Approximate Geodesics on Smooth Surfaces of Arbitrary Topology

Shuhua Lai[1] and Fuhua (Frank) Cheng[2]

[1] Department of Mathematics & Computer Science,
School of Engineering, Science and Technology,
Virginia State University, Petersburg, VA 23806, slai@vsu.edu
[2] Graphics & Geometric Modeling Lab,
Department of Computer Science,
University of Kentucky, Lexington, KY 40506, cheng@cs.uky.edu

Abstract. This paper introduces a new approach for computing large number of approximate geodesic paths from a given point to all directions on a 3D model (mesh or surface) of arbitrary topology. The basic idea is to unfold the 3D model into a flat surface so that the geodesic from a given point in a given direction can be obtained simply by drawing a straight line from the given point along the given direction on the unfolded surface. Hence our method does not require setting up any linear systems, nor any expensive matrix computation, but is simply done by iteratively extending the geodesic path along the given direction until the geodesic path reaches a certain length. The iterative process proceeds with a linear complexity. Therefore the new approach is very fast and can be used for meshes with large number of vertices. The smooth surface representation scheme used in this paper is Catmull-Clark subdivision surfaces, but the same idea can be applied to other subdivision schemes as well. Some test results obtained using this method are included. They show the effectiveness of our approach.

1 Introduction

In mathematics, a geodesic is a generalization of the notion of a "straight line" to "curved spaces". In the presence of a metric, a geodesic is defined to be (locally) the shortest path between two points in the space. In the presence of an affine connection, a geodesic is defined to be the curve whose tangent vectors remain parallel if they are transported along it. The computation of geodesic paths is needed in many computer graphics applications. As a matter of fact, recent research in finding algorithms for robot motion, terrain navigation and surface cutting has resulted in a number of interesting variants of the shortest path problem. However, most of the research work deals with the shortest path between two points in three dimensions in the presence of polyhedral obstacles. In this paper we present an approach to find geodesic paths from a given point along any given directions on smooth surfaces of arbitrary topology.

Subdivision surfaces [1] have become popular recently because of their capability in modeling/representing any complex shape with only one surface and because of their relatively high *visual quality*, *numerical stability*, *simplicity* in implementation. Subdivision surfaces cover both parametric forms [2, 3] and discrete forms. Parametric forms are good for design and representation and discrete forms are good for machining and tessellation (including FE mesh generation). Therefore we have a representation scheme that is good for almost all applications. Geodesic computation techniques using subdivision surfaces as a representation scheme certainly are needed for subdivision surface based modeling/design.

In this paper we describe a new approach for computing approximate geodesic paths from a given point to all directions on a 3D model (mesh or surface) of arbitrary topology. The basic idea is to unfold the 3D model into a flat surface so that the geodesic from a given point in a given direction can be obtained simply by drawing a straight line from the given point along the given direction on the unfolded surface. Hence our method does not require setting up any linear systems, nor any expensive matrix computation, but is done by iteratively extending the geodesic path along the given direction until the geodesic path intersects with itself or reaches a certain length. The iterative process proceeds with a linear complexity. Therefore the new approach is very fast and can be used for meshes with large number of vertices. The geodesic computation algorithm presented in this paper is obtained using the concept of Catmull-Clark subdivision surfaces, but the same idea can be applied to other subdivision schemes as well. The capability of the new approach is demonstrated with test examples shown in the paper.

2 Previous Work

2.1 Related work on computing geodesics

Mitchell, Mount, and Papadimitriou (MMP) [4] first presented an efficient implementation of the exact geodesic algorithm for triangular mesh in 1987. Their algorithm has $O(n^2 \log n)$ worst-case time complexity, but in practice can work with million-node meshes in reasonable time. The MMP algorithm [4] provides an exact solution for the "single source, all destinations" shortest path problem on a triangular mesh. Their algorithm partitions each mesh edge into a set of intervals (windows) over which the exact distance computation can be performed automically. These windows are propagated in a "continuous Dijkstra"-like manner. An exact geodesic algorithm with worst case time complexity of $O(n^2)$ was described in [6]. Kapoor [7] described an algorithm for the "single source, single destination" geodesic path between two given mesh vertices, in $O(n \log_2 n)$ time.

Approximate geodesics with guaranteed error bounds can be obtained by adding extra edges into the mesh and running Dijkstra on the one-skeleton of this augmented mesh [8]. Many extra edges are required to obtain accurate geodesics. Many of these methods require special processing of triangles with obtuse angles. In 2005, a fast algorithm for exact or approximate geodesic computation was presented in [9]. This new approach shows that MMP-based approximation algorithm yields more accurate solutions than the fast-marching method when applied to meshes. They proved that the new algorithm runs in $O(n \log n)$ time even for small error thresholds. Kobbelt etc. also presented an algorithm for efficient and accurate computation of geodesics [10]. This nice algorithm can handle arbitrary, possibly open, polygons on the mesh to define the zero set of the distance field.

An algorithm for computing geodesics on smooth surfaces can be found in [5], which introduces a novel approach for rapidly computing a very large number of geodesics on a smooth surface. Their idea is based on the phase flow method [5].

2.2 Smooth surface representation using subdivision surfaces

Given a control mesh, a subdivision surface is generated by iteratively refining (subdividing) the control mesh to form new and finer control meshes. The refined control meshes converge to a limit surface called a *subdivision surface*. So a subdivision surface is determined by the given control mesh and the mesh refining (subdivision) process. The control mesh of a subdivision surface can contain vertices whose *valences* (numbers of adjacent edges) are different from four. Those vertices are called *extra-ordinary vertices*. Popular subdivision surfaces include Catmull-Clark subdivision surfaces (CCSSs) [1], Doo-Sabin subdivision surfaces and Loop subdivision surfaces.

Subdivision surfaces can model/represent complex shape of arbitrary topology because there is no limit on the shape and topology of the control mesh of a subdivision surface. Subdivision surfaces are intrinsically discrete. Recently it was proved that subdivision surfaces can also be parametrized [2]. Therefore, subdivision surfaces cover both *parametric forms* and *discrete forms*. Parametric forms are good for design and representation, discrete forms are good for machining and tessellation (including FE mesh generation). Hence, we have a representation scheme that is good for all graphics and CAD/CAM applications. Subdivision surfaces by far are the most general surface representation scheme. They include non-uniform B-spline and NURBS surfaces as special cases. In this paper we only consider objects represented by CCSSs. But our approach works for other subdivision schemes as well.

3 Preliminaries

In this section we discuss some properties of geodesics of a given mesh M or a given smooth surface S. The proof of these properties can be found in [4, 5]. These properties are the theoretical foundation of our approach.

- Property 1: There exists a geodesic path from a given point to any other point on the given mesh M or the given smooth surface S.
- Property 2: When M or S is unfolded, a geodesic path of M or S becomes a straight line segment.
- Property 3: If S is continuous everywhere, so is any geodesic path of S.

– Property 4: When a geodesic path of S passes through a point of S, the geodesic path has the same tangent vector on the left side of the point as the right side of the point.
– Property 5: Geodesic paths starting at the same point P, but going along different directions may intersect. If they do intersect, say at point Q, then the lengths of the geodesic paths between P and Q are the same, even though they go different directions starting from point P.

4 Basic Idea

Given a mesh M, we can find its limit surface S using Catmull-Clark subdivision. The limit surface S is a smooth surface which has C^2 continuity almost everywhere except at a few extra-ordinary points. The task of this paper is to find an approximate geodesic path from a given point P on S along any initial direction vector W.

The basic idea of our approach is to unfold the surface S so that the geodesic path can be achieved directly using a straight line from the given point P along the given vector W in the flattened surface. We can compute S from a given sparse 3D mesh M using subdivision surface parametrization techniques. The question is how to flatten the smooth surface S. This is done with the following three steps.

– Discretize S to get a dense mesh $\bar{S}$;
– Unfold $\bar{S}$ along a given initial direction;
– Compute geodesic paths on $\bar{S}$.

As they will be shown below, all the three steps can be directly computed, without setting up any linear system or applying any costly matrix computation, but only uses linear combination of vertices locally. Hence this is a linear local method which is very easy to implement and can deal with meshes of large number of vertices effectively.

5 Discretization of a smooth surface S

In this section we show how to discretize a smooth surface in order to get a dense mesh approximation. Here we assume the smooth surface is a Catmull-Clark subdivision surface (CCSS). Therefore we can evaluate an ordinary or extra-ordinary CCSS patch and its tangent vectors at any given point of the smooth surface. These techniques are needed in the construction of the approximating polyhedron for the surface unfolding process. Several approaches [2, 3] have been presented for exact evaluation of an extra-ordinary patch at any parameter point (u, v). We use the parametrization technique presented in [3] here. This method is more efficient for both surface and tangent evaluation because it employs less eigen basis functions in its representation.

The parametrization technique presented in [3] works for general CCSS's, i.e., for a given vertex point $\mathbf{V}$, a new *vertex point* $\mathbf{V}'$ is computed as:

$$\mathbf{V}' = \alpha_n \mathbf{V} + \beta_n \sum_{i=1}^{n} \mathbf{E}_i + \gamma_n \sum_{i=1}^{n} \mathbf{F}_i$$

where α_n, β_n and γ_n are positive numbers and $\alpha_n + \beta_n + \gamma_n = 1$, and it is based on an $\Omega-partition$ of the parameter space [2,3]. The value of an extra-ordinary patch is evaluated as follows:

$$\mathbf{S}(u,v) = W^T \mathrm{K}^m \sum_{j=0}^{n+5} \ \lambda_j^{m-1} \ \mathrm{M}_{b,j} \ G \tag{1}$$

where n is the valance of the extra-ordinary patch [3], W is a vector containing the 16 B-spline power basis functions:

$$W^T(u,v) = [1, u, v, u^2, uv, v^2, u^3, u^2v, uv^2, v^3,$$

$$u^3v, u^2v^2, uv^3, u^3v^2, u^2v^3, u^3v^3] \ ,$$

K is a diagonal matrix:

$$\mathrm{K} = Diag(1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 16, 16, 16, 32, 32, 64),$$

and m and b are defined as follows:

$$m(u,v) = min\{\lceil log_{\frac{1}{2}} u \rceil, \lceil log_{\frac{1}{2}} v \rceil\} \ ,$$

$$b(u,v) = \begin{cases} 1, & if \ \ 2^m u \geq 1 \ \ and \ \ 2^m v < 1 \\ 2, & if \ \ 2^m u \geq 1 \ \ and \ \ 2^m v \geq 1 \\ 3, & if \ \ 2^m u < 1 \ \ and \ \ 2^m v \geq 1 \ . \end{cases}$$

λ_j, $0 \leq j \leq n+5$, are eigenvalues of the Catmull-Clark subdivision metrix and $\mathrm{M}_{b,j}$, $1 \leq b \leq 3$, $0 \leq j \leq n+5$, are matrices of dimension $16 \times (2n+8)$. λ_j and $\mathrm{M}_{b,j}$ are independent of (u,v) and their exact expressions are given in [3]. G is the vector of control points (See [3] for their labeling).

One can compute the derivatives of $\mathbf{S}(u,v)$ to any degree simply by differentiating $W(u,v)$ in Eq. (1) accordingly. For example,

$$\frac{\partial}{\partial u}\mathbf{S}(u,v) = (\frac{\partial W}{\partial u})^T \ \mathrm{K}^m \sum_{j=0}^{n+5} \ \lambda_j^{m-1} \ \mathrm{M}_{b,j} \ G. \tag{2}$$

The value and tangents at an extra-ordinary vertex are simply the limit points of the corresponding equations in (2) when $(u,v) \rightarrow (0,0)$:

$$\begin{array}{ll} \mathbf{S}(0,0) & = [1, 0, \cdots, 0] \cdot M_{2,n+1} \cdot G \\ \mathbf{D}_u(0,0) & = [0, 1, 0, 0, \cdots, 0] \cdot \mathrm{M}_{2,2} \cdot G \\ \mathbf{D}_v(0,0) & = [0, 0, 1, 0, \cdots, 0] \cdot \mathrm{M}_{2,2} \cdot G \end{array} \tag{3}$$

[3] Eq. (1) works for regular patches as well, i.e., when $n = 4$.

where $\mathbf{D}_u(0,0)$ and $\mathbf{D}_v(0,0)$ are the direction vectors of $\frac{\partial \mathbf{S}(0,0)}{\partial u}$ and $\frac{\partial \mathbf{S}(0,0)}{\partial v}$, respectively. As a result, the normal vector at $(0,0)$ is

$$\mathbf{N}(0,0) = \mathbf{D}_u(0,0) \times \mathbf{D}_v(0,0).$$

With the availability of direct evaluation of Catmull-Clark subdivision surfaces, one can discretize a CCSS patch by patch to any accuracy with a dense mesh approximation. Note that after the discretization, the resulting mesh $\bar{S}$ consists of only quadrilaterals, which is convenient for us to unfold the mesh.

6 Unfolding $\bar{S}$ in a given direction

Once we have an approximate representation of the smooth surface, we can unfold the mesh to get a flattened surface so that the geodesic path can be obtained easily. As we know, meshes with arbitrary topology may not be able to be flattened without distortion. So we cannot find the geodesic paths if the surface is distorted. Fortunately, we do not need to unfold the whole surface. Only the patches of a slice of $\bar{S}$ are needed to be unfolded in the process of computing the geodesic paths. This is done as follows.

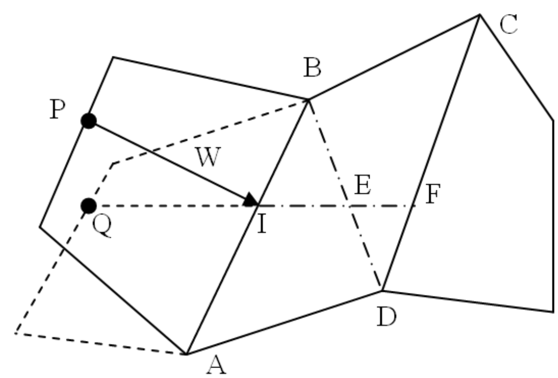


Fig. 1. Unfold the patches along the given direction W.

Given a starting point P and a starting direction W (see Figure 1), we need to find the geodesic path from P along the direction W. First we rotate the surface patch that P and W are on (i.e., ΔBPA in Figure 1) so that it is on the same plane as its neighboring patch, which is patch ΔABD in Figure 1. After the rotation around line AB, point P goes to point Q. Because now ΔQBA and ΔABD are on the same plane, we can find the intersection points I and E. Points I and E are saved to a list because they are part of the geodesic path. Again starting from point E and along the direction of vector $(E - I)$, we repeat the above process to find intersection point F, which is again part of the geodesic path and should be saved into the list. Note that here ΔABD and ΔBCD may not lie on the same plane. By repeating the same process, a slice of surface patches is unfolded and a straight line can be determined from the unfolded surface patches of the slice.

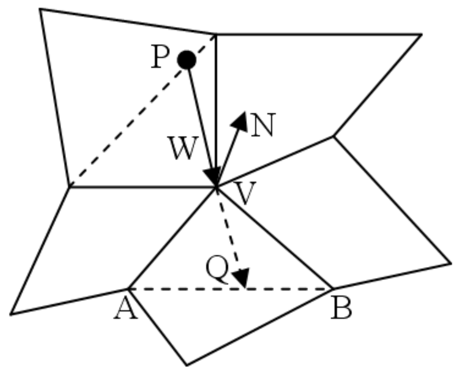


Fig. 2. Unfold the patches when the geodesic passing vertex V.

The above process would not work when a geodesic path passes through a vertex of the underlying approximate mesh $\bar{S}$ of S. See Figure 2. This is because we need an axis when we rotate a patch to flatten the patch. When a geodesic path passes through a vertex, we cannot use the above method to find the next segment of the geodesic path because we do not know which patch we should use to proceed next. We need a new approach and it is done as follows.

From the properties of geodesic paths, we know that a geodesic path is continuous if the 3D surface is continuous everywhere. Hence if a smooth geodesic path passes through vertex V (see Figure 2), then the geodesic path has the same tangent vector to the left and to the right of the vertex V. As a result, if the discretization of S is dense enough, then the entering segment to V and the departing segment from V of the geodesic path are on the same plane as the normal vector of the smooth surface at the point V. In other words, if N is the normal vector of S at V, then ΔPNV and ΔNQV are on the same plane assuming VQ is the departing segment of the geodesic path from V. With this observation, we can determine the next segment of the geodesic path and proceed and repeat the process using one of the above two approaches, depending on if the geodesic path passes through a vertex of $\bar{S}$.

7 Computing geodesic paths on $\bar{S}$

Once we know how to unfold patches of $\bar{S}$, the process of computing the geodesic paths is straightforward. Again we will use Figure 1 and Figure 2 to illustrate how to do the computation efficiently.

To get Q from rotating P about the line segment AB in Figure 1, one can use the following formula

$$Q = \begin{bmatrix} x^2(1-c)+c, & xy(1-c)-zs, & xz(1-c)+ys, & 0 \\ yx(1-c)+zs, & y^2(1-c)+c, & yz(1-c)-xs, & 0 \\ xz(1-c)-ys, & yz(1-c)+xs, & z^2(1-c)+c, & 0 \\ 0, & 0, & 0, & 1 \end{bmatrix} * P,$$

where $c = \frac{\mathbf{n_1} \cdot \mathbf{n_2}}{||\mathbf{n_1}||.||\mathbf{n_2}||}$, $s = \frac{\mathbf{n_1} \times \mathbf{n_2}}{||\mathbf{n_1}||.||\mathbf{n_2}||}$, and $||(xyz)|| = 1$. x, y and z are the normalized coordinate components of the vector $B - A$. $\mathbf{n_1}$ and $\mathbf{n_2}$ are the normal vectors of ΔBPA and ΔADB, respectively, i.e., $\mathbf{n_1} = PA \times PB$ and $\mathbf{n_2} = AD \times AB$.

When the geodesic path passes through a vertex V of $\bar{S}$, to find Q from P (See Figure 2), we first need to calculate the normal vector N of S at V, which can be precisely obtained using the formula given in the previous section. Once N is known, according to the properties of geodesics, we have that the four points P, $N + V$, Q and V are coplanar. Hence to find Q, we just need to find the intersection point of the line segment AB and the 2D plane determined by the three points P, $N + V$ and V. The line segment AB is the diagonal line of a neighboring patch. Because we do not know which patch the line segment AB belongs to, we need to loop through all the neighboring patches of vertex V. Because there is one and only one such neighboring patch (other than the one that P belongs to) whose diagonal line segment intersects the plane defined by the three points P, $N + V$ and V, the loop will stop once the intersection point Q is found. The intersection point of the line segment AB and the plane $(P, N+V, V)$ can be computed by first computing the following parameter value:

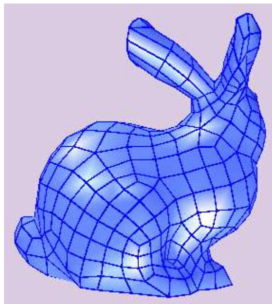
$$s = -\frac{\mathbf{n} \cdot \mathbf{w}}{\mathbf{n} \cdot \mathbf{u}},$$

where $\mathbf{n} = (P - V) \times N$, $\mathbf{w} = A - V$ and $\mathbf{u} = B - A$. If $0 <= s <= 1$, the line segment AB and the plane have an intersection point $Q = A + s(B - A)$. Otherwise, there is no intersection point between them.

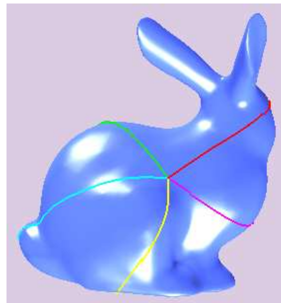
8 Test Results

The proposed approach has been implemented in *C++* using *OpenGL* as the supporting graphics system on the Windows platform. Quite a few examples have been tested with the method described here. All the examples have extra-ordinary vertices. Some of the tested results are shown in Figure 3. For all the test cases shown in this paper, the original mesh, and a bundle of geodesic paths on the smooth 3D models are given. All geodesic paths are shown within a certain length only in the examples.

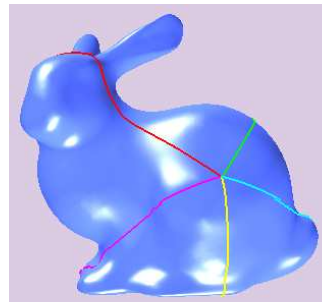
The new geodesic computation method can handle meshes with large number of vertices in a matter of almost real time on an ordinary PC (3.2GHz CPU, 2GB of RAM). For example, each of the smooth models shown in Figure 3 has more than one million vertices after discretization. It takes less than a second to find all



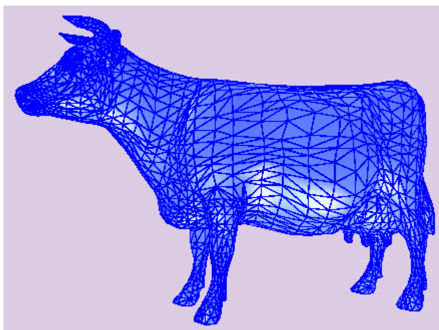
(a) Original Mesh



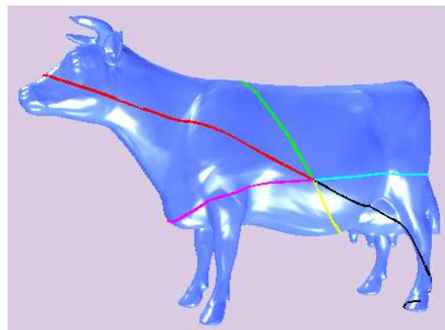
(b) Geodesic Paths



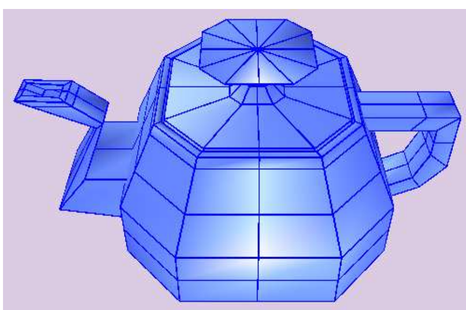
(c) Geodesic Paths



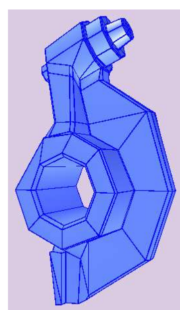
(d) Original Mesh



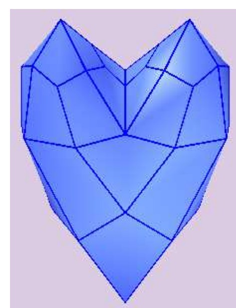
(e) Geodesic Paths



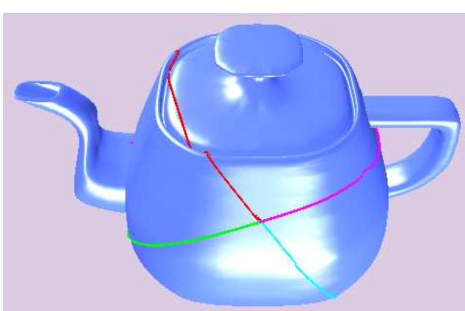
(f) Original Mesh



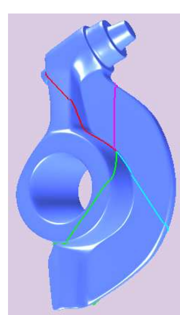
(g) Original Mesh



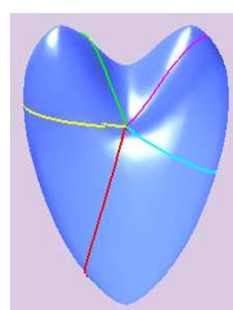
(h) Original Mesh



(i) Geodesic Paths



(j) Geodesic Paths



(k) Geodesic Paths

Fig. 3. Examples of geodesic paths starting from a given point in different directions.

the geodesic paths for each given model. Hence the new geodesic computation method is suitable for interactive applications, such as shape design, terrain navigation and so on.

9 Summary and Future Work

A new approach for computing large number of approximate geodesic paths from a given point to all directions on a 3D model (mesh or surface) of arbitrary topology is presented. The basic idea is to unfold the 3D model into a flat surface so that the geodesic from a given point in an initial given direction can be obtained simply by drawing a straight line from the given point along the given direction on the unfolded surface. The new method is fast and does not require any costly matrix computation, or linear system solving in the process of geodesic computation, hence it is very easy to implement. The new approach also is very fast with a linear time complexity and hence can be used for meshes with large number of vertices. Test examples show the effectiveness of our approach.

One of our future research objectives is to apply this new approach to find the geodesic path between two points of a smooth surface. Another subject of our future research is to compare the performance of the new approach with other geodesic computation methods in the literature to study its effectiveness and possible improvements.

ACKNOWLEDGEMENT
This work is supported by National Science Foundation of China (61020106001, 61170324), National Science Council of ROC (NSC-100-2811-E-007-021), and a joint grant of National Tsinghua University and Chang-Gung Memorial Hospital (101N2756E1).

References

1. Catmull E, Clark J, Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer-Aided Design*, 1978, 10(6):350-355.
2. Stam J, Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values, *Proceedings of SIGGRAPH* 1998:395-404.
3. Shuhua Lai, F. Cheng, Parametrization of General CCSSes and its Applications, *Computer Aided Design & Applications*, 3, 1-4, 2006.
4. J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou, The discrete geodesic problem. *SIAM J. Comput.*, 16(4):647-668, 1987.
5. Lexing Ying and Emmanuel J. Candésk, Fast geodesics computation with the phase flow method, *Journal of Computational Physics*, Volume 220, Issue 1, 20 December 2006, Pages 6-18.
6. Chen, J., and Han, Y. Shortest paths on a polyhedron; part I: computing shortest paths. *Int. J. Comp. Geom. & Appl.*, 6 (2), 127-144. 1996.
7. Kapoor, S., Efficient computation of geodesic shortest paths. In *Proc. 32nd Annual ACM Symp. Theory Comput.*, 770-779, 1999.

8. Lanthier, M., Maheshwari, A., and Sack, J.-R., Approximating weighted shortest paths on polyhedral surfaces. In *Proc. 13th Annu. ACM Symp. Comput. Geom.*, 274-283, 1997.
9. Surazhsky, V. and Surazhsky, T. Kirsanov, D. and etc., Fast exact and approximate geodesics on meshes,
10. David Bommes and Leif Kobbelt, Accurate Computation of Geodesic Distance Fields for Polygonal Curves on Triangle Meshes, In *Vision Modeling and Visualization*, 2007.