

Geodesic Paths and Distances

Report on A Survey of Algorithms for Geodesic Paths and Distances

Matthieu Pierre Boyer*
École Normale Supérieure
Paris, France
matthieu.boyer@ens.fr

Antoine Groudiev*
École Normale Supérieure
Paris, France
antoine.groudiev@ens.fr

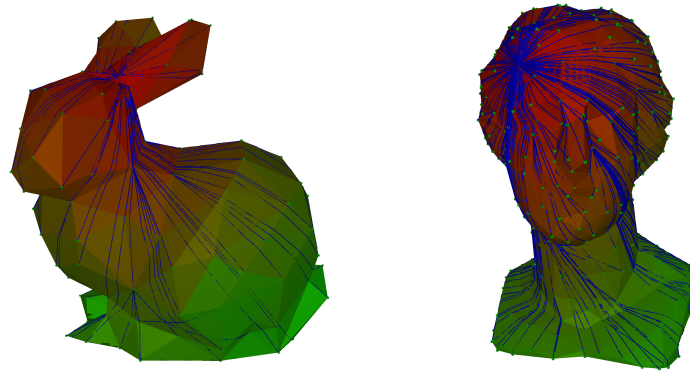


Figure 1: Geodesic paths on meshes, computed with our implementation of the Improved Chen-Han algorithm. Faces are colored from red to green according to the distance of their barycenter to the source point. The geodesic path to each vertex is drawn in blue.

ABSTRACT

In this report, we investigate different methods to compute shortest-paths on meshed 2-manifolds embedded in \mathbb{R}^3 , based on [CLPQ20]. We will most notably compare different types of methods, either coming from the resolution of PDEs on the manifold, or through the unfolding of the embedding to \mathbb{R}^2 . We also provide from-scratch implementations of three methods, namely the heat method, fast marching, and the Improved Chen-Han algorithm, and compare their performances on several datasets.

INTRODUCTION

TODO: motivate the problem, give applications, present the structure of the report.

After brief mathematical reminders in section 1, we present two methods based on the resolution of Partial Differential Equations (PDEs) in section 2. Then, in section 3, we discuss computational geometry methods, focusing on the Improved Chen-Han algorithm. Finally, we present experiments and results in section 4, and conclude in the final section.

1 MATHEMATICAL REMINDERS

Formally, a 2-manifold (without boundary) is a topological space in which all points have neighborhoods homeomorphic to disks (without boundary) in \mathbb{R}^2 . Intuitively, this means that zooming enough on any point of the manifold will make it look like a flat surface. In Figure 2, we can see that some meshes are not manifolds, as some points are locally non-flat.

*Both authors contributed equally.

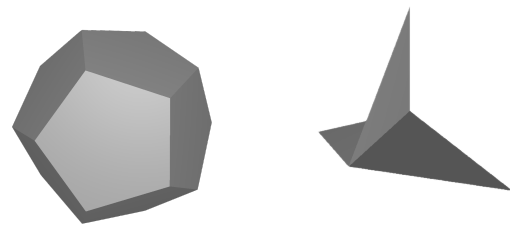


Figure 2: Examples of a 2-manifold (left) and non-manifold (right) mesh. (Images generated from our custom viewer.)

In our context, we will be given a 2-manifold already meshed¹, that is, a finite set $\mathbb{V} \subseteq \mathbb{R}^3$ of vertices (of cardinal $n_{\mathbb{V}}$) and a finite set $\mathcal{F} \subseteq \llbracket 1, n_{\mathcal{F}} \rrbracket^3$ of faces, given by the indices of the associated vertices. The edges E of the manifold are given by any subsets of size two of a face $f \in \mathcal{F}$. Because we cannot have continuous functions, functions on the manifold will be represented as functions on \mathbb{V} , on E or on \mathcal{F} . As such, we will take any function and interpolate it linearly on each face, giving us a piecewise-linear function.

A path on a piecewise-linear manifold can then be understood as interpolating on pieces of the manifold, or directly computing the curves on the mesh. The quality of the approximation by the mesh of the 2-manifold will never be taken into account in the quality results.

¹[CLPQ20] gives a few methods to create such a meshing. In particular, our implementations of the methods are restricted without loss of generality to triangular faces; more general meshes can be considered through triangulation techniques.

Reformuler
la
phrase
ci-
dessous

2 PDE-BASED METHODS

In this section we will study methods inspired by Partial Differential Equations (PDE) that arise from models of physical phenomena. Indeed, many physical phenomena propagate along the surfaces over time, and dissipate over space, thus allowing us to retrieve geodesics from solutions to the equations.

2.1 General Theory

Consider the parabolic heat equation

$$\frac{d}{dt}u_t = \Delta u_t.$$

Here, u_t is the temperature profile at time t and Δ is the laplacian operator (or divergence of the gradient operator). However, on a piecewise-linear manifold, because functions on vertices are interpolated linearly to become functions on the manifold, the gradient is piecewise-constant and can be computed explicitly from the values at each vertex. Consider a face $f \in \mathcal{F}$ with vertices $p_i, p_j, p_k \in \mathbb{R}^3$. Let $e_1 = p_j - p_i$ and $e_2 = p_k - p_i$. The face normal is

$$n_f = \frac{e_1 \times e_2}{|e_1 \times e_2|},$$

where \times is the cross product in \mathbb{R}^3 and thus the gradient, being perpendicular to level curves is

$$\nabla u|_f = \frac{1}{2A_f} \sum_{l \in f} u_l (n_f \times e_l),$$

where $A_f = \frac{1}{2} |e_1 \times e_2|$ is the area of face f and e_l is the **opposite** to vertex l . Note that we can represent the gradient as a matrix $G \in \mathbb{R}^{n_V \times 3n_{\mathcal{F}}}$, although this representation is really inefficient for practical computation. The definition of the divergence then comes from the Gauss-Ostrogradski theorem by intregation by parts

$$(\nabla \cdot X)_i = \frac{1}{A_i} \sum_{f \ni i} \sum_{e \in f} \frac{1}{2} \cot(\alpha_e^f) \langle X_f, e \rangle,$$

where A_i is the Voronoi area associated with vertex i and α_e^f is the angle at the vertex opposite to edge e in f .

Finally, we can define the Laplace-Beltrami operator (the piecewise-linear version of the continuous laplacian) as $\Delta = (\nabla \cdot) \circ \nabla : \mathbb{V} \rightarrow \mathbb{R}$ or simply

$$(\Delta u)_i = \frac{1}{2A_i} \sum_{e=(i,j)} (\cot \alpha_{i,j} + \cot \beta_{i,j}) (u_i - u_j),$$

where $\alpha_{i,j}, \beta_{i,j}$ are the two angles opposite edge (i, j) . One could then see the Laplace-Beltrami operator as a matrix $L \in \mathbb{R}^{n_V \times n_V}$.

After the spatial discretization of the laplace operator we just described, operating a time discretization in a single backward Euler step for some fixed time t will give us approximate solutions to the equation. If we want to find the distance maps from some set $X \subseteq \mathbb{V}$, we simply solve the linear equation associated to the continuous equation we need to solve.

2.2 Implementations

In our implementation, we compared multiple methods based on physical phenomena allowing to trace geodesics, or at least geodesic-like curves. Indeed, not all of those compute the true geodesic distance, but some sense of distance that can be drawn and integrated to find shortest paths.

Heat Method. This method is based on the heat equation $\nabla u_t = \frac{d}{dt}u_0$, which models the evolution of temperature profiles u_t in time in a given material, which here will be our surface, from the initial profil u_0 . It can be derived We discretize it as:

$$(\text{id} - t\Delta)u_t = u_0.$$

Retrieving the true geodesic distance can then be done by first normalizing the gradient $X = -\nabla u_t / |\nabla u_t|$ of the solution of the above equation that points along geodesics, then solving the Poisson equation $\Delta \phi = \nabla \cdot X$ to retrieve the true distance function. This fact comes from the Varadhan formula $\phi(x, y) = \lim_{t \rightarrow 0} \sqrt{-4t \log k_{t,x}(y)}$.

[CW17] suggest that the proper value of t to use for computations here is around h^2 with h the mean spacing between adjacent nodes, as $h\Delta$ is invariant with respect to scale.

Poisson Equations. The equations in this paragraph all allow to draw geodesics-like curves, although they do not give the actual metric on the manifold. They are all based on the Poisson equation

$$\Delta u = u_0,$$

which can be derived, for example, from the Maxwell equations to compute the electrostatic potential along a charge distribution, or from the momentum equation to compute the pressure in an incompressible fluid given its velocity.

Wavefront Propagation. The hyperbolic wavefront propagation equation

$$\frac{d^2}{dt^2}u_t = c^2 \Delta u_t, \text{ or, } \square u = 0,$$

where \square is the d'Alembert operator, models the propagation of a wave in a material, which will again be our surface here. It arises for example the response of the surface to some elastic deformation u considering the stress tensor $T = E \nabla u$ with E the homogenous modulus of elasticity, and considering the inertial force $\rho \frac{\partial^2 u}{\partial t^2}$ caused by the local acceleration.

3 COMPUTATIONAL GEOMETRY METHODS

While PDE-based methods provide a general framework to compute geodesic distances on arbitrary, possibly non-meshed manifolds, we can leverage the fact that our manifold is meshed to design more efficient algorithms. A whole class of such algorithms, arising from computational geometry, can be classified according to two main characteristics.

Global vs. local. Global methods compute globally optimal paths, up to numerical precision. Local methods start from an initial non-geodesic path and iteratively improve it until convergence to a local minimum.

3.4 Implementation details

We implemented the Improved Chen-Hand algorithm from scratch in Rust, following the description from [XW09]. We provide here a short description of some implementation details, data structures used, as well as some difficulties we encountered and limitations of our implementation.

3.4.1 Data structures. As other methods, the input of the algorithm is a 2-manifold represented as a triangle mesh, that is a set of vertices \mathbb{V} and a set of faces (3-tuples) \mathcal{F} . However, we also need to represent edges E explicitly, as windows are defined on edges. We therefore employ an edge data structure, storing for each edge both geometric information (length) and topological information (adjacent faces, vertices, ...). Notably, we precompute the twin edges for each edge, which is repeatedly invoked during window propagation.

The main algorithm relies on a priority queue of windows, ordered by the minimum distance from the source to any point in the window. We implement this priority queue as a binary heap, using Rust's standard library implementation. Windows are represented as structs containing all the necessary information for propagation, as described above. Information about the vertices, such as the edge from which the shortest path to the vertex originates, is also stored during the main loop of the algorithm to enable path reconstruction later on.

3.4.2 Limitations. Two main limitations of our implementation are worth mentioning: paths along the edges fail to be computed successfully, and some vertices may not receive a distance value at the end of the algorithm when the geometry is particularly complex.

4 EXPERIMENTS AND RESULTS

TODO: do experiments, provide results, compare methods.

TODO: highlight the limitations of each method, and the trade-offs.

CONCLUSION

REFERENCES

- [CH90] Jindong Chen and Yijie Han. Shortest paths on a polyhedron. In *Proceedings of the sixth annual symposium on Computational geometry*, pages 360–369, 1990.
- [CLPQ20] Keenan Crane, Marco Livesu, Enrico Puppo, and Yipeng Qin. A survey of algorithms for geodesic paths and distances, 2020.
- [CWW17] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. The heat method for distance computation. *Commun. ACM*, 60(11):90–99, October 2017.
- [MMP87] Joseph SB Mitchell, David M Mount, and Christos H Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.
- [XW09] Shi-Qing Xin and Guo-Jin Wang. Improving chen and han's algorithm on the discrete geodesic problem. *ACM Transactions on Graphics (TOG)*, 28(4):1–8, 2009.