Constructing Differentiable Homeomorphisms between Isomorphic Triangulations

Frederick Crimins*   Diane Souvaine*

April 25, 2002

1 Introduction

Let $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, p_n\}$ be two sets of triangulated points in the plane such that every triangle $p_i p_j p_k$ corresponds to a triangle $q_i q_j q_k$, *i.e.* the triangulations are isomorphic. We aim to construct a differentiable homeomorphism between the triangulated areas defined by P and Q.

A *homeomorphism* is a one-to-one, continuous mapping of R^d onto R^d with a continuous inverse. These mappings are used in such fields as computer graphics for mapping a rectangular image onto a smooth or triangulated object. Methods for constructing piecewise linear homeomorphisms are shown in [1, 2]. However, a piecewise linear homeomorphism has an order-0 (C^0) maximum achievable level of continuity on the boundaries between the linearly transformed pieces. With respect to image mapping, this means that noticeable irregularities will occur along these boundaries. Therefore, a homeomorphism with higher order continuity, a differentiable homeomorphism, is preferable for such applications. A *differentiable homeomorphism* is a one-to-one, differentiable mapping of R^d onto R^d with a differentiable inverse.

In the following section we present a homeomorphism which we believe to be differentiable. We explain the reasoning that spawned this construction as well as how to generate the actual homeomorphism. In Section 3 we outline some of the limitations of the mapping, and we discuss methods for overcoming these issues. We then describe our current implementation of the homeomorphism and where this research is headed in Sections 4 and 5.

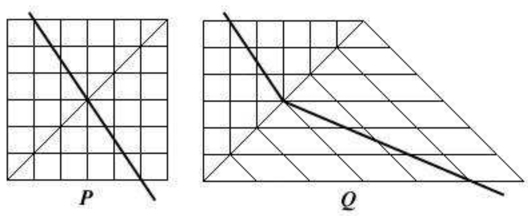

Figure 1: The effect of a piecewise linear homeomorphism mapping P onto Q.

2 A New Coordinate System

When a piecewise linear homeomorphism is applied to a triangulated polygon, each triangle undergoes a unique linear transformation in which the transformed image of a line that crosses over an edge shared by two triangles will remain continuous but not necessarily differentiable. For example, in Figure 1, a piecewise linear homeomorphism mapping P onto Q will map the line l_p to the line l_q. Clearly, l_q is not differentiable at the point where it crosses over the edge shared by Q's two triangles.

It is easier to see the full effect of the homeomorphism by examining the grid lines in P as they also become indifferentiable when mapped onto Q. A coordinate system that does not maintain smooth grid lines clearly cannot support a globally differentiable mapping. Resolving this shortcoming is the basis of

*EECS Department, Tufts University, Medford, MA 02155, fcrimins/dls@eecs.tufts.edu. Research supported in part by NSF grant EIA-99-96237

our new coordinate system, in which we will maintain smooth grid lines between adjacent triangles. More precisely, we will require that every grid line be perpendicular to the edge of a triangle at its point of intersection with that edge. However, this is not a simple construction, mainly because it is unclear how to make all of the grid lines perpendicular to triangle edges even as they approach a vertex of a triangle.

Our solution to this problem is based on the following construction (see Figure 2). First find the centroid of the triangle. This point is the intersection of the three segments between each vertex and the midpoint of the opposite edge. From the centroid, construct line segments, l_A, l_B, and l_C, perpendicular to each of the three edges of the triangle. We explain later why these segments, called *center segments*, are actually grid lines.

Notice that it is possible to construct a triangle for which one of these segments extends outside the triangle before it intersects the line containing its respective edge. For now, we will assume that this is not the case. This issue is discussed further in Section 3.

Next, construct line segments, which we will call *wing segments*, from each triangle vertex to the midpoint of both visible center segments that lie on either side of that vertex (s_{A1}, s_{A2}, s_{B1}, s_{B2}, s_{C1}, and s_{C2}). These nine segments, three center segments and six wing segments, will be known as *contour guides.* We use these contour guides to construct the grid lines of our coordinate system.

Each grid line or *contour* consists of a straight line segment c and a circular arc a. The segment c emanates from the centroid of the triangle in a polar fashion until it intersects a wing segment (*e.g.* s_{A2}). The arc a is then constructed tangent to the line containing c at its point of intersection with the wing segment; and from there, a curves toward the edge of the triangle so that it is perpendicular to that edge. The three center segments are degenerate cases of grid lines because the arcs in these contours are straight line segments. Notice that as these composite grid lines approach a vertex of the triangle, the radius of the resulting arc approaches zero. Nonetheless, all of the contours remain perpendicular as they intersect the edges of the triangle.

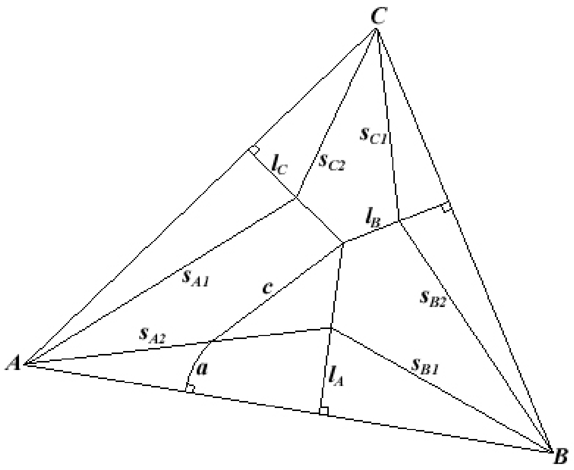

Figure 2: The contour guides aid the construction of a contour.

We use these contours to specify the coordinates (e, p, c) of a point X within a triangle. We call this coordinate system the *triangular coordinate system.* In a set of triangular coordinates, the e-coordinate identifies the edge of the triangle intersected by the contour through X. The set of edge labels, 1, 2, and 3, is therefore a sufficient domain for e. The p-coordinate may assume any value between 0 and 1 and it indicates the position at which X's contour intersects edge e. This value is specified as a proportion of the total length of e, *e.g.*, if $p = 0.5$, then X's contour will intersect e at the midpoint of e. The c-coordinate also assumes a value on the $[0, 1]$ interval. This coordinate identifies X's distance from the centroid as measured along the contour of X and as a proportion of the length of this contour. For example, if X lies on an edge of the triangle then $c = 1$ and if X is the centroid of the triangle then $c = 0$.

By use of triangular coordinates rather than Cartesian coordinates, we hypothesize that a differentiable homeomorphism can be defined between two isomorphically triangulated polygons P and Q. For a point p in P, first find the bounding triangle of p in P's triangulation. Then, with respect to this triangle, determine the triangular coordinates of p. Finally, map the image of p in the corresponding isomorphic triangle in Q's triangulation using the same triangular coordinates as p. Clearly this process also requires that the edge labellings in isomorphic triangles are isomorphic.

This may seem like a simple process. However, the complexity of one step has been severely overlooked; determining the triangular coordinates of a point X is not a trivial task. For several reasons, in particular implementation, the conversion from Cartesian to triangular coordinates must be feasible. This conversion is in fact possible and is based on determining the contour that passes through X. If X lies within the hexagon formed by the six wing segments of the triangle, this calculation is simple because two points on the line segment portion of the contour are known, namely X and the centroid. However, if X lies to the outside of this hexagon, then this computation is much more difficult, mainly because none of the properties of the contour are readily obvious. Finding the contour in this case involves solving a cubic polynomial. Due to space constraints, a full explanation of this complex calculation has been excluded from this paper.

3 Limitations

As mentioned in the last section, it is possible to construct a triangle in which one of the edge-perpendiculars drawn from the centroid extends outside the triangle before it intersects the line containing its respective edge. For example, see Figure 3.

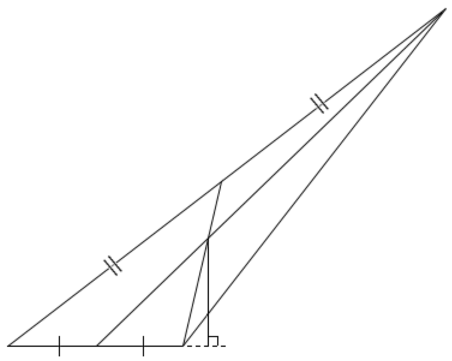

Figure 3: The perpendicular constructed from the centroid to the lower edge of this triangle extends outside of the triangle before intersecting the line containing that edge. Therefore, triangular coordinates cannot be applied to this triangle.

This suggests that using an alternate center point might be necessary. It is possible to locate a point within every triangle, such as the incenter, for which the edge-perpendicular segments are completely contained in the triangle. However, by viewing specific cases, it is evident from the implementation that any point other than the centroid will not support a differentiable homeomorphism.

In general, there are two types of triangles in which the contour guides cannot properly be constructed. Triangles containing grossly obtuse angles often cause this disfunctionality; and in addition, triangles that contain an obtuse angle between a very small and a very large side are also susceptible. However, it is easy to show that this construction can be successfully applied to all acute and isosceles triangles.

We have also shown that it is not possible to deconstruct a discordant triangle into usable triangles with the help of additional interior points or Steiner points. The complexity of this process is compounded by the requirement that additional points would be inserted into a discordant triangle's isomorphic triangle as well. Isomorphic triangulations that avoid triangles of this type are of great interest.

Achieving an optimal triangulation by minimizing the maximum angle in a triangulated mesh has been the source of extensive research. Lawson [4] has shown that the Delauney triangulation achieves this property in two dimensions. In addition, Bern and Eppstein [3] discuss optimal triangulations while examining various formulations of optimality.

4 Implementation

We have implemented this homeomorphism in C++ using the Library of Efficient Data types and Algorithms (LEDA) version 4.1, maintained by Algorithmic Solutions. Our implementation can be found at http://www.eecs.tufts.edu/r/geometry/dhm. Our program generated the images shown in Figure 4 by

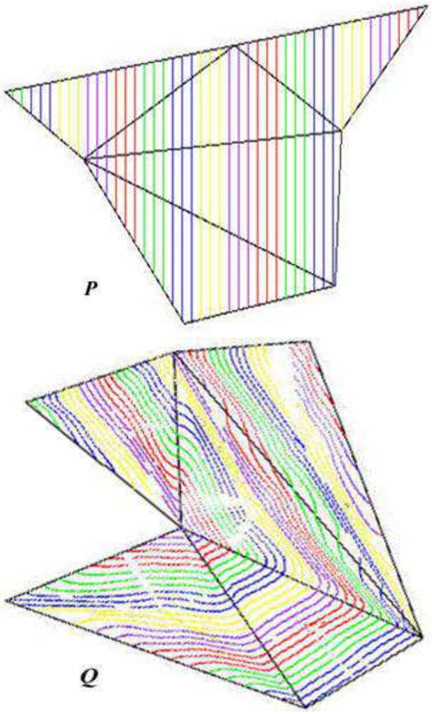

Figure 4: The effect of a homeomorphism mapping P onto Q using triangular coordinates.

applying the mapping described above between the two isomorphically triangulated polygons. The program first calculated the triangular coordinates from the Cartesian coordinates of the points along a series of vertical parallel lines in the first polygon. It then used those same triangular coordinates to compute the Cartesian coordinates of the points in the second polygon. As you can see, the image of the parallel lines is smooth as it crosses triangle boundaries.

A drawback of this implementation is that rounding and calculation errors frequently occur when converting points from Cartesian to triangular coordinates.

5 Further Research

One outstanding question has emerged from our research of this construction: How do you prove that this homeomorphism is differentiable? The conversion of a triangular coordinate to a Cartesian coordinate is rather complex; it is far more complex than the conversion from polar to Cartesian. As a result, the formulation of a function obtained using the triangular coordinate system in terms of Cartesian or polar coordinates is also unclear. In addition, the requirement that the median be used as the center point for the coordinates is not fully understood. These issues contribute substantially to the difficulty of proving the differentiability over triangular boundaries of this homeomorphism. Methods similar to those used for analyzing primal-dual relationships have been suggested, however further insights are needed.

References

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

[2] M. Babikov, D. Souvaine, and R. Wenger. Constructing piecewise linear homeomorphisms of polygons with holes. *Proc. 9th CCCG*, pages 6–10, 1997.

[3] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. In *Computing in Euclidean Geometry, Edited by Ding-Zhu Du and Frank Hwang, World Scientific, Lecture Notes Series on Computing*, volume 1. 1992.

[4] C.L. Lawson. Software for c^1 surface interpolation. In J. Rice, editor, *Mathematical Software III*, pages 161–194. Academic Press, 1977.