Straightest Paths on Meshes by Cutting Planes

Sungyeol Lee, Joonhee Han, and Haeyoung Lee

Hongik University, Dept. of Computer Engineering,
72-1 Sangsoodong Mapogu, Seoul Korea 121-791
{leesy, hanj, leeh}@cs.hongik.ac.kr

Abstract. Geodesic paths and distances on meshes are used for many applications such as parameterization, remeshing, mesh segmentation, and simulations of natural phenomena. Noble works to compute shortest geodesic paths have been published. In this paper, we present a new approach to compute the straightest path from a source to one or more vertices on a manifold mesh with a boundary. A cutting plane with a source and a destination vertex is first defined. Then the straightest path between these two vertices is created by intersecting the cutting plane with faces on the mesh. We demonstrate that our straightest path algorithm contributes to reducing distortion in a shape-preserving linear parameterization by generating a measured boundary.

1 Introduction

Calculating geodesic paths and distances on meshes is a fundamental problem in various graphics applications such as parameterization [10], remeshing [6,13,18], mesh segmentation [18], and simulations of natural phenomena [15,9].

Noble works to compute shortest geodesic paths and distances have been presented [12,1,7,19]. An algorithm for straightest geodesic paths [14] was first proposed by Polthier and Schmies. Their straightest geodesics are well defined with the initial condition (a source and direction) but not with boundary conditions (a source and a destination). The straightest path may not be the same as the shortest path on meshes and may be more appropriate for some applications like wave propagation [15] or parameterization [10] for texture mapping.

In this paper, we present a new and simple algorithm to compute straightest paths and distances between two vertices on manifold meshes with a boundary as shown in Figure 1. Our straightest path algorithm enables us to create a measured boundary for a linear parameterization and hence reduces distortion more than the parameterization with a fixed boundary as shown in Figure 3 and 4.

1.1 Related Work

There are several algorithms for geodesic computations on meshes, mostly based on shortest paths with boundary conditions i.e., source and destination vertices.

Derived from Dijkstra's algorithm, The MMP algorithm [12] is introduced to compute an exact shortest geodesic path. The CH algorithm [1] also allows the user to compute an exact shortest path with faster processing. Their implementation and extension are also proposed by [5,6,19] but they are still considered

M.-S. Kim and K. Shimada (Eds.): GMP 2006, LNCS 4077, pp. 609–615, 2006.

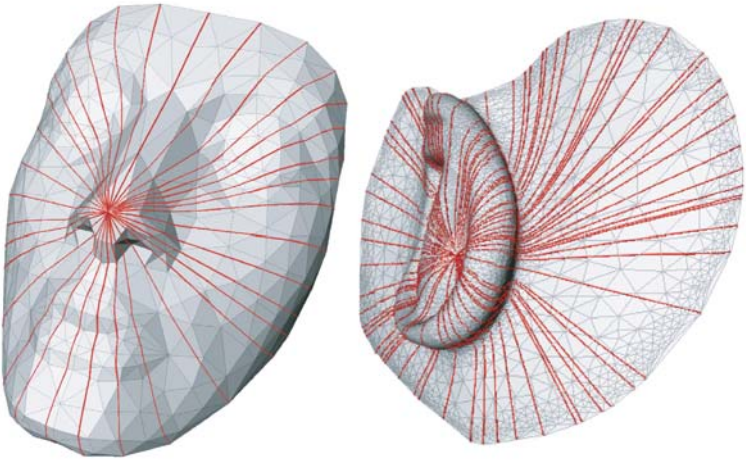

Fig. 1. Straightest paths from a source to every vertex on the boundary as determined by our method. Models are Nefertiti on the left and an Ear on the right.

hard to implement. The Fast Marching algorithm by [7] computes an approximate geodesic path on meshes and has been used for remeshing and parameterization [18,13]. However, it requires special processing for triangles with obtuse angles. A detailed overview of this approach can be seen in [11].

Another approach is to compute the straightest geodesic path. Polthier and Schimes [14] presented an algorithm to compute a straightest geodesic path on a mesh. They extended the notion of straight lines from the Euclidean plane onto the surface. It is uniquely defined with a source and direction. However it is not always defined between a source and a destination and also requires special handling of the swallow tails created by conjugate vertices [15] and triangles with obtuse angles [9].

Our method in this paper is designed to compute straightest paths between a source and one or more vertices on manifold meshes with a boundary.

2 Our Straightest Path Algorithm with Cutting Planes

We present a new and simple algorithm to compute a straightest path from a source to one or more destinations on a genus-0 surface patch. Figure 2 explains our simple steps to compute a straightest path between a source S and a destination D.

1. Specify a center S which is the point of origin and determine the vertex normal at S as follows:
   - Make virtual edges from S to every boundary vertex of the mesh.
   - Make virtual faces around S with these virtual edges.
   - Average normal vectors of virtual faces around S and assign it to the vertex normal vector at S.

   Then design the base plane B at S with the above vertex normal.