# TITLE

**Abstract**

In this paper, we present a new approach to geodesic computation on both point clouds and polyhedral surface. Our method consists of three parts: Dijkstra’s algorithm geodesic computation based on Cartesian grid, geodesic curve regression and geodesic path pull-back. In the first part, we have developed a topologic independent mechanism for point based data form in the formulation. This allows both polyhedral surface and point clouds to be stored in the unify structure. Differing from most previous methods, by which the computed geodesics algorithms heavily depends on the data type of the inputs. The second and third part of our approach contributes largely to improve the efficiency for approximate geodesic computation for single source single destination geodesic problem. This is achieved with the introduction of geodesic curve regression. The presented algorithm is in general computationally efficient, which is easily expanded to all pair computation, i.e. all sources to all destinations computation, only incurring an approximate complexity of *O*(),.

Keywords. Geodesic computation, polygon mesh, point clouds, geodesic regression

# Introduction

In mathematics, a geodesic is a generalization of the notion of the shortest path in any neighbourhoods in the curved spaces. In terms of generalization of Euclidean straight line, on implicit surface, a geodesic is a curve on the surface which at every point on the curve, the geodesic curvature is zero. Particularly in differential geometry, geodesics play a very important role. The extensive use of geodesic computation has made it become a common operation in many computer graphic related applications. Ranging from computer aided design to machine learning, from medical image analysis to computer animation.

Manual modelled polygon mesh has been a mainstream for 3D shape representative for decades, however by the fast development of shape acquisition device, such as MRI, laser scanner, with application in medical visualizations, document archival, geoscience and designing. The data provides by those shape acquisition device usually in form of unorganized point range data (point clouds) to representing shapes. Geodesics computation on mash-based data has been intensively studied for decades, many efficient algorithm has been developed [8][13][25]~[29]. However, geodesics computation on point clouds rarely dressed [14][30][34]. The current method of compute geodesic on point clouds usually involves an intermediate step which is to fit a surface to it, mostly an implicit-surface, in which the computation accuracy will be affected and computational complexity will be increased.

In this paper, we present a novel algorithm for computing geodesics directly on point clouds and also be able to process polyhedral surface. The existing methods for computing geodesics can be categorised into mesh-based and non-mesh based method. By the reason of discrete form is much easier to edit and render in computer graphic area, mash-based method has been a hot research topic for many years -. Compare to the mesh-based method, only few researchers have explored the computation of geodesics on point clouds. One reason is point cloud is a unorganized data set which does not have connection relationship between each data element required by mesh-based methods [27][32]. Rather than fit a surface onto point data and compute geodesic on the tessellation of that surface. Our main idea is to create a rough geodesic path on the Cartesian grid and pull it back onto the point clouds and refine the curve by exam the geodesic curvature of the curve at sampled point on the curve. Our contributions are as below:

1. Multi-data-type adaptability and easy to control error tolerance: well-known algorithm like Dijkstra’s , MMP and FMM can only work on mesh-based data such as polyhedron surface. They cannot process the unorganized point data like 3D scanner raw data or sampled implicit-surface data. Because all three algorithms mentioned above are Dijkstra’s like algorithm which heavily related to the connection relationship in the graph structure data.
2. High accuracy geodesics: In general, the geodesic computation developed previously is calculated on a fitted implicit surface of the point cloud and compute the geodesic on its tessellation. Although the MMP algorithm and FMM algorithm are able to provide an accurate while efficient result, however, the computation error introduced by fitted surface cannot be eliminated. The geodesic path still lies on an offsetting surface of actual point data. Also to determine the offset of the fitted surface often involves intuitive manipulations. In section 4, we will show how to gain a smooth geodesic path on the “surface” of the point clouds.
3. Low computational complexity: algorithms like MMP and FMM normally have an approximate complexity of *O*(*N*2 log *N*) for all pair geodesics. For point cloud algorithm , the implicit-surface extraction process adds more computation on top of the mesh-based approach which used to calculate geodesic path. In our method, we will show that the implementation of our approach to all pair geodesic computation does not appear the exponential increase of the time complexity throughout the increase of the number of the point in the input point cloud data.

This paper is constructed as follow: in section 2, we briefly investigate the research background and related work, in section 3 we present our algorithm in detail. Section 4 is the detail description of the implementation of our algorithm and discusses the experiment result. In section 5, the conclusion will be given.

# Previous Work

Geodesic problem is to determine the shortest path between two point *s* and *d* on a polyhedral surface or a smooth surface. There are many methods has been developed on polyhedral surface. Dijkstra's algorithm was the first method to solve the shortest path problem for a graph with nonnegative edge path costs. This method has built the foundation of solving the shortest path problem. And after it, many researches was developed their method based on it and extend from 2D graph to 3D surface. Sharir and Schorr were the first team to extend the Dijkstra's algorithm into 3D dimension with a time complexity *O* ( log *n*). However their method can only solve the problem on a convex polyhedral surface. MMP algorithm inherits the paradigm of Dijkstra’s algorithm they partitions each mesh edge into a set of windows, those windows are propagated throughout the entire mesh by using a “continuous Dijkstra” like expanding wavefront method. They have improved their time complexity to *O* ( log *n*). Rather than building a subdivision of the polyhedral surface and compute the shortest path edge sequence, Surazhsky implemented MMP algorithm in 2005 and they extend the MMP algorithm with a bounded error approximation algorithm which has improved computationally efficient. their experiments shows their algorithm runs significantly faster and cost far less memory then Chan and Hans’ , also their approximate algorithm was able to achieve better accuracy by costing similar running time to FM method. In 1990s Chen and Han introduced an non-Dijkstra-like exact geodesic method based on a key observation “one angle one split” with time complexity of *O* () which solve the problem directly on the polyhedral surface. It consists two parts calculation, in the first part, the shortest path form given source to each vertex on the mesh is computed and a set of windows which contains the information about the shortest path from that given source point to points on each edges. In the next part, those windows are used to compute the decomposition of the surface and the shortest path to any destination point on the polyhedral surface can be reported. Later, Xin and Wang proposed an improved Chan and Han algorithm by introducing a method with less windows computation. Although their algorithm remains time complexity of *O* ( log *n*), according to their experiment report, it outperforms Chen and Han’s algorithm , also, the MMP algorithm and their implementation cast far less memory space than those two algorithm.

Point clouds data is a kind of typical output data from the 3D shape acquisition device. Geodesic has a very range of application as mentioned above, by the fast expanding range of the usage of the 3D shape acquisition device, to calculate the geodesic on the point cloud data raised on many fields i.e. surface simplification[15], surface reconstruction based on point sets [6] [18], manipulation on point sampled surface[17]. To compute the geodesic problem on point clouds, there are few research has considered this problem. Memoli and Sapiro [19] proposed a method which create an offset band of the point cloud and construct a 3D Cartesian grid inside the band. Then the FMM algorithm [7] is used to calculate the approximate geodesic path on the grid. Klein and Zachmann [24] introduced a method that calculates the geodesic on the proximity graph based on the point cloud. In their paper, the proximity graph was generated by using Delaunay graph or sphere-of-influence graph. Hofer and Pottmann [22] introduced a method of compute geodesic as energy minimizing piecewise curve which is constrained on a MLS surface. Although their method is efficient on high dimensional data, the accuracy of the geodesic path is depends on the nosie of the construction of the MLS surface. Ruggeri et al [30]. proposed an approximating method of compute geodesic on point set surface, the geodesic curve was constructed by using an energy minimization function to define a piecewise linear approximations of the geodesic curve. The initial curve was created by using Dijkstra’s algorithm [1] then the curve was refined with minimizing its energy function. However, their experiment shows the computational is too high for large numbers of geodesic computation requirement.

Our approach is inspired by Memoli and Sapiro [19], instead of compute the geodesic on the Cartesian grid fitted to the point clouds, our algorithm compute the geodesic curve directly on the point clouds.

# Geodesic Computation On Point Cloud

# Implementation

# Conclusions

**Reference**

1. Dijkstra, E.W., A note on two problems in connection with graphs. Numerische Mathematik., 1959: p. 83–89.
2. Sharir, M. and A. Schorr, On shortest paths in polyhedral spaces, in Proceedings of the sixteenth annual ACM symposium on Theory of computing., 1984, ACM. p. 144-153.
3. Mitchell, J.S.B., D.M. Mount, and C.H. Papadimitriou, The discrete geodesic problem. SIAM J. Comput., 1987. 16(4): p. 647-668.
4. Chen, J. and Y. Han, Shortest paths on a polyhedron, in Proceedings of the sixth annual symposium on Computational geometry., 1990, ACM: Berkley, California, United States. p. 360-369.
5. Lanthier, M., et al., Approximating weighted shortest paths on polyhedral surfaces, in Proceedings of the thirteenth annual symposium on Computational geometry., 1997, ACM: Nice, France. p. 485-486.
6. Floater, M. and A. Iske, Thinning algorithms for scattered data interpolation. BIT Numerical Mathematics, 1998. 38(4): p. 705-720.
7. Sethian, J., Level Set Methods and Fast Marching Methods., 1999, Cambridge University Press.
8. Hershberger, J. and S. Suri, An Optimal Algorithm for Euclidean Shortest Paths in the Plane. SIAM J. Comput., 1999. 28(6): p. 2215-2256.
9. Kapoor, S., Efficient computation of geodesic shortest paths, in Proceedings of the thirty-first annual ACM symposium on Theory of computing., 1999, ACM: Atlanta, Georgia, United States. p. 770-779.
10. Kaneva, B., An Implementation of Chen & Han's Shortest Paths Algorithm. 2000: Smith College, Northampton, Mass.
11. Agarwal, P.K., S. Har-Peled, and M. Karia, Computing approximate shortest paths on convex polytopes, in Proceedings of the sixteenth annual symposium on Computational geometry., 2000, ACM: Clear Water Bay, Kowloon, Hong Kong. p. 270-279.
12. Frisken, S.F., et al., Adaptively sampled distance fields: a general representation of shape for computer graphics, in Proceedings of the 27th annual conference on Computer graphics and interactive techniques 2000, ACM Press/Addison-Wesley Publishing Co. p. 249-254.
13. Kanai, T. and H. Suzuki, Approximate Shortest Path on Polyhedral Surface Based on Selective Refinement of the Discrete Graph and Its Applications, in Proceedings of the Geometric Modeling and Processing 2000, IEEE Computer Society. p. 241.
14. Boissonnat, J.-D. and F. Cazals, Coarse-to-fine surface simplification with geometric guarantees. Computer Graphics Forum, 2001. 20(3): p. 490-499.
15. Dey, T., J. Giesen, and J. Hudson. Decimating Samples for Mesh Simplification. in Proc. 13th Canadian Conf. Comput. Geom. 2001.
16. Mémoli, F. and G. Sapiro, Fast Computation of Weighted Distance Functions and Geodesics on Implicit Hyper-Surfaces. Journal of Computational Physics, 2001. 173(2): p. 730-764.
17. Pauly, M., M. Gross, and L.P. Kobbelt, Efficient simplification of point-sampled surfaces, in Proceedings of the conference on Visualization 2002, IEEE Computer Society: Boston, Massachusetts. p. 163-170.
18. Dyn, N., M.S. Floater, and A. Iske, Adaptive thinning for bivariate scattered data. J. Comput. Appl. Math., 2002. 145(2): p. 505-517.
19. Memoli, F. and G. Sapiro. Distance functions and geodesics on points clouds. in IEEE Workshop on Variational Geometric and Level Set Methods in Computer Vision. 2003.
20. Katz, S. and A. Tal, Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Trans. Graph., 2003. 22(3): p. 954-961.
21. Ravi Kumar, G.V.V., et al., Geodesic curve computations on surfaces. Computer Aided Geometric Design, 2003. 20(2): p. 119-133.
22. Hofer, M. and H. Pottmann, Energy-minimizing splines in manifolds. ACM Trans. Graph., 2004. 23(3): p. 284-293.
23. Martinez, D., L. Velho, and P.C. Carvalho, Geodesic Paths on Triangular Meshes, in Proceedings of the Computer Graphics and Image Processing, XVII Brazilian Symposium., 2004, IEEE Computer Society. p. 210-217.
24. Klein, J. and G. Zachmann, Point cloud surfaces using geometric proximity graphs. Comput. Graph., 2004. 28(6): p. 839-850.
25. Goldberg, A.V. and C. Harrelson, Computing the shortest path: A\* search meets graph theory, in Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms., 2005, Society for Industrial and Applied Mathematics: Vancouver, British Columbia. p. 156-165.
26. Martínez, D., L. Velho, and P.C. Carvalho, Computing geodesics on triangular meshes. Computers &amp; Graphics, 2005. 29(5): p. 667-675.
27. Surazhsky, V., et al., Fast exact and approximate geodesics on meshes. ACM Trans. Graph., 2005. 24(3): p. 553-560.
28. Peyre, G. and L. Cohen, Geodesic Remeshing Using Front Propagation. Int. J. Comput. Vision, 2006. 69(1): p. 145-156.
29. Polthier, K. and M. Schmies, Straightest geodesics on polyhedral surfaces, in ACM SIGGRAPH 2006 Courses2006, ACM: Boston, Massachusetts. p. 30-38.
30. Ruggeri, M.R., et al. Approximating Geodesics on Point Set Surfaces. in Proceedings of SPBG. 2006.
31. Ying, L. and E.J. Candès, Fast geodesics computation with the phase flow method. Journal of Computational Physics, 2006. 220(1): p. 6-18.
32. Xin, S.-Q. and G.-J. Wang, Improving Chen and Han's algorithm on the discrete geodesic problem. ACM Trans. Graph., 2009. 28(4): p. 1-8.
33. Pottmann, H., et al., Geodesic patterns. ACM Trans. Graph., 2010. 29(4): p. 1-10.
34. Thielhelm, H., et al., Connecting geodesics on smooth surfaces. The Visual Computer, 2012. 28(6-8): p. 529-539.
35. Xin, S.-Q., X. Ying, and Y. He, Constant-time all-pairs geodesic distance query on triangle meshes, in Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games2012, ACM: Costa Mesa, California. p. 31-38.