HAL
open science

A Fast Sweeping Method for Computing Geodesics on Triangular Manifolds

Song-Gang Xu, Yun-Xiang Zhang, Jun-Hai Yong

To cite this version:

Song-Gang Xu, Yun-Xiang Zhang, Jun-Hai Yong. A Fast Sweeping Method for Computing Geodesics on Triangular Manifolds. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010. inria-00517321

HAL Id: inria-00517321

https://inria.hal.science/inria-00517321v1

Submitted on 14 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Fast Sweeping Method for Computing Geodesics on Triangular Manifolds

Song-Gang Xu, Yun-Xiang Zhang, and Jun-Hai Yong

Abstract—A wide range of applications in computer intelligence and computer graphics require computing geodesics accurately and efficiently. The fast marching method (FMM) is widely used to solve this problem, of which the complexity is $O(N \log N)$, where N is the total number of nodes on the manifold. A fast sweeping method (FSM) is proposed and applied on arbitrary triangular manifolds of which the complexity is reduced to $O(N)$. By traversing the undigraph, four orderings are built to produce two groups of interfering waves, which cover all directions of characteristics. The correctness of this method is proved by analyzing the coverage of characteristics. The convergence and error estimation are also presented.

Index Terms—Geodesics, fast sweeping methods, fast marching methods, Eikonal equation, triangular manifold.

1 Introduction

The geodesics computation is commonly seen in many applications of computer intelligence and computer graphics. It is widely used in many fields, such as robotic optimal path planning [1], [2], [3], modeling representation [4], surface parameterization [5], and seismic travel time estimation [6], [7], etc. A general procedure on this problem often includes two parts. The first one is to develop partial differential equations (PDEs) that define the geodesics properly. The second one is to track the geodesics from the solution of the PDE. The Eikonal equation is a competitive candidate.

The Eikonal equation is extensively discussed as a particular class of first order nonlinear PDEs. Its general form is $|\nabla T(\mathbf{x})| = f(\mathbf{x})$, where T is the distance function tracking the motion of a monotonically advancing front and $f(\mathbf{x})$ is the advancing velocity of fronts along normals. $f(\mathbf{x})$ is usually a constant in computing the geodesics. Since it is difficult to find the exact solution, numerical methods [8], [9], [10], [11], [12], [13], [14], [15] are employed to solve it. Their refined implementations [16], [15], [17], [18], [19] are also presented. All of them require the decomposition of the computing domain with discrete models [20], [21], [22]. Generally, these methods are classified into two categories, the fast marching method (FMM) and the fast sweeping method (FSM).

The FMM [1], [8], [9], [16], [11], [15], [17], [18], [19], [23] treats the Eikonal equation as a time variant problem. With the discretization of Godunov Hamiltonian, the FMM constructs an upwind scheme to evaluate the nodes on computing domain. By sorting the values of nodes at upwind front, this scheme maintains a sequence that directs the order of evaluating nodes. FMM can be applied to the Cartesian domain [8], [9], [16], [17], [19], the anisotropic domain [14], and the triangular domain of manifolds [11]. It has the complexity of $O(N \log N)$, where N is the number of nodes on discretization. Recently, some refined schemes improve the primitive FMM on the accuracy or optimal complexity. The accuracy is improved by introducing a high-order finite difference [16], [17] or a high-accuracy local solver [19]. Meanwhile, the optimal complexity is reduced from $O(N \log N)$ to $O(N)$ by a faster sort procedure [18] or a heuristic group marching scheme [15].

The FSM [24], [25], [12], [26], [27], [13], [28], [29], which was first proposed by Danielsson [30], treats the Eikonal equation as a stationary boundary problem and solves it directly by sweeping the computing domain. Generally speaking, the FSM consists of two parts. One is a group of local solvers which is used to evaluate nodes on discretization. The other is a devised process generating alternating sequences that direct nodes' evaluation. The FSM is adaptive to the discretization of both Godunov Hamiltonian [24], [12], [13], [28] and Lax-Friedrichs Hamiltonian [25]. Meanwhile, it could be applied on both Cartesian domain [24], [12] and triangular domain [13], [28]. The FSM has a computing complexity of $O(N * F(f(\mathbf{x})))$ [24], [12], [31], where F depends on the data complexity $f(\mathbf{x})$. Because $f(\mathbf{x})$ is a constant in computing the geodesics, the complexity of FSM is $O(N)$. In addition, the FSM has also been improved by a high-order discretization [22] or a high-order local scheme [27] to produce high accurate numerical solution.

The FSM is faster and more adaptive to different kinds of discretizations than the FMM in computing geodesics. However, its computing domain is constrained to be simple geometries, i.e., the realistic grids and triangular plane. In other words, it is difficult to apply the FSM on anisotropic or arbitrary manifolds. It is even hard to keep the low-complexity $O(N)$.

• *The authors are with the Institute of Computer Graphics and Computer Aided Design Laboratory (CG&CAD), School of Software, Tsinghua University, Room 824, The Main Building, Beijing 100084, PR China, and the Key Laboratory for Information System Security, Ministry of Education of China, Beijing 100084, PR China.*
E-mail: {xsg06, zhangyunxiang06}@mails.tsinghua.edu.cn, yongjh@tsinghua.edu.cn.

Manuscript received 21 Apr. 2008; revised 17 July 2008; accepted 28 Oct. 2008; published online 7 Nov. 2008.
Recommended for acceptance by G. Sapiro.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-04-0234.
Digital Object Identifier no. 10.1109/TPAMI.2008.272.

In this paper, a method is proposed to apply the FSM on a triangular manifold. Since it is represented by a triangular mesh, the manifold can be viewed as an undigraph naturally. By traversing this undigraph, the FSM will produce predetermined sweeping sequences, which will direct the order of nodes' evaluation in Gauss-Seidel iterations. The method is optimal in the sense that a finite number of iterations is needed. Its computing complexity is as well $O(N)$. In addition, both theoretical analysis and experiments are presented to verify its correctness. The convergence and error estimation are also included.

The rest of this paper is organized as follows: In Section 2, we review some related preliminaries on the Eikonal equation and meanings of characteristics. The local solver for each grid on the triangular manifold is addressed in Section 3. In Section 4, the sweeping sequences obtained by traversing the graph are explained and the FSM algorithm is presented. Section 5 provides proofs on the convergence and error estimation for the distance function. Meanwhile, the correctness of the proposed method to solve the Eikonal equation is presented. In Section 6.1, several numerical experiments are designed to test the method. Finally, Section 6.3 concludes the presented work and discusses future work.

2 Preliminary

2.1 Eikonal Equation

Let us consider a boundary Γ, which could be a curve in 2D or a surface in 3D. Γ moves in its normal direction at a given speed F. $T(\mathbf{x})$ represents the arrival time that Γ reaches each point $\mathbf{x}$. In one dimension, using the fact that $distance = rate \times time$, this problem can be expressed as

$$1 = F\frac{dx}{dT}.$$

In multiple dimensions, ∇T is orthogonal to the level sets of T. Similar to the one-dimensional case, its magnitude is inversely proportional to the speed. Hence,

$$|\nabla T|F = 1, \quad T(\Gamma) = 0. \tag{1}$$

The general form of the stationary Eikonal equation [16], [12], [13] is

$$|\nabla T(\mathbf{x})| = f(\mathbf{x}) \quad \mathbf{x} \in R^n, \tag{2}$$

with boundary condition $T(\mathbf{x}) = \phi(\mathbf{x})$, $\mathbf{x} \in \Gamma \subset R^n$. $T(\mathbf{x})$ is a function defined on R^n and $\mathbf{x}$ is the independent variable in R^n. The boundary condition is enforced on Γ, i.e., a set of lines, or some discrete points, or even a single source point, where ϕ is zero.

2.2 Characteristics

Characteristics are the trajectories of a high-frequency wave propagation. Considering (2), we denote $H(\mathbf{p}, \mathbf{x}) = |\mathbf{p}| - f(\mathbf{x})$, where $\mathbf{p} = \nabla T$. The characteristic equation for the Eikonal equation is

$$\begin{cases} \dot{\mathbf{x}} = \nabla_{\mathbf{p}} H = \frac{\nabla T}{f(\mathbf{x})}, \\ \dot{\mathbf{p}} = -\nabla_{\mathbf{x}} H = \nabla f(\mathbf{x}), \\ \dot{T} = \nabla T \cdot \dot{\mathbf{x}} = f(\mathbf{x}), \end{cases} \tag{3}$$

where the dot "$\dot{}$" above the letter denotes the derivative along characteristics parametrized by the arc length s. $\varphi(\mathbf{x})$ is a smooth solution of T in (3) and then $\nabla\varphi$ is the direction of the characteristics. Apparently these characteristics identify the directions in which $\varphi(\mathbf{x})$ increases or decreases.

The characteristics are an analytical representation of the causality in the Eikonal equation. There are infinite characteristics, which intersect or not according to $f(\mathbf{x})$. To our knowledge, most of the numerical methods are required to *cover* all of the characteristics. *Cover* means that, when evaluating nodes on the domain, the front forms an acute angle to a specified characteristic.

Furthermore, all directions of characteristics can be divided into a finite number of groups. One characteristic can be decomposed into a finite number of segments that belong to one of those groups. This implies that if it covers the directions of the characteristics in each group simultaneously, this numerical method will produce correct numerical solutions of (2).

3 Local Solvers on Triangular Manifolds

3.1 Representation of Triangular Manifolds

A manifold is tessellated if it is represented by a triangular mesh. When a discretization is built on the triangulation $\mathcal{T}_h$, a group of schemes for evaluating distance function on each node is derived. These schemes are often referred to as local solvers. However, a triangular manifold Ω_h from which local solvers can be obtained should satisfy the following requirements:

- The manifold Ω_h consists of nonoverlapping, nonempty, and closed triangles. The maximum value of their circumcircle diameters is $h_{\mathcal{T}}$.
- Any vertex of a triangle is neither an interior point nor a point on edges except for two end points.
- The intersection of two triangles $\mathcal{T}_1, \mathcal{T}_2 \in \mathcal{T}_h$ is empty, a common vertex, or a common edge.

Local solvers focus on evaluating distance function $T(\mathbf{x})$ in a triangle. Here, nodes on the manifold are referred as vertices. If the right angle is considered to be obtuse one, the inner angle of a vertex is either acute or obtuse. The local solver of a vertex is also either an obtuse one or an acute one with respect to its inner angle. In Fig. 1a, two angles $\angle ACB$ and $\angle ECF$ are presented; suppose that $CE \perp CB$ and $CF \perp CA$. If the direction of causality is marked by a black arrow, local solvers for both acute and obtuse angles should be considered in three situations: The causality falls in both angles (See Figs. 1a and 1b), the causality falls out of both angles (See Figs. 1c and 1d), and the causality falls in one of them (See Figs. 1e and 1f).

3.2 Acute Solver

An acute solver that covers the causality entirely was presented in [13]. We adopt it and explain it in brief. In the triangle of Fig. 2, the vertex C is to be evaluated. Its value is T_C. $\angle ACB$ is acute. Vertices A and B have already been evaluated. Their values are T_A and T_B, respectively. Without loss of generality, we assume that $T_A < T_B$.

When the causality falls in both $\angle ACB$ and $\angle ECF$, shown in Fig. 2a, the evaluating ordering should be A, B, and C.

Authorized licensed use limited to: Tsinghua University Library. Downloaded on April 06,2010 at 08:51:35 EDT from IEEE Xplore. Restrictions apply.

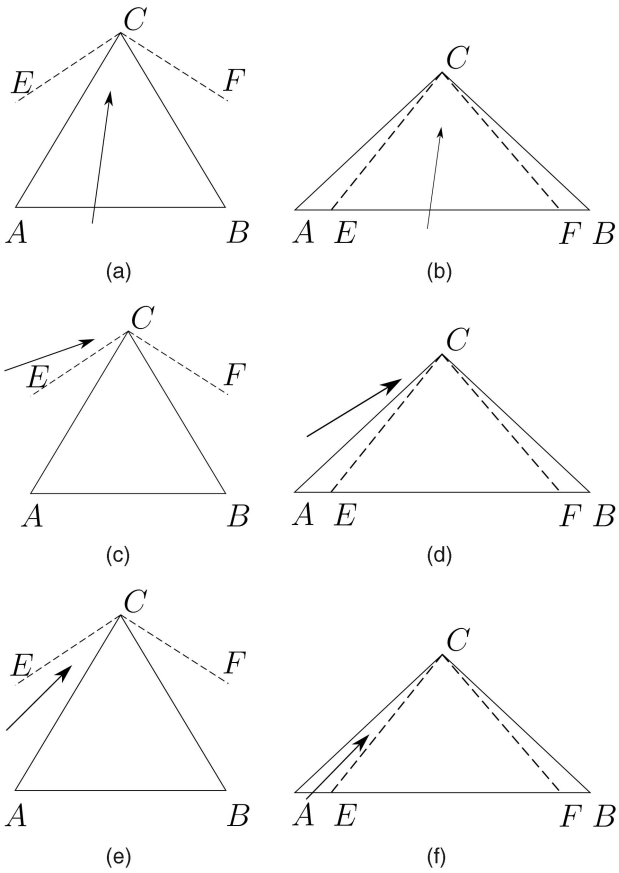

Fig. 1. (a), (c), (e) show the cases of the acute triangle; (b), (d), (f) show the cases of the obtuse triangle.

This fact can be substantiated by Huygens principle if the following requirements are satisfied:

- $(T_B - T_A)/f_C \leq \overline{AB}$. $\overline{AB}$ is the length of edge AB. This ensures that the front of the distance function propagates from A to B at a given speed f_C.
- $\theta \leq \beta$. This ensures that the evaluating ordering is A, B, and C.
- $\theta + \alpha < \frac{\pi}{2}$. This ensures that the causality is satisfied.

Then, we have

$$T_C = \min\{T_C, \overline{CQ} \cdot f_C + T_A\}, \tag{4}$$

which is the case in Fig. 1a.

Otherwise, the causality does not fall inside $\triangle ABC$, and we have $T_C = \min\{T_C, T_A + bf_C, T_B + af_C\}$, where $a = \overline{BC}$ and $b = \overline{AC}$. Figs. 2a and 2b are the cases in Figs. 1c and 1e, respectively.

The acute solver is summarized as

if $|T_B - T_A| \leq cf_C$ **then**
  $\theta = \arcsin(\frac{T_B - T_A}{cf_C})$;
  if $\max(0, \beta - \frac{\pi}{2}) \leq \theta \leq \frac{\pi}{2} - \alpha$ or $\beta - \frac{\pi}{2} \leq \theta \leq \min(0, \frac{\pi}{2} - \alpha)$ **then**
    $\overline{CP} = a\sin(\beta - \theta)$; $\overline{CQ} = b\sin(\alpha + \theta)$;
    $T_C = \min\{T_C, \overline{CQ} \cdot f_C + T_A\}$
  else
    $T_C = \min\{T_C, T_A + bf_C, T_B + af_C\}$
  end if

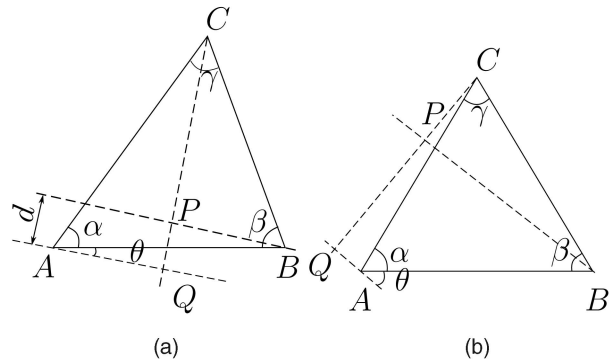

Fig. 2. The causality falls inside in (a) and outside in (b) of the triangle.

else
  $T_C = \min\{T_C, T_A + bf_C, T_B + af_C\}$
end if

3.3 Obtuse Solver

When $\angle ACB$ is an obtuse angle, T_C in Fig. 1b can be evaluated using the acute solver. T_C in Fig. 1d can be evaluated using $T_C = \min\{T_C, T_A + bf_C, T_B + af_C\}$ because the causality falls outside. Fig. 1f is a special one. Several methods have been provided to evaluate T_C under this situation. However, the technique of triangles unfolding in [16], [11] might increase the computing complexity. The technique of angle splitting in [13] cannot be used on triangular manifolds. We propose an obtuse solver that is discussed in the following two cases, respectively:

- The assumption of $T_A < T_B$ implies that T_A has been evaluated. If T_B has also been evaluated, the method shown in Fig. 3a is adopted. T_C is computed by $T_C = T_A + f_c \cdot \overline{CQ}$. The computing method is similar to the cases in Fig. 2a.
- Otherwise, T_B is infinite so that the method shown in Fig. 3b is adopted. We find two neighboring triangles $\triangle AFC$ and $\triangle ABE$ of $\triangle ABC$. $\triangle ABE$ is flipped so as to be on the same plane of $\triangle ABC$. Meanwhile, according to the algorithm in Section 4, T_E, T_F, or both have been evaluated. If T_F has been evaluated, we evaluate the vertex C in $\triangle AFC$ with the known T_A and T_F. If T_E has been evaluated, we evaluate C in $\triangle AEC$ with the known T_A and T_E.

Proof on the cover of the causality is presented in Section 5. Summarily, the obtuse solver is concluded as follows:

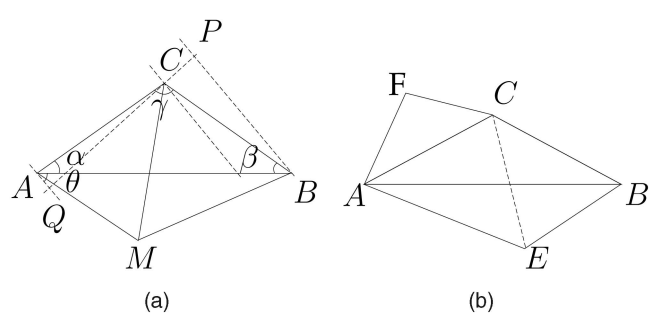

Fig. 3. (a) is the obtuse local solver when B has been evaluated; (b) is the one when B is infinity.

Authorized licensed use limited to: Tsinghua University Library. Downloaded on April 06,2010 at 08:51:35 EDT from IEEE Xplore. Restrictions apply.

if T_B is infinity **then**
  Find B' which is the vertex E or F, of which value is not infinity. This form a new $\triangle AB'C$
end if
$a = \overline{B'C}; b = \overline{AC};$
$\theta = \arcsin(\frac{T_B - T_A}{cf_C})$
if $|T_{B'} - T_A| \leq cf_C$ **then**
  if $\max(0, \beta - \frac{\pi}{2}) \leq \theta \leq \frac{\pi}{2} - \alpha$ or $\beta - \frac{\pi}{2} \leq \theta \leq \min(0, \frac{\pi}{2} - \alpha)$ **then**
    $\overline{CP} = a\sin(\beta - \theta); \overline{CQ} = b\sin(\alpha + \theta);$
    $T_C = \min\{T_C, \overline{CQ} \cdot f_C + T_A\}$
  else if $\theta > \frac{\pi}{2} - \alpha$ or $\theta < \beta - \frac{\pi}{2}$ **then**
    $T_C = \min\{T_C, T_A + bf_C, T_{B'} + af_C\}$
  else
    $T_C = \min\{T_C, T_A + \overline{CQ} \cdot f_C\}$
  end if
else
  $T_C = \min\{T_C, T_A + bf_C, T_{B'} + af_C\}$
end if

The situation $\angle ACB = \pi/2$ can be solved by either the acute solver or the obtuse solver. The latter is adopted in experiments.

4 FSM on Triangular Manifolds

This section is about the algorithm that applies the FSM to an arbitrary manifold. The basic problem is the construction of sweeping sequences. In [12], these sequences are derived from orderings of rectangular grids. However, when it comes to domain of triangular plane, the natural orderings do not exist. Qian et al. [13] introduced l^p distances taking multiple reference points in sweeping to generate sequences. Though the complexity of Gauss-Seidel iterations is $O(N)$, the extra sorting procedure becomes the bottleneck. Moreover, no l^p distances can be measured on an arbitrary manifold, which means that neither above is available.

We develop a method that constructs sweeping sequences from topological orderings of a triangular manifold. Because it is represented by a triangular mesh, the manifold can be viewed as an undigraph naturally. Nodes of the mesh are viewed as vertices of the undigraph. Edges of the mesh represent the connections between vertices. With this preparation, the method produces a tree by a breadth-first traversing on the undigraph from one vertex. Usually this vertex is the initial position of geodesics. Then, it produces two topological orderings by two different breadth-first traversings on this tree. These two orderings and their reverses are the four sweeping sequences. We explain the whole process via an example.

Fig. 4a shows a triangular manifold of which the nodes are numbered. It is viewed as an undigraph. The initial position is Point 1. First, the method produces a tree by a breadth-first traversing on the undigraph. There is more than one breadth-first tree that can be derived from this undigraph. Any one is appropriate. We choose the one shown in Fig. 4b as an example. It is considered as a directed one in which positions of siblings sharing a common parent cannot be swapped. Second, we have two different breadth-first traversings on this tree. One traverses siblings of a common parent from left to right and the other

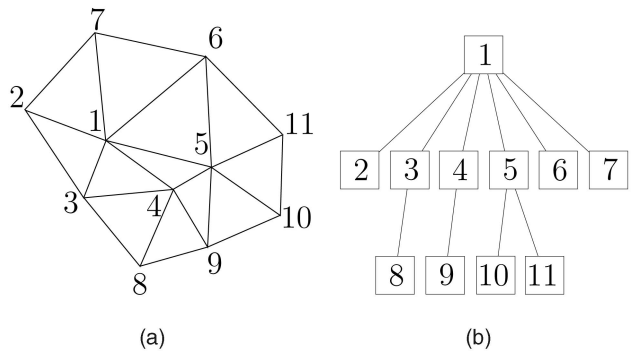

Fig. 4. (a) is a triangular manifold; (b) is a certain breadth-first traversing tree.

traverses from right to left. Apparently, the first traversing produces two orderings, $S_1^+ = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ and its reverse $S_1^- = \{11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$. And the second one produces two orderings, $S_2^+ = \{1, 7, 6, 5, 4, 3, 2, 11, 10, 9, 8\}$ and $S_2^- = \{8, 9, 10, 11, 2, 3, 4, 5, 6, 7, 1\}$. These four orderings are the sweeping sequences that are used in the Gauss-Seidel iterations.

This algorithm generates arrival time at each vertex of the undigraph. Without loss of generality, the boundary condition of the initial position is assumed to be one vertex of the undigraph. Thus, the algorithm is composed of two parts, as follows.

Part I. Constructing orderings.

1. Construct the undigraph.
2. Set distance function value T of the starting vertex V_0 as zero and set that of other vertices as infinite.
3. Have a breadth-first traversing on the undigraph starting from V_0, which produces a traversing ordering S_1^+ and a corresponding directed breadth-first tree B.
4. Have another breadth-first traversing on B that accesses the children of each node from right to left, which produces ordering S_2^+.
5. Reverse S_1^+ and S_2^+ to produce two orderings S_1^- and S_2^-.

Part II. Gauss-Seidel iterations.
for $j = +, -$ **do**

1. For every vertex $C \in S_1^j$ and every triangle associated with $C, f_C = f(C)$, and apply local solvers.
2. For every vertex $C \in S_2^j$ and every triangle associated with $C, f_C = f(C)$, and apply local solvers.

end for

The computing model is similar to the wavefronts one described in [13], though these waves depicted in Fig. 5a might not be concentric circles. If one edge of these circles overlaps some parts of a characteristic, the vertices there cannot be evaluated correctly. That is why two groups of interfering sweeping sequences are constructed. Fig. 5b depicts these two breadth-first traversing waves visually, which seem like two groups of interfering water waves. If the trajectory of a characteristic is tangent with one wave in one group, it cannot be tangent with any wave in the other. The formal proof is presented in Section 5.

Authorized licensed use limited to: Tsinghua University Library. Downloaded on April 06,2010 at 08:51:35 EDT from IEEE Xplore. Restrictions apply.

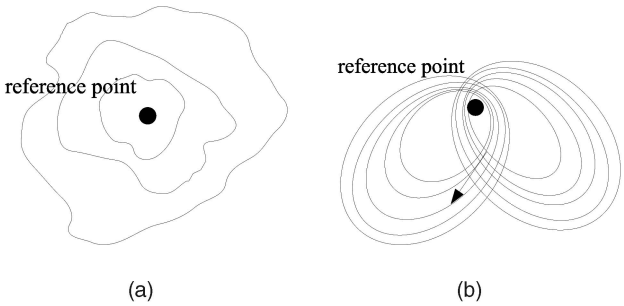

Fig. 5. (a) shows the loops formed by breadth-first traversing; (b) shows the interferences of two breadth-first traversings.

The computing complexity is linear to the number of vertices. In Part I of the algorithm, the complexities of Steps 1 and 2 are $O(C)$ and $O(N)$, respectively, where C is a constant and N is the number of the vertices. The complexity of Steps 3 and 4 is $O(N+E)$, where E is the number of edges. Step 5 runs in $O(N)$. In the procedure of Gauss-Seidel iterations, the complexity is $O(N)$. Thus, this method runs in the linear time to the size of the undigraph. Furthermore, the undigraph is a topological representation of a manifold, which is a considerably sparse one. Considering a regular mesh with $E \simeq \frac{3}{2}N$, the complexity of this method is $O(N+E) = O(N+\frac{3}{2}N) = O(N)$.

5 Algorithm Analysis

5.1 Godunov Numerical Hamiltonian

Given $\triangle ABC$, let $a = |BC|$, $b = |CA|$, $c = |AB|$, $p_1 = (T_C - T_A)/b, p_2 = (T_C - T_B)/a$, and $p_3 = (T_B - T_A)/c$. The discretization for the Eikonal equation we used is based on the Godunov numerical Hamiltonian:

$$\hat{H}_C\left(\frac{T_C - T_A}{b}, \frac{T_C - T_B}{a}\right) = f_C, \tag{5}$$

where

$$\hat{H}_C(p_1, p_2) = \begin{cases} \frac{1}{\sin\gamma}\sqrt{p_1^2 - 2p_1p_2\cos\gamma + p_2^2} \\ \quad \text{if } |p_3| \le f_C \text{ and} \\ \quad \beta - \frac{\pi}{2} \le \arcsin(\frac{p_3}{f_C}) \\ \quad \le \min(0, \frac{\pi}{2} - \alpha) \\ \min\{T_C, T_A + bf_C, T_B + af_C\} \\ \quad \text{otherwise.} \end{cases} \tag{6}$$

Lemma 1 (Godunov numerical Hamiltonian [13]). *The Godunov numerical Hamiltonian is consistent, namely,*

$$\hat{H}_C\left(\frac{T_C - T_A}{b}, \frac{T_C - T_B}{a}\right) = |\mathbf{p}| \tag{7}$$

if $\nabla T = \mathbf{p} \in R^2$.

Lemma 2 (Discretization of the acute solver [13]). *If the causality condition is guaranteed, the acute solver satisfies the discretization of Godunov numerical Hamiltonian.*

Lemmas 1 and 2 have been proven in [13]. The following lemma shows that the obtuse solver satisfies the discretization of the Godunov numerical Hamiltonian.

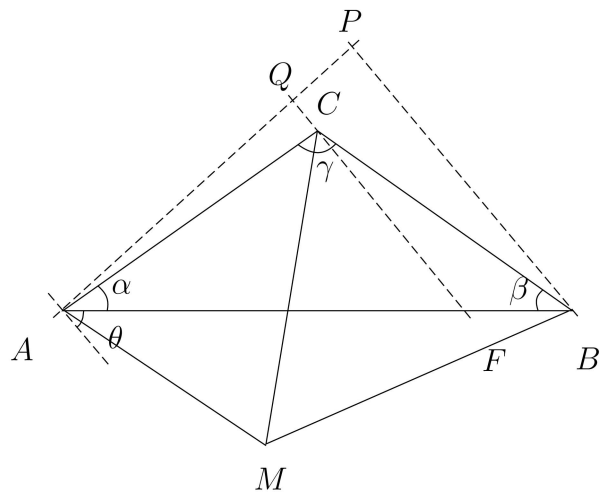

Fig. 6. Two ways to evaluate T_C.

Lemma 3 (Discretization of the obtuse solver). *Assuming that the causality condition is guaranteed, the obtuse solver satisfies the discretization of Godunov numerical Hamiltonian.*

Proof. Since the case of the acute solver has been proved in Lemma 2, only the obtuse one depicted in Fig. 3a is presented here.

In Fig. 6, it is possible to find a point M that can construct two acute angles, $\angle ABM$ and $\angle ACM$. Then, T_C can be obtained from points A and M by

$$T_C = F_{acute_1}(T_A, T_M). \tag{8}$$

By solving (8), we have

$$T_C = F_{acute_1}(T_A, T_M) = \min\{T_C, T_A + \overline{AQ} \cdot f_C\}.$$

Note that another value T'_C of Point C can be evaluated in an obtuse local solver,

$$T'_C = F_{obtuse}(T_A, T_B). \tag{9}$$

Since $\angle ABM$ is an acute triangle, T_B is determined by the points A and M too, namely,

$$T_B = F_{acute_2}(T_A, T_M). \tag{10}$$

By solving (9) and (10), we have

$$\begin{aligned} T'_C &= F_{obtuse}(T_A, T_B) \\ &= F_{obtuse} \circ F_{acute_2}(T_A, T_M) \\ &= \min\{T_C, F_{obtuse}(T_A, T_A + \overline{AP} \cdot f_B)\} \\ &= \min\{T_C, T_A + \overline{AF} \cdot \sin\theta \cdot f_C\} \\ &= \min\{T_C, T_A + \overline{AQ} \cdot f_C\}. \end{aligned}$$

Now, reach the conclusion $T_C = T'_C$, which indicates that the obtuse solver is equivalent to backtracking the acute solver. Thus, the obtuse local solver can use the Godunov numerical model for the discretization. □

It has been proven in [13] that the Godunov numerical model can be used if all of the triangles are acute on the manifold. It has also been proven that the numerical solution is convergent on this discretization. According to Lemmas 2 and 3, the following theorem is proved.

Authorized licensed use limited to: Tsinghua University Library. Downloaded on April 06,2010 at 08:51:35 EDT from IEEE Xplore. Restrictions apply.

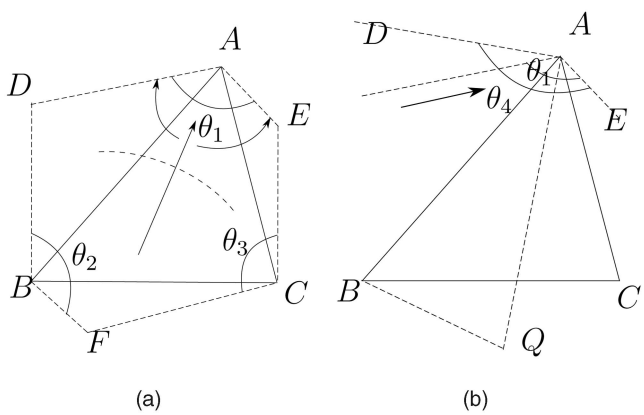

Fig. 7. The causality is bounded by θ in (a). The critical conditions are covered by local solvers.

Theorem 1 (Convergence). *The solution of the fast sweeping algorithm converges monotonically to the solution of the discretized system.*

5.2 Causality and Characteristic

All of the lemmas and the theorem above are built on the basis that the causality condition is guaranteed. As we know, the characteristics never intersect with each other on the manifold. The causality condition stands as long as the algorithm can cover any direction of characteristics locally on the computing domain.

Lemma 4. *Six sweeping orderings in a triangle cover all trajectories of characteristics from any direction of it.*

Proof. Providing a triangle illustrated in Fig. 7a, a black arrow marks the wave propagating direction, which is also the direction of a characteristic. If it is bounded in θ_1, then this direction implies that the value T_A should be obtained from T_B and T_C. The sweeping ordering of the three vertices is either T_B, T_C, and T_A or T_C, T_B, and T_A. In other words, if these three vertices can be evaluated in either ordering, all of the directions of characteristics bounded by θ_1 can be covered. An operator $\odot$ is defined to simplify the further discussion. The domain of $\odot$ is an evaluating sequence in a triangle, and the field is an angle in which the direction of characteristic should be bounded. For example, in Fig. 7a, $\odot(\overrightarrow{BCA})$ represents that the sweeping sequence is T_B, T_C, and T_A. Then, the characteristic direction marked as a black arrow is bounded in θ_1. We have

$$\odot(\overrightarrow{BCA}) = \theta_1.$$

Similarly, we have

$$\odot(\overrightarrow{CBA}) = \theta_1,$$
$$\odot(\overrightarrow{ACB}) = \theta_2,$$
$$\odot(\overrightarrow{CAB}) = \theta_2,$$
$$\odot(\overrightarrow{ABC}) = \theta_3,$$

and

$$\odot(\overrightarrow{BAC}) = \theta_3.$$

Apparently, we have

$$\theta_1 + \theta_2 + \theta_3 = 2\pi. \tag{11}$$

Equation (11) shows that all of the directions are covered by these operations except for six ones along the edges AD, AE, BD, BF, CE, and CF. However, these odd directions will be covered by the obtuse solver. Consider the case of AD in Fig. 7b. Providing that T_A is to be evaluated and there is a wavefront with $\overrightarrow{DA}$ as the direction of its normal, T_A and T_C should be evaluated simultaneously. If T_C is not infinity, T_A can be evaluated correctly by the scheme in Fig. 3a; otherwise, another evaluated vertex Q is introduced, as depicted in Fig. 7b, and then the normal of wavefronts is bound in θ_4. □

Lemma 5. *For a specific $\triangle$ ABC, at most three sweeping orderings are needed to cover trajectories from any direction.*

Proof. In Lemma 4, six sweeping orderings can be divided into three groups:

$$\theta_1 = \odot(\overrightarrow{CBA}) = \odot(\overrightarrow{BCA}),$$
$$\theta_2 = \odot(\overrightarrow{CAB}) = \odot(\overrightarrow{ACB}),$$

and

$$\theta_3 = \odot(\overrightarrow{ABC}) = \odot(\overrightarrow{BAC}).$$

Without loss of generality, we choose vertex A to be evaluated. Then the ordering must be either T_B, T_C, and T_A or T_C, T_B, and T_A. If the ordering is the first one, then the operation $\odot(\overrightarrow{BCA})$ is chosen to evaluate T_A; otherwise, the operation $\odot(\overrightarrow{CBA})$ is chosen. It is noted that both $\odot(\overrightarrow{BCA})$ and $\odot(\overrightarrow{CBA})$ belong to the group θ_1. In other words, only one operation in group θ_1 is needed when T_A is to be evaluated. Similarly, if either T_B or T_C is to be evaluated, one operation belong to either group θ_2 or θ_3 is chosen. It indicates that the three orderings that belong to the three groups in Lemma 4 are enough to cover trajectories from any direction. □

Lemma 6. *Four sweeping orderings cover either of two operations in three groups of Lemma 5.*

First, we present intuitive explanation of this lemma. Letting point A be the initial position of geodesics in Fig. 8a, two orderings are produced by a breadth-first tree. One is $ABCDE$, the other is $ACBED$. Considering $ABCED$ in $\triangle ABC$, the subordering ABC indicates the operation $\odot(\overrightarrow{ABC})$ that belongs to the group θ_3 while its reverse subordering CBA indicates the operation $\odot(\overrightarrow{CBA})$ that belongs to the group θ_1. In the same way, the ACB , which is the subordering of $ACBED$, belongs to the group θ_2 and its reverse ordering BCA belongs to the group θ_1. It is obviously observed that the groups θ_1, θ_2, and θ_3 are included at least once by the two orderings. According to Lemma 5, the four sweeping orderings are produced to cover trajectories from any direction.

Proof. We propose that, within any ordering of three nodes, if the positions of two nodes are swapped to produce a new ordering, these two orderings and their reverse

Authorized licensed use limited to: Tsinghua University Library. Downloaded on April 06,2010 at 08:51:35 EDT from IEEE Xplore. Restrictions apply.

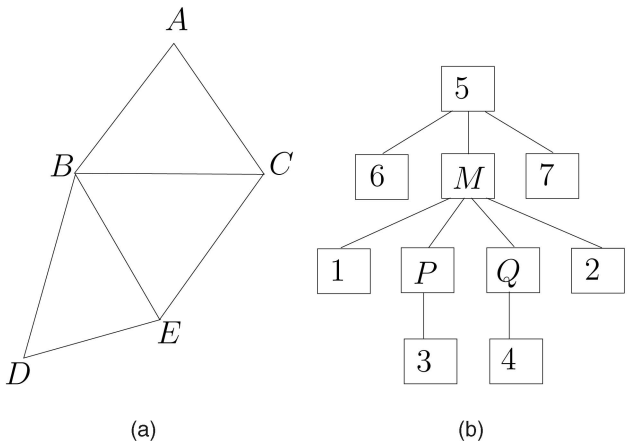

Fig. 8. (a) is a triangular manifold; (b) is a layout of a breadth-first traversing tree.

orderings indicate four different operations, which can cover all the three groups in Lemma 5. This proposition can be verified by an enumeration.

Furthermore, we verify that these two orderings produce correct suborderings for all triangles. If $\triangle MPQ$ is a triangle on a manifold, then one vertex must be the parent of the other two vertices in a traversing tree. Meanwhile, the two children vertices must be neighbors. Fig. 8b shows one of many possible results. The boxes numbered in Fig. 8b could be empty, a vertex, or a tree. The ordering of the first breadth-first tree is $(5,6)M(7,1)PQ(2,3,4)$. The ordering of the second breadth-first tree is $(5,7)M(6,2)QP(1,4,3)$. Two correct suborderings MPQ and MQP are produced. □

Theorem 2. *The FSM covers trajectories of characteristics from any direction.*

Proof. With the combination of Lemmas 4, 5, and 6, the theorem is proven to be correct. □

Corollary 1. *If the initial boundary condition is a single point, the FSM evaluates the arrival time of each vertex on the manifold in four times sweeping.*

Proof. This proposition is correct because, first, there are no intersections of the characteristics on each triangles, and second, any direction of the characteristics can be covered by the four sweeping orderings when evaluating vertices in a triangle. □

5.3 Error Estimation

In this section, the error estimation of geodesics at each node on the manifold is proposed. The error analysis model

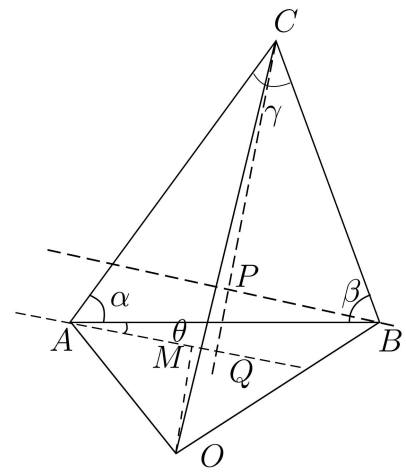

Fig. 9. The error analysis model. O is the initial position and we apply the local solvers in $\triangle ABC$ to evaluate C.

is given in Fig. 9. The exact distance function $d(\mathbf{x})$ satisfies the Eikonal equation $|\nabla d(\mathbf{x})| = 1$ everywhere except for the initial data point. Let h_{max} represent the maximal length of edges and h_{min} represent the minimal one on the manifold. Using a Godunov numerical Hamiltonian at the vertex C in $\triangle ABC$, we have

$$\frac{d_C - d_A}{b} = \nabla d(C) \cdot \frac{\overrightarrow{AC}}{|AC|} + O\big(h_{max}^2\big). \tag{12}$$

Meanwhile, the numerical solution is

$$T_C = T_A + f_c \cdot b \cdot \sin(\alpha + \theta), \tag{13}$$

where $f_c = 1$. Since this discretization produces monotonic numerical solutions, the absolute error Δ_C is

$$\begin{aligned}\Delta_C &= T_C - d_C \\ &= T_A - d_A + b\sin(\alpha+\theta) - b\nabla d(C)\cdot\frac{\overrightarrow{AC}}{|AC|} \\ &\quad + b\cdot O\big(h_{max}^2\big) \\ &= T_A - d_A + b(\cos\angle ACQ - \cos\angle ACO) \\ &\quad + b\cdot O\big(h_{max}^2\big) \\ &\le T_A - d_A + b\left|\sin\frac{\angle QCO}{2}\right| + b\cdot O(h_{max}^2).\end{aligned}$$

Noticing that $\angle QCO = \angle MOC$, $\sin\angle MOC \le \sin\angle AOC = \frac{|AM|}{|AO|} \le \frac{|AB|}{2|AO|}$, and $|AO| \le n \cdot h_{max}$, where n is the number of triangles, we further have

TABLE 1
Accuracy on Planes

					FMM			FSM		
Model	**Vertices**	**Faces**	h_{max}	h_{min}	**Time (ms)**	**Max**	**Avg**	**Time (ms)**	**Max**	**Avg**
Plane 1	441	800	0.7071	0.5	21	0.51239	0.27365	3	0.51239	0.27365
Plane 2	1681	3200	0.3536	0.25	45	0.310614	0.171475	16	0.310614	0.171475
Plane 3	6561	12800	0.1767	0.125	184	0.183772	0.103797	66	0.183772	0.103797
Plane 4	25921	51200	0.08838	0.0625	856	0.106541	0.061226	270	0.106541	0.061226
Plane 5	103041	204800	0.04419	0.03125	4513	0.060633	0.035395	1089	0.060633	0.035395

Authorized licensed use limited to: Tsinghua University Library. Downloaded on April 06,2010 at 08:51:35 EDT from IEEE Xplore. Restrictions apply.

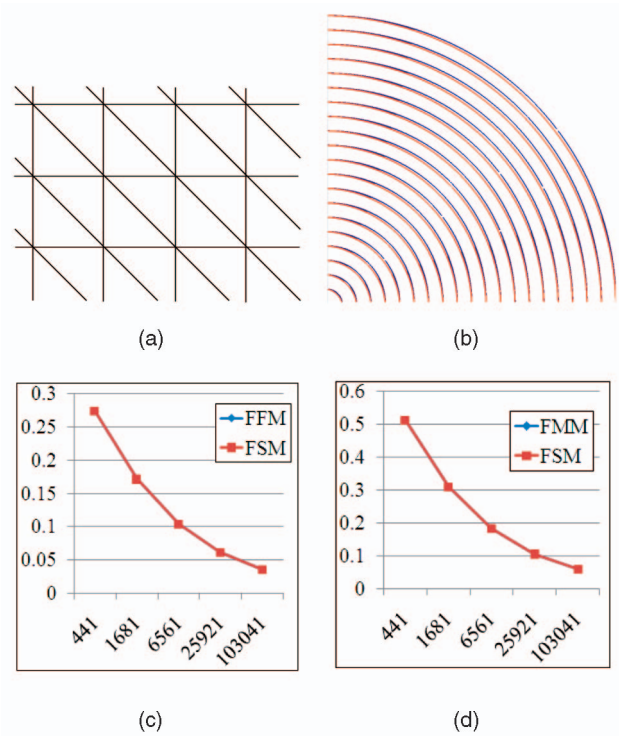

Fig. 10. (a) is a triangulated plane, (b) is the result of computing geodesics on the plane, (c) is the average error, and (d) is the maximal error of the nodes on the plane.

$$\begin{aligned}\Delta_C &\le T_A - d_A + h_{max}\cdot\frac{|AB|}{2|AO|} + O\big(h_{max}^3\big)\\ &\le T_O - d_O + \frac{1}{2}h_{max}\sum_{i=1}^{n}\frac{1}{i} + n\cdot O\big(h_{max}^3\big).\end{aligned}$$

We give an upper bound of n such that $n < \frac{S}{\frac{\sqrt{3}}{8}h_{min}^2}$, where S is the area of the computing domain. Thus,

$$\Delta_C \le \frac{1}{2}h_{max}\ln\frac{8S}{\sqrt{3}h_{min}^2} + O\left(S\cdot\frac{h_{max}^3}{h_{min}^2}\right). \tag{14}$$

From (12), there is a $\delta > 0$ which satisfies $O(|S\cdot\frac{h_{max}^3}{h_{min}^2}|) < \delta\cdot|S\cdot\frac{h_{max}^3}{h_{min}^2}|$. Hence, the error is upbounded. This upper bound is only provided for analytic reason. This worst result happens when all the triangles are juxtaposed along a line. A reasonable error bound is as follows:

Since a tree is constructed in the algorithm and the evaluating ordering follows the upwind direction, we have $\hat{n} = O(\ln n)$, then (14) can be rewritten as

$$\begin{aligned}\Delta_C \le{}& \ln\ln\frac{8S}{\sqrt{3}h_{min}^2}\cdot O(h_{max})\\ &+ \ln\frac{8S}{\sqrt{3}h_{min}^2}\cdot O\big(h_{max}^3\big).\end{aligned} \tag{15}$$

6 Numerical Experiments

6.1 Accuracy on 2D Domains

This experiment compares the accuracy of the proposed FSM with the FMM in 2D domain. Planes are divided into five levels, which are different in the number of vertex, the number of edge, the length of the maximum edge, and so on. These information are summarized in Table 1, where h_{max} and h_{min} mean the maximal and the minimal length of edges, respectively. Time costs are also recorded in the table. The domain is a certain triangulated plane, which is bounded in the region $[-5,5]\times[-5,5]$. As shown in Fig. 10a, five planes are created. Their boundary conditions are all $T(-5,-5) = 0$. Both h_{max} and h_{min} are a half of that in its prior level. To measure the error, two error normals are introduced. In order to compare the accuracy and complexity between the FMM in [11] and the proposed FSM, several numerical experiments have been conducted on an Intel Core Duo E4500@ 2.2 GHz PC with 2 GB RAM.

$$\max_{i\in V} T_i - d_i \tag{16}$$

and the absolute average error with

$$\frac{1}{|V|}\sum_{i\in V} T_i - d_i, \tag{17}$$

where V is the vertex set of the given manifold, T_i is the numerical solution, and d_i is the exact solution of the vertex i. The result is shown in Fig. 10b, where the blue line stands for the exact solution and the red one is calculated by the FSM.

Figs. 10c and 10d are the line charts of the absolute average error and absolute maximum error that are both caused by the FMM and FSM, respectively. In the line chart, the X-axis stands for the number of the vertices in five different planes and the Y-axis stands for the absolute errors. Although the same plane is processed by the two different algorithms independently, nearly the same data are obtained in this process. The evidence is provided in Table 1. That is why only one line can be seen in each chart. This phenomenon demonstrates that the FSM has a comparative accuracy with that of the FMM in these plane

TABLE 2
Accuracy on Spheres

					FMM			FSM		
Model	**Vertices**	**Faces**	h_{max}	h_{min}	**Time (ms)**	**Max**	**Avg**	**Time (ms)**	**Max**	**Avg**
Sphere 1	4682	9360	1.752	0.1231	255	0.217530	0.0203613	62	0.244528	0.0220840
Sphere 2	14042	28080	1.752	0.0320	1132	0.254583	0.0219177	189	0.266872	0.0243771
Sphere 3	42122	84240	1.752	0.00597	5510	0.299209	0.0345620	658	0.295206	0.0249671
Sphere 4	126362	252720	1.752	0.00149	28587	0.416340	0.0558753	2239	0.313131	0.0256877

Authorized licensed use limited to: Tsinghua University Library. Downloaded on April 06,2010 at 08:51:35 EDT from IEEE Xplore. Restrictions apply.

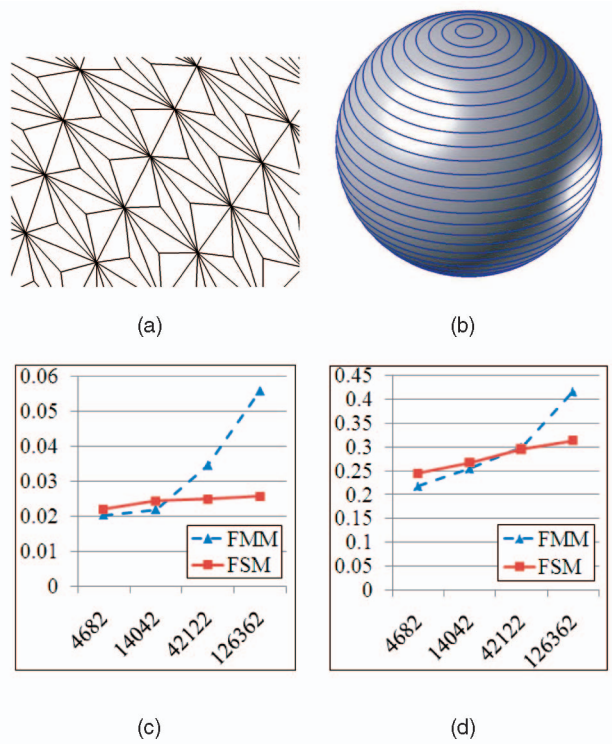

Fig. 11. (a) is a triangulated sphere model, (b) is the result of geodesics on the sphere, (c) is the average error, and (d) is the maximal error of the nodes on the sphere.

domains. The trends that both absolute average error and absolute maximum error decrease when the number of vertices increases are represented in the two line charts.

6.2 Accuracy on Spheres

This experiment compares the accuracy of the proposed FSM and FMM with the exact geodesics on a manifold of sphere. The computing domain represents several triangulated spheres bounded in the box $[-10, 10] \times [-10, 10] \times [-10, 10]$. The same error measurement of (16) and (17) is used. Similar to the prior experiment, the spheres are divided into four levels according to their varied degrees of subdivision. The information also can be observed in Table 2. A start point is selected as a source point arbitrarily. Slightly different from that in Table 1, h_{max} is a constant in all levels. Each triangle in a sphere manifold model is divided by connecting the three vertices to the center point. After the operation of division, these models are all mixed with acute and obtuse triangles. Fig. 11a has shown the detail of a sphere model.

Fig. 11b demonstrates the result of the FSM. Figs. 11c and 11d are the absolute average error and absolute maximum error comparisons between the FMM and FSM, respectively. The X-axis represents the vertex number and the Y-axis represents the absolute error. One common in these two line charts is the trend that the errors increase as the number of vertices increases. In the first chart, the difference between the two algorithms is trivial at the beginning and is enlarged at Level 3, which has 42,122 vertices in the model. Similar result also appears in the second chart, where the difference approximates 0.1 at Level 4. Noticing that error accumulation in the FSM is relatively slower than that in the FMM, which shows that the local geometric changes have less influence on the proposed method than that of the FMM.

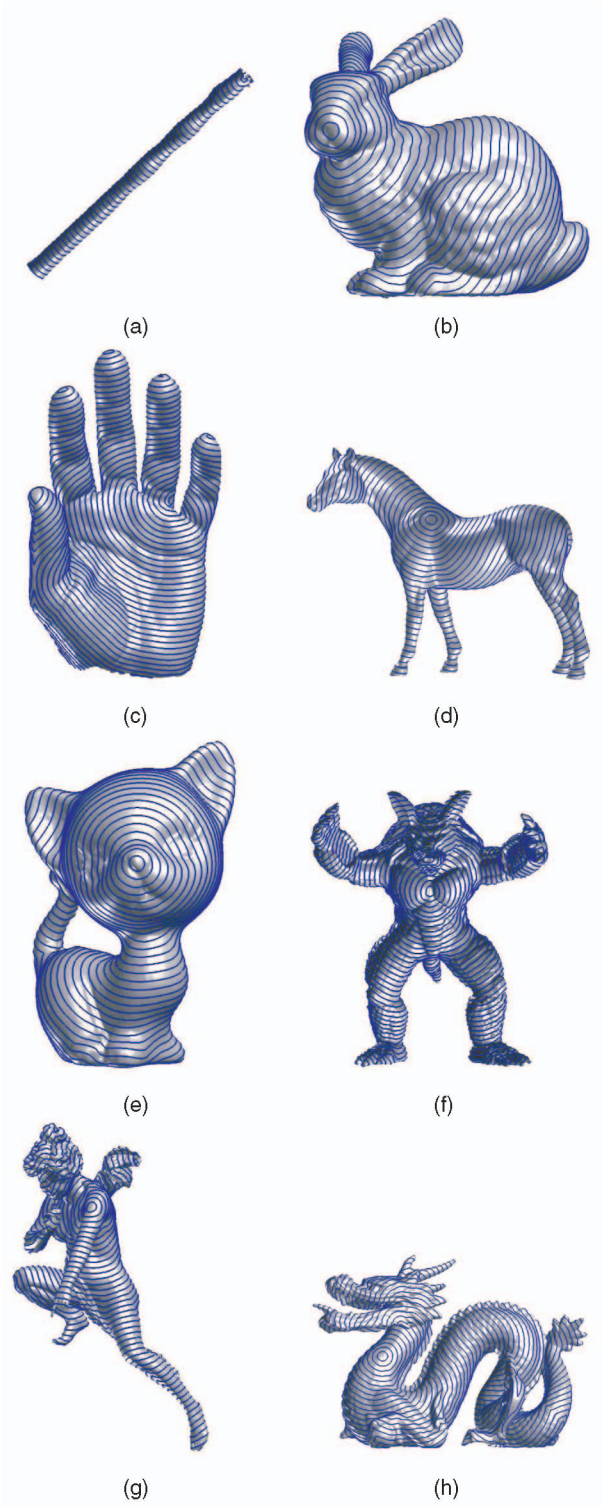

Fig. 12. Results of computing geodesics on the models of arbitrary manifolds. (a) Drill, (b) Bunny, (c) Hand, (d) Horse, (e) Kitty, (f) Armadillo, (g) Angel, and (h) Dragon.

6.3 Efficiency on Arbitrary Manifolds

Differently from the two experiments above, this experiment is conducted to identify the time costs of the two algorithms in a series of arbitrary mesh models. Contours

Authorized licensed use limited to: Tsinghua University Library. Downloaded on April 06,2010 at 08:51:35 EDT from IEEE Xplore. Restrictions apply.

TABLE 3
Performance on Manifolds

Model	Vertices	Faces	FMM (ms)	FSM (ms)
Drill	1961	3855	69	21
Bunny	10680	21348	506	138
Hand	24795	49432	1373	344
Horse	48485	96966	3297	696
Kitty	134448	268896	11632	1856
Armadillo	172974	345944	20801	2839
Angel	237018	474048	33677	3892
Dragon	437645	871414	79611	5888

are drawn on all mesh models in Fig. 12 to exhibit the solutions by performing the FSM. The time costs that represent their performances are recorded in Table 3. Fig. 13 demonstrates this comparison in a line chart. In Table 3, the scale of the vertex number ranges from 1,961 to 437,645. The difference on the time costs of the two methods is obvious. This trend is shown clearly in Fig. 13. When performed on the bunny model, the FMM takes 506 ms to complete the calculation, which is almost 4 times longer than 138 ms the FSM takes. The performance gap is widen to more than 10 times when these two algorithms are performed on the model dragon, which has more than 437,000 points. It is trivial to see that the FSM has a linear cost to the scale of the manifolds while the FMM does not. We conclude that the FSM performs more efficiently than the FMM.

This paper extends the FSM to the domain of arbitrary manifolds for the computation of geodesics and reduce the computing complexity to $O(N \log N)$. In order to facilitate the construction of sweeping orderings, the local solver for obtuse triangles is improved. Since the triangular manifold can be viewed as an undigraph that takes the natural topology from the primitive model, four orderings for sweeping are produced via the breadth-first traversing on the undigraph. These orderings are proven to satisfy the basic requirement of characteristics' coverage, which ensures that reasonable numerical results are to be produced. The computational complexity is linear to the scale of vertices on the manifold. Extensive numerical experiments demonstrate the accuracy and efficiency of the proposed algorithm.

7 Conclusion

In the future, we intend to study the FSM for the general Eikonal equation in the domain of arbitrary manifolds.

Acknowledgments

This research was supported by the Chinese 973 Program (2004CB719400), the National Science Foundation of China (60625202, 60533070, 90715043, 60911130368), and the Chinese 863 Program (2007AA040401). J.-H. Yong was supported by the Fok Ying Tung Education Foundation (111070). Furthermore, in Fig. 12, Model *Bunny* was courteously provided by MPII, Model *Hand* was courteously provided by INRIA and IMATI, and Model *Kitty* was courteously provided by Frank_ter_Haar, from the AIM@-SHAPE Shape Repository.

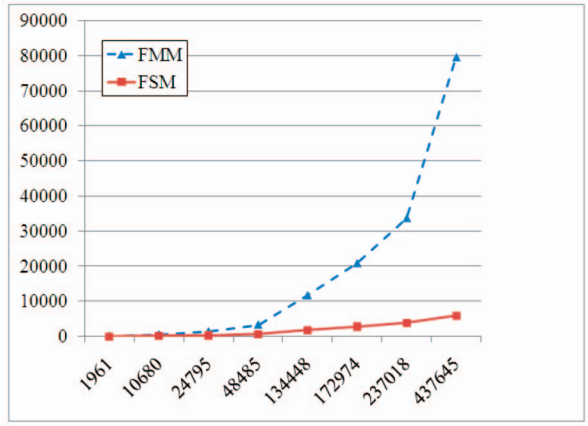

Fig. 13. Comparison of performance.

References

[1] J. Sethian, *Level Set Methods and Fast Marching Methods*, second ed. Cambridge Univ. Press, 1999.
[2] R. Kimmel and J. Sethian, "Optimal Algorithm for Shape from Shading and Path Planning," *J. Math. Imaging and Vision*, vol. 14, no. 3, pp. 237-244, 2001.
[3] M. Hassouna, A. Abdel-Hakim, and F. Farag, "Robust Robotic Path Planning Using Level Sets," *Proc. IEEE Int'l Conf. Image Processing*, Sept. 2005.
[4] R. Kimmel, D. Shaked, N. Kiryati, and A. Bruckstein, "Skeletonization via Distance Maps and Level Sets," *Computer Vision and Image Understanding*, vol. 62, no. 3, pp. 382-391, 1995.
[5] V. Kirshnamurthy and M. Levoy, "Fitting Smooth Surfaces to Dense Polygon Meshes," *Proc. ACM SIGGRAPH*, pp. 313-324, 1996.
[6] J. van Trier and W. Symes, "Upwind Finite-Difference Calculation of Traveltimes," *J. Geophysics*, vol. 56, no. 6, pp. 812-821, 1991.
[7] R. Rawlinson and M. Sambridge, "Wave Front Evolution in Strongly Heterogeneous Layered Media Using the Fast Marching Method," *Geophysical J. Int'l*, vol. 156, no. 3, pp. 631-647, 2004.
[8] J. Sethian, "A Fast Marching Level Set Method for Monotonically Advancing Fronts," *Proc. Nat'l Academy of Sciences USA*, vol. 93, no. 4, pp. 1591-1595, 1996.
[9] J. Tsitsiklis, "Efficient Algorithms for Globally Optimal Trajectories," *IEEE Trans. Automatic Control*, vol. 40, no. 9, pp. 1528-1538, Sept. 1995.
[10] J. Sethian, "Fast Marching Methods and Level Set Methods for Propagating Interfaces," technical report, Von Karman Inst. Lecture Series on Computational Fluid Mechanics, 1998.
[11] R. Kimmel and J. Sethian, "Computing Geodesic Paths on Manifolds," *Proc. Nat'l Academy of Sciences USA*, vol. 95, no. 15, pp. 8431-8435, 1998.
[12] H. Zhao, "Fast Sweeping Method for Eikonal Equations," *J. Math. and Computing*, vol. 74, no. 250, pp. 603-627, 2005.
[13] J. Qian, Y. Zhang, and H. Zhao, "Fast Sweeping Methods for Eikonal Equations on Triangular Meshes," *SIAM J. Numerical Analysis*, vol. 45, no. 1, pp. 83-107, 2005.
[14] J. Sethian and A. Vladimirsky, "Ordered Upwind Methods for Static Hamilton-Jacobi Equations: Theory and Algorithms," *SIAM J. Numerical Analysis*, vol. 41, no. 1, pp. 325-363, 2003.
[15] S. Kim and D. Folie, "The Group Marching Method: An O(N) Level Set Eikonal Solver," research report, Dept. of Math., Univ. of Kentucky, http://www.ms.uky.edu/~math/MAreport/PDF/00-03.pdf, 2000.
[16] J. Sethian, "Fast Marching Methods," *SIAM J. Rev.*, vol. 41, no. 2, pp. 199-235, 1999.
[17] J. Rickett and S. Fomel, "A Second-Order Fast Marching Eikonal Solver," 2000.
[18] A.B.L. Yatziv and G. Sapiro, "O(N) Implementation of the Fast Marching Algorithm," *J. Computational Physics*, vol. 212, no. 2, pp. 393-399, 2006.
[19] A. Faraq and M.S. Hassouna, "Multistencils Fast Marching Methods: A Highly Accurate Solution to the Eikonal Equation on Cartesian Domains," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1563-1574, June 2007.

Authorized licensed use limited to: Tsinghua University Library. Downloaded on April 06,2010 at 08:51:35 EDT from IEEE Xplore. Restrictions apply.

[20] M. Bardi and L. Evans, "On Hopf's Formulas for Solutions of Hamilton-Jacobi Equations," *J. Nonlinear Analysis: Theory, Methods and Application,* vol. 8, no. 11, pp. 1373-1381, 1984.
[21] R. Abgrall, "Numerical Discretization of the First-Order Hamilton-Jacobi Equations on Triangular Meshes," *Comm. Pure and Applied Math.,* vol. 9, no. 12, pp. 1339-1373, 1998.
[22] S. Augoula and R. Abgrall, "High Order Numerical Discretization for Hamilton-Jacobi Equations on Triangular Meshes," *J. Scientific Computing,* vol. 15, no. 2, pp. 197-229, 2000.
[23] A. Bronstein, M. Bronstein, and R. Kimmel, "Weighted Distance Maps Computation on Parametric Three-Dimensional Manifolds," *J. Computational Physics,* vol. 225, no. 1, pp. 771-784, 2007.
[24] R. Tsai, H. Zhao, and S. Osher, "Fast Sweeping Algorithms for a Class of Hamilton-Jacobi Equations," *SIAM J. Numerical Analysis,* vol. 41, no. 2, pp. 673-694, 2003.
[25] C. Kao, S. Osher, and J. Qian, "Lax-Friedrichs Sweeping Scheme for Static Hamilton-Jacobi Equations," *J. Computational Physics,* vol. 196, no. 1, pp. 367-391, 2004.
[26] C. Kao, S. Osher, and R. Tsai, "Fast Sweeping Methods for Hamilton-Jacobi Equations," *SIAM J. Numerical Analysis,* vol. 42, no. 1, pp. 2612-2632, 2005.
[27] H.Z.Y. Zhang and J. Qian, "High Order Fast Sweeping Methods for Static Hamilton-Jacobi Equations," *J. Computational Physics,* vol. 29, no. 1, pp. 25-56, 2006.
[28] J. Qian, Y. Zhang, and H. Zhao, "A Fast Sweeping Method for Static Convex Hamilton-Jacobi Equations," *SIAM J. Scientific Computing,* vol. 31, no. 2, pp. 237-271, 2007.
[29] H. Zhao, S. Osher, B. Merrimanb, and M. Kang, "Implicit and Non-Parametric Shape Reconstruction from Unorganized Points Using Variational Level Set Method," *Computer Vision and Image Understanding,* vol. 80, no. 3, pp. 295-314, 2000.
[30] P.E. Danielsson, "Euclidean Distance Mapping," *Computer Graphics and Image Processing,* vol. 14, no. 3, pp. 227-248, 1980.
[31] A. Bronstein, M. Bronstein, Y. Dvir, R. Kimmel, and O. Weber, "Parallel Algorithms for Approximation of Distance Maps on Parametric Surfaces," technical report, Dept. of Computer Science, Technion, 2007.

Song-Gang Xu received the BSc degree in computer science from Hefei University of Technology, China, in 2005. He joined the research and development group of the Institute of Computer Graphics and Computer Aided Design in the School of Software at Tsinghua University, China, as a master's candidate in 2006. His research interests include manifold reconstruction, path problem, and point set surface.

Yun-Xiang Zhang received the BSc degree in software engineering from Wuhan University, China, in 2004. He joined the research and development group of the Institute of Computer Graphics and Computer Aided Design in the School of Software at Tsinghua University, China, as a master's candidate in 2006. His research interests include geodesics computing, mesh segmentation, and 3D model retrieval.

Jun-Hai Yong received the BS and PhD degrees in computer science from Tsinghua University, China, in 1996 and 2001, respectively. He is currently a professor in the School of Software at Tsinghua University, China. He held a visiting researcher position in the Department of Computer Science at the Hong Kong University of Science and Technology in 2000. He was a postdoctoral fellow in the Department of Computer Science at the University of Kentucky from 2000 to 2002. His research interests include computer-aided design, computer graphics, computer animation, and software engineering.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: Tsinghua University Library. Downloaded on April 06,2010 at 08:51:35 EDT from IEEE Xplore. Restrictions apply.