Parameterization of 3D Surface Patches by Straightest Distances

Sungyeol Lee and Haeyoung Lee

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

Abstract. In this paper, we introduce a new piecewise linear parameterization of 3D surface patches which provides a basis for texture mapping, morphing, remeshing, and geometry imaging. To lower distortion when flatting a 3D surface patch, we propose a new method to locally calculate straightest distances with cutting planes. Our new and simple technique demonstrates competitive results to the current leading parameterizations and will help many applications that require one-to-one mapping.

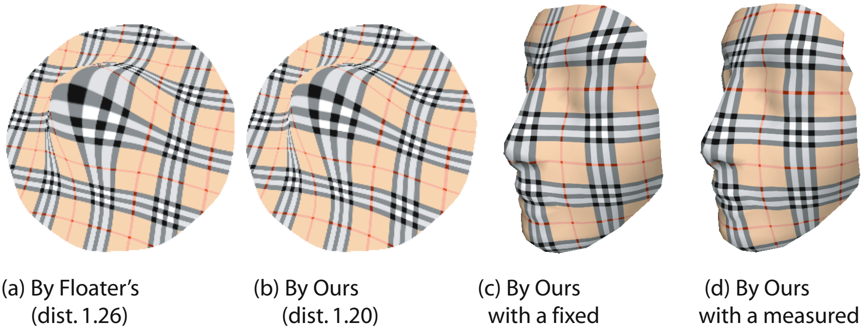
1 Introduction

A 3D mesh parameterization provides a piecewise linear mapping between a 3D surface patch and an isomorphic 2D patch. It is a widely used or required operation for texture-mapping, remeshing, morphing or geometry imaging. Guaranteed one-to-one mappings that only requires a linear solver have been researched and many algorithms [4,5,11,8,10] were proposed. To reduce inevitable distortions when flattening, a whole object is usually partitioned into several genus 0 surface patches. Non-linear techniques [19] are also presented with good results in some applications but they require more computational time than linear methods.

Geodesics on meshes have been used in various graphics applications such as parameterization [10], remeshing [14,20], mesh segmentation [20,6], and simulations of natural phenomena [16,9]. Geodesics provide a distance metric between vertices on meshes while the Euclidean metric can not. Straightest geodesic path on meshes was introduced by Polthier and Schmies [15] and used for parameterization by [10]. However their straightest geodesics may not be defined between a source and a destination and require special handling of the swallow tails created by conjugate vertices [16] and triangles with obtuse angles [9].

In this paper we present a new linear parameterization of 3D surface patches. Our parameterization is improved upon [10] by presenting a new way to locally calculate straightest geodesics. Our method demonstrates visually and statistically competitive results to the current leading methods [5,10] as shown in Figure 1, 3, 5, and Table 1.

Y. Shi et al. (Eds.): ICCS 2007, Part II, LNCS 4488, pp. 73–80, 2007.



(a) By Floater's (dist. 1.26)

(b) By Ours (dist. 1.20)

(c) By Ours with a fixed

(d) By Ours with a measured

Fig. 1. Comparisons with texture-mapped models, Hat and Nefertiti: (a) is resulted by Floater's [5] with a distortion of 1.26. (b) is by our new parameterization with a distortion of 1.20, less than by Floater's. The distortion is measured by the texture stretch metric [19]. (c) is by ours with a fixed boundary and (d) is also by ours with a measured boundary.We can see much less distortion in (d) than (c).

1.1 Related Work

Parameterization. There has been an increased need for a parameterization for texture-mapping, remeshing, morphing or geometry imaging. Many piecewise linear parameterization algorithms [4,5,11,8,10] were proposed. Generally the first step for parameterization is mapping boundary vertices to a fixed position. Usually the boundary is mapped to a square, a circle, or any convex shape while respecting the 3D-to-2D length ratio between adjacent boundary vertices. The positions of the interior vertices in the parameter space are then found by solving a linear system. The linear system is generated with coefficients in a convex combination of 1-ring neighbors for each interior vertex. These coefficients characterize geometric properties such as angle and/or area preserving.

Geodesic Paths. There are several algorithms for geodesic computations on meshes, mostly based on shortest paths [13,1,7] and have been used for remeshing and parameterization [20,14]. However, still special processing for triangles with obtuse angles is required. A detailed overview of this approach can be seen in [12].

Another approach is to compute the straightest geodesic path. Polthier and Schmies first introduced an algorithm for the straightest geodesic path on a mesh [15]. Their straightest geodesic path is uniquely defined with the initial condition i.e., a source vertex and direction but not with boundary conditions i.e., a source and a destination. A parameterization by straightest geodesics was first introduced in [10]. They used locally calculated straightest geodesic distances for a piecewise linear parameterization. Our parameterization is improved upon [10] by presenting a new way to calculate straightest geodesics.

2 Our Parameterization by Straightest Distances

A 3D mesh parameterization provides a piecewise linear mapping between a 3D surface patch and an isomorphic 2D patch. Generally the piecewise linear parameterization is accomplished as follows: for every interior vertex V_i of a mesh, a linear relation between the (u_i, v_i) coordinates of this point and the (u_j, v_j) coordinates of its 1-ring neighbors $\{V_j\}_{j \in \mathcal{N}(i)}$, is set of the form:

$$\sum_{j \in \mathcal{N}(i)} a_{ij}(U_j - U_i) = 0 \tag{1}$$

where $U_i = (u_i, v_i)$ are the coordinates of vertex V_i in the parameter space, and a_{ij} are the *non-negative* coefficients of matrix A. The boundary vertices are assigned to a circle, or any convex shape while respecting the 3D-to-2D length ratio between adjacent boundary vertices. The parameterization is then found by solving the resulting linear system $AU = B$. A is sparse because each line in the matrix A contains only a few non-zero elements (as many as the number of its neighbors). A preconditioned bi-conjugate gradient (PBCG) method [17] is used to iteratively solve this sparse linear system.

As long as the boundary vertices are mapped onto a convex shape, the resulting mapping is guaranteed to be one-to-one. The core of this shape-preserving parameterization is how to determine non-negative coefficients a_{ij}. In this paper, we propose a new algorithm to determine these coefficients.

2.1 Our Local Straightest Distances

The core of this piecewise linear parameterization is finding *nonnegative* coefficients a_{ij} in the equation 1. Our new parameterization proposes to determine these coefficients by using locally straightest paths and distances with local cutting planes. The work by Lee et. al. [10] uses local straightest geodesics by Polthier and Schmies's [15] for these coefficients, however the tangents of the straightest geodesics by this previous method are determined by gaussian curvatures at vertices and may not be intuitively straightest especially when the gaussian curvature is not equal to 2π. In Figure 2, V_{ps} is determined by having the same left and right angle at V_i by [10], while V_{our} is determined intuitively straightest by our local cutting plane.

Our new method for local straightest paths and distances is determined as follows. As shown in Figure 2, a base plane B is created locally at each interior vertex. To preserve shape better, the normal $Normal_B$ of the base plane$B$ is calculated by area-weighted averaging of neighboring face normals of V_i as shown in equation 2 and normalized later.

$$Normal_B = \sum_{j \in \mathcal{N}(i)} w_j Normal_j \tag{2}$$

In this way, we found that the distortion is lower than a simple averaged normal of neighboring faces. A local cutting plane P passing with V_i, V_j is also calculated. Two planes intersect in a line as long as they are not parallel. Our cutting plane P pierces a neighboring face (for example j-th neighboring face) on the mesh. Therefore there is a line segment which is the straightest path by

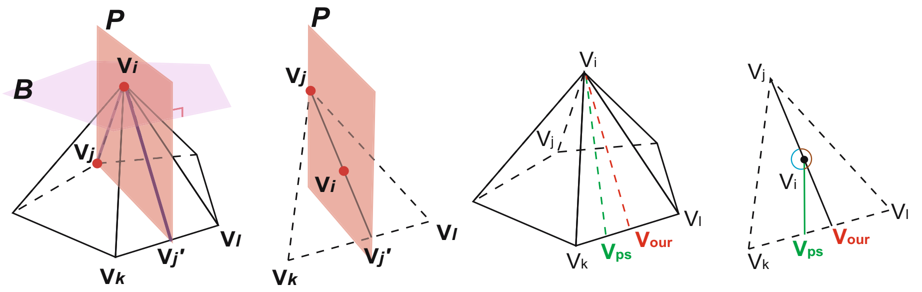


Fig. 2. Our new local straightest path: For each interior vertex V_i, a local base B and a cutting plane P with V_i, V_j is created. A local straightest path is computed by cutting the face $V_iV_kV_l$ with P. The intersection $V_{j'}$ is computed on the edge V_kV_l and connected to V_i to form a local straightest path. V_{ps} is determined by the Polthier and Schimes's [15] and V_{our} is determined by our new method.

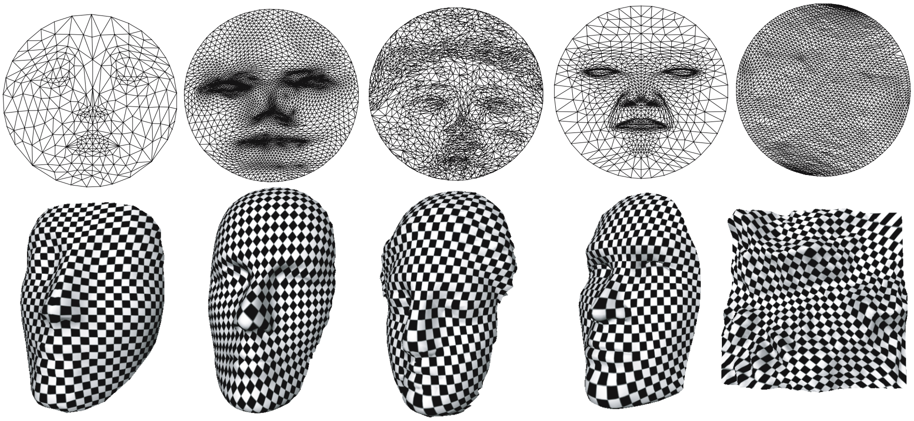


Fig. 3. Results by our new parameterization: models are Nefertiti, Face, Venus, Man, Mountain from the left to the right

our method. There may be multiple line intersections where the plane P may pierce multiple neighboring faces. As a future work, we will explore how to select a line segment.

A local straightest path is computed by intersecting the face $V_iV_kV_l$ and the cutting plane P. The tangent a for this intersecting line segment $V_jV_{j'}$ can be easily calculated from the normal $Normal_j$ of the face $V_iV_kV_l$ and the normal $Normal_p$ of the cutting plane P as follows:

$$a = Normal_j X Normal_c \tag{3}$$

Then, the intersection vertex $V_{j'}$ is computed on the edge V_kV_l and connected to V_i for the local straightest path $V_jV_iV_{j'}$. Finally barycentric coordinates for the weights of V_j, V_k, V_l are computed, summed, normalized and then used to fill up the matrix A. Figure 3 shows the results of our new parameterization.

2.2 Discussion

Floater's [5] is considered as the widely used parameterization and LTD's [10] also used a straightest geodesic path algorithm by [15]. Therefore we compare our method to the two existing parameterizations.

The visual results achieved by our new parameterization are shown in Figure 3. The distortion with the texture-stretch metric in [19] is also measured and shown in Table 1. Notice that our parameterization produces competitive results to the current leading linear parameterizations. With measured boundary The previous algorithms and the distortion metric (L^2-norm, the mean stretch over all directions) are all implemented by us.

3 Measured Boundary for Parameterization

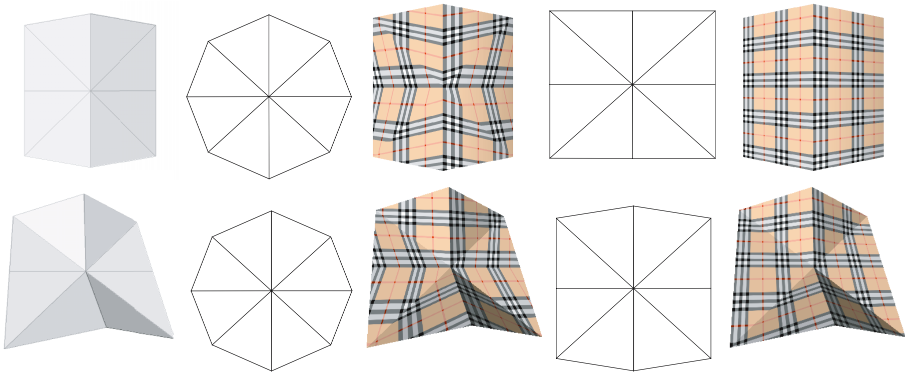
As shown in Figure 4 (b) and (c), and the 1st and 2nd figures in Figure 3, high distortion always occurs near the boundary. To reduce this high distortion, we attempt to derive a boundary by our straightest geodesic path algorithm.

An interior source vertex S can be specified by a user or calculated as a center vertex of the mesh from the boundary vertices. A virtual edge is defined as an edge between S and a vertex on the boundary. Straightest paths and distances of virtual edges to every vertex on the boundary will be measured as follows:

1. Make virtual edges connecting from S to every boundary vertex of the mesh.
2. Map each virtual edge onto the base plane B by a polar map, which preserves angles between virtual edges such as [4]. The normal of the base plane B is calculated as previously mentioned in 2.
3. Measure the straightest distance for each virtual edge on B from S to each boundary vertices with corresponding cutting planes.
4. Position each boundary vertex at the corresponding distance from S on B.
5. If the resulted boundary is non-convex shaped, change it to a convex. Find the edges having minimum angle with the consecutive edge (i.e., concaved part of the boundary) and move the boundary vertex to form a convex.

In the linear system $AU = B$, the boundary vertices in B is simply set to the measured position (u_i, v_i) and $(0, 0)$ for inner vertices. Then PBCG as mentioned in 2 is used to find positions in the parameterized space.

Figure 4 (d) and (e) clearly shows the utility of our straightest geodesic paths with the simple models Testplane on the top and Testplane2 on the bottom. With a circular boundary, previous parameterizations [5,10] produce the same results in (b) for two different models. In (c), there is also a high distortion in the texture-mapping by using (b). Our straightest path algorithm contributes to deriving two distinct measured boundaries and results in very low distortion in (d) and much better texture-mapping in (e).



(a) Models (b) Circular boundary (c) Textured by (b) (d) Measured boundary (e) Textured by (d)

Fig. 4. Comparisons between parameterizations with a fixed boundary and a measured boundary by our new method: With a circular boundary, previous parameterizations [5,10] produce the same results in (b) for two different models in (a). Notice in (c) that there are a lot of distortion in the texture-mapping by the results in (b). Our straightest path algorithm contributes to creating a measured boundary to reduce distortion by distinct results in (d) and much better texture-mapping in (e).

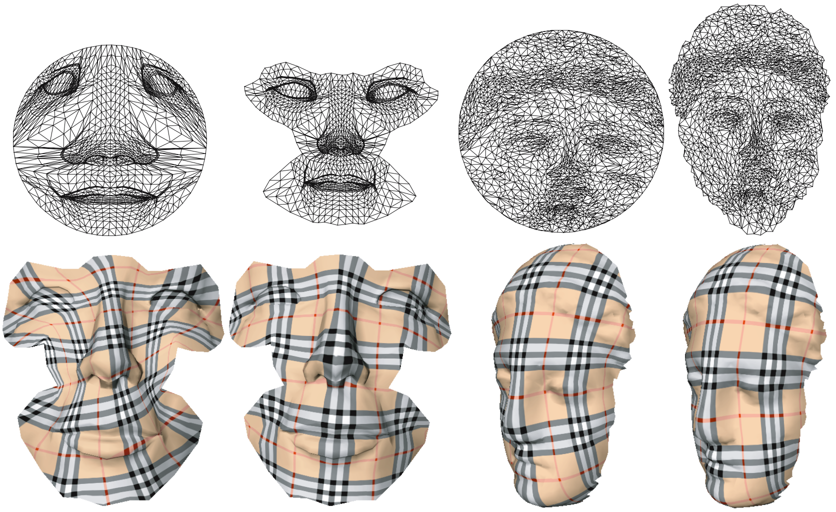


Fig. 5. Results by our new parameterization with different boundaries. Models are Face in the two left and Venus on the two right columns. The tip of the nose on each model is chosen as S.

Results with more complex models are demonstrated in Figure 5. Notice that there is always a high level of distortion near the fixed boundary but a low level of distortion near the measured boundary by using our method. The straightest distances to the boundary vertices are actually dependent on the selection on the source vertex S. We simply use a vertex centered on the mesh from the boundary as for the source S. As a future work, we will explore how to select the vertex S.

4 Results

The visual results by our method are shown in Figure 1, 3, and 5. The statistical results comparing our parameterization with other methods are listed in Table 1. Notice that visually and statistically our methods produce competitive results than the previous methods.

Table 1. Comparisons of distortion measured by the texture stretch metric [19]: The boundary is fixed to a circle. Combined with measured boundaries by our straightest path algorithm, our new parameterization in the 6th column produces better results than the current leading methods.

Models	No. of Vertices	Floater's [5] fixed bound.	LTD's [10] fixed bound.	Our Param. fixed bound.	Our Param. measured bound.
Nefertiti	299	1.165	1.165	1.164	1.146
Man	1208	1.244	1.241	1.240	1.226
Face	1547	1.223	1.222	1.221	1.334
Venus	2024	2.159	2.162	2.168	1.263
Mountain	2500	1.519	1.552	1.550	1.119

The performance complexity of our algorithm is all linear to the number of vertices, i.e., $O(V)$. The longest processing time among our models in Table 1 is 0.53 sec, required for the Mountain having the highest number of vertices. The processing time is measured on a laptop with a Pentium M 2.0GHz 1GB RAM.

5 Conclusion and Future Work

In this paper, we introduce a new linear parameterization by locally straightest distances. We also demonstrate the utility of our straightest path algorithm to derive a measured boundary for parameterizations with better results.

Future work will extend the utility of our straightest path algorithm by applying it to other mesh processing techniques such as remeshing, subdivision, or simplification.

Acknowledgement

This work was supported by grant No. R01-2005-000-10120-0 from Korea Science and Engineering Foundation in Ministry of Science & Technology.

References

1. Chen J., Han Y.: "Shortest Paths on a Polyhedron; Part I: Computing Shortest Paths", Int. J. Comp. Geom. & Appl. 6(2), 1996.
2. Desbrun M., Meyer M., Alliez P.: "Intrinsic Parameterizations of Surface Meshes", Eurographics 2002 Conference Proceeding, 2002.
3. Floater M., Gotsman C.: "How To Morph Tilings Injectively", J. Comp. Appl. Math., 1999.
4. Floater M.: "Parametrization and smooth approximation of surface triangulations", Computer Aided Geometric Design, 1997.
5. Floater M.: "Mean Value Coordinates", Comput. Aided Geom. Des., 2003.
6. Funkhouser T., Kazhdan M., Shilane P., Min P.,Kiefer W., Tal A., Rusinkiewicz S., Dobkin D.: "Modeling by example", ACM Transactions on Graphics, 2004.
7. Kimmel R., Sethian J.A.: "Computing Geodesic Paths on Manifolds", Proc. Natl. Acad. Sci. USA Vol.95 1998, 1998.
8. Lee Y., Kim H., Lee S.: "Mesh Parameterization with a Virtual Boundary", Computer and Graphics 26 (2002), 2002.
9. Lee H., Kim L., Meyer M., Desbrun M.: "Meshes on Fire", Computer Animation and Simulation 2001, Eurographics, 2001.
10. Lee H., Tong Y. Desbrun M.: "Geodesics-Based One-to-One Parameterization of 3D Triangle Meshes", IEEE Multimedia January/March (Vol. 12 No. 1), 2005.
11. Meyer M., Lee H., Barr A., Desbrun M.: "Generalized Barycentric Coordinates to Irregular N-gons", Journal of Graphics Tools, 2002.
12. Mitchell J.S.B.: "Geometric Shortest Paths and network optimization", In Handbook of Computational Geometry, J.-R. Sack and J. Urrutia, Eds. Elsevier Science 2000.
13. Mitchell J.S.B., Mount D.M., Papadimitriou C.H.: "The Discrete Geodesic Problem", SIAM J. of Computing 16(4), 1987.
14. Peyré G., Cohen L.: "Geodesic Re-meshing and Parameterization Using Front Propagation", In Proceedings of VLSM'03, 2003.
15. Polthier K., Schmies M.: "Straightest Geodesics on Polyhedral Surfaces", Mathematical Visualization, 1998.
16. Polthier K., Schmies M.: "Geodesic Flow on Polyhedral Surfaces", Proceedings of Eurographics-IEEE Symposium on Scientific Visualization '99, 1999.
17. Press W., Teuklosky S., Vetterling W., Flannery B.: "Numerical Recipes in C, second edition", Cambridge University Press, New York, USA, 1992.
18. Riken T., Suzuki H.: "Approximate Shortest Path on a Polyhedral Surface Based on Selective Refinement of the Discrete Graph and Its Applications", Geometric Modeling and Processing 2000 (Hongkong), 2000.
19. Sander P.V., Snyder J., Gortler S.J., Hoppe H.: "Texture Mapping Progressive Meshes", Proceedings of SIGGRAPH 2001, 2001.
20. Sifri O., Sheffer A., Gotsman C. : "Geodesic-based Surface Remeshing", In Proceedings of 12th Intnl. Meshing Roundtable, 2003.