EUROGRAPHICS 2013 / I. Navazo, P. Poulin
(Guest Editors)

Volume 32 (2013), Number 2

Exploring Local Modifications for Constrained Meshes

Bailin Deng[1,2] Sofien Bouaziz[1] Mario Deuss[1] Juyong Zhang[3] Yuliy Schwartzburg[1] Mark Pauly[1]

[1]École Polytechnique Fédérale de Lausanne [2]Technische Universität Wien [3]University of Science and Technology of China

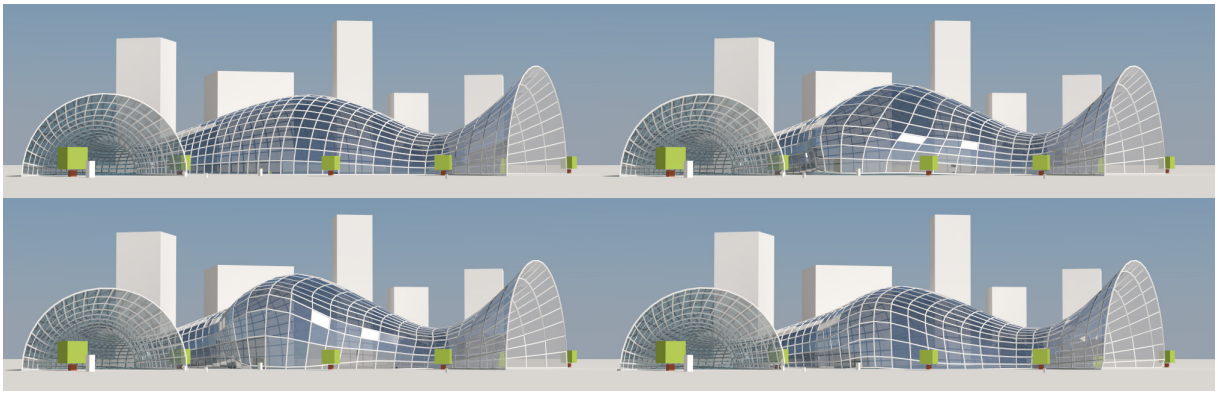


Figure 1: *Local modifications of a constrained mesh. In this example a glass structure composed of planar quads is locally deformed by exploring a subspace encoding local planar modifications of its central zone.*

Abstract

Mesh editing under constraints is a challenging task with numerous applications in geometric modeling, industrial design, and architectural form finding. Recent methods support constraint-based exploration of meshes with fixed connectivity, but commonly lack local control. Because constraints are often globally coupled, a local modification by the user can have global effects on the surface, making iterative design exploration and refinement difficult. Simply fixing a local region of interest a priori is problematic, as it is not clear in advance which parts of the mesh need to be modified to obtain an aesthetically pleasing solution that satisfies all constraints.

We propose a novel framework for exploring local modifications of constrained meshes. Our solution consists of three steps. First, a user specifies target positions for one or more vertices. Our algorithm computes a sparse set of displacement vectors that satisfies the constraints and yields a smooth deformation. Then we build a linear subspace to allow realtime exploration of local variations that satisfy the constraints approximately. Finally, after interactive exploration, the result is optimized to fully satisfy the set of constraints. We evaluate our framework on meshes where each face is constrained to be planar.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

In geometric modeling, constraints are often used to control important properties of designs that relate to fabrication, structural stability, or aesthetics of a surface, among numerous other objectives. Effective algorithms for constrained deformation that can handle the complex and often global coupling of constraints are therefore of central importance. *Local control* is an important feature of a constraint-based modeling system, as it allows the user to change a design without influencing regions that are already satisfactory. In current tools, local modifications are typically achieved by manually selecting editable vertices. However, global inter-

dependencies of constraints make it difficult to determine in advance which vertices need to be modified. In this paper we address the question of how to automatically determine a local deformation that satisfies all the constraints, while maintaining the quality of the shape.

1.1. Overview

Given an input mesh and a set of geometric constraints, e.g. planarity of polygonal faces, our goal is to explore local modifications of the mesh that respect all the prescribed constraints. To facilitate effective editing, we separate our solution into three main steps: Given the target positions of one or more vertices specified by the user, we first compute a smooth local modification of the mesh that satisfies the constraints. This allows the user to interactively explore local modifications using vertex handles. In the second step, we construct a subspace of high quality local deformations that satisfies the constraints approximately. This enriches the solutions of the first step with correlated local variations, which the user can explore in realtime. In the final step, we optimize the deformed meshes generated in the second step to fully satisfy the constraints. The details of each step are presented in Section 2. In each step of our framework, the results should satisfy the following conditions:

1. The deformation is local (confined to a small region of the mesh close to the user input).
2. The modified mesh has a nice shape, i.e. its fairness energy should be small.
3. The modified mesh satisfies all constraints (at least approximately).

In our approach, the desired deformation is achieved by enforcing sparsity and smoothness over the deformation vectors. Sparsity is necessary as we want to displace a minimal set of vertices, while smoothness helps us to obtain nice deformed shapes. We apply our framework on meshes where each face is constrained to be planar. Planar meshes are of great interest in architectural geometry due to their advantages for fabrication.

Contribution. During the first step of handle-based editing, contrary to previous work, our approach computes local modifications satisfying the constraints, without manually selecting the deformable region. We instead provide a single scalar value for the user to control the level of locality, by formulating an optimization problem with a sparsity-inducing penalty. In the second step, we build a linear subspace characterizing high-quality local variations, by introducing a function that measures correlation between local displacements. Finally, we demonstrate how these problems can be split into a set of simple and tractable subproblems that can be solved efficiently. This is achieved by combining variable splitting and the augmented Lagrangian method.

1.2. Related Work

Decomposing freeform surfaces into a set of shapes with constraints is a subject of numerous publications in geometry processing [ACSD*03, BZK09, LXW*11]. Recent work on surface rationalization allows computing panel layouts of architectural freeform surfaces [EKS*10, FLHCO10]. Optimization of panel shapes can lead to interesting structural properties. Circular and conical meshes have been shown to be of great interest for offset surfaces [LPW*06]. Planar quad (PQ) meshes are easy to fabricate, and various methods have been developed for their optimization [LPW*06, ZSW10]. Surfaces can also be decomposed into other types of components that are efficient to manufacture, such as planar, circular, or geodesic curves [DPW11].

One drawback of decoupling design and rationalization is that post-optimizing the shape of surface elements may lead to large deviation from the original design. Shape space exploration methods [HSTP11, YYPM11, Vax12, BDS*12, ZTY*12] address this issue by allowing a user to explore a space of surfaces that nearly satisfy the constraints. However, these approaches lack locality and thus potentially lead to global modifications during exploration, which can make form-finding difficult to control. We propose a shape space exploration method that keeps the modifications local, thus allowing finer control over the deformation.

Recently, [Hof10] provided a solution to local modification of PQ meshes with a given displacement for one vertex. His solution modifies only the one-ring neighborhood of this vertex. However, the aesthetics of the mesh are usually destroyed by such overly local edits. [HK12] presented a method that enables local and intuitive control over the deformation. In this approach, the constraints are linearized to allow interactive editing. In contrast to the greedy method proposed by [HK12], we present in Section 2.1 a solution that directly solves the non-linear constrained problem. This works well for larger displacements and does not need to reproject onto the constrained mesh manifold.

[BDS*12] presents a general framework to enforce shape constraints on discrete mesh elements. Their optimization algorithm is closely related to recent proximal optimization methods [CP11] from which we also draw some inspiration. However, contrary to [BDS*12], our framework handles hard constraints, which is an important feature of a constraint-based modeling system.

Our work is based on recent advances in sparsity-inducing penalties [BJMO12] that have been successfully applied in compressive sensing [CW08]. We apply a mixed ℓ_2/ℓ_1 penalty [EM09] over the deformation vectors to find a minimal set of vertices that need to be moved to satisfy the user requirement. It has been shown that minimization of mixed ℓ_2/ℓ_1 norms leads to block-sparse solutions, i.e., the non-zero entries of the solution vectors appear in a small number of blocks [EKB10]. In our case, the blocks are triplets of coordinates for each vertex displacement vector.

In geometry processing, sparsity-inducing penalties were used for surface reconstruction [DGP09, ASGCO10], construction of splines [Lav02], construction of multiscale kernels [Rus11] and deformation [GZL12]. In [GZL12], the sparsity-inducing penalty is applied on the error function itself, while in our approach it is applied on the displacement vectors. The sparsity-inducing penalty we use is most related to [ASGCO10] that uses a mixed ℓ_2/ℓ_1 norm regularizer in the context of surface reconstruction.

Previous work on sparsity-enhancing optimization usually considers convex formulations [BJMO12]. However, in our case we need to deal with non-convex constraints which are common in geometry processing. We propose an efficient algorithm based on the augmented Lagrangian method [Ber96, NW06] and variable splitting [Eck89]. The splitting scheme allows us to decompose our complex problem into a set of subproblems that can be solved easily and efficiently.

2. Formulation

In this section, we introduce our framework for exploring local deformations of a constrained mesh. Each step of our framework involves a constrained optimization problem. Numerical solutions to these problems are then discussed in Section 3. For a mesh with n vertices and fixed connectivity, we represent it as a vector $\mathbf{p} = [\mathbf{p}_1^T, \ldots, \mathbf{p}_n^T]^T \in \mathbb{R}^{3n}$, where $\mathbf{p}_i$ is the position for vertex v_i ($i = 1, \ldots, n$). We consider meshes that satisfy a set of constraints $E_j(\mathbf{p}) = 0$ ($j = 1, \ldots, m$) about the vertex positions.

2.1. Handle-based exploration

Starting from a given mesh $\mathbf{p}^0$ with all constraints satisfied, we use a small set of mesh vertices $\{v_i \mid i \in \Gamma\}$ as handles to deform the mesh. Based on the target positions of these handles, we compute a new shape $\mathbf{p}^0 + \mathbf{d}$ of the mesh, where $\mathbf{d} = [\mathbf{d}_1^T, \ldots, \mathbf{d}_n^T]^T \in \mathbb{R}^{3n}$ is a mesh displacement with $\mathbf{d}_k \in \mathbb{R}^3$ being the displacement for vertex v_k. $\mathbf{d}$ is determined by solving a constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{d}} \quad & \frac{\omega_\mathrm{h}}{2} F_\mathrm{handle}(\mathbf{d}) + \frac{\omega_\mathrm{s}}{2} F_\mathrm{sparse}(\mathbf{d}) + \frac{\omega_\mathrm{f}}{2} F_\mathrm{fair}(\mathbf{d}) \\ \text{s.t.} \quad & E_j(\mathbf{p}^0 + \mathbf{d}) = 0, \quad \text{for } j = 1, \ldots, m. \end{aligned} \tag{1}$$

Here F_handle penalizes deviation between handle vertices and their target positions, F_sparse measures sparsity of the displacement, F_fair is a fairness energy, and the ω are nonnegative weights. Each function is formulated as follows.

Handle positions. The target position of a handle vertex v_i gives a target displacement vector $\widetilde{\mathbf{d}}_i$, hence

$$F_\mathrm{handle}(\mathbf{d}) = \sum_{i \in \Gamma} \|\mathbf{d}_i - \widetilde{\mathbf{d}}_i\|_2^2.$$

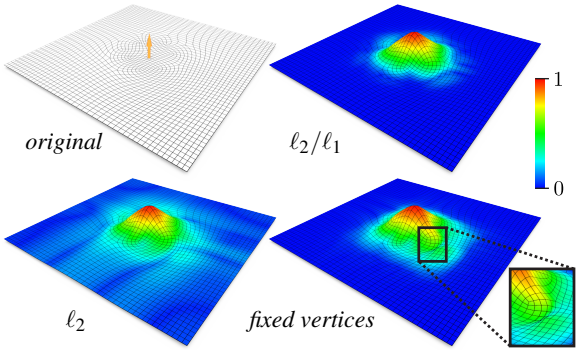


Figure 2: *Comparison of our sparsity regularization with classical local editing approaches. With equal fairness energy, our ℓ_2/ℓ_1 regularizer provides a more local edit than an ℓ_2-norm closeness term and exhibits less distortion than fixing vertices manually. The color-coding shows the lengths of verticex displacements, normalized by the input displacement length for the handle vertex.*

Sparsity. For local deformations, we require $\mathbf{d}$ to be sparse, i.e., there are many vertices with zero displacement. If we collect the vertex displacement norms into a vector $\delta = [\,\|\mathbf{d}_1\|_2, \ldots, \|\mathbf{d}_n\|_2\,]$, then sparsity of $\mathbf{d}$ means a small number of nonzero coefficients in δ. Such a property can be achieved by minimizing the ℓ_1 norm of δ [CRT06], giving a sparsity-inducing penalty for $\mathbf{d}$

$$\|\mathbf{d}\|_{2,1} = \|\delta\|_1 = \sum_{i=1}^{n} \|\mathbf{d}_i\|_2.$$

$\|\mathbf{d}\|_{2,1}$ is called the *mixed ℓ_2/ℓ_1 norm* of $\mathbf{d}$ [EM09]. We define F_sparse based on $\|\mathbf{d}\|_{2,1}$:

$$F_\mathrm{sparse}(\mathbf{d}) = \sum_{i \notin \Gamma} \|\mathbf{d}_i\|_2.$$

Here we ignore handle vertices as they are already constrained by F_handle. As shown in Figure 2, using the mixed ℓ_2/ℓ_1 norm provides a high-quality local deformation that is advantageous over manually fixing vertices or using an ℓ_2-norm closeness energy term.

Fairness energy. Assuming that the original mesh already has a nice shape, we only require that $\mathbf{d}$ defines a smooth vertex displacement field. We choose a positive semidefinite quadratic form of $\mathbf{d}$ which measures its smoothness

$$F_\mathrm{fair}(\mathbf{d}) = \|\mathbf{E}\mathbf{d}\|_2^2. \tag{2}$$

We can use Laplacian-based energies, or the second/third order difference energies in [YYPM11].

Using the above formulations, Problem (1) reads

$$\begin{aligned} \min_{\mathbf{d}} \quad & \frac{\omega_\mathrm{h}}{2} \sum_{i \in \Gamma} \|\mathbf{d}_i - \widetilde{\mathbf{d}}_i\|_2^2 + \frac{\omega_\mathrm{s}}{2} \sum_{i \notin \Gamma} \|\mathbf{d}_i\|_2 + \frac{\omega_\mathrm{f}}{2} \|\mathbf{E}\mathbf{d}\|_2^2 \\ \text{s.t.} \quad & E_j(\mathbf{p}^0 + \mathbf{d}) = 0, \quad j = 1, \ldots, m. \end{aligned} \tag{3}$$

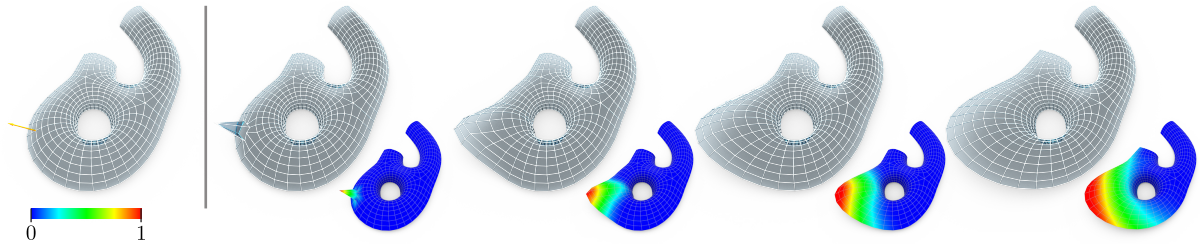


Figure 3: *Handle-based local modifications of a PQ mesh, showing the tradeoff between fairness and sparsity. Left: the input displacement for the original mesh. Right: results with increasing fairness weight from left to right. The color-coding shows the lengths of vertex displacements, clamped and normalized by the input displacement length for the handle vertex.*

Figure 3 shows the effect of varying the fairness weight ω_f when deforming a PQ mesh. When $\omega_f = 0$, only the one ring neighborhood of the handle vertex is modified, which recovers the solution given in [Hof10]. With increasing values of ω_f, the modification becomes more global with smoother shape. This shows the tradeoff between sparsity and smoothness of the deformation. Local edits for general polygonal meshes with planar faces are shown in Figures 8 and 9.

Counting degrees of freedom. To gain an overview of the level of locality that can be achieved by smooth deformations, we can count the degrees of freedom for constrained meshes. The numbers of faces and vertices in a mesh are related by $\sum_i i \cdot F_i = \sum_j j \cdot V_j$, where F_i is the number of faces with i vertices, and V_j is the number of vertices with j incident faces. Each vertex provides three degrees of freedom, while the planarity condition of a face with i vertices induces $i-3$ constraints. Therefore, for a mesh where all faces are planar, the expected number for degrees of freedom is

$$d = 3N_V - \sum_i (i-3)F_i,$$

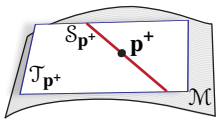
where N_V is the number of vertices. On a regular PQ mesh, the number of faces is approximately the same as the number of vertices, and $d \approx 2N_V$. For a regular planar tri-hex mesh, the number of triangles is approximately two times the number of hexagons, and $d \approx 2N_V$. An example can be seen in Figure 8. For a regular planar hexagonal mesh, $d \approx \frac{3}{2}N_V$. This is an example of a constrained mesh with fewer degrees of freedom. In these meshes, smooth deformations tend to be more global (see Figure 9).

2.2. Subspace exploration

The optimization approach in the previous section computes a single solution $\mathbf{d}_\omega$ from given weight parameters $\omega = (\omega_h, \omega_s, \omega_f)$. This solution, however, may not be the modification the user intended. In many cases, the user may want to explore different shapes that satisfy the constraints. One possibility to do so is to run the optimization with different weight parameters, which can be a time-consuming and tedious task. Furthermore, such an approach is not able to expose interesting shapes that are suboptimal with respect to the optimization objective function.

In this section, we present a different approach to explore viable shapes. For a local modification $\mathbf{d}$ on the original mesh $\mathbf{p}^0$, we denote the modified mesh $\mathbf{p}^0 + \mathbf{d}$ as $\mathbf{p}^+$. We compute a space $S_{\mathbf{p}^+}$ of meshes with nice shapes that are local modifications of $\mathbf{p}^0$. The meshes in $S_{\mathbf{p}^+}$ satisfy the constraints approximately. By computing such a space for each optimization solution $\mathbf{d}_\omega$, our approach provides considerably more choices of shapes for the user to explore than changing ω only.

Given a fixed connectivity, the *constrained mesh manifold* $\mathcal{M} = \{\mathbf{p} \in \mathbb{R}^{3n} \mid E_j(\mathbf{p}) = 0, j = 1, \ldots, m\}$ is the manifold of meshes that satisfy the constraints, embedded in the space of meshes with the given connectivity. For a mesh $\mathbf{p} \in \mathcal{M}$, the space $\mathcal{T}_\mathbf{p}$ of tangent vectors of $\mathcal{M}$ at $\mathbf{p}$ is $\mathcal{T}_\mathbf{p} = \{\mathbf{t} \mid \nabla E_j(\mathbf{p}) \cdot \mathbf{t} = 0, j = 1, \ldots, m\}$. $\mathcal{T}_\mathbf{p}$ represents displacements from $\mathbf{p}$ that satisfy the constraints up to first order [YYPM11]. For our problem, we assume that $\mathbf{p}^+ \in \mathcal{M}$ (which is the case if $\mathbf{d}$ is computed using the algorithm in Section 2.1). We compute the target space as $S_{\mathbf{p}^+} = \{\mathbf{p}^+ + \mathbf{t} \mid \mathbf{t} \in \mathcal{S}_{\mathbf{p}^+}\}$, where $\mathcal{S}_{\mathbf{p}^+}$ is a linear subspace of $\mathcal{T}_{\mathbf{p}^+}$ (see inset). Meshes in $S_{\mathbf{p}^+}$ satisfy the constraints approximately. We search for the subspace $\mathcal{S}_{\mathbf{p}^+}$ based on the following set of requirements.



Shape quality and vertex positions. The first requirement for $\mathcal{S}_{\mathbf{p}^+}$ is that every displacement $\mathbf{t} \in \mathcal{S}_{\mathbf{p}^+}$ leads to a mesh $\mathbf{p}^+ + \mathbf{t}$ with nice shape. Assuming a nice shape of mesh $\mathbf{p}^+$, we apply the fairness energy (2) to $\mathbf{t}$ for measuring the shape quality of $\mathbf{p}^+ + \mathbf{t}$. Optionally, the user may want some vertices to stay close to their current positions during exploration (for example the handle vertices in Section 2.1). This requirement is enforced with a function $\sum_{i \in \Psi} \|\mathbf{t}_i\|_2^2$, where $\mathbf{t}_i \in \mathbb{R}^3$ is the component of $\mathbf{t}$ for vertex v_i, and Ψ is the index set of vertices subject to the closeness condition. Combining

these two functions, we derive a quality measure for $\mathbf{t}$

$$Q(\mathbf{t}) = \frac{\beta_f}{2}\|\mathbf{E}\mathbf{t}\|_2^2 + \frac{\beta_h}{2}\sum_{i\in\Psi}\|\mathbf{t}_i\|_2^2, \tag{4}$$

where β_f, β_h are nonnegative weights.

Sparsity. Another requirement is that $\mathcal{S}_{\mathbf{p}^+}$ represents sparse displacements w.r.t. $\mathbf{p}^0$, meaning that the mixed ℓ_2/ℓ_1 norm $\|\mathbf{d}+\mathbf{t}\|_{2,1}$ should be small for all $\mathbf{t}\in\mathcal{S}_{\mathbf{p}^+}$. Suppose that $\mathcal{S}_{\mathbf{p}^+}$ is spanned by an orthonormal basis $\mathbf{t}^1,\ldots,\mathbf{t}^s$, then $\mathbf{t}=\sum_{i=1}^s x_i\mathbf{t}^i$ with $x_i\in\mathbb{R}$. Note that $\|\cdot\|_{2,1}$ satisfies the triangle inequality

$$\|\mathbf{d}+\mathbf{t}\|_{2,1} \le \|\mathbf{d}\|_{2,1} + \sum_{i=1}^{s}|x_i|\|\mathbf{t}^i\|_{2,1}.$$

In order to have a small $\|\mathbf{d}+\mathbf{t}\|_{2,1}$, it is necessary that $\|\mathbf{t}^i\|_{2,1}$ is small for all i, namely $\mathbf{t}^i$ represents a sparse displacement.

However, sparsity for all $\mathbf{t}^i$ does not guarantee sparsity of $\mathbf{d}+\mathbf{t}$. For example, if the influence regions of $\mathbf{d}$, $\mathbf{t}^1$, …, $\mathbf{t}^s$ are disjoint, $\mathbf{d}+\mathbf{t}$ may modify all of these regions, resulting in a non-sparse displacement. We must therefore make sure that the influence regions of $\mathbf{d}$, $\mathbf{t}^1$, …, $\mathbf{t}^s$ have a large common overlap. For this we define a *correlation function*: For two *unit* vectors $\mathbf{u},\mathbf{v}\in\mathbb{R}^{3n}$ representing mesh displacements, their correlation is defined as

$$\xi(\mathbf{u},\mathbf{v}) = \sum_{i=1}^{n}\|\mathbf{u}_i\|_2\|\mathbf{v}_i\|_2, \tag{5}$$

where $\mathbf{u}_i,\mathbf{v}_i\in\mathbb{R}^3$ are components of $\mathbf{u}$, $\mathbf{v}$ for vertex v_i. $\xi(\mathbf{u},\mathbf{v})$ attains maximum value 1 when $\|\mathbf{u}_i\|_2=\|\mathbf{v}_i\|_2$ for all i (the influence regions of $\mathbf{u}$, $\mathbf{v}$ overlap completely, with identical displacement length for each vertex). And it reaches minimum value 0 when $\|\mathbf{u}_i\|_2\|\mathbf{v}_i\|_2=0$ for all i (the influence regions of $\mathbf{u}$, $\mathbf{v}$ are disjoint). Furthermore, there is a nice interpretation of $\xi(\mathbf{u},\mathbf{v})$ regarding the triangle inequality

$$\|x\mathbf{u}+y\mathbf{v}\|_{2,1} \le |x|\|\mathbf{u}\|_{2,1} + |y|\|\mathbf{v}\|_{2,1} \tag{6}$$

with $x,y\in\mathbb{R}$:

Proposition 1 For unit vectors $\mathbf{u}$, $\mathbf{v}$, function

$$D(\mathbf{u},\mathbf{v}) = \oint_{x^2+y^2=1}(|x|\|\mathbf{u}\|_{2,1}+|y|\|\mathbf{v}\|_{2,1}-\|x\mathbf{u}+y\mathbf{v}\|_{2,1})\,ds$$

has the following properties

(a) $D(\mathbf{u},\mathbf{v})=0\Leftrightarrow\xi(\mathbf{u},\mathbf{v})=0$;
(b) $D(\mathbf{u},\mathbf{v})\ge c\cdot\xi(\mathbf{u},\mathbf{v})$, where

$$c = \frac{4}{\max\limits_i \max\limits_{x^2+y^2=1}(\|x\mathbf{u}_i\|_2+\|y\mathbf{v}_i\|_2+\|x\mathbf{u}_i+y\mathbf{v}_i\|_2)}.$$

For a proof, see Appendix A. Here $D(\mathbf{u},\mathbf{v})$ is an integral of the difference function between the two sides of (6). If $\mathbf{u}$ and $\mathbf{v}$ are orthogonal, then $D(\mathbf{u},\mathbf{v})$ is integrated over all unit vectors in $\mathrm{span}(\mathbf{u},\mathbf{v})$. For $\|\cdot\|_{2,1}$ to be small over the space $\mathrm{span}(\mathbf{u},\mathbf{v})$, we need a large integral $D(\mathbf{u},\mathbf{v})$ together with small $\|\mathbf{u}\|_{2,1}$ and $\|\mathbf{v}\|_{2,1}$. Proposition 1 shows that correlation function $\xi(\mathbf{u},\mathbf{v})$ is related to a lower bound for $D(\mathbf{u},\mathbf{v})$, and they reach minimum values at the same time.

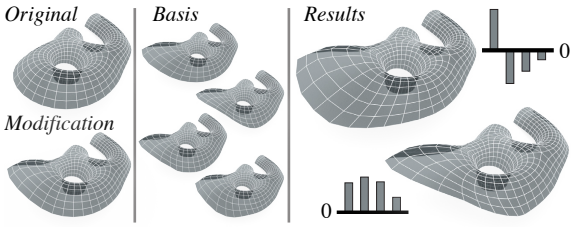


Figure 4: *Subspace of local modifications for the mesh in Figure 3, using four basis vectors. Left: original mesh with initial local modification. Middle: the basis vectors, illustrated by adding each vector to the initial modification. Right: two local modifications in the subspace, computed by adding linear combinations of basis vectors to the initial modification (shown with bar charts).*

Optimization problem. Based on the above observation, we compute $\mathcal{S}_{\mathbf{p}^+}$ as follows. First the user chooses the dimension s of $\mathcal{S}_{\mathbf{p}^+}$. We then construct an orthonormal basis $\mathbf{B}=[\mathbf{t}^1,\ldots,\mathbf{t}^s]$ of $\mathcal{S}_{\mathbf{p}^+}$ incrementally in s steps. $\mathbf{B}$ is initially empty. At step k ($k=1,\ldots,s$), a unit vector $\mathbf{t}^k\in\mathcal{T}_{\mathbf{p}^+}$ is added to $\mathbf{B}$. $\mathbf{t}^k$ is computed by searching in the orthogonal complement of $\mathrm{span}(\mathbf{t}^1,\ldots,\mathbf{t}^{k-1})$ for a vector which

1. is of high quality according to the measure in (4);
2. represents sparse displacements;
3. has large correlations with $\mathbf{d}$ and $\mathbf{t}^1$, …, $\mathbf{t}^{k-1}$.

We find this vector by optimizing

$$\begin{aligned}\min_{\mathbf{t}}\quad & Q(\mathbf{t}) + \frac{\beta_s}{2}\|\mathbf{t}\|_{2,1} - \frac{\beta_c}{2}C(\mathbf{t})\\ \text{s.t.}\quad & \|\mathbf{t}\|_2=1,\ \ \mathbf{J}\mathbf{t}=\mathbf{0},\ \ \mathbf{B}^T\mathbf{t}=0.\end{aligned} \tag{7}$$

Here $C(\mathbf{t})=\xi(\bar{\mathbf{d}},\mathbf{t})+\sum_{1\le i<k}\xi(\mathbf{t}^i,\mathbf{t})$ with $\bar{\mathbf{d}}=\mathbf{d}/\|\mathbf{d}\|_2$, $\mathbf{J}=[\nabla E_1(\mathbf{p}^+),\ldots,\nabla E_m(\mathbf{p}^+)]^T$, and β_s, β_c are weights. The constraint $\mathbf{J}\mathbf{t}=\mathbf{0}$ means $\mathbf{t}\in\mathcal{T}_{\mathbf{p}^+}$, and the term $-C(\mathbf{t})$ penalizes small correlations between $\mathbf{t}$ and $\bar{\mathbf{d}}$, $\mathbf{t}^1$, …, $\mathbf{t}^{k-1}$.

After computing the basis $\mathbf{B}$, the user can explore modifications in $\mathcal{S}_{\mathbf{p}^+}$ in realtime by specifying linear combination coefficients for the basis vectors. Additionally, to find a subspace of the highest quality w.r.t. $Q(\mathbf{t})$, we can perform eigen-decomposition on the reduced Hessian of $Q(\mathbf{t})$ over $\mathcal{S}_{\mathbf{p}^+}$ [YYPM11], and use eigenvectors of the smallest eigenvalues as the exploration basis. In our experiments, the number of basis vectors is no more than 1% of the tangent space dimension. Figure 4 shows the exploration basis for a PQ mesh, and two linear combinations of them. In Figure 5, we show two sets of local modifications of a PQ mesh, each starting from a local deformation computed according to Section 2.1. A realtime exploration of this space can be seen in the accompanying video.

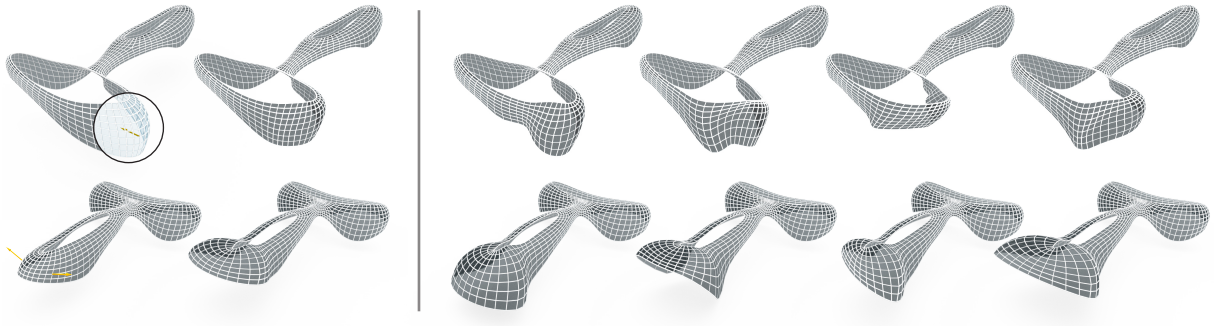


Figure 5: *Subspace exploration of local modifications for a PQ mesh. Left: input displacements with the corresponding handle-based local deformations. Right: local modifications using different linear combinations of the subspace basis vectors.*

2.3. Projection onto constrained mesh manifold

Since the tangent space is only a linear approximation to the constrained mesh manifold $\mathcal{M}$, large displacements in subspace $\mathcal{S}_{\mathbf{p}^+}$ may lead to a mesh that is no longer close to $\mathcal{M}$. To obtain a mesh that fully satisfies the constraints, we project the modified mesh after Section 2.2, which we denote as $\tilde{\mathbf{p}}$, onto $\mathcal{M}$. This is done by computing a displacement $\tilde{\mathbf{t}}$ from $\tilde{\mathbf{p}}$ to $\mathcal{M}$ with an optimization

$$\begin{aligned} \min_{\tilde{\mathbf{t}}} \quad & \alpha_{\mathrm{f}} F_{\mathrm{fair}}(\tilde{\mathbf{t}}) + \alpha_{\mathrm{d}} F_{\mathrm{dist}}(\tilde{\mathbf{t}}) \\ \text{s.t.} \quad & E_j(\tilde{\mathbf{p}} + \tilde{\mathbf{t}}) = 0, \quad \text{for } j = 1, \ldots, m, \end{aligned} \tag{8}$$

where α_{f}, α_{d} are weights, and function F_{dist} penalizes the distance between $\tilde{\mathbf{p}}$ and its projection on $\mathcal{M}$. A natural choice for such a function is $F_{\mathrm{dist}} = \|\tilde{\mathbf{t}}\|_2^2$. Alternatively, we can use a weighted distance function $F_{\mathrm{dist}} = \sum_i^n w_i \|\tilde{\mathbf{t}}_i\|_2$ to put higher penalty for vertices with small displacements between the original mesh $\mathbf{p}^0$ and the current mesh $\tilde{\mathbf{p}}$, in order to prevent global displacement from the original mesh $\mathbf{p}^0$ to the final projected mesh $\tilde{\mathbf{p}} + \tilde{\mathbf{t}}$. In Figure 6, we project two quad meshes obtained from subspace exploration onto the PQ mesh manifold, so that they are well within the fabrication limit. Figure 1 shows several architectural designs produced by our framework.

3. Numerical Solutions

In this section, we present a common strategy to solve the optimization problems in Section 2. We use problem (3) as an example, but a similar formulation also applies to problems (7) and (8) as discussed in Section 3.4. We introduce an efficient algorithm based on the augmented Lagrangian method (ALM) [Ber96] and variable splitting [Eck89]. Using auxiliary variables, we convert the minimization into a constrained optimization problem with separable objective function and linear constraints, which is then solved using ALM. Due to the structure of the converted problem, each iteration of ALM is decomposed into subproblems that can be solved efficiently.

3.1. Converted problem for ALM

To present the solution for (3), we use local modification of PQ meshes as an example. All faces of a PQ mesh need to be kept planar after modification. For solving (3), we convert it into a problem with separable objective function and linear constraints. First, the constraints are converted into linear constraints involving auxiliary variables. For a quad mesh $\mathbf{p} \in \mathbb{R}^{3n}$, the planarity of a face f_j with vertices $v_{j_1}, v_{j_2}, v_{j_3}, v_{j_4}$ is equivalent to the existence of three linearly dependent vectors $\mathbf{z}_{j,1}, \mathbf{z}_{j,2}, \mathbf{z}_{j,3} \in \mathbb{R}^3$ such that

$$\mathbf{p}_{j_1} - \mathbf{p}_{j_4} = \mathbf{z}_{j,1}, \quad \mathbf{p}_{j_2} - \mathbf{p}_{j_4} = \mathbf{z}_{j,2}, \quad \mathbf{p}_{j_3} - \mathbf{p}_{j_4} = \mathbf{z}_{j,3}. \tag{9}$$

Collecting this condition for all faces, we obtain an equivalent condition for the constraints

$$\mathbf{z} = \mathbf{M}\mathbf{p}. \tag{10}$$

Here matrix $\mathbf{M} \in \mathbb{R}^{9m \times 3n}$ maps vertex coordinates $\mathbf{p}$ to the left hand side vectors in (9) for all faces. $\mathbf{z} = [\mathbf{z}_{1,1}^T, \mathbf{z}_{1,2}^T, \mathbf{z}_{1,3}^T, \ldots, \mathbf{z}_{m,1}^T, \mathbf{z}_{m,2}^T, \mathbf{z}_{m,3}^T]^T \in \mathbb{R}^{9m}$ is an auxiliary variable subject to the linear dependence constraint of $\mathbf{z}_{j,1}$, $\mathbf{z}_{j,2}$, $\mathbf{z}_{j,3}$ for all j. Furthermore, in F_{handle} and F_{sparse}, we replace variable $\mathbf{d}$ with an auxiliary variable $\mathbf{y} \in \mathbb{R}^{3n}$ under a constraint $\mathbf{y} = \mathbf{d}$. Then (3) is converted to an equivalent problem

$$\begin{aligned} \min_{\mathbf{d},\mathbf{y},\mathbf{z}} \quad & \frac{\omega_{\mathrm{h}}}{2} \sum_{i \in \Gamma} \|\mathbf{y}_i - \tilde{\mathbf{d}}_i\|_2^2 + \frac{\omega_{\mathrm{s}}}{2} \sum_{i \notin \Gamma} \|\mathbf{y}_i\|_2 + \frac{\omega_{\mathrm{f}}}{2} \|\mathbf{E}\mathbf{d}\|_2^2 + \sigma(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{d}, \quad \mathbf{z} = \mathbf{M}(\mathbf{p}^0 + \mathbf{d}). \end{aligned} \tag{11}$$

where $\sigma(\mathbf{z})$ is an indicator function

$$\sigma(\mathbf{z}) = \begin{cases} 0 & \text{if } \forall j,\ \mathbf{z}_{j,1}, \mathbf{z}_{j,2}, \mathbf{z}_{j,3} \text{ are linear dependent,} \\ +\infty & \text{otherwise.} \end{cases}$$

In (11), the objective function is separable in $\mathbf{d}$, $\mathbf{y}$, $\mathbf{z}$, and the constraints can be written in linear form $\mathbf{x} = \mathbf{A}\mathbf{d} + \mathbf{b}$, where

$$\mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{I} \\ \mathbf{M} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}\mathbf{p}^0 \end{bmatrix},$$

with identity matrix $\mathbf{I} \in \mathbb{R}^{3n \times 3n}$. Now (11) has a form

$$\min_{\mathbf{d},\mathbf{x}} F(\mathbf{d},\mathbf{x}) \quad \text{s.t. } \mathbf{h}(\mathbf{d},\mathbf{x}) = \mathbf{0}, \tag{12}$$

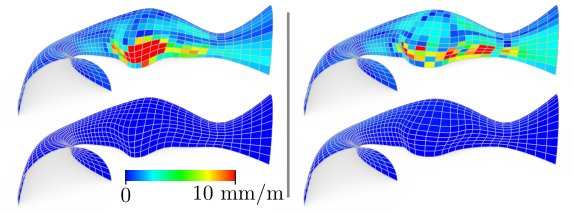


Figure 6: *Projection onto the PQ mesh manifold of two quad meshes designed with the subspace exploration approach. The color-coding shows the distance between the diagonals of each quad, normalized by the maximum edge length of the quad (with 10mm/m being the fabrication limit).*

where $\mathbf{h}(\mathbf{d},\mathbf{x}) = \mathbf{x} - \mathbf{A}\mathbf{d} - \mathbf{b} \in \mathbb{R}^{3n+9m}$. The *augmented Lagrangian function* for (12) is

$$L(\mathbf{d},\mathbf{x},\boldsymbol{\lambda};\mu) = F(\mathbf{d},\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{h}(\mathbf{d},\mathbf{x}) + \frac{\mu}{2}\|\mathbf{h}(\mathbf{d},\mathbf{x})\|_2^2,$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{3n+9m}$ is the multiplier vector, and $\mu > 0$ is the penalty parameter. ALM searches for a local saddle point of L by iteratively updating the *primal variables* $\mathbf{d},\mathbf{x}$ and the *dual variable* $\boldsymbol{\lambda}$ until convergence. In each iteration, new values $(\widehat{\mathbf{d}},\ \widehat{\mathbf{x}},\ \widehat{\boldsymbol{\lambda}},\ \widehat{\mu})$ are computed from the current values $(\overline{\mathbf{d}},\overline{\mathbf{x}},\overline{\boldsymbol{\lambda}},\overline{\mu})$ using the following steps:

1. *Primal update*: $(\widehat{\mathbf{d}},\widehat{\mathbf{x}}) = \operatorname{argmin}_{\mathbf{d},\mathbf{x}} L(\mathbf{d},\mathbf{x},\overline{\boldsymbol{\lambda}};\overline{\mu})$.
2. *Dual update*: $\widehat{\boldsymbol{\lambda}} = \overline{\boldsymbol{\lambda}} + \overline{\mu}\,\mathbf{h}(\widehat{\mathbf{d}},\widehat{\mathbf{x}})$.
3. *Penalty update*: choose $\widehat{\mu} \geq \overline{\mu}$.

In primal update, L is minimized over the primal variables only. In dual update, $\boldsymbol{\lambda}$ is updated according to current penalty parameter and the new value of constraint function $\mathbf{h}$. We design a primal update and penalty update suitable for our problem as explained below.

3.2. Primal update

For the primal update, the objective function $\widehat{L}(\mathbf{d},\mathbf{x}) = L(\mathbf{d},\mathbf{x},\overline{\boldsymbol{\lambda}};\overline{\mu})$ is nonconvex and nonsmooth. We perform alternating minimization to compute a stationary point of $\widehat{L}$. Starting from values $\mathbf{d}^{(0)} = \overline{\mathbf{d}}$, $\mathbf{x}^{(0)} = \overline{\mathbf{x}}$, we iteratively perform the following updates until convergence:

1. Fix $\mathbf{d}$, update $\mathbf{x}$: $\quad \mathbf{x}^{(l+1)} = \operatorname{argmin}_{\mathbf{x}} \widehat{L}(\mathbf{d}^{(l)},\mathbf{x})$.
2. Fix $\mathbf{x}$, update $\mathbf{d}$: $\quad \mathbf{d}^{(l+1)} = \operatorname{argmin}_{\mathbf{d}} \widehat{L}(\mathbf{d},\mathbf{x}^{(l+1)})$.

Updating x. This problem is separable in $\mathbf{y}$ and $\mathbf{z}$, so we can solve for them separately. The subproblem for $\mathbf{y}$ is

$$\min_{\mathbf{y}} \omega_{\mathrm{h}} \sum_{i\in\Gamma} \|\mathbf{y}_i - \widetilde{\mathbf{d}}_i\|_2^2 + \omega_{\mathrm{s}} \sum_{i\notin\Gamma} \|\mathbf{y}_i\|_2 + \mu\|\mathbf{y} - \mathbf{a}\|_2^2,$$

with $\mathbf{a} = \mathbf{d}^{(l)} - \overline{\boldsymbol{\lambda}}_{\mathbf{y}}/\overline{\mu}$ where $\overline{\boldsymbol{\lambda}}_{\mathbf{y}} \in \mathbb{R}^{3n}$ is the part of $\overline{\boldsymbol{\lambda}}$ that corresponds to constraints involving $\mathbf{y}$. This is separable in each vertex component $\mathbf{y}_i$, with closed-form solution

$$i \in \Gamma:\ \mathbf{y}_i = \frac{\omega_{\mathrm{h}}}{\omega_{\mathrm{h}}+\mu}\widetilde{\mathbf{d}}_i + \frac{\mu}{\omega_{\mathrm{h}}+\mu}\mathbf{a}_i, \tag{13}$$

$$i \notin \Gamma:\ \mathbf{y}_i = \begin{cases} \mathbf{0} & \text{if } \|\mathbf{a}_i\| \leq \frac{\omega_{\mathrm{s}}}{2\mu} \\ (1 - \frac{\omega_{\mathrm{s}}}{2\mu\|\mathbf{a}_i\|_2})\mathbf{a}_i & \text{otherwise} \end{cases}, \tag{14}$$

where $\mathbf{y}_i,\ \mathbf{a}_i \in \mathbb{R}^3$ are components of $\mathbf{y}$, $\mathbf{a}$ on the same position as $\mathbf{d}_i$ in $\mathbf{d}$. The subproblem for $\mathbf{z}$ is

$$\min_{\mathbf{z}} \sigma(\mathbf{z}) + \frac{\mu}{2}\|\mathbf{z} - \mathbf{c}\|_2^2, \tag{15}$$

where $\mathbf{c} = \mathbf{M}(\mathbf{p}^0 + \widetilde{\mathbf{d}}^{(l)}) - \overline{\boldsymbol{\lambda}}_{\mathbf{z}}/\overline{\mu}$ and $\overline{\boldsymbol{\lambda}}_{\mathbf{z}} \in \mathbb{R}^{9m}$ is the part of $\overline{\boldsymbol{\lambda}}$ that corresponds to $\mathbf{z}$. This means finding the closest projection from $\mathbf{c}$ to the set $\{\mathbf{z} \in \mathbb{R}^{9m} \mid \sigma(\mathbf{z}) = 0\}$. It is separable into subproblems each involving components $\mathbf{z}_{j,1},\mathbf{z}_{j,2},\mathbf{z}_{j,3}$ of one face f_j, with solution

$$\mathbf{z}_{j,k} = \mathbf{c}_{j,k} - \mathbf{N}_j(\mathbf{N}_j \cdot \mathbf{c}_{j,k}), \quad \text{for } j = 1,\ldots,m \text{ and } k = 1,2,3, \tag{16}$$

where $\mathbf{c}_{j,k} \in \mathbb{R}^3$ is the component of $\mathbf{c}$ with the same position as $\mathbf{z}_{j,k}$ in $\mathbf{z}$, and $\mathbf{N}_j$ is the right singular vector of matrix $[\mathbf{c}_{j,1},\mathbf{c}_{j,2},\mathbf{c}_{j,3}]^T$ for the smallest singular value.

Updating d. We update $\mathbf{d}$ by solving

$$\min_{\mathbf{d}} \ \omega_{\mathrm{f}}\|\mathbf{E}\mathbf{d}\|_2^2 + \mu(\|\mathbf{A}\mathbf{d} + \mathbf{b} - \mathbf{x} - \boldsymbol{\lambda}/\mu\|_2^2).$$

This reduces to a symmetric positive definite linear system

$$[\mu\mathbf{A}^T\mathbf{A} + \omega_{\mathrm{f}}\mathbf{E}^T\mathbf{E}]\mathbf{d} = \mu\mathbf{A}^T(\mathbf{x} - \mathbf{b} + \boldsymbol{\lambda}/\mu). \tag{17}$$

Termination criteria for primal update. The deviation of $(\mathbf{d}^{(l)},\mathbf{x}^{(l)})$ from a stationary point of $\widehat{L}$ can be measured with the norm of the *dual residual* $\mathbf{r}^{(l)} = \overline{\mu}\mathbf{A}(\mathbf{d}^{(l)} - \mathbf{d}^{(l-1)})$. We use $(\mathbf{d}^{(l)},\mathbf{x}^{(l)})$ as the solution $(\widehat{\mathbf{d}},\ \widehat{\mathbf{x}})$ for the primal update, when $\|\mathbf{r}^{(l)}\|_2$ is smaller than a given threshold, or l reaches the maximum number of iterations $l_{\max}$. In the following sections, we denote by $\widehat{\mathbf{r}}$ the dual residual associated with the solution $(\widehat{\mathbf{d}},\ \widehat{\mathbf{x}})$.

3.3. Penalty update and convergence of ALM

For an updated primal variable value $(\widehat{\mathbf{d}},\widehat{\mathbf{x}})$ to be close to a solution of (12), it is necessary that $\|\mathbf{h}(\widehat{\mathbf{d}},\widehat{\mathbf{x}})\|_2$ and $\|\widehat{\mathbf{r}}\|_2$ are both small. We accept $(\widehat{\mathbf{d}},\widehat{\mathbf{x}})$ as a solution when $\|\mathbf{h}(\widehat{\mathbf{d}},\widehat{\mathbf{x}})\|_2 \leq \varepsilon_{\mathrm{p}}$ and $\|\widehat{\mathbf{r}}\|_2 \leq \varepsilon_{\mathrm{d}}$, where ε_{p} and ε_{d} are tolerance values.

The penalty parameter μ has an influence on the primal update: on one hand, a larger penalty parameter μ tends to produce updates with smaller norms of $\mathbf{h}$; on the other hand, if μ is too large, the primal update problem may become ill-conditioned. We start with a relatively small μ, and increase it by a fixed ratio each time. μ is increased only when the decrease of $\|\mathbf{h}(\widehat{\mathbf{d}},\widehat{\mathbf{x}})\|_2$ is too small compared with the decrease of $\|\widehat{\mathbf{r}}\|_2$ and the new value of μ does not exceed a given upper bound. Using this strategy, every limit point is a KKT point of the original problem [Tse01, ABMS07].

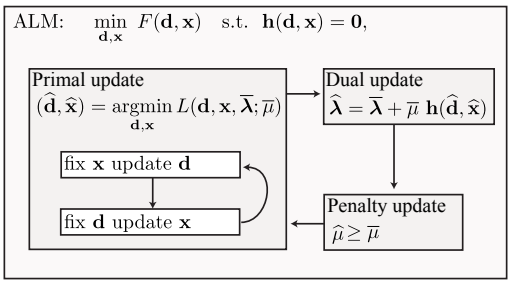


Figure 7: *An overview of the numerical solving approach described in Section 3.*

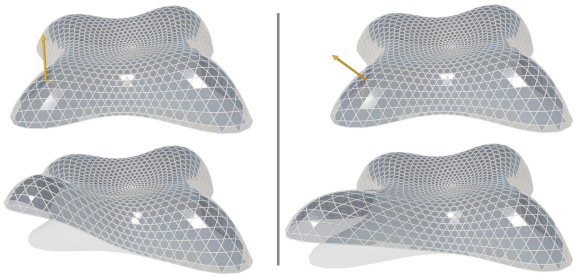


Figure 8: *Local modifications of a planar tri-hex model. The input displacements are shown on the top.*

3.4. General pattern

The above framework can be adapted to solve the subspace exploration (7) and projection problems (8). The key step in applying this approach is to convert the original problem into an equivalent one, of the form (12) with separable objective function and linear constraints. The converted problem can then be solved as shown in Figure 7. Problem (8) can be solved without introducing auxiliary variables for nonsmooth terms. The solution to (7) is given in Appendix B.

Remark. When we convert constraints into an equivalent linear condition (10), the constraints are transferred to auxiliary variables. The primal update subproblem (15) is then a projection onto the manifold of auxiliary variables that satisfy the constraints. For many shape constraints in geometry processing, we can derive their equivalent conditions with efficient projection operators. For a thorough discussion of projection operators for shape constraints, the reader is referred to [BDS*12].

3.5. Efficiency

Our algorithm solves problem (3) efficiently. The primal update steps for auxiliary variables ((13), (14), (16)) are independent and can be easily parallelized. The $3n \times 3n$ linear system (17) can be decomposed into three $n \times n$ symmetric positive definite sparse linear systems for each of the x, y and z components, and solved with the same coefficient matrix. This coefficient matrix depends on μ only, so for a particular value μ we can prefactorize the matrix and solve the system efficiently. Moreover, since μ starts from a specified value and increases exponentially, there is only a limited number of values for μ before it reaches a predefined upper bound. We can therefore prefactorize all possible coefficient matrices. Our algorithm is well suited for GPUs and multicore CPUs. For handle-based deformation of a PQ mesh with 1K vertices (>10K variables) on a quad core CPU, our algorithm converges within a few seconds to a mesh where all constraints are satisfied up to fabrication tolerance. For subspace exploration, the computation time for each basis vector is similar to a handle-based deformation. Afterwards, the exploration itself is realtime. The computation time for projection onto constrained mesh manifold is shorter than handle-based modification, due to the absence of auxiliary variables for the mixed ℓ_2/ℓ_1 norm.

3.6. Discussion

Our method is closely related to the Alternating Direction Method of Multipliers (ADMM) [BPC*11], which is popular for convex optimization problems with separable objective function and linear constraints. ADMM is an augmented Lagrangian algorithm where the primal update is done by only one iteration of alternating minimization. For nonconvex problems, ADMM may fail to converge because it can produce primal updates far from local minima. Our primal update performs multiple iterations of alternating minimization and terminates when it is close enough to a local minimum. This leads to more accurate results for the primal update, enabling our method to handle nonconvex problems.

4. Limitation and Future Work

Our handle-based approach can be directly generalized to inequality constraints. However, the subspace exploration approach is not applicable in this case. In the future, we want to investigate the exploration of meshes with inequality constraints. Our method does not detect or prevent self-intersections. Nevertheless, the fairing terms in our optimizations tend to avoid self-intersections for deformations that are not too large. As a future work, we would like to introduce a collision avoidance constraint to prevent the surface to self-intersect or to intersect surrounding elements. Additionally, our approach keeps a fixed mesh topology during the exploration. Changing the topology may produce a better result, and this is an interesting avenue of research. Our framework enforces sparsity of the displacement vector using a mixed ℓ_2/ℓ_1 norm regularizer. However, the mixed ℓ_2/ℓ_1 norm regularization may not completely prevent small vertex displacements. An alternative approach is to use a l_0-norm based sparsity-inducing penalty [XLXJ11]. But the convergence guarantee given by [ABMS07] does not hold in

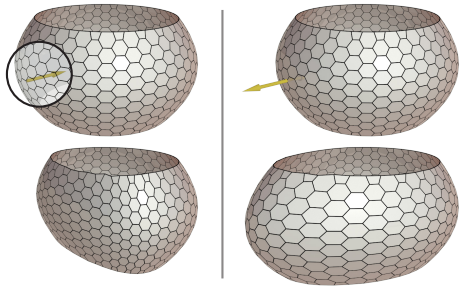


Figure 9: *Local modifications of a planar hexagonal model, with input displacements shown on the top. Due to fewer degrees of freedom, the displacements affect a larger region.*

this case because the l_0-norm is not continuous. In the future, we would like to investigate other sparsity-inducing norms and continuous relaxations for the l_0-norm. We also noticed that by removing the correlation term from the subspace basis calculation, the system proposed many local edits that are not in the region of the original deformation. While this is not applicable to subspace exploration, this provides an interesting avenue for future work.

5. Conclusion

We introduced a novel framework for exploring local modifications of globally constrained meshes. First, we presented an approach to compute smooth and local deformations of a constrained mesh using handles. Second, we showed how a linear subspace of high quality local deformations can be constructed using a correlation function. This subspace enriches the solutions of the first step and can be explored in realtime by a user. Finally, we introduced an algorithm to reproject the deformed meshes generated in the second step onto the constrained mesh manifold. For each of these steps we presented a computationally tractable algorithm making use of variable splitting. We applied our framework on the design of meshes with planar faces that are of great interest for architectural geometry.

Acknowledgements. The authors thank Helmut Pottmann, Amir Vaxman, Chunlin Wu, José Mario Martínez, Zhouchen Lin, David Russell Luke and the reviewers for their valuable comments. The models are provided by Asymptote Architecture and Waagner Biro (Figure 5), Zaha Hadid Architects and Amir Vaxman (Figure 8). The model in Figure 3 and 4 is taken from [PSB*08]. This work has been supported by Swiss National Science Foundation (SNSF) grants 20PA21L_129607 and 200021_137626, ERC Starting Grant 257453 COSYM, and Austrian Science Fund (FWF) grant P23735-N13.

References

[ABMS07] ANDREANI R., BIRGIN E. G., MARTÍNEZ J. M., SCHUVERDT M. L.: On augmented lagrangian methods with general lower-level constraints. *SIAM J. Optim. 18*, 4 (2007), 1286–1309. 7, 8

[ACSD*03] ALLIEZ P., COHEN-STEINER D., DEVILLERS O., LÉVY B., DESBRUN M.: Anisotropic polygonal remeshing. *ACM Trans. Graph. 22*, 3 (2003). 2

[ASGCO10] AVRON H., SHARF A., GREIF C., COHEN-OR D.: ℓ_1-sparse reconstruction of sharp point set surfaces. *ACM Trans. Graph. 29*, 5 (2010), 135:1–135:12. 3

[BDS*12] BOUAZIZ S., DEUSS M., SCHWARTZBURG Y., WEISE T., PAULY M.: Shape-up: Shaping discrete geometry with projections. *Comput. Graph. Forum 31*, 5 (2012), 1657–1667. 2, 8

[Ber96] BERTSEKAS D. P.: *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996. 3, 6

[BJMO12] BACH F. R., JENATTON R., MAIRAL J., OBOZINSKI G.: Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning 4*, 1 (2012), 1–106. 2, 3

[BPC*11] BOYD S., PARIKH N., CHU E., PELEATO B., ECKSTEIN J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning 3*, 1 (2011), 1–122. 8

[BZK09] BOMMES D., ZIMMER H., KOBBELT L.: Mixed-integer quadrangulation. *ACM Trans. Graph. 28*, 3 (2009). 2

[CP11] COMBETTES P. L., PESQUET J.-C.: Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer, 2011. 2

[CRT06] CANDÈS E. J., ROMBERG J., TAO T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory 52*, 2 (2006), 489–509. 3

[CW08] CANDÈS E. J., WAKIN M. B.: An introduction to compressive sampling. *IEEE Signal Process. Mag. 25*, 2 (2008). 2

[DGP09] DOBREV V., GUERMOND J.-L., POPOV B.: Surface reconstruction via L_1-minimization. In *Numerical Analysis and Its Applications*. 2009, pp. 32–43. 3

[DPW11] DENG B., POTTMANN H., WALLNER J.: Functional webs for freeform architecture. *Comput. Graph. Forum 30*, 5 (2011), 1369–1378. 2

[Eck89] ECKSTEIN J.: *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis, MIT, 1989. 3, 6

[EKB10] ELDAR Y. C., KUPPINGER P., BOLCSKEI H.: Block-sparse signals: Uncertainty relations and efficient recovery. *IEEE Trans. Signal Process. 58*, 6 (2010), 3042–3054. 2

[EKS*10] EIGENSATZ M., KILIAN M., SCHIFTNER A., MITRA N. J., POTTMANN H., PAULY M.: Paneling architectural freeform surfaces. *ACM Trans. Graph. 29* (2010). 2

[EM09] ELDAR Y. C., MISHALI M.: Robust recovery of signals from a structured union of subspaces. *IEEE Trans. Inf. Theory 55*, 11 (2009), 5302–5316. 2, 3

[FLHCO10] FU C.-W., LAI C.-F., HE Y., COHEN-OR D.: K-set tilable surfaces. *ACM Trans. Graph. 29*, 4 (2010). 2

[GZL12] GAO L., ZHANG G., LAI Y.: L_p shape deformation. *Science China Information Sciences 55*, 5 (2012), 983–993. 3

[HK12] HABBECKE M., KOBBELT L.: Linear analysis of nonlinear constraints for interactive geometric modeling. *Comput. Graph. Forum 31*, 2 (2012), 641–650. 2

[Hof10] HOFFMANN T.: On local deformations of planar quad-meshes. In *Mathematical Software – ICMS 2010*. 2010. 2, 4

[HSTP11] HILDEBRANDT K., SCHULZ C., TYCOWICZ C. V., POLTHIER K.: Interactive surface modeling using modal analysis. *ACM Trans. Graph. 30*, 5 (2011), 119:1–119:11. 2

[Lav02] LAVERY J. E.: Shape-preserving, multiscale interpolation by univariate curvature-based cubic L_1 splines in cartesian and polar coordinates. *Comput. Aided Geom. Design 19*, 4 (2002), 257–273. 3

[LPW*06] LIU Y., POTTMANN H., WALLNER J., YANG Y.-L., WANG W.: Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph. 25*, 3 (2006). 2

[LXW*11] LIU Y., XU W., WANG J., ZHU L., GUO B., CHEN F., WANG G.: General planar quadrilateral mesh design using conjugate direction field. *ACM Trans. Graph. 30*, 6 (2011). 2

[NW06] NOCEDAL J., WRIGHT S. J.: *Numerical Optimization*, 2nd ed. Springer, 2006. 3

[PSB*08] POTTMANN H., SCHIFTNER A., BO P., SCHMIEDHOFER H., WANG W., BALDASSINI N., WALLNER J.: Freeform surfaces from single curved panels. *ACM Trans. Graph. 27*, 3 (2008). 9

[Rus11] RUSTAMOV R. M.: Multiscale biharmonic kernels. *Comput. Graph. Forum 30*, 5 (2011). 3

[Tse01] TSENG P.: Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications 109*, 3 (2001), 475–494. 7

[Vax12] VAXMAN A.: Modeling polyhedral meshes with affine maps. *Comput. Graph. Forum 31*, 5 (2012), 1647–1656. 2

[XLXJ11] XU L., LU C., XU Y., JIA J.: Image smoothing via l0 gradient minimization. *ACM Trans. Graph. 30*, 6 (2011). 8

[YYPM11] YANG Y.-L., YANG Y.-J., POTTMANN H., MITRA N. J.: Shape space exploration of constrained meshes. *ACM Trans. Graph. 30*, 6 (2011). 2, 3, 4, 5

[ZSW10] ZADRAVEC M., SCHIFTNER A., WALLNER J.: Designing quad-dominant meshes with planar faces. *Comput. Graph. Forum 29*, 5 (2010), 1671–1679. 2

[ZTY*12] ZHAO X., TANG C.-C., YANG Y.-L., POTTMANN H., MITRA N. J.: Intuitive design exploration of constrained meshes. In *Advances in Architectural Geometry 2012* (2012). 2

Appendix A: Proof of Proposition 1

For (a), using the triangle inequality (6) we have

$$D(\mathbf{u},\mathbf{v}) = 0 \Leftrightarrow \|x\mathbf{u}+y\mathbf{v}\|_{2,1} \equiv |x|\|\mathbf{u}\|_{2,1} + |y|\|\mathbf{v}\|_{2,1} \Leftrightarrow \xi(\mathbf{u},\mathbf{v}) = 0.$$

For (b), note that for $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{R}^3$ and curve $C: x^2+y^2=1$,

$$\begin{aligned}\|\mathbf{u}_i\|_2\|\mathbf{v}_i\|_2 &= \frac{1}{4}\oint_C \left[(\|x\mathbf{u}_i\|_2 + \|y\mathbf{v}_i\|_2)^2 - \|x\mathbf{u}_i + y\mathbf{v}_i\|_2^2\right] ds \\ &= \frac{1}{4}\oint_C A_i(x,y)B_i(x,y)\, ds,\end{aligned}$$

where $A_i(x,y) = \|x\mathbf{u}_i\|_2 + \|y\mathbf{v}_i\|_2 + \|x\mathbf{u}_i + y\mathbf{v}_i\|_2$, $B_i(x,y) = |x|\|\mathbf{u}_i\|_2 + |y|\|\mathbf{v}_i\|_2 - \|x\mathbf{u}_i + y\mathbf{v}_i\|_2$. From the mean value theorem, there exist x_i, y_i with $x_i^2 + y_i^2 = 1$ such that

$$\oint_C A_i(x,y)B_i(x,y)\, ds = A_i(x_i,y_i)\oint_C B_i(x,y) ds.$$

Since $\oint_C B_i(x,y)ds \geq 0\ \forall i$, and $D(\mathbf{u},\mathbf{v}) = \sum_i \oint_C B_i(x,y)ds$, $\xi(\mathbf{u},\mathbf{v}) = \frac{1}{4}\sum_i \oint_C A_i(x,y)B_i(x,y)ds$, it follows that

$$\begin{aligned}\xi(\mathbf{u},\mathbf{v}) &= \frac{1}{4}\sum_i \oint_C A_i(x,y)B_i(x,y)ds \\ &\leq \frac{1}{4}\left[\max_i A_i(x_i,y_i)\right]\sum_i \oint_C B_i(x,y)ds \\ &\leq \frac{1}{4}\left[\max_i \max_{x^2+y^2=1} A_i(x,y)\right] D(\mathbf{u},\mathbf{v}).\end{aligned}$$

Appendix B: Numerical solution to (7)

We reformulate the problem in the form of (12)

$$\begin{aligned}\min_{\mathbf{t},\mathbf{y},\mathbf{q},\mathbf{m}}\ & Q(\mathbf{t}) + \frac{\beta_s}{2}\|\mathbf{y}\|_{2,1} - \frac{\beta_c}{2}C(\mathbf{y}) + \sigma_1(\mathbf{q}) + \sigma_2(\mathbf{m}) \\ \text{s.t.}\ & \mathbf{y}=\mathbf{t},\ \mathbf{q}=\mathbf{t},\ \mathbf{m}=\mathbf{t},\ \mathbf{J}\mathbf{t}=\mathbf{0},\end{aligned}$$

where $\mathbf{y},\mathbf{q},\mathbf{m}$ are auxiliary variable, and $\sigma_1(\mathbf{q})$, $\sigma_2(\mathbf{m})$ are indicator functions for conditions $\|\mathbf{q}\|_2 = 1$ and $\mathbf{B}^T\mathbf{m} = \mathbf{0}$, respectively. We solve it using the augmented Lagrangian method in Section 3. In the primal update, we first fix $\mathbf{t}$ and update ($\mathbf{y}$, $\mathbf{q}$, $\mathbf{m}$), then fix ($\mathbf{y}$, $\mathbf{q}$, $\mathbf{m}$) and update $\mathbf{t}$:

Updating y, q, m. This involves the following subproblems

$$\min_{\mathbf{y}} \beta_s\|\mathbf{y}\|_{2,1} - \beta_c C(\mathbf{y}) + \mu\|\mathbf{y}-\mathbf{a}^{\mathbf{y}}\|_2^2,$$

$$\min_{\mathbf{q}} \sigma_1(\mathbf{q}) + \mu\|\mathbf{q}-\mathbf{a}^{\mathbf{q}}\|_2^2, \quad \min_{\mathbf{m}} \sigma_2(\mathbf{m}) + \mu\|\mathbf{m}-\mathbf{a}^{\mathbf{m}}\|_2^2,$$

where μ is the penalty parameter, $\mathbf{a}^{\mathbf{y}} = \mathbf{t} - \boldsymbol{\lambda}^{\mathbf{y}}/\mu$, $\mathbf{a}^{\mathbf{q}} = \mathbf{t} - \boldsymbol{\lambda}^{\mathbf{q}}/\mu$, $\mathbf{a}^{\mathbf{m}} = \mathbf{t} - \boldsymbol{\lambda}^{\mathbf{m}}/\mu$, with $\boldsymbol{\lambda}^{\mathbf{y}}$, $\boldsymbol{\lambda}^{\mathbf{q}}$, $\boldsymbol{\lambda}^{\mathbf{m}}$ being the components of the multiplier vector for the constraints involving $\mathbf{y}$, $\mathbf{q}$, $\mathbf{m}$ respectively. The solutions are

$$\mathbf{y}_i = \begin{cases} \mathbf{0} & \text{if } \|\mathbf{a}_i^{\mathbf{y}}\| \leq \frac{C^i}{2\mu} \\ (1 - \frac{C^i}{2\mu\|\mathbf{a}_i^{\mathbf{y}}\|_2})\mathbf{a}_i^{\mathbf{y}} & \text{otherwise.}\end{cases}$$

$$\mathbf{q} = \frac{\mathbf{a}^{\mathbf{q}}}{\|\mathbf{a}^{\mathbf{q}}\|_2}, \quad \mathbf{m} = (\mathbf{I} - \mathbf{B}\mathbf{B}^T)\mathbf{a}^{\mathbf{m}},$$

where subscript i indicates components for the i-th vertex, $C^i = \beta_s - \beta_c(\|\bar{\mathbf{d}}_i\|_2 + \sum_{1\leq j<k}\|\mathbf{t}_i^j\|_2)$. Note that the $\mathbf{y}$-subproblem has a unique solution only if $C^i \geq 0$ for all i. It is therefore required that $\beta_s \geq \beta_c \cdot \max_i(\|\bar{\mathbf{d}}_i\|_2 + \sum_{1\leq j<k}\|\mathbf{t}_i^j\|_2)$.

Updating t. We minimize the following function about $\mathbf{t}$

$$Q(\mathbf{t}) + \frac{\mu}{2}(\|\mathbf{t}-\mathbf{a}^{\mathbf{y}}\|_2^2 + \|\mathbf{t}-\mathbf{a}^{\mathbf{q}}\|_2^2 + \|\mathbf{t}-\mathbf{a}^{\mathbf{m}}\|_2^2 + \|\mathbf{J}\mathbf{t} + \boldsymbol{\lambda}^{\mathbf{t}}/\mu\|_2^2),$$

where $\boldsymbol{\lambda}^{\mathbf{t}}$ is the component of the multiplier vector for the constraint $\mathbf{J}^T\mathbf{t} = \mathbf{0}$. This reduces to solving a sparse (if $\mathbf{E}$ is sparse) symmetric positive definite linear system

$$[\mu(3\mathbf{I} + \mathbf{J}^T\mathbf{J}) + \omega_f\mathbf{E}^T\mathbf{E} + \omega_h\mathbf{D}]\mathbf{d} = \mu(\mathbf{a}^{\mathbf{y}} + \mathbf{a}^{\mathbf{q}} + \mathbf{a}^{\mathbf{m}}) - \mathbf{J}^T\boldsymbol{\lambda}^{\mathbf{t}},$$

where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = 1$ for $i \in \Psi$ and $\mathbf{D}_{ii} = 0$ otherwise.