Visualization of Floater and Gotsman's Morphing Algorithm

Ivaylo Ilinkin
Gettysburg College
300 N Washington St
Gettysburg, PA 17325, US
iilinkin@gettysburg.edu

ABSTRACT

This video provides a visualization of an algorithm proposed by Floater and Gotsman for morphing two polygonal tilings. The algorithm represents the interior vertices of the tilings as convex combinations of their neighbors. At each time step the convex coefficients are linearly interpolated and the interior vertices of the intermediate tilings are found as the solutions to a system of linear equations.

Categories and Subject Descriptors

I.3.5 [**Computational Geometry and Object Modeling**]: Geometric algorithms, languages, and systems

Keywords

Morphing, tiling, triangulation

1. INTRODUCTION

Morphing is the process of continuous transformation of one shape into another. It is a popular technique in graph drawing, solid modeling, and computer graphics. The video accompanying this paper provides a visualization of an algorithm due to Floater and Gotsman [5] for morphing two polygonal tilings, where a *tiling* is defined as a planar subdivision whose faces are convex polygons and whose boundary forms a convex polygon. A triangulation of a point set is a special case of a tiling whose faces are triangles. A desirable property of a morphing transformation is to avoid self-crossings and to ensure that the intermediate shapes are also valid tilings.

Let T^0 and T^1 be tilings with vertices $U^0 = (u_1^0, u_2^0, ..., u_N^0)$ and $U^1 = (u_1^1, u_2^1, ..., u_N^1)$, where the n interior vertices are listed first, followed by the m boundary vertices. The order of the vertices provides an implicit correspondence between the two shapes. A morphing transformation, ϕ, maps corresponding vertices and faces between the two tilings, i.e. $f_i^0 = (u_{i_1}^0, u_{i_2}^0, ..., u_{i_k}^0)$ is a face in tiling T^0, if and only if, $\phi(f_i^0) = (\phi(u_{i_1}^0), \phi(u_{i_2}^0), ..., \phi(u_{i_k}^0))$ is a face in tiling T^1. The morphing transformation produces a sequence of intermediate points $U^t = (u_1^t, u_2^t, ..., u_N^t)$, for $t \in [0, 1]$, and the goal is to ensure that U^t is a valid tiling free of self-crossings.

In Section 2 the tilings T^0 and T^1 are assumed to have identical boundaries. Section 3 outlines Floater and Gotsman's approach for morphing two tilings with different boundaries. Section 4 concludes with discussion of related work.

SoCG'14, June 8–11, 2014, Kyoto, Japan.
ACM 978-1-4503-2594-3/14/06.
http://dx.doi.org/10.1145/2582112.2595649

2. CONVEX MORPH

Floater and Gotsman [5] propose a morphing transformation that uses convex combinations to represent the interior vertices of the given tilings T^0 and T^1. The convex coefficients are interpolated at each time step t and are used to set up a system of linear equations whose solutions are the interior vertices of the intermediate tiling T^t.

2.1 Convex Combinations

Consider an interior vertex, u_i, of a given tiling T and let $U_i = (u_{i1}, u_{i2}, ..., u_{in_i})$ be the vertices of the faces incident to u_i listed in anti-clockwise order around u_i. Since the faces of T are convex polygons, the vertices in U_i form a star-shaped polygon, P_i, whose kernel contains u_i. Since u_i is in the kernel of P_i, the line through u_i and a vertex, u_{ij}, of P_i will intersect and edge (u_{ip}, u_{iq}) of P_i. The triangle $\tau_i^{(j)} = (u_{ij}, u_{ip}, u_{iq})$ contains u_i, and therefore, u_i can be expressed as a convex combination using barycentric coordinates:

$$u_i = \lambda_{ij}^{(j)} u_{ij} + \lambda_{ip}^{(j)} u_{ip} + \lambda_{iq}^{(j)} u_{iq},$$

where $\lambda_{ij}^{(j)} > 0$. (For convenience, let $\lambda_{ir}^{(j)} = 0$ for $r \neq j, p, q$.)

Overall there are n_i triangles (one for each vertex of P_i) each of which yields a convex representation of u_i. Letting

$$\lambda_{ij} = \frac{1}{n_i} \sum_{h=1}^{n_i} \lambda_{ij}^{(h)}, \qquad (\lambda_{ij} > 0 \text{ since } \lambda_{ij}^{(j)} > 0)$$

gives a convex combination of u_i in terms of its neighbors:

$$u_i = \sum_{j=1}^{n_i} \lambda_{ij} u_{ij},$$

with only positive coefficients [3, 5].

2.2 Algorithm

The morphing algorithm in [5] works as follows: Given the tiling T^0 (resp. T^1), compute the convex coefficients, λ_{ij}^0 (resp. λ_{ij}^1), for each interior vertex u_i^0 (resp. u_i^1). Next, at each time step t, obtain the convex coefficients of the

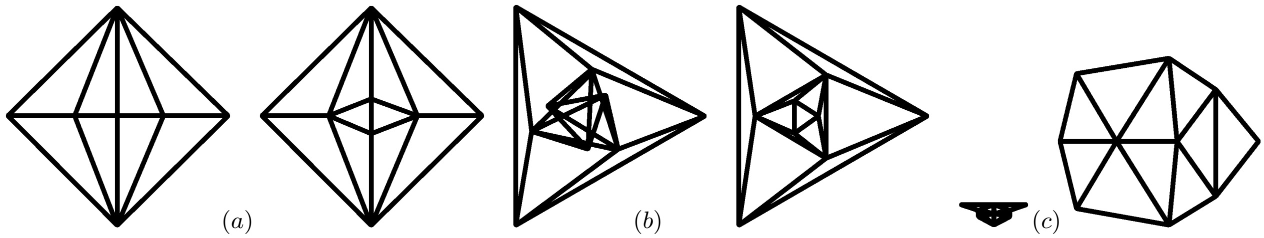

Figure 1: Comparison between (*linear*, *convex*) morph results for three different configurations [5] at $t = 1/2$.

intermediate tiling T^t via linear interpolation:

$$\lambda_{ij}^t = (1-t)\lambda_{ij}^0 + t\lambda_{ij}^1.$$

The convex coefficients can now be used to find the interior vertices of the intermediate tiling T^t by solving a system of linear equations, one for each interior vertex u_i^t:

$$\sum_{j=1}^{n_i} \lambda_{ij}^t u_{ij}^t = u_i^t.$$

It is shown that T^t is a valid tiling, i.e. free of self-crossings [5, 4, 3, 9]. Figures 1 (a) and (b) show a comparison of the intermediate tilings obtained by a *linear morph* and the *convex combinations* algorithm at time $t = 1/2$.

3. DIFFERENT BOUNDARIES

The algorithm in Section 2.2 does not apply directly when the two tilings, T^0 and T^1, have different boundaries, since it is not clear how to compute the intermediate boundary vertices. Floater and Gotsman adapt their algorithm by borrowing an approach suggested by Shapira and Rappoport [7]. The idea is to express the boundary vertices in polar coordinates with respect to their centroids c^0 and c^1.

Consider the m boundary vertices $(u_1^0, u_2^0, ..., u_m^0)$ of T^0 listed in anti-clockwise order. Then, $c^0 = \frac{1}{m}\sum_{i=1}^m u_i^0$. Next, each boundary vertex u_i^0 is expressed in polar coordinates, (r_i^0, θ_i^0), such that $r_i^0 = \|c^0 u_i^0\|$ and

$$0 \le \theta_1^0 < \theta_2^0 < ... < \theta_m^0 < \theta_1^0 + 2\pi.$$

The boundary vertices of T^1 are expressed similarly relative to c^1. Then, at each time step the centroids and the polar coordinates are interpolated:

$$c^t = (1-t)c^0 + tc^1$$

$$r_i^t = (1-t)r_i^0 + tr_i^1$$

$$\theta_i^t = (1-t)\theta_i^0 + t\theta_i^1$$

to obtain the boundary vertices $(u_1^t, u_2^t, ..., u_m^t)$ of T^t:

$$u_i^t = c^t + r_i^t(\cos\theta_i^t, \sin\theta_i^t).$$

After the boundary vertices of T^t have been determined, the interior vertices of T^t can be computed by applying the algorithm in Section 2.2. Figure 1(c) shows a comparison of the intermediate tilings obtained by a *linear morph* and the modified *convex combinations* algorithm at time $t = 1/2$.

4. DISCUSSION

The algorithm presented here has been extended in subsequent work. Observing that the method in Section 2.1 is just one possible approach to obtain convex combinations for the interior vertices of a given tiling, Surazhsky and Gotsman [8] discuss other schemes that offer better control over the morph in terms of vertex trajectories or areas of intermediate tiles. They also present an approach for intersection-free morphing of simple polygons [6] — the polygons are enclosed in convex shells with corresponding vertices, and then the polygons, as well as the regions between the polygons and their shells, are triangulated compatibly using an extension of the method in [2], which makes it possible to apply the algorithm presented here.

One disadvantage of the algorithm is that it does not provide a guarantee on the number of required steps and does not produce explicit vertex trajectories, but only snapshots at each step in the interpolation. In recent work, Alamdari et al. [1] show that two compatible triangulations can be morphed using $O(n^2)$ steps, where each step is a *linear morph*, and therefore, the resulting vertex trajectories are piece-wise linear.

5. REFERENCES

[1] S. Alamdari, P. Angelini, T. M. Chan, G. D. Battista, F. Frati, A. Lubiw, M. Patrignani, V. Roselli, S. Singla, and B. T. Wilkinson. Morphing planar graph drawings with a polynomial number of steps. In *Proc. 24th Symp. on Disc. Algo. (SODA)*, pages 1656–1667, 2013.

[2] B. Aronov, R. Seidel, and D. L. Souvaine. On compatible triangulations of simple polygons. *Computational Geometry: Theory and Applications*, 3:27–35, 1993.

[3] M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231–250, 1997.

[4] M. S. Floater. Parametric tilings and scattered data approximation. *International Journal of Shape Mdeling*, 4(3,4):165–182, 1998.

[5] M. S. Floater and C. Gotsman. How to morph tilings injectively. *Journal of Computational and Applied Mathematics*, 101(1-2):117–129, 1999.

[6] C. Gotsman and V. Surazhsky. Guaranteed intersection-free polygon morphing. *Computers & Graphics*, 25(1):67–75, 2001.

[7] M. Shapira and A. Rappoport. Shape blending using the star-skeleton representation. *IEEE Computer Graphics and Applications*, 15(2):44–50, Mar. 1995.

[8] V. Surazhsky and C. Gotsman. Controllable morphing of compatible planar triangulations. *ACM Transactions on Graphics*, 20(4):203–231, 2001.

[9] W. T. Tutte. How to draw a graph. *Proc. London Math. Soc.*, 3(13):743–768, 1963.