

Modeling Polyhedral Meshes with Affine Maps—Without the Actual Affine Maps

Amir Vaxman

September 20, 2016

Abstract

We present a novel and more efficient formulation to the affine-map based polyhedral mesh editing, first presented in [1]. The new implementation reduces the amount of variables considerably, while keeping the same functionality.

1 Modeling Polyhedral Meshes with Affine Maps

In the following, we give a brief account of [1]; for full details, refer to the paper.

We deal with a polygonal (non-triangular) 2-manifold mesh $M = (V, E, F)$. The positions of each of the vertices in the original (base) mesh are defined as $p_v, v \in V$. We wish to obtain a deformed, or edited mesh, defined by new positions $q_v \in V$. We do so by three methods:

Handle-based deformation We choose a set of vertices, denoted as *handles*: $v_h \in H \subset V$, and prescribe their new positions q_h . The position of the *free* vertices in $V \setminus H$ are determined automatically by the constraints and the objectives of the problem.

Interpolation Given source positions p and target positions q , we seek new positions r so that, for a time variable $t \in [0, 1]$, we obtain $r(0) = p$ and $r(1) = q$, and expect a smooth and intuitive transition of every other t . An extrapolation is when we allow t to vary outside of these boundaries.

Shape-Space Exploration For a given set of constraints (validity), and an objective (niceness), we wish to explore all the *valid* shapes that are as nice as possible. Exploring here means the ability to navigate between a reduced set of possibilities in an intuitive way.

The constraint that is targeted in [1] is face planarity. A polygonal mesh with planar faces is often called *polyhedral*. We use “near-polyhedral” to denote

faces and meshes that are planar up to a tolerance. Using the positions q as variables, the planarity condition is cubic (involves a triple product). Adding an auxiliary variable that represents a normal to the faces makes it possible to form quadratic conditions instead.

The main insight in [1] is that since planarity is invariant to affine transformations, we can preserve planarity by constraining each face to deform by a single affine map. By controlling the isotropy of the map, near-planar faces result in near-planar faces as well. With that in mind, they assign a single affine map per face: $B_f, f \in F$, represented by a 3×3 matrix, and enforce this affine map to deform the mesh with the following set of conditions (see Figure 1):

$$\forall f \in F, \forall e_{jk} = (p_k - p_j) \in f, (p_k - p_j)B_f = q_k - q_j, \quad (1)$$

where j, k are consecutive vertices on each face. Assuming the variables in the system are (q, B) , the system is linear.

Within the space of valid *piecewise-affine* transformations, there are two objectives minimized by their algorithm:

- Map Prescription: given an input set of maps T_f , minimize $\sum_f |B_f - T_f|^2$. These maps can come from a computation resulting from the interpolation, or as the result of a local iteration in an as-rigid-as-possible algorithm.
- Map smoothness: minimizing the difference between neighboring maps for smooth deformation: $\sum_{f,g \in F} |B_f - B_g|^2$, for all adjacent f, g .

These energies can be controlled by factors α, β s.t. the total objective function (energy) is then:

$$E = \alpha \sum_{f \in F} |B_f - T_f|^2 + \beta \sum_{f,g \in F} |B_f - B_g|^2. \quad (2)$$

There are two disadvantages to this method:

1. The space is often over-constrained when there is not enough boundary. But this does not seem to affect the quality of the results. It is conjectured and empirically witnessed in [1] that planarity is well-behaved even for approximately piecewise-affine spaces.
2. The coordinates q contribute V variables per dimensions, and the auxiliary affine maps contribute $3F$ more variables, which is approximately almost 3 times as much for quad meshes. That makes the matrices involved in solving quite cumbersome.
3. If a face is perfectly planar, there is a redundant single degree of freedom per dimension that needs to be regulated.

In light of the points above, we provide an alternative formulation of the same space in the following Section.

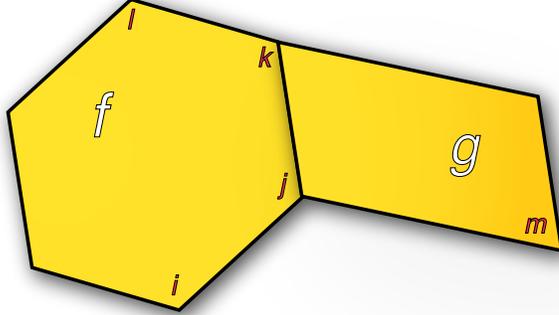


Figure 1: The setting used in our formulation.

2 Modeling Polyhedral Meshes without Actual Affine Maps

Instead of using B_f , we only use a single normal variable per face n_f . The original normal for positions p , denoted N_f , is initialized with the normal to the plane, if the face is perfectly planar, or an average of the normals to each consecutive pair of edges in the face. The specific choice is not of much consequence, since we are assuming near-polyhedral meshes. The affine map transforming a pair of consecutive edges e_{ij}, e_{jk} and the normal n_f is:

$$B_{ijk,f} = \begin{pmatrix} p_j - p_i \\ p_k - p_j \\ N_f \end{pmatrix}^{-1} \cdot \begin{pmatrix} q_j - q_i \\ q_k - q_j \\ n_f \end{pmatrix} \quad (3)$$

Our compatibility conditions make sure that for each consecutive four vertices i, j, k, l in a face, creating three consecutive edges, the affine maps of each two identify:

$$\begin{pmatrix} p_j - p_i \\ p_k - p_j \\ N_f \end{pmatrix}^{-1} \cdot \begin{pmatrix} q_j - q_i \\ q_k - q_j \\ n_f \end{pmatrix} = \begin{pmatrix} p_k - p_j \\ p_l - p_k \\ N_f \end{pmatrix}^{-1} \cdot \begin{pmatrix} q_k - q_j \\ q_l - q_k \\ n_f \end{pmatrix} \quad (4)$$

Note that as (q, n) are our new variables, the system is still linear. In every face of degree d , we only need $d - 1$ such equation, as the last one is linearly dependent on the other conditions in that face. We note that there are several equivalent ways to express the same constraints, but their degree of freedom is similar.

The objective function is done in a similar manner:

- Map prescription: we have an expression for combination of two consecu-

tive edges on every face f , and prescribed map T_f :

$$\sum_{i,j,k,f} \left| \begin{pmatrix} p_j - p_i \\ p_k - p_j \\ N_f \end{pmatrix}^{-1} \cdot \begin{pmatrix} q_j - q_i \\ q_k - q_j \\ n_f \end{pmatrix} - T_f \right|^2 \quad (5)$$

- Map smoothness: for each adjacent face pair (f, g) and their mutual edge i, j , we choose the consecutive i, j, k on face f , and the consecutive k, j, m on face g (see Figure 1), and minimize:

$$\sum_{i,j,k,f} \left| \begin{pmatrix} p_j - p_i \\ p_k - p_j \\ N_f \end{pmatrix}^{-1} \cdot \begin{pmatrix} q_j - q_i \\ q_k - q_j \\ n_f \end{pmatrix} - \begin{pmatrix} p_j - p_k \\ p_m - p_j \\ N_g \end{pmatrix}^{-1} \cdot \begin{pmatrix} q_j - q_k \\ q_m - q_j \\ n_g \end{pmatrix} \right|^2 \quad (6)$$

The matrix inverses are always well-defined, assuming that the original faces are not degenerate (no zero edges, and no consecutive collinear edges). n_f is in general neither unit length, nor orthogonal to the modified face q , as affine maps do not preserve these properties. We do not use it other than an auxiliary variable. Using n_f adds $|F|$ variables to the system, which is about 50% improvement from using B_f as variables. This is the version that is implemented in libhedra [2].

2.1 Why the normal auxiliary variable is necessary

Readers might wonder why the normal variable is entirely necessary. In theory, there is an alternative: gather all the edges for all consecutive vertices $1, \dots, d$ in a face of degree d , and form the constraint:

$$\begin{pmatrix} p_2 - p_1 \\ p_3 - p_2 \\ \dots \\ p_1 - p_d \end{pmatrix}^{-1} \cdot \begin{pmatrix} q_2 - p_1 \\ q_3 - q_2 \\ \dots \\ q_1 - q_d \end{pmatrix} = 0, \quad (7)$$

where the Penrose-Moore inverse is used. This expression could then be used for the objective function as well, replacing B_f . If the face is near-polyhedral, the solution would be unique.

However, this is not the case for exactly-planar faces (or near enough to have bad conditioning): out of all the possible affine maps that transform the source face to the target face, the PM inverse would choose the minimum norm one. The 3-dimensional linear space of such maps is discussed in [1], and is exactly related to the operation of the affine map on the orthogonal component, i.e., the normal. The minimum-norm solution would most likely account for a map that takes the normal into zero. For instance, applying a global rotation to two adjacent (and non co-planar) faces, and then retrieving the affine map

by Equation 7 would result in two different matrices. This makes the objective functions problematic to formulate.

One option is to throw away the problem of minimum-norm solutions and only use 6 variables to uniquely encode the transformation of a perfectly-planar faces in a local basis on the supporting plane of a face. Perfect rotations, for instance, are easy to detect and compare. Unfortunately, we are not aware of any way to properly handle near-planar faces with this approach, and lose this practical flexibility and generality.

In light of this, using an auxiliary normal variable is a relatively painless way to solve the ambiguity, without having to distinguish between planar and near-planar faces. The transformations are now always unique and comparable with faces on different planes, since they are extrinsic.

References

- [1] Amir Vaxman. Modeling polyhedral meshes with affine maps. *Computer Graphics Forum*, 31(5):1647–1656, 2012.
- [2] Amir Vaxman. libhedra: A polygonal mesh processing library, 2016.