. \tag{12}$$

with $\Delta^3(v_{k,j*}) = v_{k,j-1} - 3v_{k,j} + 3v_{k,j+1} - v_{k,j+2}$.

The final fairness energy of a mesh is the sum of energies of the selected polygons. Our mesh energies constitute just examples. Depending on the application, other selections may be preferable.

Orthogonal and tangential energies. Deformation of the original mesh in direction mainly orthogonal to the reference surface results in large visible shape changes, while directions tangential to the reference surface lead to self-slippage and vertex movement on the underlying surface. Both types are of interest, and are captured by respective energy formulations,

$$\begin{aligned} F_{ortho}(\mathbf{x}) &:= \sum_i \|n^0_i \times (v_i - v^0_i)\|^2 \\ F_{tang}(\mathbf{x}) &:= \sum_i \|n^0_i \cdot (v_i - v^0_i)\|^2, \end{aligned} \tag{13}$$

where n^0_i is the surface normal at the current mesh vertex v^0_i.

Deviation from reference surface. Sometimes it is desirable to stay close to the original reference surface. To this end, we introduce the closeness energy

$$F_{close} = \sum_{i,j} \|v_{i,j} - v^*_{i,j}\|^2, \tag{14}$$

where $v^*_{i,j}$ denotes the closest point on the reference mesh from $v_{i,j}$. Note that, instead of a point-to-point distance measure, one can also use a point-to-plane variant.

Combined energy functions. The function can be any energy we are interested in like the ones presented above. These energies F_i-s can be combined into a single energy $F = \sum_i \lambda_i F_i$ for a specific task. Equation 10 for the intrinsic Hessian takes the form

$$H^I(F) = \textstyle\sum_i \lambda_i H^I(F_i), \tag{15}$$

with desired weights λ_i. We precompute the individual Hessians, and then allow the user to appropriately combine the various energies in a shape space exploration session.

4 Mesh Manifold Exploration

In this section, we describe ways to explore mesh manifolds using tangent spaces and osculants, coupled with the various energy functions. In our illustrations, we use PQ meshes, and in most cases, we employ a combination of (third order) absolute fairness and orthogonal energies to explore the space of desirable meshes. By default, chosen energies are given equal weights.

Subspace (sub-osculant) exploration. Spectral analysis of the intrinsic Hessian identifies the directions of locally extremal second directional derivatives of the chosen energy, with the corresponding eigenvalues being these derivatives. Hence, it serves to quickly identify the good subspaces (passing through $\mathbf{x}_0$) in the tangent space, and hence on the osculant. We restrict navigation to that subspace of the tangent space $\mathrm{T}_{\mathcal{M}}(\mathbf{x}_0)$ which is spanned by the eigenvectors to the lowest few eigenvalues of the intrinsic Hessian (lowest 5% in our examples). For example, using only a pair of eigen-directions, i.e., navigating in the corresponding 2D parameter plane T^2, the user can directly explore only meshes with good aesthetic behavior.

For PQ mesh manifold exploration, we provide guidance as follows: (i) We compute a boundary curve on the plane T^2 around the origin $\mathbf{x}_0$ to indicate where the planarity bound crosses an acceptable threshold. We approximate the boundary using a polygon constructed by marching along equally spaced radial rays and searching for the tolerance crossings. (ii) We color code the parameter space based on the considered energy. Note that the user navigates in the tangent plane T^2, but in fact directly explores meshes on that part of the precomputed osculant (sub-osculant) which is parameterized over T^2 (see Figure 9 and supplementary demo). The exploration being interactive is suited for contraint-aware shape design.

Regular sampling of the sub-osculant. Although the above metaphor extends to exploration in $d = 3$, direct 3D-parameter space navigation can be difficult. Instead, we automatically compute the range of various models in this space using the boundary polytope P of the region with admissible planarity measure. We let the user simply browse through the space of variations, and then locally navigate the parameter space for refinements (see supplementary demo). We start by sampling the polytope P, and then relax the samples using Lloyd iterations to regularly distribute the points, i.e., meshes, on the corresponding part of the sub-osculant.

Handle driven exploration. Traditional handle driven shape manipulation is easily incorporated in our framework. Suppose the

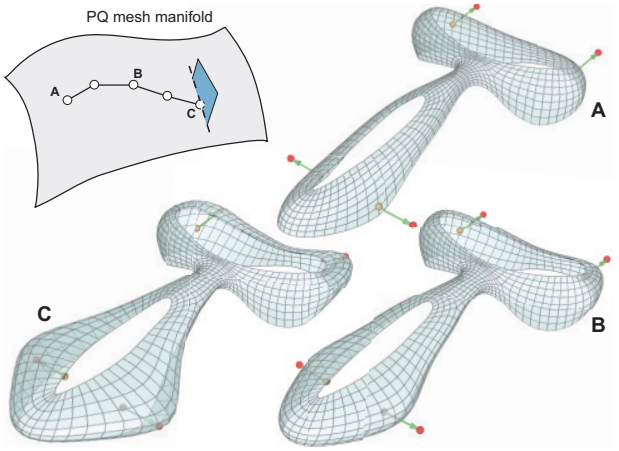


Figure 10: *Starting from an initial mesh (A), the user prescribes four vertex constraints (in red), which correspond to a subspace in $\mathbb{R}^D$ (blue region). We take steps by solving quadratic optimization with linear constraints, thus traversing along the PQ mesh manifold to reach the target region, while minimizing a chosen energy functions. Stepping through the intermediate meshes, e.g., B, we reach the final mesh (C) (for all meshes $\max_i |E_i| < 10mm$).*

user prescribes a vertex deformation $v_j \to v'_j$ and an influence region in which m constraints are active.

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

$$\min_{\mathbf{t}} F(\mathbf{x}_0 + \mathbf{t}) \quad \text{such that,}$$
$$\nabla E_i^T \cdot \mathbf{t} = 0, \ \forall\ i = 1, \ldots, m; \quad t_j = v'_j - v_j, \qquad (16)$$

where t_j denotes the j-th 'coordinate' of $\mathbf{t}$ (actually 3 coordinates, associated with vertex v_j). While Equation 16 adds $m+3$ constraints, the energy function is optimized with the remaining degrees of freedom. For the fairness energy, the above form is quadratic and convex, and minimization reduces to solving a sparse linear system [Nocedal and Wright 2006]. In order to handle larger deformations, we walk using small steps in the tangent space, re-estimate the normal space in each step, and achieve the deformation in several steps. Simultaneously conferring to multiple displacements amounts to adding multiple positional constraints. In order to maintain interactivity for large deformations, we use the tangent approximation, to avoid the costly update of the osculant.

Note that we reach, in a greedy way, a possible deformation meeting the input, but this is not the only solution (see Figure 10 and the accompanying demo for local deformations). Additionally, we can use other (presented) exploration tools to explore further possibilities, while keeping v_j fixed.

The feasibility of a local influence region depends on the nature of the constraints. Globally effective constraints (as for example planarity of polygons extending over large portions of the mesh) may contradict the choice of a local influence region. In such cases and for highly constrained meshes, handle-driven deformation may not be the right tool to use.

Stiffness estimation. Constraints may make sections of a model difficult to manipulate, leaving little freedom for deformations. We illustrate this at hand of PQ meshes, but the basic ideas apply in a much more general setting. Given an input model, we compute both *local* and *global* stiffness maps to provide the user a sense of design space restrictions. For local stiffness, at each vertex v_j, with a fixed neighborhood size (8-rings in our experiments), we estimate how far v_j can be displaced along its normal direction before either

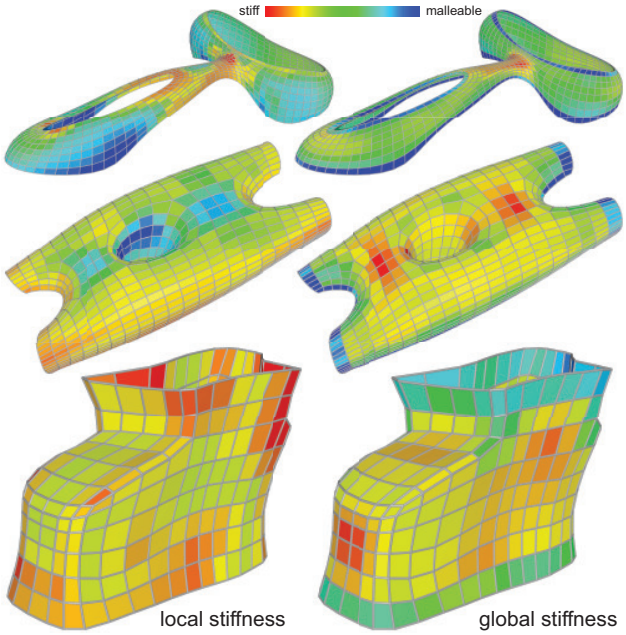


Figure 11: *We compute* local*, 8-ring neighborhoods in this example, and* global *stiffness estimates for input models to provide the user with relative estimates of flexibility in the neighboring design space (see Section 4). Intuitively, the Opus model (bottom) is the most restrictive given the large near-planar regions.*

(i) planarity bound $\max_i |E_i|$ is reached, or (ii) the combined fairness and orthogonality energy becomes unacceptable. For global stiffness maps, we use the intrinsic Hessians H_F^I computed for the prescribed energy function. Let $\{\mathbf{h}_1, \mathbf{h}_2, \ldots\}$ be the lowest eigenvectors (lowest 5% in our experiments) of H_F^I expressed in $\mathbb{R}^D$. Then we assign a stiffness score at vertex v_j as $\sum_i \|(\mathbf{h}_i)_j\|$, where $(\mathbf{h}_i)_j$ denotes the j-th component of $\mathbf{h}_i$. Figure 11 shows local and global stiffness maps for three models. The models are sorted top down based on their relative stiffness measure, the most restrictive model being the last. Note that according to the global measure, the open boundaries are more malleable — the same behavior is observed in Figures 4 and 5 by the long vectors at the periphery.

Newton relaxation. We have a second order approximation of the energy F while moving on the osculant (Equation 9). When the intrinsic Hessian is positive definite, we can hope to be sufficiently close to the minimizer, and thus take a Newton step. Here, we greatly benefit from having a parametrization of the osculant, and not requiring side conditions to stay on the mesh manifold. Thus, we can improve the considered energy functions, while maintaining the quality of constraint satisfaction (see Figure 12). This is difficult with a penalty-based optimization. Note that except in Figure 12, we skip this optimization to show only the basic results.

5 Further Examples and Discussion

We tested our framework on a variety of designs. Please refer to the supplementary material for navigation and exploration in PQ and circular mesh manifolds on selected models. In this section, we demonstrate how commonly used architectural design and fabrication constraints can be easily integrated into our framework.

Avoiding obstacles. Design spaces are often restricted by space constraints and boundary conditions. For example, a new building has to respect space constraints as dictated by existing structures. Specifically, the designed surfaces should avoid specified obstacle

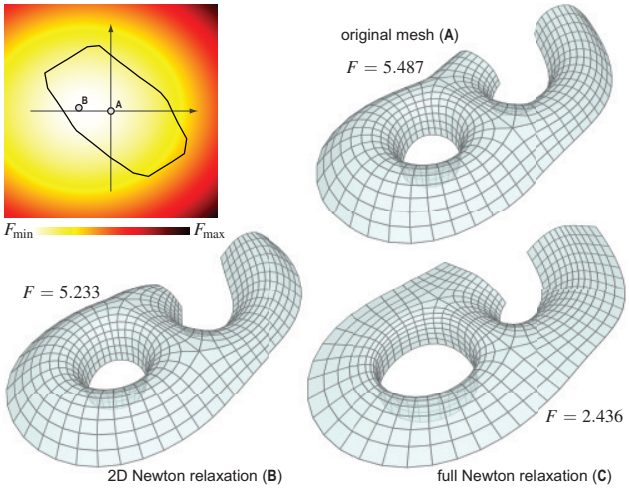


Figure 12: *At a mesh* **A**, *if the intrinsic Hessian* H_F^I *of the prescribed energy function (here, third order fairness combined with orthogonality to the reference mesh) is positive definite, we use a Newton relaxation, restricted to the PQ osculant surface, to improve the mesh quality without compromising the planarity* ($\max |E_i| < 10mm$ *for all the above meshes). Meshes* **B** *and* **C** *are obtained by minimizing in a 2D tangent subspace (top-left) and in the full tangent space, respectively.*

regions $\mathcal{R}$, e.g., the green box as shown in Figure 13. Given a set of user prescribed vertex displacements, we deform the input shape by taking small steps in the mesh manifold. In each step, starting from the current mesh $\mathbf{x}$, we compute the next mesh position $\mathbf{x}'$ towards the target deformations using Equation 16. If the mesh $\mathbf{x}'$ does not intersect the obstacle, i.e., $\mathbf{x}' \cap \mathcal{R} = \emptyset$, we continue with further steps. Otherwise, we identify the intersecting vertex set $\{v_k' | v_k' \in \mathbf{x}' \cap \mathcal{R}\}$, add the corresponding vertex inequality constraints of the form $v_k \notin \mathcal{R}$ to Equation 16, and solve the resultant quadratic program [Coleman and Li 1996] to get a new $\mathbf{x}'$. If the current solution $\mathbf{x}'$ still intersects the obstacle, we detect and add additional constraints, and iterate; otherwise, we remove all the inequality constraints, and proceed with further steps. Since we try to walk around the obstacles, we may fail to find a solution in complicated configuration spaces. Note that adding inequality constraints

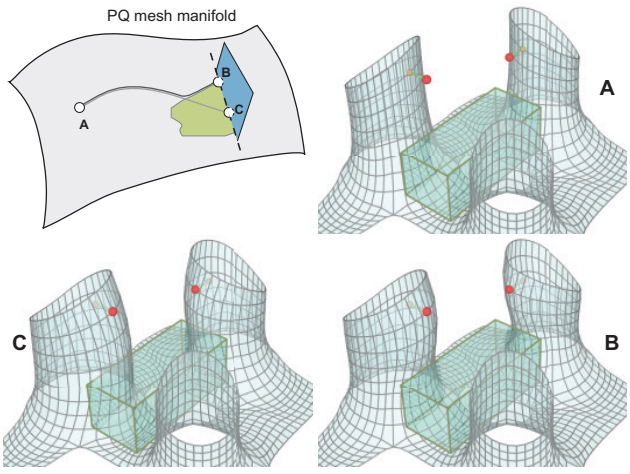


Figure 13: *PQ mesh manifold deformation in the presence of an obstacle (in green). Given two user prescribed vertex constraints (in red), mesh B and mesh C are the deformed PQ meshes obtained with and without obstacle constraints, respectively. The obstacle (green box) acts as a forbidden region in* $\mathbb{R}^D$.

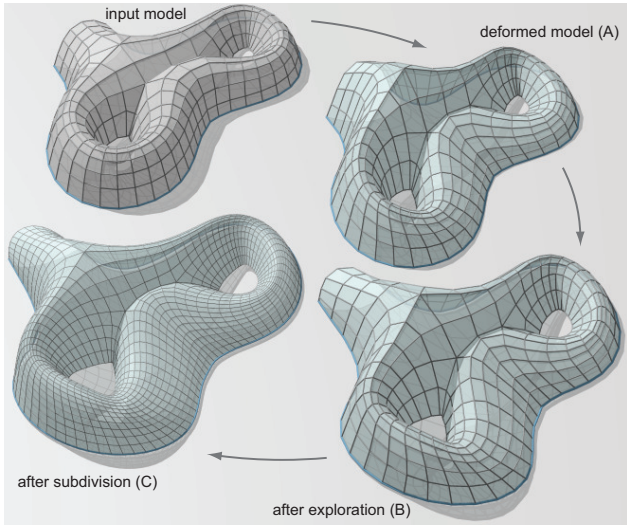


Figure 14: *In architectural design, preserving planarity of prescribed curves, e.g., floors, in course of design exploration is desirable. Our framework can easily incorporate such constraints. Here we show the result of handle-based deformation (A), subsequent shape exploration based on eigenanalysis of the respective intrinsic Hessian (B), and subdivision combined with optimization for planarity (C). Floor curves marked in blue.*

for all vertices is not practical, since the resultant quadratic program becomes too complex, and computationally expensive.

Planarity of selected polygons. In architecture, alignment of certain mesh polygons to prescribed planes, e.g., horizontal floor levels (see Figure 14), is an important design constraint. It may also be interesting to keep selected polygons (support elements) in vertical planes. If the planes, or at least their normals are prescribed, these are just linear constraints. Planarity of a polyline in an unspecified plane can be expressed by planarity of all quads formed by successive polygon vertices. This would still allow for an overall non-planar curve if 4 or more consecutive vertices lie on a straight line, a rare event which may be detected by a function defined on the resulting constrained mesh manifold.

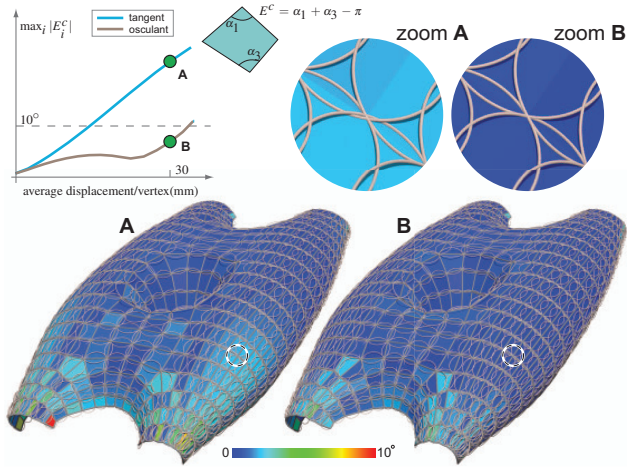


Figure 15: *Tangent and osculant space meshes to a circular mesh manifold. For equal displacements, the osculant surface produces better quality circular meshes, e.g., in the highlighted (zoomed) area, the average deviations of the associated circles from the mesh vertex are* 13*mm and* 0.5*mm, for meshes A and B, respectively.*

Circular mesh manifold. As another example, we explore circular meshes, which – together with conical meshes – are discretizations of the network of principal curvature lines and possess offset properties important for supporting structures and multilayer constructions [Pottmann et al. 2007]. A quad mesh is *circular* if each face has a circumcircle, i.e., it is a special type of PQ mesh. If $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ are the angles of a planar quad face, the additional face constraint is $\alpha_1 + \alpha_3 = \pi$ *or* $\alpha_2 + \alpha_4 = \pi$. Thus, we have two constraints per face, implying that the circular mesh manifold $\mathcal{M}^c$ is of dimension $D - 2m$ (m being the number of faces), and of course contained in the corresponding PQ mesh manifold. Exploration and navigation methodologies described before apply directly to this mesh manifold (see Figures 1, 15, 17, 18, and 19).

Projection to the constrained mesh manifold. Although navigation using the osculant approximation allows us to take longer steps (e.g., in PQ mesh manifolds, we regularly observed 2-4x improvement over tangent space navigation even along directions of low-moderate curvature) while staying close to an implicitly prescribed shape space, large steps of manifold navigation can still take us too far off the underlying manifold. Additionally, many steps lead to accumulation errors resulting in a mesh $\mathbf{x}$ off the initial shape space (see Figure 6). In such cases, one can project the current mesh using a constrained optimization since we have a good starting guess, e.g., for PQ mesh manifold one can use a few iterations of the method by Liu et al. [2006]. Since mesh $\mathbf{x}$ is close to the PQ space, the optimization usually has little visual difference (in contrast to Figure 2), but helps to lower the accumulated error. In our experience, the osculant surface navigation accumulates error only over large steps.

New design possibilities. Shape space exploration opens up new ways for direct constrained mesh design. We start with a trivial example of a mesh $\mathbf{x}_0$ satisfying the chosen constraints; this defines the connectivity and the corresponding mesh manifold. We can then design real freeform variants by deformation and/or exploration.

Due to the interactive performance on smaller models, we suggest to start the design of new shapes at lower resolution (<500 faces in our current implementation). Then, manifold navigation can be based on osculant computation at multiple intermediate meshes. Exploration combined with handle-based deformation offer an effective toolbox for the design of coarse models, which themselves provide a perfect input for the subdivision and optimization approach of Liu et al. [2006] (Figures 14 and 16). Note that in presence of increasing number of constraints, our method becomes more useful (dimension of constrained mesh manifold decreases); in contrast, other alternatives (e.g., optimization based approximation) become less predictable.

Interesting shapes result from flat circular meshes with nonregular combinatorics, reflecting the fact that the network of principal curvature lines on a surface usually has singularities. One will first analyze the shape variety at various types of singularities (see Figure 17) and then proceed towards the design of more complex

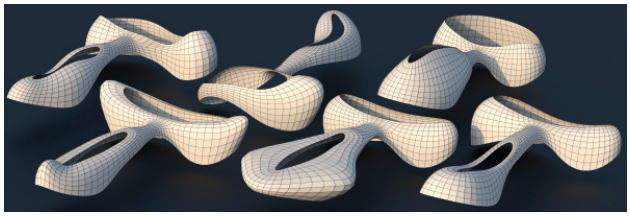


Figure 16: *Starting from a single PQ mesh (starting model shown in Figure 11-top), our geometric framework allows navigation and exploration of the shape space of PQ meshes, enabling easy creation of aesthetic model variants.*

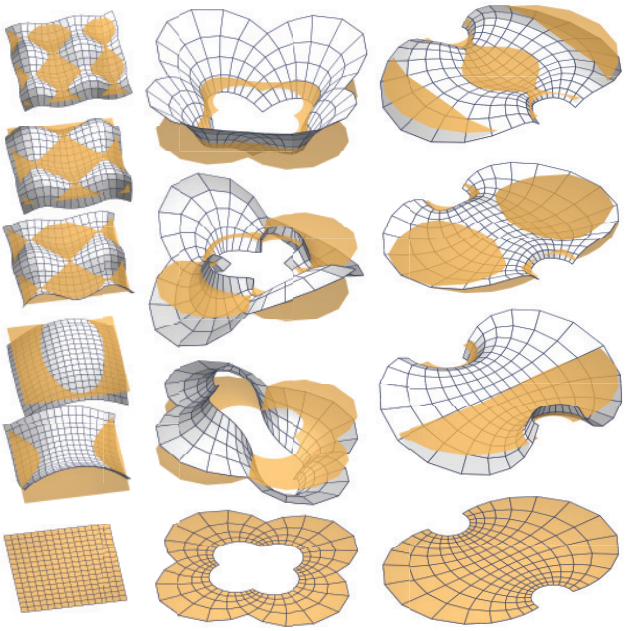


Figure 17: *Eigenanalysis of the intrinsic Hessian is suitable to generate a catalog of basic forms. We analyse the mesh manifold in the neighborhood of a (here circular) mesh $\mathbf{x}_0$ that lies entirely in a plane. The basic types arising from a regular grid $\mathbf{x}_0$ are expected. The more interesting forms come from circular meshes $\mathbf{x}_0$ with or around singularities, and have been generated with conformal mappings (complex functions $z \mapsto 1/z$ (middle), $z \mapsto \tanh z$ (right)).*

circular meshes. Planar circular meshes discretize planar orthogonal curve networks and can be designed in various ways. Note that this allows us to *design surfaces by directly manipulating their network of principal curvature lines*. The combinatorics of the network (circular mesh) defines the manifold $\mathcal{M}^c$ in which the designer can navigate (see Figures 1, 18, and 19). In Figure 1, additional polyline planarity constraints were used for three of the pillars. In Figure 19, the input is a circular mesh obtained from a simple rotational surface, with two additional constraints: (i) polyline planarity prescribed for the floor curve, (ii) the polylines at the two ends of the tunnel are each constrained to lie on (unspecified) circles.

Performance. The most costly part of shape exploration is the osculant computation, where we have to solve the $(D-m)^2$ linear

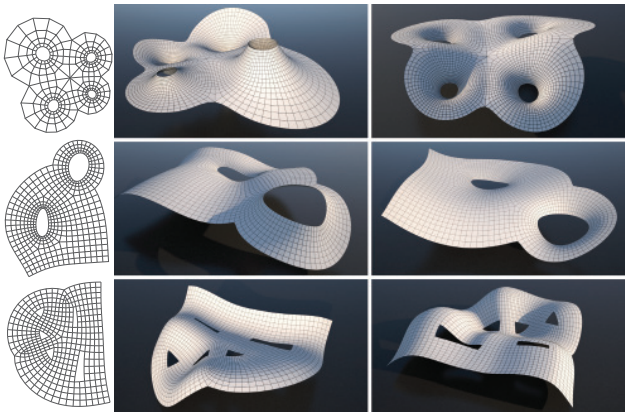


Figure 18: *Starting from flat circular meshes, circular mesh manifold exploration allows quick creation of designs. Typically, in each case, we require 1-2 minutes of interaction.*

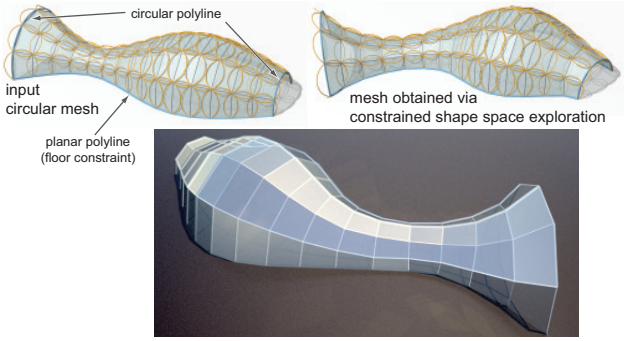


Figure 19: *Designed mesh obtained via shape space exploration, starting from a circular mesh with polyline planarity (floor) constraint, and two polylines constrained on (unspecified) circles.*

$(m \times m)$ systems of Equation 5. We used CUDABLAS, a GPU matrix library with Nvidia Quadro 5000, that takes a few seconds for $m = 300$ faces, but about several minutes for models with around 1K faces for computing the osculant and the spectral analysis of the associated intrinsic Hessian matrix. Subsequent exploration is real-time (see supplementary video). Handle driven deformation, requiring only gradients of prescribed constraints, works at interactive rates (see accompanying demo).

Limitations. In order to allow efficient design space exploration, we precompute the osculant and the associated matrices. Even with a GPU based parallel solver, the precomputation step is time consuming for large models involving upwards of 3K faces. One option is to investigate *multiresolution approaches* to efficiently refine the model and enlarge the resultant shape manifold. In our setting, a constrained mesh manifold is defined only based on specified constraints, and by itself does not prevent meshes from self-intersecting. While a suitable energy function can help identify and steer away from such undesirable meshes, it is more efficient to roll-back during the path traversal once such events are detected, rather than slow down the exploration phase with a global energy.

Conclusion and future research. We introduced constrained shape spaces specified by a single mesh along with a collection of non-linear constraints. We provided necessary mathematical formulations to analyze such high dimensional and co-dimensional surfaces, obtained approximations using tangent space and osculant surface, which respectively share first and second order contact with the mesh manifold. We also presented intrinsic Hessian approximations of energy functions directly on the osculant surface, and use the same for navigation and exploration restricted to desirable regions as dictated by prescribed quality measures. Using PQ and circular mesh manifolds as typical test scenarios, we evaluated the framework on a variety of designs and demonstrated its capabilities towards *design exploration*.

We see multiple avenues for future research, such as, investigating other constrained mesh manifolds, e.g., functional webs [Deng et al. 2011], encoding other mesh qualities as suitable energy functions, and develop further tools for constrained manifold navigation.

Acknowledgements. We thank Daniel Piker for providing the starting "rheotomic" mesh used as the input mesh in Figure 1 and for the starting flat meshes in the upper two rows of Figure 18. We thank Johannes Wallner for his many useful comments and suggestions, Alexander Schiftner, Mathias Höbinger and Michael Eigensatz for their help and valuable comments, and the anonymous reviewers for their feedback. We are grateful to Heinz Schmiedhofer for the final renderings. The work has been partially supported by Austrian Science Fund (FWF) grant P23735-N13 and Austrian Science Promotion Agency (FFG) grant 813391.

References

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *Proc. ACM SIGGRAPH*, 187–194.

BOBENKO, A. I., AND SURIS, Y. B. 2008. *Discrete Differential Geometry: Integrable Structure*, vol. 98. Graduate Studies in Mathematics.

BOTSCH, M., AND SORKINE, O. 2008. On linear variational surface deformation methods. In *IEEE Trans. on Vis. and Comp. Graphics*, vol. 14, 213–230.

BOTSCH, M., PAULY, M., GROSS, M., AND KOBBELT, L. 2006. Primo: coupled prisms for intuitive surface modeling. In *Proc. of Symp. of Geometry Processing*, 11–20.

CECCATO, C., HESSELGREN, P., PAULY, M., POTTMANN, H., AND WALLNER, J., Eds. 2010. *Advances in Architectural Geometry*. Springer.

COLEMAN, T., AND LI, Y. 1996. A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization 6*, 4, 1040–1058.

DENG, B., POTTMANN, H., AND WALLNER, J. 2011. Functional webs for freeform architecture. In *Proc. of Symp. of Geometry Processing*, 1369–1378.

DO CARMO, M. P. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ.

GAL, R., SORKINE, O., MITRA, N. J., AND COHEN-OR, D. 2009. iwires: An analyze-and-edit approach to shape manipulation. *ACM TOG 28*, 3, #33, 1–10.

GU, X. D., AND YAU, S.-T. 2008. *Computational Conformal Geometry*. International Press.

HOFFMANN, T. 2011. *On local deformations of planar quad meshes*, vol. 6327 of *LNCS*. 167–169.

HUANG, Q., WICKE, M., ADAMS, B., AND GUIBAS, L. 2009. Shape decomposition using modal analysis. *Computer Graphics Forum (Proc. EUROGRAPHICS) 28*, 2, 407–416.

KILIAN, M., MITRA, N. J., AND POTTMANN, H. 2007. Geometric modeling in shape space. *ACM TOG 26*, 3, #64, 1–8.

LIPMAN, Y., LEVIN, D., AND COHEN-OR, D. 2008. Green coordinates. *ACM TOG 27*, 78:1–78:10.

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

NOCEDAL, J., AND WRIGHT, S. J. 2006. *Numerical Optimization*, 2nd ed. Springer, New York.

POTTMANN, H., LIU, Y., WALLNER, J., BOBENKO, A., AND WANG, W. 2007. Geometry of multi-layer freeform structures for architecture. *ACM TOG 26*, 3, # 65,1–11.

POTTMANN, H., SCHIFTNER, A., BO, P., SCHMIEDHOFER, H., WANG, W., BALDASSINI, N., AND WALLNER, J. 2008. Freeform surfaces from single curved panels. *ACM TOG 27*, 3, # 76,1–10.

SCHOUTEN, J. A., AND STRUIK, D. J. 1931. *Einführung in die Neuen Methoden der Differentialgeometrie*. Groningen.

SUMNER, R. W., ZWICKER, M., GOTSMAN, C., AND POPOVIĆ, J. 2005. Mesh-based inverse kinematics. *ACM TOG 24*, 3, 488–495.

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