Geodesic Distance Propagation Across Open Boundaries

Shuangmin Chen[1], Zijia Yue[1], Wensong Wang[2], Shiqing Xin[2], Changhe Tu[2]

[1]Qingdao University of Science and Technology
[2]Shandong University

Abstract

The computation of geodesic distances on curved surfaces stands as a fundamental operation in digital geometry processing. Throughout distance propagation, each surface point assumes the dual role of a receiver and transmitter. Despite substantial research on watertight triangle meshes, algorithms designed for broken surfaces—those afflicted with open-boundary defects—remain scarce. Current algorithms primarily focus on bridging holes and gaps in the embedding space to facilitate distance propagation across boundaries but fall short in addressing large open-boundary defects in highly curved regions. In this paper, we delve into the prospect of inferring defect-tolerant geodesics exclusively within the intrinsic space. Our observation reveals that open-boundary defects can give rise to a "shadow" region, where the shortest path touches open boundaries. Based o n such an observation, we have made three key adaptations to the fast marching method (FMM). Firstly, boundary points now exclusively function as distance receivers, impeding any further distance propagation. Secondly, bidirectional distance propagation is permitted, allowing the prediction of geodesic distances in the shadow region based on those in the visible region (even if the visual region is a little more distant from the source). Lastly, we have redefined priorities to harmonize distance propagation between the shadow and visible regions. Notably intrinsic, our algorithm distinguishes itself from existing counterparts. Experimental results showcase its exceptional performance and accuracy, even in the presence of large and irregular open boundaries.

CCS Concepts

• ***Computing methodologies*** → ***Mesh models; Mesh geometry models;***

1. Introduction

The geodesic distance between two points on a mesh surface represents the minimum distance one must travel along the surface to connect these points. Geodesic distances find applications in various fields, including computer graphics, simulations, and computational geometry. Fast and accurate estimation of geodesic distances is crucial in applications such as shape analysis [PZP*20], path planning [DDG*18], and mesh parameterization [LAKH22]. Given that polygonal surfaces are the most popular representation in the computer graphics community, the design of geodesic algorithms has to consider the discrete nature of mesh structures.

Most existing geodesic algorithms, such as the fast marching algorithm [SV00] and the heat method [CWW13], operate on the inherent geometry since geodesic distances are independent of the embedding space. Each surface point assumes the dual role of a receiver and transmitter. Despite significant progress in computing geodesics on watertight triangle meshes, algorithms designed for handling surfaces with missing parts remain limited. The challenge arises when the geodesic path reaches a boundary point, where it has to advance along the boundary curve, resulting in computed distances larger than those on the ground-truth surface. Existing defect-tolerant geodesic algorithms bridge holes and gaps in the embedding space to facilitate distance propagation across open boundaries. However, these algorithms have two common disadvantages. First, they depend on the degree to which the guessed "cap" is close to the real missing part, making accuracy hard to guarantee. Second, they cannot address large open-boundary defects in highly curved regions.

We observe that open-boundary defects may create a "shadow" region during the propagation, where the shortest path touches open boundaries. To address this issue, we have adapted the fast marching method (FMM) in three aspects. Firstly, boundary points solely serve as distance receivers, preventing further distance propagation. Secondly, we permit bidirectional distance propagation, enabling the prediction of geodesic distances in the shadow region based on those in the visible region, albeit potentially farther. Lastly, we have redefined priorities to coordinate distance propagation between shadow and visible regions. Unlike existing algorithms that bridge holes or gaps, our algorithm, being completely intrinsic, distinguishes itself from current approaches. Figure 1 demonstrates that our algorithm is efficient and accurately reports geodesic distances, even in the presence of large and irregular open boundaries.

Our contributions are three-fold:

1. We develop an efficient algorithm for estimating geodesic dis-

DOI: 10.2312/pg.20241284

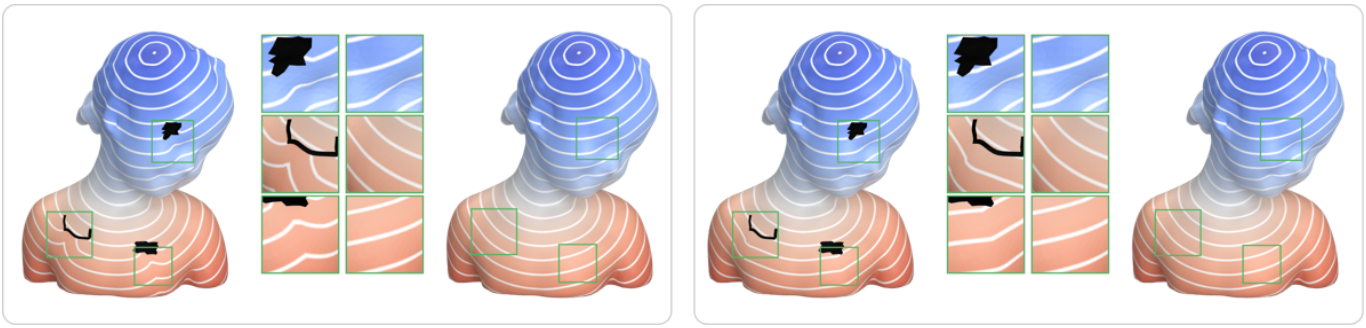


Figure 1: *FMM (left) struggle with missing parts, resulting in geodesic distance fields that are significantly different from the ground truth. In contrast, our algorithm (right) can tolerate missing parts, producing an outcome that closely resembles the ground truth.*

tances for 3D shapes with open boundaries. To the best of our knowledge, this is the first intrinsic algorithm for computing open-boundary-tolerant geodesics.
2. We propose a bidirectional distance propagation mechanism, enabling the computation of geodesic distances in the shadow region based on those in the visible region.
3. We coordinate distance propagation priorities to enable synchronous estimation of geodesic distances in visible and shadow regions.

2. Related Work

There is a large body of literature on geodesic computation. In this section, we review the existing research works in two categories. The first category of algorithms assumes that the input surface (typically a triangle mesh) is complete and does not contain missing parts. The second category, instead, aims to report defect-tolerant geodesic distances.

2.1. Exact and approximate geodesic algorithms

Exact algorithms. Given a surface with analytic representation, the query of a geodesic path can be formulated as the Euler-Lagrange partial differential equation in differential geometry, but the PDE has no closed-form solution generally - one has to find a numerical solution instead.

Considering a smooth surface can be discretized into a polygonal mesh, researchers began to study computing an exact geodesic distance field on a polygonal surface forty years ago. Representative algorithms include the MMP algorithm [MMP87], the CH algorithm [CH90], and some variant versions [SSK*05, XW09, Liu13, XWL*15, QHY*16]. These algorithms commonly use a window to encode those geodesic paths sharing the same edge sequence, which enables one to represent infinitely many geodesic paths by finite many discrete windows. The key spirit is to propagate a dynamic outward wavefront across mesh faces in a Dijkstra-like sweep. Exact geodesic algorithms are not easy to be extended to other geometric domains such as point clouds or tetrahedral meshes.

Graph-based algorithms. Graph-based methods solve the discrete geodesic problem by transforming the polygonal mesh M into a dense graph G so that any geodesic path on the mesh M has an approximate counterpart encoded by G. [ALMS98] suggested adding a logarithmic number of Steiner points along each edge of the mesh, which are placed in a geometric progression along an edge. Then it takes $O(mn\log(mn)+nm^2)$ time to compute a $(1+\varepsilon)$-approximation geodesic path, where m is the total number of Steiner points and n is the number of mesh edges. [XYH12] suggested adding m uniformly distributed auxiliary points on the mesh surface, which takes $O(mn^2\log n)$ time to construct G with $O(m^2+n)$ edges. With the help of G, the approximate geodesic distance between any pair of points (not necessarily mesh vertices) can be queried in constant time. [AFH20] is a new accuracy-aware window propagation algorithm, which directly and effectively calculates the edges of the graph. Compared with the thermal method, in terms of accuracy, for anisotropic meshes, it is necessary to manually adjust the smoothing parameters. The graph method can guarantee the true distance metric, while the thermal method cannot. The segmentation method of the algorithm may cause the geodesics between adjacent triangles to be not smooth, thus affecting the coherence of the geodesics. [YWH13] proposed a saddle vertex graph (SVG) for encoding all-pair geodesic distances. Constructing SVG takes $O(nK^2\log K)$ time, where K is the maximal degree to define the proximity between saddle vertices. [WFW*17] proposed a discrete geodesic graph (DGG) to approximate the geodesic distance between a pair of points with specified accuracy, which runs in empirically linear computational time. [MP23] introduced a direct method for computing the Riemannian center of mass on triangle meshes, which operates under the polyhedral metric and utilizes piecewise-linear interpolation of gradients from distance fields of geodesics emanating from a set of control points. The approach is demonstrated to be more stable and significantly faster than existing state-of-the-art methods.

Approximation algorithms based on $\|\nabla\mathbf{D}\|=1$. Most of PDE-based methods solve the Eikonal equation $\|\nabla\mathbf{D}\|=1$ with boundary condition $\mathbf{D}(s)=0$, where s is the source point. For example, the fast marching method (FMM) [KS98, Set96, SV00] uses an upwind finite difference scheme to approximate the accurate

geodesic distance field in $O(n \log n)$ time. [WDB*08] proposed an $O(n)$-time parallel FMM on geometry images. There are also other approaches to solve the eikonal equation indirectly. For example, the heat method [CWW13], as well as its parallel implementation [TZD*19], utilizes an observation that the heat field and the geodesic distance field share the common isolines (but may have different values). They need to normalize the gradient field into a unit vector field before the reconstruction of the distance field.

Smoothness driven geodesic algorithms. Although there are some smoothness-driven geodesic algorithms, such as commute distances [FPRS07], diffusion distances [CL06], biharmonic distances [LRF10], heat distances [CWW13], and QGDF [CZX*20], quite few research works systematically discuss the relationship between smoothness and accuracy. In fact, a geodesic distance field is quite different from a general scalar field. The difference lies in that an exact geodesic distance field is naturally smooth at non-ridge points but not at ridge points. In order to carefully maintain the balance between smoothness and accuracy, the smooth effect should be enforced in the near-ridge area, rather than all over the surface. The goal of this paper is to keep the overall accuracy while optionally smoothening the geodesic distance field in the near-ridge area.

2.2. Defect-tolerant geodesic algorithms

Existing algorithms predominantly concentrate on rectifying holes and gaps within the embedded space, which aids in the propagation of distances across boundaries. However, they encounter limitations when it comes to tackling extensive open-boundary defects, particularly in regions characterized by high curvature.

Campen and Kobbelt [CK11] pioneered defect-tolerant geodesic distance algorithms on broken grids. This algorithm performs well on meshes with various defects, such as gaps, holes, disconnected components, noise, polygon soup, and non-manifold structures. However, the quality of the defect-tolerant geodesic distance is highly contingent on the user-specified tolerances. Moreover, if the sizes of the defects exceed the local feature sizes, the morphological operators may fail to bridge the holes or gaps. Xin et al. [XQYH12]proposed a global algorithm for calculating the defect tolerance geodesic distance in an iterative and global manner. It enforces unit gradient constraints everywhere, which leads to a globally smooth and isotropic distance field. The algorithm can provide a good guess for small defects (e.g., small holes, small gaps), but for larger gaps, this strategy will fail.

N.Feng and K.Crane [FC24] proposed a method to compute a signed geodesic distance field from a broken input. The algorithm for computing the Signed Distance Function (SDF) on imperfect geometries involves diffusing the input shape's normals using a vector heat equation, normalizing the resulting vectors to approximate the SDF's gradient, and then solving a scalar Poisson equation to obtain the SDF that best fits this gradient. Even if the input surface is broken, one can still get a faithful distance field as long as the underlying SDF can be effectively estimated. However, this depends on the embedding space and thus makes the approach not totally intrinsic.

3. Algorithm

3.1. Overview

In this subsection, we first give the overview of our algorithm and then elaborate the algorithmic details. Let $M = (V, E, F)$ be a triangle mesh with k holes $h_1, \ldots, h_k$ and V, E, F be its vertex, edge, and face sets. Consider a scalar field $d : V \to \mathbb{R}$ defined on V,

$$d = \{d_i | d_i \text{ is a scalar value at vertex } v_i\}. \quad (1)$$

Given a source point $s \in M$, our algorithm initially computes the geodesic distance field on M using the Fast Marching Method. We then categorize the mesh vertices into two disjoint groups: those that can be "illuminated" by light located at s and those that remain in the "shadow" due to the holes. Our objective is to provide a meaningful approximation of the geodesic distances within the "shadow" region.

To address these challenges, our algorithm enhances the FMM, specifically tailoring it for models with defects such as holes or gaps. It is capable of predicting geodesic distances on the original model (without defects) without the need to physically fill in the defects. The pseudo-code of our approach is provided in Algorithm 1. First, boundary points function exclusively as receivers of distance information, preventing further propagation of distance from these points. Second, by leveraging distance information from visible regions, our algorithm can predict the gradient direction of the distance field in the "shadow" regions. This allows for the potential continuation of propagation from areas that are somewhat more distant from the source. Lastly, we have redefined priority settings to coordinate distance propagation between shadow and visible regions more effectively. Figure 2 clearly illustrates the differing processes of distance propagation after adapting the FMM.

In the following of this section, we first review the FMM in Section 3.2, then show bidirectional distance propagation technique in Section 3.3. Finally, we present the adjustments made to the propagation priorities in Section 3.4.

3.2. Review on FMM

The Fast Marching Method (FMM) for computing geodesics is an efficient numerical technique, particularly suited for dealing with discrete surface data structures, such as meshes composed of triangles. Here are the basic steps for calculating geodesics on a triangular mesh surface using FMM:

1. **Initialization:** Select one or more source points and initialize their geodesic distances to zero. At the same time, mark the status of the source points as "Closed".
2. **Constructing the priority queue:** Build a priority queue (also known as the front) around the source points, which includes all points adjacent to the source points. The status of these points is marked as "Alive".
3. **Propagation:** Select a point from the priority queue for updating until empty, typically choosing the point with the smallest estimated distance, marked as "Closed". For all neighbors of this point, calculate the geodesic distance from the source point to these neighbors. For each neighbor of a given point, calculate

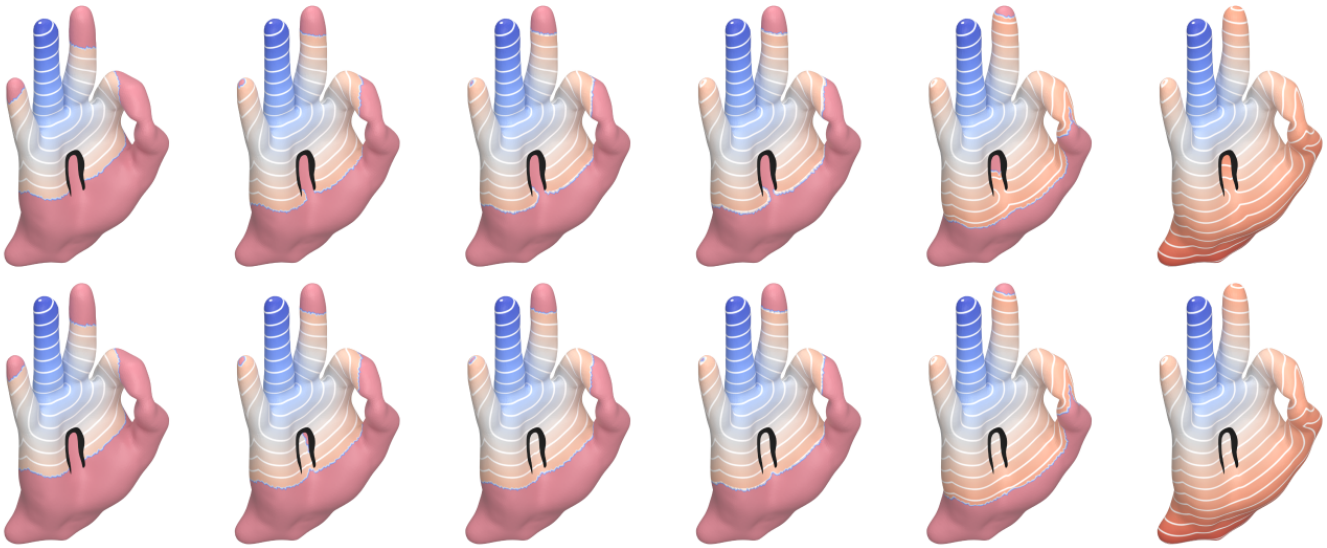


Figure 2: *Comparison of FMM (top) and our algorithm (bottom) in propagating geodesic distances through the same node, adhering to a fixed sequence.*

Algorithm 1: Geodesic Distance Computation for Models with Holes

Input: A model M with open boundaries and a source vertex s;
Output: Geodesic distance field $\mathcal{D}$ on M;
Apply the Fast Marching Method (FMM) to compute the initial geodesic distance field $\mathcal{D}_1$ on M;
Archive $\mathcal{D}_1$ as the reference geodesic distance field;
Initialize an empty priority queue Q, prioritized by $\lambda\mathcal{D}_1 + (1-\lambda)\mathcal{D}$;
Identify all vertices v_i adjacent to s, set $d(v_i) = \|e_{sv_i}\|$, where e_{sv_i} is the edge connecting s and v_i, and enqueue event h_i with the information of v_i into Q;
while *Q is not empty* **do**
  Dequeue the event h_i from Q;
  For each vertex v_j adjacent to v_i, perform forward propagation to update $d(s, v_j)$;
  if *v_i is a boundary vertex* **then**
    Compute the back propagation distance d_b;
    Define $\vec{n_1}$ as the forward propagation direction;
    Define $\vec{n_2}$ as the backward propagation direction;
    Define $\vec{n_3}$ as the distance field direction of the opposite triangle ;
    if $\vec{n_1} \cdot \vec{n_3} < \vec{n_2} \cdot \vec{n_3}$ **then**
      Update $d(s, v_j) = d_b$;
    end
  end
end

the distance as follows, where p_1 and p_2 are vertices of a triangle with known distances from the source $d(p_1)$ and $d(p_2)$, and q is the third vertex:

- **Case 1:** $d(q) = d(p_1) + \|p_1q\|$;
- **Case 2:** solve $d(q)$ using the condition $|\nabla d(q)| = 1$;
- **Case 3:** $d(q) = d(p_2) + \|p_2q\|$.

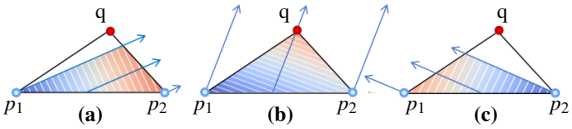


Figure 3: *q in (a) and (c) is not directly illuminated by the source point. The distance to q is determined by the geodesic paths from p_1 and p_2, respectively.*

As depicted in Figure 3, in Cases 1 and 3, vertex q lies within the "shadow" region. Consequently, the geodesic path to q must pass through either p_1 or p_2. In contrast, in Case 2, vertex p is directly "illuminated" by light from s, allowing for the application of the Eikonal equation to determine the distance.

3.3. Forward and backward propagation

The fast marching algorithm [SV00] leverages the intrinsic geometry of surfaces for computing geodesic distances, as these distances are independent of the embedding space. In this algorithm, each surface point acts simultaneously as both a receiver and a transmitter of distance information no matter open-boundary defects. A notable challenge arises when a geodesic path encounters a boundary point, requiring progression along the boundary curve. This often results in computed distances that are greater than those on the actual surface, as illustrated in Figure 3.

To tackle this challenge, we have refined the FMM at boundary points, which also determine the geodesic distances by solving the Eikonal equation:$|\nabla d(q)| = 1$. Boundary points are configured to act exclusively as receivers of distance information, effectively

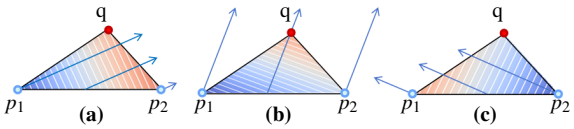


Figure 4: *If p_1 or p_2 is a point on the boundary (a)(c), our forward propagation strategy is modified such that the geodesic distance value at point p is also calculated using $|\nabla d(q)| = 1$.*

preventing further propagation of distances into the surface. This modification leads to a more accurate computation of geodesic distances, as demonstrated in Figure 4 (a) and (c).

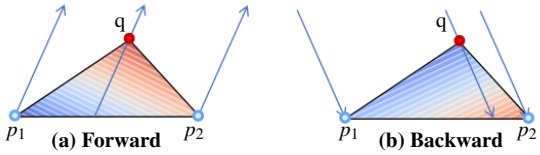


Figure 5: *The Eikonal equation $|\nabla d(q)| = 1$ yields two solutions, and the gradient are symmetric with respect to edges p_1p_2.*

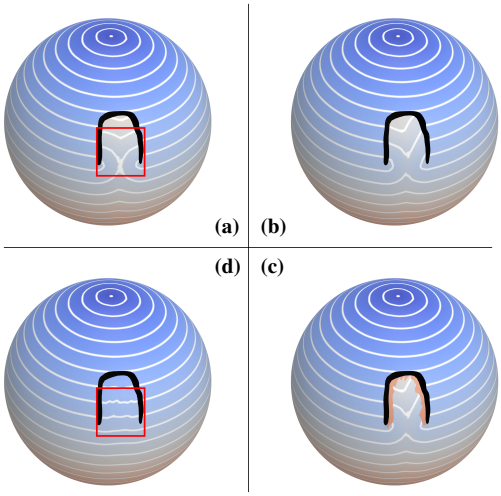


Figure 6: *(a) shows the distance field using the FMM on a broken sphere. (b) presents the distance field using a modified forward propagation at the boundary. (c) highlights the shadow region in orange, where backward propagation is allowed. (d) displays the distance field after applying bidirectional propagation, with corrections visible in the red-bordered area.*

Besides, we present bidirectional distance propagation, allowing the geodesic distance in the shadow region to be calculated based on the geodesic distance in the visible region, which may be farther away. Suppose we are considering the propagation in a triangle $\triangle\, p_1p_2q$, $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$with known distances from the source $d(p_1)$ and $d(p_2)$, the key criterion is to determine how the opposite vertex $q(x3, y3)$ gets its shortest distance d(q). According to the distance field being a linear field and the distance gradient equal to 1, we can easily derive the following equations:

$$\begin{aligned} ax_1 + by_1 + c &= d(p_1) \\ ax_2 + by_2 + c &= d(p_2) \\ a^2 + b^2 &= 1 \end{aligned} \tag{2}$$

As depicted in Figure 5 and Eq. 2, $d(q) = ax_3 + by_3 + c$ has two feasible solutions, denoting 'forward propagation' and 'backward propagation', respectively. When employing FMM to calculate geodesic distances, propagation typically proceeds from regions of lower to higher distances, thus directing the gradient forward. However, due to the presence of boundaries, within parts of the shadow region—as illustrated in Figure 6 (c), the orange region—it becomes feasible to use visible regions with known gradient directions as references for backward propagation. We adapt the FMM to alternate between 'forward propagation' and 'backward propagation'. The decision to use either 'forward' or 'backward propagation' is based on which option ensures that the gradients within the two adjacent triangles remain more coherent (as illustrated in Figure 7). It is important to note that the wavefront propagates from one triangle to one of its neighboring triangles.

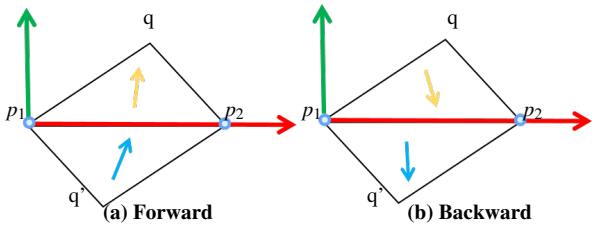


Figure 7: *In the $\triangle\, p_1p_2q$, the gradient of the distance field has two possible directions for propagation shown in yellow: forward and backward. The choice depends on the direction of the gradient, shown in blue, in the adjacent $\triangle\, p_1p_2q'$. In (a), the gradient in $\triangle\, p_1p_2q'$ propagates in the forward direction, thus the gradient direction in $\triangle\, p_1p_2q$ is chosen to be forward; (b) follows the same principle.*

3.4. Priority setting

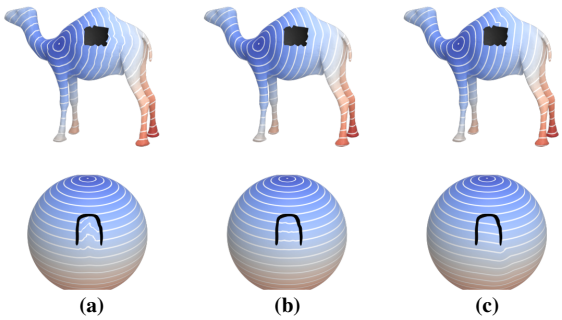


Figure 8: *Comparison of distance fields with different priority settings: (a) Effect of using FMM-calculated distances as priorities. (b) Distance field derived by partly referencing FMM priorities. (c) is over-relying on FMM priorities.*

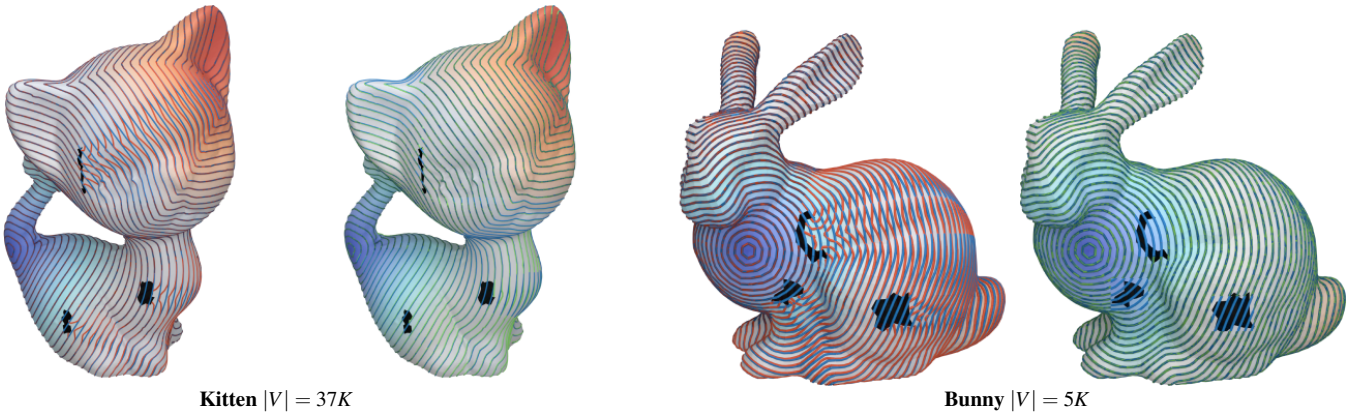


Figure 9: *To visualize the quality, we overlay the results with the groudtruth, a more precise solution computed on a no hole model. The red, green and blue curves are the results of FMM, our method and the baseline.*

We refine the assignment of priorities to ensure balanced distance propagation between the shadow and visible regions. Using computed distance values directly as priorities would disproportionately favor the shadow region, as illustrated in Figure 8 (c). To counter this, we propose using the original priorities from the visible region as a reference and introduce a parameter λ to achieve an equitable distribution of propagation priorities across both regions as Algorithm 1 shown. The need for this priority adjustment strategy is demonstrated in Figure 8.

4. Evaluation

4.1. Experimental setting

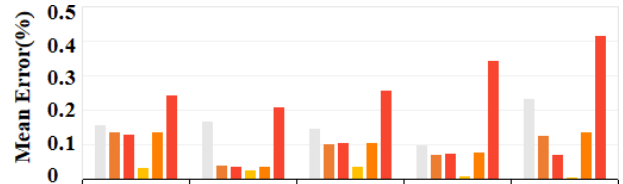
We test our algorithm on commonly used models in graphic and geometric modeling community. We implemented our algorithm in C++ on a 64-bit workstation that equipped with a 1.80 GHz Intel Core i5 CPU and 8 GB memory.

4.2. Accuracy

As discussed in Section 3.4, we introduced a parameter λ to balance distance propagation between shadow and visible regions. Tests were conducted to measure mean error across various λ values, as depicted in Figure 10. Results indicate that a $\lambda = 0.5$, blending equal parts FMM distance and our method, yields a geodesic distance closest to the ground truth.

We calculate the approximation error by comparing distances at points on a model with defects to exact distances, then computing the mean of all relative errors. This mean error effectively reflects discrepancies in the shadow region.

Note that results shown only use $\lambda = 0.5$ for subsequent experiments, as this value yields a low mean error. Our method computes nearly exact geodesic distances on triangular meshes, outperforming conventional algorithms that fail in areas with missing parts. For a visual comparison, see Figure 9 where our results are overlaid with the ground truth from a model without holes.



	Bunny 5K	Camel 9K	Bottle 10K	Bimada 16K	Kitten 20K
FMM	0.158%	0.168%	0.148%	0.098%	0.232%
λ=0	0.137%	0.041%	0.103%	0.071%	0.127%
λ=0.3	0.130%	0.037%	0.105%	0.075%	0.070%
λ=0.5	0.034%	0.026%	0.036%	0.009%	0.006%
λ=0.7	0.135%	0.038%	0.106%	0.077%	0.135%
λ=1	0.244%	0.210%	0.257%	0.344%	0.414%

Figure 10: *Comparison of mean error between the FMM and our method across various λ values. Our method in lighter orange significantly outperforms FMM in gray, with optimal performance at $\lambda = 0.5$.*

The parameter λ is influenced by the intrinsic geometry of the missing regions and the spatial distribution of sources. While certain models demonstrate improved performance with λ values either greater than or less than 0.5, our findings suggest that the overall results are generally insensitive to variations in λ. Consequently, we adopt 0.5 as the default value. However, it is acknowledged that in particularly challenging scenarios, fine-tuning of the λ parameter may be necessary.

4.3. Performance

The running time of our method is composed of two parts: the distance calculation using FMM and the bidirectional distance prop-

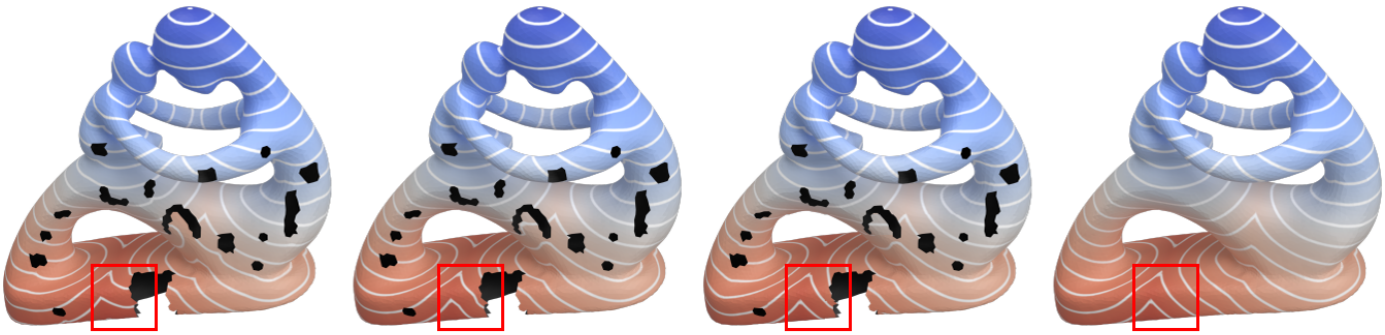


Figure 11: *Campen et al. [CK11] address ambiguities caused by substantial missing parts of the model, leading to geodesic distance fields that deviate significantly from the ground truth.*

agation. For an example of a bunny model with 9,996 faces, the timings for these two components are 1.4s and 0.56s, respectively, with λ set to 0.5 in these tests. Table 1 compares our algorithm on a hole model against the vertex-oriented triangle propagation (VTP) exact algorithm [DHF*21], Campen et al.'s [CK11], and the FMM [KS98]. Campen et al.'s [CK11] presents a computational approach that enables the estimation of geodesic distances and paths on polygonal meshes with inherent defects like holes and non-manifold configurations. This method also offers a defect-tolerant alternative to traditional mesh repair processes, which are often complex and fraught with ambiguities. By transforming the input mesh into a cubical complex and applying morphological operations, which can effectively handle gaps and holes as specified by the user. Utilizing an enhanced FMM, it computes a distance field that approximates geodesic distances. This implies that Campen et al.'s approach needs sufficiently many cells to predict the distance, which requires more memory and time as Table 1 shown. While the method is generally applicable, it may not resolve ambiguities due to large missing parts of the model as Figure 11 shown. Our method computes distances with significantly lower mean error than the other methods and demonstrates consistent performance across different mesh resolutions as shown in Figure 12.

We conducted experiments on various models including Happy Buddha, Lucy, Bottle, and Bunny. For each model, we created various holes randomly, in smooth regions and in regions with geometric features. From the visualizations in Figure 13, by focusing on the shadow regions altered through bidirectional distance propagation and examining the texture mapping of the resulting geodesic distance field, it is evident that our results closely resemble the ground truth.

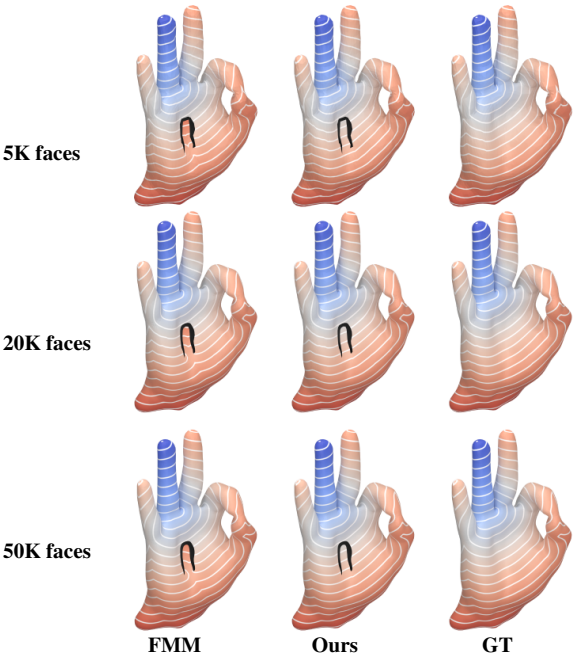


Figure 12: *Our result is isometry invariant and insensitive to the resolution.*

5. Conclusion

In this study, we presented a novel algorithm for estimating geodesic distances on 3D shapes with open boundaries. By introducing a bidirectional distance propagation mechanism and adjusting propagation priorities, our approach significantly enhances precision, particularly at boundary points which act solely as receivers to halt further propagation. This method not only improves the accuracy of geodesic distance calculations but also represents the first intrinsic method tailored to 3D shapes with open boundaries. It operates independently of external influences, relying only on the model's internal geometric features to deliver more precise results.

Our approach has significant implications for fields such as computer graphics, image processing, computational geometry, and computer vision, where accurate distance measurements are crucial. By refining the existing Fast Marching Method (FMM), our algorithm provides a marked improvement in accuracy compared to other available methods.

However, despite its advantages in accuracy, the current implementation of our algorithm does not surpass the original FMM in

Table 1: *Runtime performance across five representative models of varying resolutions is detailed, where Time represents the total time, encompassing both FMM and bidirectional propagation durations, and RAM indicates both average and peak memory usage. The value of E_{avg} denotes the mean error.*

Model	Vtp			Campen et al.'s			FMM			Ours		
	Time/s	RAM/MB	E_{avg}	Time/s	RAM/MB	E_{avg}	Time/s	RAM/MB	E_{avg}	Time/s	RAM/MB	E_{avg}
Bunny (10K faces)	0.97	11/11	0.148%	2.71	20/23	0.057%	1.40	11/14	0.158%	1.4+**0.56**	11/14	**0.034%**
Camel (18K faces)	1.87	18/18	0.056%	3.87	30/37	0.021%	1.80	17/28	0.068%	1.8+**0.63**	18/28	**0.016%**
Bottle (20K faces)	2.03	20/20	0.136%	3.72	42/46	**0.021%**	1.81	17/28	0.149%	1.81+**0.62**	18/28	0.026%
Bimba (32K faces)	2.98	28/28	0.090%	4.67	49/51	0.021%	2.88	25/41	0.098%	2.88+**0.98**	25/41	**0.009%**
Kitten (38K faces)	3.26	34/34	0.130%	5.05	51/57	0.027%	4.18	29/47	0.132%	4.18+**0.71**	29/47	**0.013%**

Figure 13: *Our results with* $\lambda = 0.5$. *Our results and the results of the fast marching method are right and left, respectively. Top Left: Buddha* $|V| = 50K$, $\varepsilon = 0.015\%$, *3 holes. Top Right: Lucy* $|V| = 262K$, $\varepsilon = 0.008\%$, *3 holes. Bottom Left: Bottle* $|V| = 10K$, $\varepsilon = 0.026\%$, *3 holes. Bottom Right: Bunny* $|V| = 5K$, $\varepsilon = 0.034\%$, *2 holes.*

terms of time efficiency. This limitation points to a critical area for future research. Subsequent efforts should focus on optimizing the algorithm's computational efficiency to broaden its applicability and utility in real-time systems. Enhancing time efficiency will not only make the algorithm more practical for extensive usage but also expand its potential applications across various disciplines.

References

[AFH20] ADIKUSUMA Y. Y., FANG Z., HE Y.: Fast construction of discrete geodesic graphs. *ACM Transactions on Graphics (TOG) 39* (2020), 1 – 14. URL: https://api.semanticscholar.org/CorpusID:216516491. 2

[ALMS98] ALEKSANDROV L., LANTHIER M., MAHESHWARI A., SACK J.-R.: An ε—approximation algorithm for weighted shortest paths on polyhedral surfaces. In *Scandinavian Workshop on Algorithm Theory* (1998), Springer, pp. 11–22. 2

[CH90] CHEN J., HAN Y.: Shortest paths on a polyhedron. In *Proceedings of the sixth annual symposium on Computational geometry* (1990), ACM, pp. 360–369. 2

[CK11] CAMPEN M., KOBBELT L.: Walking on broken mesh: Defect-tolerant geodesic distances and parameterizations. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 623–632. 3, 7

[CL06] COIFMAN R. R., LAFON S.: Diffusion maps. *Applied and computational harmonic analysis 21*, 1 (2006), 5–30. 3

[CWW13] CRANE K., WEISCHEDEL C., WARDETZKY M.: Geodesics

in heat: A new approach to computing distance based on heat flow. *ACM Transactions on Graphics (TOG) 32*, 5 (2013), 152. 1, 3

[CZX*20] CAO L., ZHAO J., XU J., CHEN S., LIU G., XIN S., ZHOU Y., HE Y.: Computing smooth quasi-geodesic distance field (QGDF) with quadratic programming. *Computer-Aided Design 127* (2020). 3

[DDG*18] DAS A., DATTA S., GKIOXARI G., LEE S., PARIKH D., BATRA D.: Embodied question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 1–10. URL: https://api.semanticscholar.org/CorpusID:35985986. 1

[DHF*21] DU J., HE Y., FANG Z., MENG W., XIN S.: On the vertex-oriented triangle propagation (vtp) algorithm: Parallelization and approximation. *Comput. Aided Des. 130* (2021), 102943. URL: https://api.semanticscholar.org/CorpusID:224859921. 7

[FC24] FENG N., CRANE K.: A heat method for generalized signed distance. *ACM Trans. Graph. 43*, 4 (2024). 3

[FPRS07] FOUSS F., PIROTTE A., RENDERS J. M., SAERENS M.: Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering 19*, 3 (2007), 355–369. 3

[KS98] KIMMEL R., SETHIAN J. A.: Computing geodesic paths on manifolds. *Proceedings of the national academy of Sciences 95*, 15 (1998), 8431–8435. 2, 7

[LAKH22] LIU R., AIGERMAN N., KIM V. G., HANOCKA R.: Da wand: Distortion-aware selection using neural mesh parameterization. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 16739–16749. URL: https://api.semanticscholar.org/CorpusID:254591602. 1

[Liu13] LIU Y.-J.: Exact geodesic metric in 2-manifold triangle meshes using edge-based data structures. *Computer-Aided Design 45*, 3 (2013), 695–704. 2

[LRF10] LIPMAN Y., RUSTAMOV R. M., FUNKHOUSER T. A.: Biharmonic distance. *ACM Transactions on Graphics (TOG) 29*, 3 (2010), 27. 3

[MMP87] MITCHELL J. S., MOUNT D. M., PAPADIMITRIOU C. H.: The discrete geodesic problem. *SIAM Journal on Computing 16*, 4 (1987), 647–668. 2

[MP23] MANCINELLI C., PUPPO E.: Computing the riemannian center of mass on meshes. *Computer Aided Geometric Design 103* (2023), 102203. URL: https://www.sciencedirect.com/science/article/pii/S0167839623000353, doi:https://doi.org/10.1016/j.cagd.2023.102203. 2

[PZP*20] POTAMIAS R. A., ZHENG J., PLOUMPIS S., BOURITSAS G., VERVERAS E., ZAFEIRIOU S.: Learning to generate customized dynamic 3d facial expressions. *ArXiv abs/2007.09805* (2020). URL: https://api.semanticscholar.org/CorpusID:220647347. 1

[QHY*16] QIN Y., HAN X., YU H., YU Y., ZHANG J.: Fast and exact discrete geodesic computation based on triangle-oriented wavefront propagation. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 125. 2

[Set96] SETHIAN J. A.: A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences 93*, 4 (1996), 1591–1595. 2

[SSK*05] SURAZHSKY V., SURAZHSKY T., KIRSANOV D., GORTLER S. J., HOPPE H.: Fast exact and approximate geodesics on meshes. *ACM transactions on graphics (TOG) 24*, 3 (2005), 553–560. 2

[SV00] SETHIAN J. A., VLADIMIRSKY A.: Fast methods for the eikonal and related hamilton–jacobi equations on unstructured meshes. *Proceedings of the National Academy of Sciences 97*, 11 (2000), 5699–5703. 1, 2, 4

[TZD*19] TAO J., ZHANG J., DENG B., FANG Z., PENG Y., HE Y.: Parallel and scalable heat methods for geodesic distance computation. *IEEE transactions on pattern analysis and machine intelligence* (2019). 3

[WDB*08] WEBER O., DEVIR Y. S., BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Parallel algorithms for approximation of distance maps on parametric surfaces. *ACM Transactions on Graphics (TOG) 27*, 4 (2008), 1–16. 3

[WFW*17] WANG X., FANG Z., WU J., XIN S.-Q., HE Y.: Discrete geodesic graph (dgg) for computing geodesic distances on polyhedral surfaces. *Computer Aided Geometric Design 52* (2017), 262–284. 2

[XQYH12] XIN S.-Q., QUYNH D. T., YING X., HE Y.: A global algorithm to compute defect-tolerant geodesic distance. In *SIGGRAPH Asia 2012 Technical Briefs* (2012), ACM, p. 23. 3

[XW09] XIN S.-Q., WANG G.-J.: Improving chen and han's algorithm on the discrete geodesic problem. *ACM Transactions on Graphics (TOG) 28*, 4 (2009), 104. 2

[XWL*15] XU C., WANG T. Y., LIU Y.-J., LIU L., HE Y.: Fast wavefront propagation (FWP) for computing exact geodesic distances on meshes. *IEEE transactions on visualization and computer graphics 21*, 7 (2015), 822–834. 2

[XYH12] XIN S.-Q., YING X., HE Y.: Constant-time all-pairs geodesic distance query on triangle meshes. In *Proceedings of the ACM SIGGRAPH symposium on interactive 3D graphics and games* (2012), ACM, pp. 31–38. 2

[YWH13] YING X., WANG X., HE Y.: Saddle vertex graph (SVG): a novel solution to the discrete geodesic problem. *ACM Transactions on Graphics (TOG) 32*, 6 (2013), 170. 2