

On Linear Spaces of Polyhedral Meshes

Roi Poranne, Renjie Chen, and Craig Gotsman

Abstract—Polyhedral meshes (PM)—meshes having planar faces—have enjoyed a rise in popularity in recent years due to their importance in architectural and industrial design. However, they are also notoriously difficult to generate and manipulate. Previous methods start with a smooth surface and then apply elaborate meshing schemes to create polyhedral meshes approximating the surface. In this paper, we describe a reverse approach: given the topology of a mesh, we explore the space of possible planar meshes having that topology. Our approach is based on a complete characterization of the maximal linear spaces of polyhedral meshes contained in the curved manifold of polyhedral meshes with a given topology. We show that these linear spaces can be described as nullspaces of differential operators, much like harmonic functions are nullspaces of the Laplacian operator. An analysis of this operator provides tools for global and local design of a polyhedral mesh, which fully expose the geometric possibilities and limitations of the given topology.

Index Terms—Polyhedral meshes

1 INTRODUCTION

PM's, i.e. meshes with planar faces, have gained popularity in recent years due to several new methods that render their construction relatively easy. Typically, a designer creates a traditional free-form surface and then applies a meshing scheme that generates an approximating mesh consisting of only planar faces. Naturally, the focus of these schemes, e.g. [15], [28], is to generate good approximations, and this is done using very specific (regular) types of mesh topologies. It may well be that these are the *only* topologies that can approximate general smooth surfaces well. However, the topology of the mesh itself has its own artistic value: a triangular meshing of a surface will not have the same "look" as a quad or hex meshing. Yet, as mentioned, the cases where a smooth surface can be faithfully meshed into a PM are limited. Hence, we propose a different strategy: instead of constructing the final PM based on a design of a surface, we explore the space of possible PM's with a given topology. Such a PM is called a *realization* of the topology.

Our goal is to gain quick and intuitive understanding of the manifold that is attached to mesh topology designed by the user. Our approach is based on the observation that the complicated manifold of PM's with a given topology can be decomposed into overlapping, linear spaces, each of which is *maximal*—adding a base PM to the space will introduce non-PMs to the space. The advantage of linear spaces lies in the simplicity of exploring them: PM's in such a space may be designed by forming linear combinations of a spanning set of basic PM's. The disadvantage is that the dimensionality of these spaces is much smaller than that of the complete manifold of PM's. Thus, showing that they are indeed

maximal is crucial. By switching between spaces, it is possible to reach any PM in the manifold. We will refer to the PM's of a spanning set simply as *shapes*.

The use of linear spaces can be incorporated into well-known mesh deformation methods, such as as-rigid/similar-as-possible [12]. In addition, we propose three types of shapes aiming at different levels of design, exposing the possibilities and limitations for deforming a given PM; the reason for their names will subsequently become clear. *Eigenshapes* are globally smooth shapes at different *frequencies* akin to the eigenvectors of the Laplacian. *Sparse shapes* are based on the smallest groups of vertices that can move together without impairing the planarity of the faces of the PM. Finally, *fundamental shapes* allow a single vertex to be moved with minimal change to other vertices while preserving planarity.

1.1 Related work

Meshing and planarization. The creation of polyhedral meshes is an active field of research. The most common problem is to mesh, or remesh, a free-form into a PM. The approach used by Cohen-Steiner et al. [5] is to try to fit a limited number of planes to the surface and then intersect them. The surface is first partitioned into a user-defined number of *almost* flat regions, for each of which a plane is fitted. These planes, called *shapes proxies*, will generally not have well-defined intersection points. Thus, the faces they produce are only *close* to being planar. Cutler and Whiting [6] added an iterative optimization process to the algorithm that guarantees that the resulting faces are planar.

In both of these systems, the user can control the number of faces and their density in the result, but cannot dictate the mesh topology (its edge structure), which can essentially be arbitrary. While this is not necessarily a drawback, in some cases a regular mesh is desirable. Liu et al. [15] and Wang et al. [26] showed how a surface may be meshed into a planar quad-dominant (PQ) mesh and a planar hexagonal (P-Hex) mesh, respectively. The two algorithms are quite similar: an almost polyhedral mesh is first generated from

• The authors are with Technion—Israel Institute of Technology, Haifa, Israel. E-mail: {roip, renjie, gotsman}@cs.technion.ac.il.

Manuscript received 17 Mar. 2013; revised 3 Nov. 2014; accepted 19 Dec. 2014. Date of publication 11 Jan. 2015; date of current version 1 Apr. 2015.

Recommended for acceptance by B. E. E. Levy.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TVCG.2014.2388205

the surface, based on differential geometric entities (PQ meshes are based on conjugate networks and P-Hex meshes on the Dupin indicatrix. Zadavec et al. [28] and Liu et al. [16] elaborated on how to design better conjugate networks.) A subsequent step involves the planarization of the result: a non-linear optimization, where the vertices of the mesh are repositioned to make the faces planar. This latter step seems to dominate the runtime, and does not scale well with mesh size. Alexa and Wardetzky [2] demonstrated the construction of a Laplacian operator on non-triangular meshes. As a side effect of their construction, they were able to devise a related operator that measures the planarity of faces. With this new operator, they obtained a *planarizing flow*, that is, a geometric flow that flattens faces. In Poranne et al. [17], a local/global based alternating algorithm was used to solve the planarization problem very efficiently. The improved performance enables interactive deformation of PM's.

Mesh deformation. The problem of editing and deforming mesh geometry is one of the most studied topics in geometry processing. Most mesh deformation methods are intended to work exclusively with triangle meshes. See [4] for a thorough introduction. These methods may be classified into two types, based on the type of user interaction employed. In the first type, the user directly modifies the surface using one of common design metaphors. The most relevant to us are the handle-based methods (e.g. [3], [12], [20], [21]), where the user controls the deformation by moving a small number of points on the mesh. These points generate constraints for an optimization problem, whose solution is the deformed mesh. Other common design metaphors includes skeleton-based and cage-based. Jacobson et al. [13] noted the differences between these methods and provided a hybrid method incorporating both. More intricate approaches for mesh deformation use direct control of the mesh normal and curvature instead of vertex positions [7], [8].

Handle-based deformation has also been used in the context of PM's. In [27], the *manifold of polyhedral meshes* was discussed in detail. The idea was to approximate this manifold by an osculate, which is much easier to explore. In this framework, deformation of a PM using positional constraints was made possible; however, computing the osculate is time-consuming and the deformation only *approximately* preserves the planarity of faces. Zhao et al. [29] use the same technique to derive a *curve-based* deformation.

In [25], Vaxman described a linear space of PM's by allowing affine transformations per face. He proposed to use the space of affine transformations instead of the entire manifold, simplifying the math considerably. In fact, this space is a special case of the linear spaces to be described in this paper. The main drawback of using this space is its small number of degrees of freedom (dimension). For example, the number of degrees of freedom of a quad PM is about half the size of its boundary, so when the mesh has no boundary, only the trivial, global, transformations are possible (i.e. global affine maps). Hexagonal PM's will have only 12 degrees of freedom, regardless of the existence of a boundary. In other words, specifying the geometry of four vertices of a PM with hexagonal topology uniquely

determines the rest of the PM. Pottmann et al. [18] described another linear space of PM's, called parallel meshes. It is also a special case of the spaces to be described in this paper.

A second type of mesh deformation is *indirect*. These include various methods that improve the quality of a mesh, such as smoothing and enhancing features. More relevant to us are methods that are used to add variation to a mesh, or to create a collection of meshes based on a single mesh (e.g. [24]). Yang et al. [27] have also contributed an indirect deformation approach, by designing a user interface which allows to traverse the osculate with ease. In this paper we propose eigenshapes as a way of indirectly adding variation to a PM.

1.2 Contribution and Overview

We extend the work of Vaxman [25] by providing a theoretical characterization of the maximal linear spaces covering the manifold of PM's. In Section 2 we discuss linear spaces of PM's in detail, characterize all of the possible maximal linear spaces, and show how to construct them. In Section 3 we employ this theory to describe a number of meaningful ways for editing PM's. In Section 4 we discuss practical consideration and limitation of this methodology.

2 LINEAR SUBSPACES

Preliminaries. In our context, a mesh is defined by a list of vertex geometry and a list of faces. The vertex geometry can be arranged in a $3 \times N_V$ matrix, where N_V is the number of vertices. We will usually denote this matrix by an uppercase letter, such as X or Y and the positions of the vertices in boldface. For example, the vertex geometry is given by

$$X = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_V}),$$

where \mathbf{x}_i are column 3-vectors. We denote by $F = \{\mathbf{f}_j\}_{j=1}^{N_F}$ the set of faces of the mesh, where each face is described as an ordered (oriented) list of vertices. F will be common to meshes that share the same connectivity, and we will refer to them only by their vertex matrices. We will also denote the coordinates of the vertices of the face $\mathbf{f} \in F$, which is a submatrix of X , by $X_{\mathbf{f}}$.

The generation of subspace V requires a single realization Y of F , and that each face is designated a *type* out of three types. We denote this designation by $T = \{t_j\}_{j=1}^{N_F}$, where t_j encode the types to be defined later. We call the triplet $\mathcal{C} = (Y, F, T)$ the *configuration* for V . Our first task is to find a description of a linear subspace of PM's based on the configuration. This will be given as the solution space of

$$M_{\mathcal{C}} \text{vec}(X) = 0, \tag{1}$$

where $M_{\mathcal{C}}$ is a matrix that depends on the configuration \mathcal{C} only, and $\text{vec}(X)$ is a (column) vectorization of X . In other words, the linear subspace will be the null space of $M_{\mathcal{C}}$. In the following, we will also encounter intermediate systems such as $M_L X = 0$ and $X M_R = 0$ for some matrices M_L, M_R . These systems are substantially different—the former operates on each face separately while the latter operates on each coordinate across all faces—but they can both be written in the form of Eq. (1).

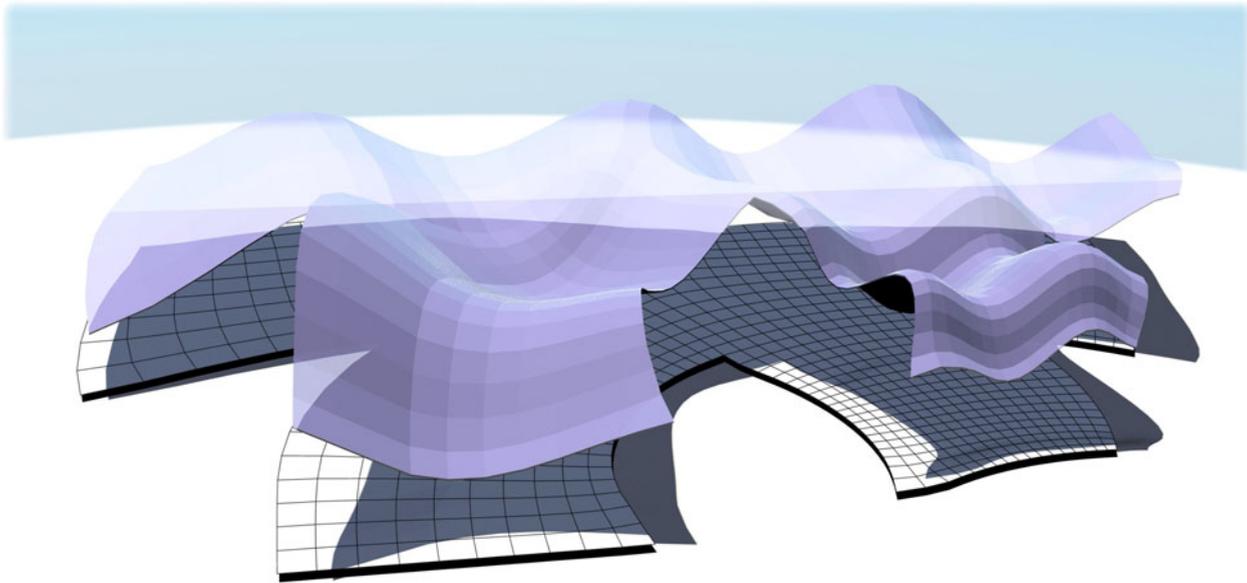


Fig. 1. A polyhedral mesh constructed from a planar graph using maximal linear subspaces.

Manifolds of meshes. When two meshes have the same topology, their linear combination can be defined simply as a linear combination of their vertex geometries. In other words, two meshes X and Y span a linear subspace of meshes defined by

$$\alpha X + \beta Y, \quad \alpha, \beta \in \mathbb{R}.$$

We can consider the set of all meshes with N_V vertices and a given topology to be vectors in \mathbb{R}^{3N_V} . The dimension of this space is $3N_V$ and is isomorphic to \mathbb{R}^{3N_V} .

Linearly combining two meshes is meaningful because the set of all possible meshes (with a given topology) is a linear space. PM's, on the other hand, reside in a complicated, curved submanifold in this space. Linearly combining two PM's will usually not result in a PM, which is the cause of many of the problems in designing them. It so happens that the manifold of PM's may be covered by linear submanifolds, which we discuss next. By replacing the non-linear constraints defining the manifold of PM by linear ones, many of the problems related to PM design disappear. We emphasize an important point that the dimensions of the linear subspaces are much smaller than $3N_V$, so making sure that a linear subspace contains the largest possible part of the space is crucial. We make a formal definition as follows:

Definition 1. A linear subspace V of polyhedral meshes is called maximal if for any $X \notin V$ the space $V + \text{span}\{X\}$ contains non-polyhedral meshes.

Centering. To make the discussion easier, all the faces of the mesh will be centered, namely, their centroids will be moved to the origin. This will not harm our claims, since centering is a linear operation that preserves planarity. Indeed, if X_f are the vertices of an uncentered face f , then,

$$X_f^c = X_f J = X_f \left(I - \frac{1}{n} E \right), \quad (2)$$

is centered, where $J := I - \frac{1}{n} E$ is the centering matrix, I and E are the identity matrix and matrix of ones respectively,

both are $n \times n$ square matrices, and n is the number of vertices in the face.

Conditions for linear subspaces. To investigate the linear subspaces of PM's we first examine the linear subspaces of much simpler entities: planar polygons. We will assume that the polygons are not degenerate, since while degenerate polygons have a place in this theory, they do not appear in practice, and therefore cause an unnecessary complication.

Lemma 1. Let $F = \{f\}$ be a mesh with a single face f with $k > 3$ vertices. Let X and Y be two $3 \times k$ matrices representing two geometries of F , both being planar k -gons in \mathbb{R}^3 , and let X^c and Y^c be their centerings. Let \mathbf{n}_X and \mathbf{n}_Y be the unit (1×3) row vectors normal to the planes defined by X and Y respectively. Then X and Y span a linear subspace of planar polygons iff at least one of the following holds:

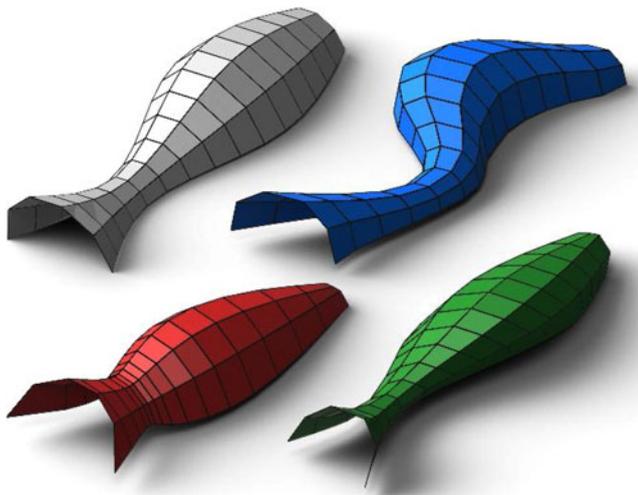


Fig. 2. Deforming the (gray) PM in various linear subspaces. Each result is not achievable in the other two subspaces.

- Relationship of type 1. X^c is an affine transformation of Y^c : $X^c = AY^c$ for some 3×3 matrix A .
- Relationship of type 2. There exists a scalar d such that
$$\mathbf{n}_Y X^c = d\mathbf{n}_X Y^c \quad (3)$$

Proof. By definition,

$$\mathbf{n}_X X^c = \mathbf{n}_Y Y^c = \mathbf{0}, \quad (4)$$

where $\mathbf{0}$ is the zero vector. First assume that X and Y span a linear subspace of planar polygons, which also means so does X^c and Y^c . Then every linear combination of X and Y (and X^c and Y^c) defines a plane and therefore has a normal vector. In other words, for each α and β , there exists a vector $\mathbf{n}_{\alpha,\beta}$ such that

$$\mathbf{n}_{\alpha,\beta}(\alpha X^c + \beta Y^c) = \mathbf{0}. \quad (5)$$

Consider the set of normal vectors $\mathbf{n}_{\alpha,\beta}, \alpha, \beta \in R$. There are two possibilities for the dimensionality of the set:

Case 1. The set has three dimensions. Then there exist $\mathbf{n}_k = \mathbf{n}_{\alpha_k, \beta_k}, k = 1, 2, 3$ such that the \mathbf{n}_k 's are not collinear. We denote by $[\alpha\mathbf{n}]$ the 3×3 matrix whose rows are $\alpha_k \mathbf{n}_k$ and similar for $[\beta\mathbf{n}]$. By eq. (5), we can write

$$[\alpha\mathbf{n}]X^c = -[\beta\mathbf{n}]Y^c. \quad (6)$$

Since the \mathbf{n}_i 's are linearly independent, we can invert $[\alpha\mathbf{n}]$ and get

$$X^c = -[\alpha\mathbf{n}]^{-1}[\beta\mathbf{n}]Y^c. \quad (7)$$

Hence, X^c is an affine transformation of Y^c , which is a relationship of type 1.

Case 2. The set of normals has dimension less than 3. Then this set must be spanned by \mathbf{n}_X and \mathbf{n}_Y . Thus, for each α, β there exist a, b such that we can write (5) as

$$(a\mathbf{n}_X + b\mathbf{n}_Y)(\alpha X^c + \beta Y^c) = \mathbf{0}. \quad (8)$$

Expanding the LHS and using (4) we obtain

$$\mathbf{n}_Y X^c = -\frac{a\beta}{b\alpha} \mathbf{n}_X Y^c, \quad (9)$$

which, noting that $\frac{a\beta}{b\alpha}$ is a constant, is the relationship of type 2, and this concludes the first direction of the proof.

In the other direction, first assume that X and Y are planar and X^c is an affine transformation of Y^c (type 1 relationship). Due to the planarity, the rank of each of the matrices X^c and Y^c is 2. Furthermore, there exists a 3×3 matrix A such that $X^c = AY^c$. Hence, their combination

$$\alpha X^c + \beta Y^c = (\alpha A + \beta I)Y^c \quad (10)$$

has rank ≤ 2 , and thus is planar, which means that $\alpha X + \beta Y$ is planar. Second, assume that X and Y are planar and $\mathbf{n}_Y X^c = d\mathbf{n}_X Y^c$ for some scalar d (type 2 relationship), then for any scalars α and β we can find a and b such that $d = \frac{a\beta}{b\alpha}$. Working our way backwards, this implies that

$$(a\mathbf{n}_X + b\mathbf{n}_Y)(\alpha X^c + \beta Y^c) = \mathbf{0}$$

which again means that $\alpha X + \beta Y$ is planar, concluding that X and Y spans a linear subspace of planar polygons. \square

The following corollaries follow immediately:

Corollary 1. If X_f and Y_f are parallel planar polygons, then they have a type 2 relationship, and hence span a linear subspace of planar polygons.

Corollary 2. Suppose X_f and Y_f span a linear subspace of planar polygons, and let $\Pi_{\alpha,\beta}$ be the plane that contains $\alpha X_f^c + \beta Y_f^c$. Then X_f and Y_f have a type 2 relationship and are not parallel iff $\mathbf{v} := \frac{\mathbf{n}_{X_f} \times \mathbf{n}_{Y_f}}{\|\mathbf{n}_{X_f} \times \mathbf{n}_{Y_f}\|}$ is the unique unit vector, up to sign, such that $\mathbf{v} \in \Pi_{\alpha,\beta}$ for any α, β .

Using Lemma 1, it is easy to prove an analogous result for PM's:

Theorem 3. Let X and Y be two PM's in R^3 with common topology. Then X and Y span a linear space of PM's iff each non-triangular face of X has a type 1 or type 2 relationship with the corresponding face of Y .

Generating linear subspaces. We now turn to the task of generating linear subspaces of PM's. We say that a PM with topology F and geometry Y generates a linear subspace V of PM's if $Y \in V$ and any $X \in V$ is also a PM with the topology F . Given that V is generated by Y , Theorem 1 tells us that for any $\mathbf{f}_j \in F$, $X_{\mathbf{f}_j}$ and $Y_{\mathbf{f}_j}$ must have a relationship of type 1 or type 2. Therefore, before we can generate a subspace V from Y , we must specify the type of relationship t_j that each face \mathbf{f}_j should have, that is, complete a configuration \mathcal{C} for V . Then we may find, for each $\mathbf{f}_j \in F$, all $X_{\mathbf{f}_j}$ that are related to $Y_{\mathbf{f}_j}$ by relationship type t_j . As before, this is done for each face separately. Specifically, we will construct the matrix $M_{\mathcal{C}}$ from Eq. (1) by first constructing matrices $M_{\mathcal{C}_j}, j = 1, \dots, N_F$, where $\mathcal{C}_j = (Y_{\mathbf{f}_j}, \mathbf{f}_j, t_j)$ are the configurations for the subspaces V_j generated by single faces. For the sake of brevity, hereinafter, X_j and Y_j are used instead of $X_{\mathbf{f}_j}$ and $Y_{\mathbf{f}_j}$.

We begin again with the affine relationship (type 1). This means that $X_{\mathbf{f}_j}^c$ must be an affine transformation of $Y_{\mathbf{f}_j}^c$. Thus $X_{\mathbf{f}_j}^c$ must satisfy

$$X_j^c(Y_j^{c+} Y_j^c - I) = 0 \quad (11)$$

$$\Rightarrow X_j J((Y_j J)^+ Y_j J - I) = 0, \quad (12)$$

where Y_j^{c+} is the pseudo-inverse of Y_j , and J is the centering matrix from (2). This equation can easily be transformed into the form of (1). Indeed, if

$$A_{\mathbf{f}_j} := J((Y_j J)^+ Y_j J - I),$$

then eq. (11) is equivalent to

$$(A_{\mathbf{f}_j} \otimes I) \text{vec}(X_j) = 0, \quad (13)$$

where \otimes is the Kronecker product. PM's related in this manner were explored by Vaxman [25].

For the type 2 relationship, it is necessary to define an additional unit vector \mathbf{v}_j which will play the same role as in Corollary 2. We will show in Theorem 4 that any choice of \mathbf{v}_j generates a maximal linear subspace. Based on the choice of \mathbf{v}_j , we consider two cases:

- 1) $\mathbf{v}_j = \mathbf{n}_{Y_j}$. In this case, X_j and Y_j must be parallel and so

$$\mathbf{n}_{Y_j} X_j^c = \mathbf{n}_{X_j} X_j^c = 0.$$

- 2) $\mathbf{v}_j \perp \mathbf{n}_{Y_j}$. In this case, there exists a vector \mathbf{n}_{X_j} such that $\mathbf{v}_j = \frac{\mathbf{n}_{X_j} \times \mathbf{n}_{Y_j}}{\|\mathbf{n}_{X_j} \times \mathbf{n}_{Y_j}\|}$. We can write X_j^c and Y_j^c in the following basis:

$$X_j^c = \mathbf{v}_j^T X_j^1 + (\mathbf{v}_j \times \mathbf{n}_{X_j})^T X_j^2 \quad (14)$$

$$Y_j^c = \mathbf{v}_j^T Y_j^1 + (\mathbf{v}_j \times \mathbf{n}_{Y_j})^T Y_j^2, \quad (15)$$

where X_j^i, Y_j^i for $i = 1, 2$ are row vectors which contain the projections for each vertex on the appropriate vector. Multiplying (14) by \mathbf{n}_{Y_j} and (15) by $d\mathbf{n}_{X_j}$ we get

$$\mathbf{n}_{Y_j} X_j^c = \mathbf{n}_{Y_j} (\mathbf{v}_j \times \mathbf{n}_{X_j})^T X_j^2 \quad (16)$$

$$d\mathbf{n}_{X_j} Y_j^c = d\mathbf{n}_{X_j} (\mathbf{v}_j \times \mathbf{n}_{Y_j})^T Y_j^2. \quad (17)$$

The LHS of eqs. (16) and (17) are equal by the type 2 relationship, and hence so are the RHS. This implies that

$$X_j^2 = \frac{d\mathbf{n}_{X_j} (\mathbf{v}_j \times \mathbf{n}_{Y_j})^T}{\mathbf{n}_{Y_j} (\mathbf{v}_j \times \mathbf{n}_{X_j})^T} Y_j^2 = d' Y_j^2$$

and substituting in (14) gives

$$X_j^c = \mathbf{v}_j^T X_j^1 + d' (\mathbf{v}_j \times \mathbf{n}_{X_j})^T Y_j^2. \quad (18)$$

We apply the cross product by \mathbf{v}_j to both sides of the equation to get

$$\mathbf{v}_j \times X_j^c = d' \mathbf{v}_j \times (\mathbf{v}_j \times \mathbf{n}_{X_j}) Y_j^2 = d'' \mathbf{n}_{X_j}^T Y_j^2.$$

Let B be a matrix whose columns span the null space of Y_j^2 , i.e. $Y_j^2 B = 0$. We finally have that $(\mathbf{v}_j \times X_j^c) B = 0$, which can also be written in the form (1).

To better understand what this last space contains, we look at eq. (18). We note that $d' (\mathbf{v}_j \times \mathbf{n}_{X_j})$ can be replaced by any vector \mathbf{v}_j^\perp perpendicular to \mathbf{v}_j , and we can choose X_j^1 to be equal to Y_j^1 , so

$$\hat{Y}_j = \mathbf{v}_j^T Y_j^1 + \mathbf{v}_j^\perp Y_j^2.$$

From this we see that the space contains all rotations of Y_j around \mathbf{v}_j . It is easy to see that it also contains their scalings. In addition, we observe that if X_j is contained in the space, so is $X_j + \mathbf{v}_j^T X_j^1$ for any X_j^1 . Geometrically this means that the vertices of X_j are free to move in the direction of \mathbf{v}_j and still remain in the space.

We now proceed to prove that each type of space generated for a face is maximal. Again we start with the simpler case of planar polygons.

Theorem 4. *Let $\mathcal{C} = (Y, F, t)$ be a configuration for a single-face mesh $F = \{\mathbf{f}\}$ with geometry Y , and let $V := \text{null}(M_{\mathcal{C}})$, where $M_{\mathcal{C}}$ is constructed as described above. Then V is a maximal linear subspace of planar polygons.*

Proof. We will divide the proof into two parts, depending on the type of relationship t encodes. First, assume that V is the space of all affine transformations of Y . Let X be a planar polygon such that $X \notin V$. Then X and Y must have a relationship of type 2, that is, satisfy eq. (3). We can assume w.l.o.g that \mathbf{n}_X and \mathbf{n}_Y are not collinear. Otherwise, we may simply rotate Y , as any rotation of it will still be in V . Let R be a (unrelated) rotation matrix around \mathbf{n}_Y . Then $RY \in V$ and RY and X also have a type 2 relationship, namely

$$\mathbf{n}_Y X^c = d' \mathbf{n}_X RY^c. \quad (19)$$

By subtracting eq. (3) from eq. (19) we get

$$\mathbf{n}_X (dI - d'R) Y^c = 0. \quad (20)$$

Hence \mathbf{n}_X is orthogonal to the plane defined by $(dI - d'R)Y^c$. This plane is exactly the same plane defined by Y_f^c , which means that \mathbf{n}_Y is also orthogonal to $(dI - d'R)Y^c$. This in turn implies that \mathbf{n}_X and \mathbf{n}_Y are collinear, which contradicts our previous assumption. Hence, V is maximal.

The second part of the proof is further subdivided into two cases. First, we consider the case where V is the space of all polygons which are *parallel* to Y and define a planar polygon $X \notin V$. X cannot be related to all polygons in V by an affine transformation, so we assume w.l.o.g. that X and Y have the relationship of type 2. In addition, since X and Y are not parallel, \mathbf{n}_X and \mathbf{n}_Y are not collinear. Then by applying the same rotation strategy of Y used in the proof for the affine case we infer that \mathbf{n}_X and \mathbf{n}_Y are collinear and reach contradiction again.

Finally, we consider the case where V is the space of all polygons with type 2 relationship to Y . To define this space we need to set the vector \mathbf{v}_j in the plane of Y , which we recall is shared among the planes of all polygons in this space. Again, X has w.l.o.g. a relationship of type 2 to Y . X cannot contain \mathbf{v}_j since it would mean that $X \in V$. Therefore, there is another vector \mathbf{v}'_j that the planes of X and Y share. Let $R_{\mathbf{v}_j}$ be a rotation matrix around \mathbf{v}_j . Then the plane of $R_{\mathbf{v}_j} Y$ does *not* contain \mathbf{v}'_j , but still has to have a type 2 relationship with X . Hence, there is another vector \mathbf{v}''_j that the planes of $R_{\mathbf{v}_j} Y$ and X share. In a similar manner, we can find yet another polygon whose plane shares a different vector \mathbf{v}'''_j with X . These three vectors are not coplanar, yet each vertex of X is free to move in each of their directions and still remain in the space. However, this way X can be made non-planar, contradicting our assumptions. Thus we conclude that for any configuration, the space V is maximal. \square

In reality, to avoid having to specify an explicit normal for each face having a relationship of type 2, we used three cases when specifying relationships types for faces. The first case, which we call the *affine case*, is simply when all faces have type 1 relationship. In the second case, \mathbf{v}_j is set to be equal to $\mathbf{n}_{Y_{f_j}}$ for every j . Thus, the subspace generated in this case is that of all polygons which are parallel to the generating polygons, hence, the *parallel case*. In the third case, the *vertical case*, for all faces, $\mathbf{v}_j = z \times \mathbf{n}_{Y_{f_j}}$. The justification

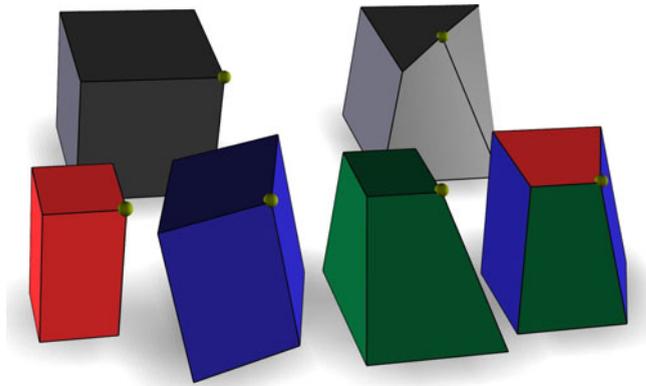


Fig. 3. Hexahedron in different subspaces generated by the (gray) cube on the top left. They are the closest ones in their subspaces to the (gray) non-PM on the top right, subject to the hard constraint imposed by the yellow vertex.

for this is the fact that many meshes, especially architectural meshes, have a prominent up direction.

Theorem 4 tells us that by using this construction to generate a space of PM's, every face potentially generates a maximal space. However, the linear subspaces of the whole PM may not be always maximal. This can happen when two neighboring faces generate spaces which do not "match", causing the two shared vertices to be overly restricted. For example, the red cube in Fig. 3 was deformed in the parallel subspace. In this space, each face can only be stretched in the obvious directions, which is a subset of the affine transformations of the face. Thus, the parallel subspace in that case is not maximal, since it is contained in the affine subspace. Note that by removing a single face from the cube, the linear subspaces become different. These situations are easily detectable, and the face can be reassigned.

It is now easy to show that there is a piecewise linear path between any two PM's in the manifold: using the affine space generated by the two meshes, they can be projected to the same plane, where they share the parallel space. This construction however is not very useful as it does not provide any insight into the manifold itself. Nevertheless, it forms a loose "lower bound".

In our examples, the relationship types per face were color coded by blue, red and green for the affine, parallel and vertical cases, respectively. When more than a single relationship type is used to generate the subspace, it is referred to as a *mixed space*.

Degrees of freedom. The number of degrees of freedom (NDOF) of a linear subspace of PM's is exactly the dimension of the nullspace B . We can estimate the NDOF in some specific cases, such as when the space is not mixed. The NDOF is then exactly the co-rank of M_C . However, this value depends too much on the current embedding of the PM and does not give any insight into the relation to its topology. We instead provide a lower bound on the NDOF for a given PM, which can be inferred from the topology alone.

Denote by N_v, N_b, N_e, N_f, N_c the number of vertices, boundary vertices, edges, faces and corners (i.e. face-vertex pairs) of the PM, respectively. The number of variables (the mesh vertex geometry) is always $3N_v$. In the affine case, the number of equations is $3N_c$, but each face is determined by

TABLE 1
Minimal Number of Free Vertices (NFB) in Different Subspaces

	Quad mesh	Hex mesh
Affine	$\frac{N_b}{2} + b + 2g$	$\frac{-N_v}{2} + \frac{3}{4}N_b + \frac{3}{2}b + 3g$
Parallel	$\frac{N_b}{2} + b + 2g$	$\frac{N_v}{6} + \frac{5N_b}{12} + \frac{5b}{6} + \frac{5g}{3}$
Vertical	$-\frac{N_v}{3} + \frac{2N_b}{3} + \frac{4b}{3} + 8g/3$	

just three vertices. Hence a lower bound on the NDOF is $3(N_v + 3N_f - N_c)$. Similarly, in the parallel case the lower bound is $3N_v - N_c + N_f$, and in the vertical case it is $3N_v - 2(N_c - 2N_f)$.

We can use the generalized Euler formula, $N_v - N_e + N_f - b = 2g$, where b is the number of boundaries, and g is the genus of the mesh, and the fact that $N_c = 2N_e - N_b$ to obtain

$$N_c = 2(N_v - 2g + N_f - b) - N_b.$$

Plugging this into the formulas for the NDOF yields an expression that does not depend on N_c and N_e . For (semi-) regular graphs, N_f can also be expressed using N_v and N_b and vice-versa, which may give more intuitive results. Additionally, we define the *number of free vertices* (NFB) as the NDOF divided by 3. The NFB roughly gives the number of vertices that can be fixed independently. We list the minimal NFB for quad and hex meshes for both cases in Table 1.

The table shows that the minimal NFB for quad meshes in the affine and parallel cases is determined by the size of the boundary. See Appendix 1 for further details. In fact, our experiments show that, apart from very symmetric cases like spheres or tori, the minimal NFB for the affine case is the true NFB, up to a global transformation. This means that there is very little that can be done with closed quad meshes in the affine case. The situation is even worse for hex meshes: unless the mesh is just a strip of hexagons, the minimal NFB will be negative. In fact, we prove in Appendix 2 that the actual NFB is 3 for any 3-regular mesh without a boundary. A trick that can be used to increase the NFB is to apply a half-edge subdivision to the hex mesh (see inset). Technically, the new mesh will not be a hex mesh, but it might retain the "look" of the original hex mesh, and the minimal NFB will be much higher. As for the parallel case, it is easy to show that for closed 3-regular PM's, the NFB is exactly N_f .



3 EXPLORING LINEAR SUBSPACES

Overview. Once all faces of the mesh have relationship types assigned to them and the matrix M_C is computed, we can begin the exploration of $null(M_C)$. While we can do this by simply computing an orthogonal basis for $null(M_C)$, it may not be very useful: this basis will contain random PM's. Instead, we discuss ways to create more meaningful shapes, which are targeted toward different levels of editing.

Eigenshapes. Yang et al. [27] proposed to explore the manifold of PM's not by explicitly setting positional constraints, but by traversing the neighborhood of the PM. This is done by choosing a few directions (two or three for easy navigation) on the osculate which match the manifold the best. Using linear subspaces, we do not have to worry about going far away from the manifold, which allows us to be more adventurous with the exploration. We propose using the PM's "harmonics" as a basis for exploration. More precisely, we use the eigenvectors of the Laplacian L of the generating mesh Y , constrained to the linear space, which we call eigenshapes. These are defined by the constrained Rayleigh quotient:

$$\max_X \frac{X^T L X}{X^T X} \quad \text{s.t.} \quad M_C X = 0. \quad (21)$$

The solution to this problem is found in [9] as the eigenvectors of PLP where

$$P = I - M_C^T (M_C M_C^T)^{-1} M_C.$$

See implementation details in Section 4 on how to compute the eigenshapes efficiently. To effectively visualize the eigenshapes and to explore them efficiently, we suggest the following idea: add the eigenshapes to the source PM and apply a "band-pass-filter" to it. By sliding the filter we can quickly see how eigenshapes of different frequencies affect the PM (Fig. 10)

Sparse shapes. Habbecke and Kobelt [10] discussed editing of constrained meshes, where their goal was to be able to reposition a vertex while making as little as possible change to the rest of the mesh and satisfy the constraints. This addresses the well-known problem of editing with constraints, where making a change in one portion of a mesh damages the work that was already done elsewhere in the mesh. Their approach is based on linearizing the constraints and finding sparse solutions to the linearized system. The same strategy can be used to deform PM's and in fact, one of the constraints treated in [27] is the planarity of faces. In terms of basic shapes, in order to be able to move just a small set of vertices, a shape where most of the vertices lie on the origin is needed. These *sparse shapes* are just sparse vectors in $\text{null}(M_C)$. To find sparse solutions, Habbecke and Kobelt employ the Orthogonal Matching Pursuit (OMP) algorithm [22], and the same can be done to find sparse shapes.

For many subspaces, the only sparse shapes that can be found are not sparse at all. For example, the affine space for quad meshes contains truly sparse shapes only for very symmetric cases (Fig. 4). In these cases *approximate* sparse shapes—shapes that are not in the linear subspace but close to it—can be found instead. For comparison, the accurate sparse shape in the middle of Fig. 5 has $\|M_C X\| \approx 10^{-12}$, and the approximate sparse shape has $\|M_C X\| \approx 10^{-4}$. The original PM was produced by planarizing a deformed torus, which had $\|M_C X\| \approx 0.1$.

Fundamental shapes. While a sparse shape changes only a small number of vertices, it can still be non-local, moving vertices on opposite sides of the PM. In many cases a shape with more locality is required; one that perhaps moves *all*

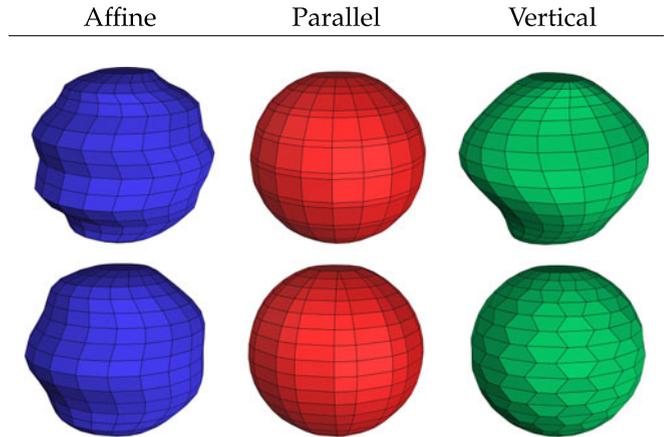


Fig. 4. Adding eigenshapes of different subspaces to a simple spherical quad PM. See also accompanying video.

vertices, but to a lesser extent ([11]). To elaborate, suppose a vertex v_i has been selected. We may then define the fundamental shape associated with v_i as the solution to the optimization problem

$$\min_X \|X - \delta_i\|^2 + \lambda \|LX\|^2 \quad (22)$$

$$\text{s.t.} \quad M_C X = 0,$$

where δ_i is a vector whose only non-zero elements are the ones corresponding to v_i and LX is a regularization term. Of course, both the distance function and the regularization terms can be replaced by other similar functions.

Handle-based deformation. PM's can be deformed directly, and the handle-based approach is probably the most natural metaphor to use (excluding, perhaps, the recent curve-based approach [29]). This was studied in detail in [25] and [17] for the case of PM's in the affine case only, where an As-Rigid/Similar-As-Possible (ARAP/ASAP) deformation was computed within the resulting subspace. The well-known solution to the ARAP/ASAP deformation problem uses an alternating local/global scheme [14], [21]. The only difference when applying this to PM's is that the constraints defining the linear subspace must be satisfied when solving the global steps. In Fig. 7 we used the same method as in [25] to deform in an ASAP way a half sphere hex mesh in the non-mixed spaces. The boundary was kept fixed and one vertex on the top was moved slightly higher. The affine subspace allows only global transformations and the parallel subspace produced self-intersections almost immediately. The vertical subspace produced pleasing, non-trivial results.

Dual exploration. Every polyhedron admits a family of dual polyhedra, most notably the polar dual [19], having the property that the vector to each of the dual vertices is normal to the corresponding primal face. Usually polar duals are associated only with star-shaped polyhedrons, since otherwise the polar dual may self-intersect. Here we ignore this and associate polar duals with general, non-convex PM's. Obviously the polar dual associated with a PM is itself a PM, so the ideas presented in this paper apply also to the space of polar duals of a given PM. This essentially means that we can explore the subspace of the PM based on its *face normals* instead of the vertex positions. Although the

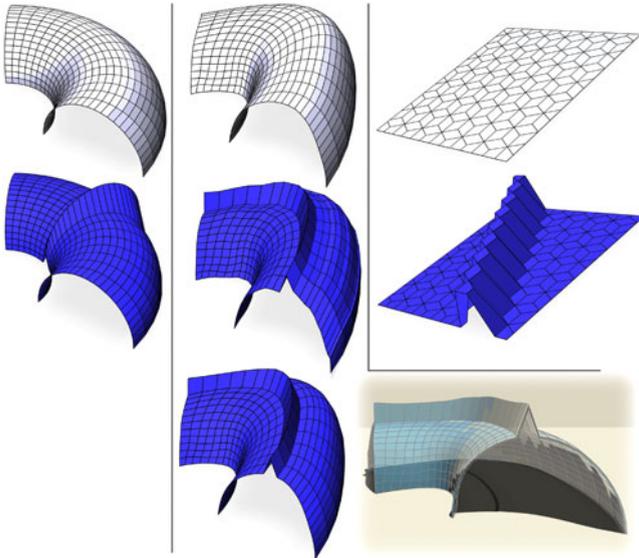


Fig. 5. Sparse shapes. (Left) Part of a symmetric torus quad PM, having an accurate sparse shape. (Middle) Deformed torus. Its accurate sparse shape is not sparse at all, but it has an inaccurate sparse shape. (Right) Sparse shape of a flat PM.

subspaces defined using the face normals are linear, since they are the same as the linear spaces of polar duals, they are not linear with respect to the vertices of the primal mesh. The reason is that the duality transformation is not linear. Still, it involves only solving a sparse linear system and can be done in real time.

The benefit of dual exploration of PM subspaces is that this gives a completely different number of DOFs compared to the primal space, based on the normal of the faces instead of the vertices. As an extreme example, the duals of any 3-regular meshes are triangle meshes, which trivially preserve planarity. Hence, editing a 3-regular mesh in the normal domain is also trivial: any choice of normal will result in a valid PM. Fig. 8 shows the dual deformation of two PM's. The results there could not have been achieved using only one primal linear space.

4 DISCUSSION AND FUTURE WORK

Implementation details. Most of the software implementation was done in MATLAB, and was wrapped as a plugin for Autodesk Maya, for its user interface. The matrix M_C was built by constructing M_{C_j} face-by-face. M_{C_j} as defined here is already not full rank, so we reduced the number of equations per-face using SVD. The construction takes less than a second for meshes with approximately a thousand faces.

To compute the eigenshapes, a sparse QR decomposition was used to generate an orthonormal basis N of $null(M_C)$,

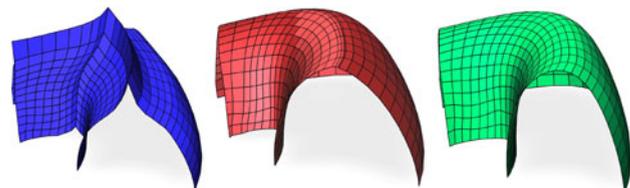


Fig. 6. Fundamental shapes of the deformed torus.

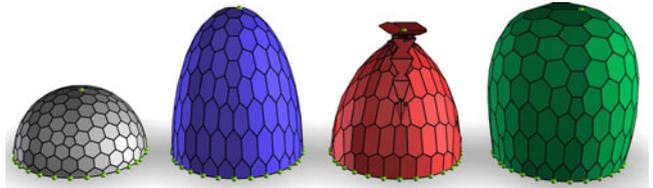


Fig. 7. ASAP deformation of a hexagonal half sphere. Note that in the (blue) affine subspace, only global transformations are possible.

then any X in $null(M_C)$ can be written as Nw for some w , and

$$\begin{aligned} \max_{X \in null(M_C)} \frac{X^T L X}{X^T X} &= \max_w \frac{w^T N^T L N w}{w^T N^T N w} \\ &= \max_w \frac{w^T N^T L N w}{w^T w}, \end{aligned} \tag{23}$$

which is solved using the eigendecomposition of $N^T L N$. This approach gives much better precision and performance than the formula in [9], since pseudoinverse computation is avoided and full size singular value decomposition is replaced with a much smaller eigenvalue decomposition. For the handle-based deformation, the relevant matrices were decomposed in a preprocessing step. We did not invest much effort to use the best possible decomposition and carefully tune the parameters. Specifically, we used LDL decomposition for the initial mesh approximation step, but a sparse QR decomposition for the global steps in the ARAP/ASAP deformation, due to numerical instabilities caused by LDL there.

Limitations. Our assumption is that the initial PM has planar faces. Otherwise, many of the calculations made are not well-defined. Of course, the planarity of faces can only be up to some numerical precision. We have found that the affine case is less sensitive to non-planar faces than the other cases. The mesh in Fig. 10 does not have planar faces, yet the eigenshapes computed for it in the affine space do not cause them to be "less" planar. On the other hand, the eigenshapes of the parallel case (not shown) quickly deteriorate the quality of the mesh.

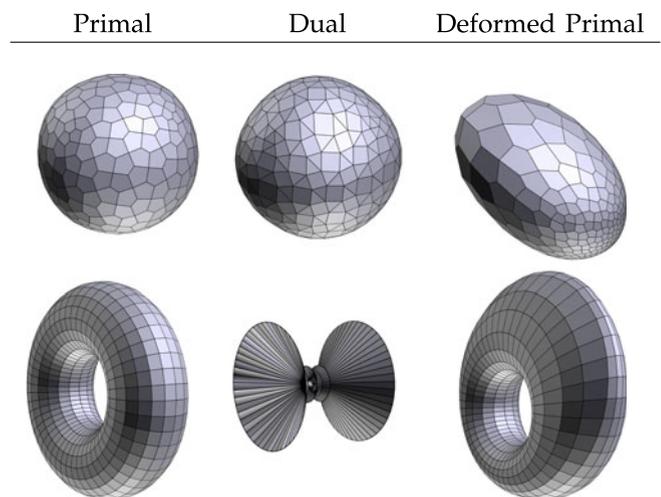


Fig. 8. Deformation of a (left) sphere and a torus using the (middle) polar dual. In both cases an eigenshape of low frequency was added to the dual mesh, and a new (right) primal mesh results.

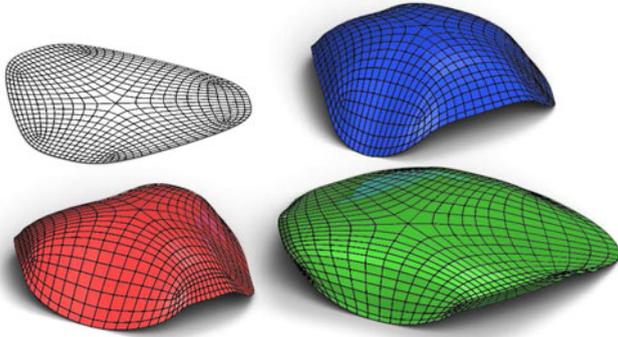


Fig. 9. Designing a PM from a planar graph. The graph was created by projecting a non-PM to the plane. It was then given height using the affine subspace, and then deformed using eigenshapes in the parallel and vertical spaces.

Creating an initial PM. The linear subspaces described here need an initial PM realizing the given topology. The simplest way to generate such a PM is to take a non-polyhedral mesh with the given topology and project it to a plane. The original mesh can then be projected into a linear space generated from the flat mesh. The result of this, however, is usually unsatisfactory and we did not use it. Most of the PM's in this paper were created by experimenting with the TopMod 3.0 software [1], where we used the variety of subdivision schemes implemented there to create elaborate meshes from simple solids. If only the mesh topology is given, then a simple "spring-based" planar embedding, such as Tutte's [23], should suffice. Figs. 1, 9 and 11 show PM's with initial PM's being planar embeddings.

Selecting the right space. There are, literally, infinite number of linear subspaces available for a single PM. Even if we limit ourselves to the three cases mentioned above, the number of possibilities to assign them to faces is exponential and manually assigning them is tedious. We did not investigate methods to find the optimal linear subspace to work with, or even attempt to define what exactly optimal means. A simple definition could be: the subspace with the highest dimension. Experimentally we observed that in many cases the parallel space had the largest dimension. However, this subspace does not generate much visual variation in the overall look of the PM, compared to the other spaces. This problem remains open for now, and we reserve it for future work. In practice, switching between the non-mixed cases provided sufficient variation.

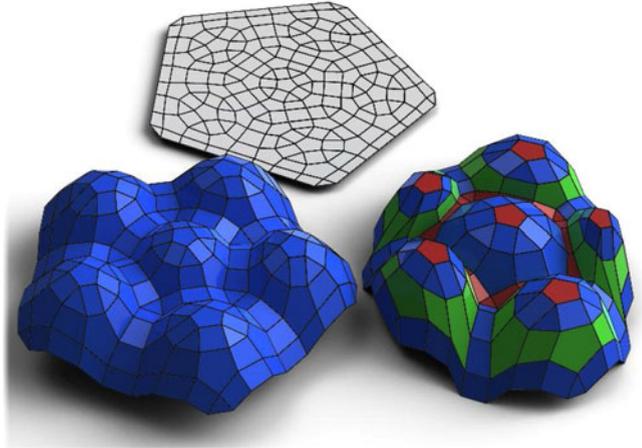


Fig. 11. Another example of designing a PM from a planar graph. The graph was created by subdividing a pentagon using several schemes until the desired result was achieved. It was given height in the affine subspace and then deformed in a mixed subspace.

Currently we use a number of heuristics while experimenting with our system. The affine space is easier to work with when there are many DOFs, as is the case for quad meshes with boundaries. Figs. 1, 2 and 6 show results of such quad meshes. In situations where the number of DOFs is too small, this is usually caused by faces with more than four edges or vertices of degree three. These can be automatically reassigned to the other two cases to achieve more freedom. Fig. 12 shows some deformation results with hexagonal mesh, for which the affine space has only 12 DOF, while the other two cases have several hundreds DOF. On the other hand, when using the parallel or the third case, some faces may enjoy too much freedom and misbehave while deforming. These can be reassigned to the parallel case, since it better preserves the shape of a polygon.

A related problem is how to interpolate PM's that are not related by a single linear space. We have shown that any two can be connected by a succession of three linear spaces, which is not very useful for interpolation. An interesting direction to explore is to approximate paths in the manifold of PM's by linear segments using the linear subspaces.

Design pipeline. Our experiments led us to the following pipeline for designing a PM. For flat meshes, the first step is to afford them some height. This is done by regular deformation followed by a planarization step, or by using the affine linear subspace and applying the handle-based deformation or using the eigenshape band-pass-filter technique. The reason for not using the parallel or vertical subspaces is that they cannot "unflatten" the PM. However, mixed spaces can also be used. Once we have a PM with some volume, the rest depends on the effect we aim to achieve. For large deformations we use the affine subspace when working on quad meshes with boundaries, and the other subspaces otherwise. To add variation or *waviness* to the PM, we use the eigenshapes. The affine eigenshapes are useful when the overall look of the PM needs to be maintained but the shapes of individual faces need to be changed. Using the parallel eigenshapes is an efficient way of adding variation to meshes having uniformly-sized faces. We show a variety of results in Figs. 8, 9, and 10 (see also the accompanying video).

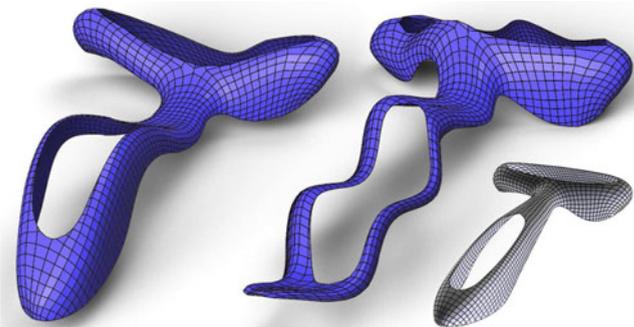


Fig. 10. The Yas model deformed using eigenshapes of different frequency in the affine subspace.

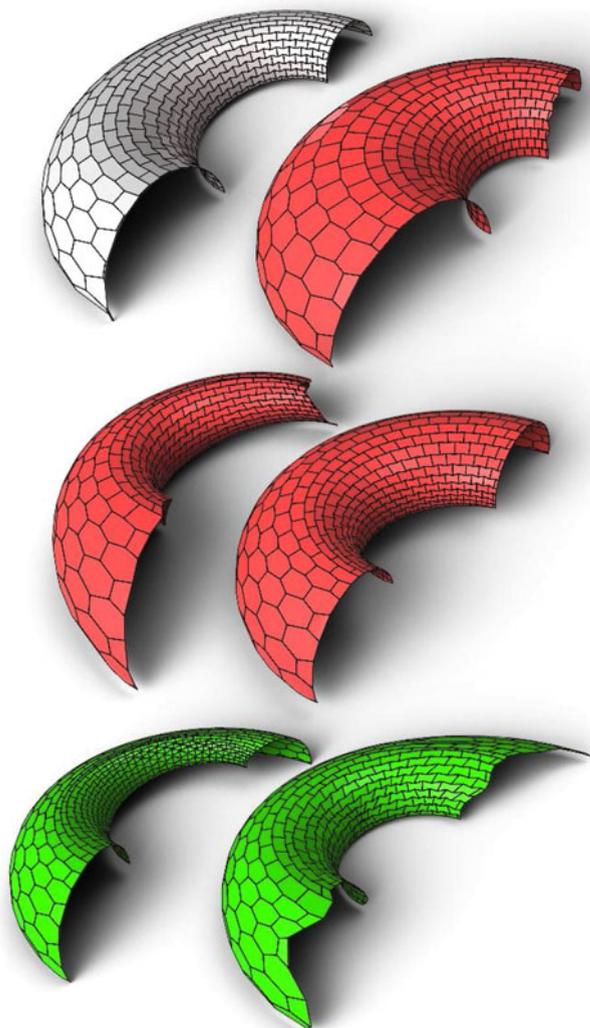


Fig. 12. Deformation of a planar hex mesh [26] using eigenshapes in the parallel and vertical spaces. Note that the affine space had only 12 DOF, while the other cases had a few hundreds.

The sparse and fundamental shapes, while helping to visualize the limitations of various subspaces, have not proven to be very useful for the design process. The reason is, by definition, they can only make the PM less smooth, which usually means less visually pleasing. However, we believe they are valuable as a theoretical tool for studying PM's. One future research direction could be to use them to decide where to make small adjustments to the topology of the mesh in order to add more freedom to specific places.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for valuable suggestions. The work of R. Chen was partially supported by the Aly Kaufmann postdoctoral fellowship at the Technion. Renjie Chen is the corresponding author of the article.

REFERENCES

[1] E. Akleman, V. Srinivasan, E. Mandal, J. Chen, Z. Melek, and E. Lendreneau, "Topmod: Topological mesh modeling system," Texas A&M Univ., College Station, TX, USA, Tech. Rep., 2004.

[2] M. Alexa and M. Wardetzky. (2011). Discrete laplacians on general polygonal meshes. New York, NY, USA, pp. 102:1–102:10. [Online]. Available: <http://doi.acm.org/10.1145/1964921.1964997>

[3] M. Ben-Chen, O. Weber, and C. Gotsman. (2009, Jul.). Variational harmonic maps for space deformation. *ACM Trans. Graph.*, vol. 28, no. 3, pp. 34:1–34:11. [Online]. Available: <http://doi.acm.org/10.1145/1531326.1531340>

[4] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. uno Levy, *Polygon Mesh Processing*, AK Peters, 2010.

[5] D. Cohen-Steiner, P. Alliez, and M. Desbrun. (2004). Variational shape approximation. in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH'04, New York, NY, USA: ACM, pp. 905–914. [Online]. Available: <http://doi.acm.org/10.1145/1186562.1015817>

[6] B. Cutler and E. Whiting. (2007). Constrained planar remeshing for architecture. in *Proceedings of the Graphical Interface 2007*, ser. GI'07, New York, NY, USA: ACM, pp. 11–18. [Online]. Available: <http://doi.acm.org/10.1145/1268517.1268522>

[7] M. Eigensatz and M. Pauly, "Positional, metric, and curvature control for constraint-based surface deformation." *Comput. Graph. Forum*, vol. 28, no. 2, pp. 551–558, 2009.

[8] M. Eigensatz, R. W. Sumner, and M. Pauly, "Curvature-domain shape processing." *Comput. Graph. Forum*, vol. 27, no. 2, pp. 241–250, 2008.

[9] W. Gander, G. H. Golub, and U. von Matt, "A constrained eigenvalue problem," *Linear Algebra its Appl.*, vol. 11415, pp. 815–839, 1989.

[10] M. Habbecke and L. Kobbelt, "Linear analysis of nonlinear constraints for interactive geometric modeling," *Comput. Graph. Forum*, vol. 31, no. 2pt3, pp. 641–650, May. 2012.

[11] T. Hoffmann. (2010). On local deformations of planar quad-meshes. in *Proc. Third Int. Congress Conf. Math. Softw.*, ser. ICMS'10, Berlin, Germany: Springer-Verlag, pp. 167–169. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1888390.1888431>

[12] T. Igarashi, T. Moscovich, and J. F. Hughes. (2005). As-rigid-as-possible shape manipulation. in *ACM SIGGRAPH 2005 Papers*, ser. SIGGRAPH'05, New York, NY, USA: ACM, pp. 1134–1141. [Online]. Available: <http://doi.acm.org/10.1145/1186822.1073323>

[13] A. Jacobson, I. Baran, J. Popović, and O. Sorkine, "Bounded biharmonic weights for real-time deformation," in *ACM SIGGRAPH 2011 papers*, ser. SIGGRAPH'11, New York, NY, USA: ACM, 2011, pp. 78:1–78:8.

[14] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler, "A local/global approach to mesh parameterization," in *Proc. Symp. Geometry Process.*, 2008, pp. 1495–1504.

[15] Y. Liu, H. Pottmann, J. Wallner, Y.-L. Yang, and W. Wang, "Geometric modeling with conical meshes and developable surfaces," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 681–689, Jul. 2006.

[16] Y. Liu, W. Xu, J. Wang, L. Zhu, B. Guo, F. Chen, and G. Wang, "General planar quadrilateral mesh design using conjugate direction field," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 140:1–140:10, Dec. 2011.

[17] R. Poranne, E. Ovreiu, and C. Gotsman, "Interactive planarization and optimization of 3d meshes," *Comp. Graph. Forum*, vol. 32, no. 1, pp. 152–163, 2013.

[18] H. Pottmann, Y. Liu, J. Wallner, A. Bobenko, and W. Wang. (2007, Jun.). Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graph.*, vol. 26, no. 3. [Online]. Available: <http://doi.acm.org/10.1145/1276377.1276458>

[19] J. Richter-Gebert. (1996). *Realization spaces of polytopes*. ser. Lecture Notes in Mathematics, New York, NY, USA: Springer. [Online]. Available: <http://books.google.co.za/books?id=K9UZAQAIAAJ>

[20] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. (2004). Laplacian surface editing. in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, ser. SGP'04, New York, NY, USA: ACM, pp. 175–184. [Online]. Available: <http://doi.acm.org/10.1145/1057432.1057456>

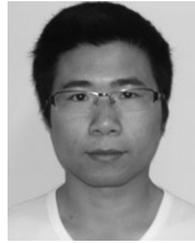
[21] O. Sorkine and M. Alexa. (2007). As-rigid-as-possible surface modeling. in *Proceedings of the Fifth Eurographics Symposium on Geometry Process.*, ser. SGP '07, Aire-la-Ville, Switzerland: Eurographics Association, pp. 109–116. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1281991.1282006>

[22] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

- [23] W. T. Tutte. (1963). How to Draw a Graph. *Proc. London Math. Soc.*, vol. s3-13, no. 1, pp. 743–767. [Online]. Available: <http://dx.doi.org/10.1112/plms/s3-13.1.743>
- [24] B. Vallet and B. Lévy. (2008, Apr.). Spectral geometry processing with manifold harmonics. *Comput. Graph. Forum*, vol. 27, no. 2, pp. 251–260. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2008.01122.x>
- [25] A. Vaxman. (2012, Aug.). Modeling polyhedral meshes with affine maps. *Comp. Graph. Forum*, vol. 31, no. 5, pp. 1647–1656. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2012.03170.x>
- [26] W. Wang, Y. Liu, D. Yan, B. Chan, R. Ling, and F. Sun, “Hexagonal meshes with planar faces,” Dept. Comput. Sci., Univ. Hong Kong, Hong Kong, Tech. Rep. TR-2008-13, 2008.
- [27] Y.-L. Yang, Y.-J. Yang, H. Pottmann, and N. J. Mitra. (2011, Dec.). Shape space exploration of constrained meshes. *ACM Trans. Graph.*, vol. 30, no. 6, pp. 124:1–124:12. [Online]. Available: <http://doi.acm.org/10.1145/2070781.2024158>
- [28] M. Zdravec, A. Schiftner, and J. Wallner. (2010). Designing quad-dominant meshes with planar faces. *Comput. Graph. Forum*, vol. 29, no. 5, pp. 1671–1679. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2010.01776.x>
- [29] X. Zhao, C.-C. Tang, Y.-L. Yang, and H. Pottmann, “Intuitive design exploration of constrained meshes,” in *Advances in Architectural Geometry 2012*, L. Hesselgren, Eds., New York, NY, USA: Springer, 2013, pp. 305–318.



Roi Poranne received the PhD degree from the Technion, Haifa, Israel, in 2013, and is currently a postdoctoral fellow at the Weizmann Institute, Rehovot, Israel. His research interests include computer graphics and geometry processing.



Renjie Chen received the BSc and PhD degrees in applied mathematics from Zhejiang University, Hangzhou, China, in 2005 and 2010, respectively. He was a postdoctoral fellow at the Center for Graphics and Geometric Computing, Technion—Israel Institute of Technology, Haifa, Israel. His research interests include computer graphics, geometry processing, computational geometry and geometric modeling.



Craig Gotsman received the PhD degree from the Hebrew University of Jerusalem, Jerusalem, Israel, in 1991. He was the first and founding director of the Jacobs Technion-Cornell Innovation Institute at Cornell Tech, New York, NY, where he is currently a professor. Holding Technion’s Hewlett-Packard Chair in Computer Engineering, he co-founded the Technion Center for Graphics and Geometric Computing and is active in research on 3D computer graphics, geometric modeling, animation and computational geometry. Straddling academia and industry, he holds ten U.S. patents, and started two companies. He also has consulted for numerous small and large companies, including Hewlett-Packard, Intel, Nokia, Shell Oil, Autodesk and Disney. He has been a visiting professor at Harvard University, INRIA Sophia Antipolis (France) and ETH Zurich (Switzerland), and a research scientist at MIT. He has published more than 150 papers in the professional literature, won eight best paper awards at leading conferences and mentored more than 50 postgraduate level students (MS, PhD and postdoc).

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.