Laboratoire Bordelais de Recherche en Informatique
UMR 5800 - Université Bordeaux I, 351, cours de la Libération,
33405 Talence CEDEX, France

Research Report RR-1447-08

Constraint shortest path computation on polyhedral surfaces

R.Synave, S.Gueorguieva and P.Desbarats
{synave,stefka,desbarats}@labri.fr

July, 2008

Constraint shortest path computation on polyhedral surfaces

Contents

Abstract

A new method to compute the shortest path over polyhedral surfaces is presented. Based on the consensus plane constrained subdivision of the initial surface, a discrete geodesic approximation is generated with respect to both metric and shape criterion optimization.

1 Shortest path problem on a polyhedron

The shortest path computation is a well known optimization geometric problem [2, 4, 11, 15]. The collision-free robot motion planning is one of the most famous application of this problem. Since, new application areas as shape analysis and medical image processing emerged and motivated a regain of interest towards further solution optimization.

The 3D Euclidean shortest-path problem is defined as follows [1]: given pairwise-disjoint polyhedral objects in $\mathbb{R}^3$ and two points s and t, compute the shortest path between s and t which avoids the interiors of the given polyhedral "obstacles". The problem received considerable attention in computation geometry community and it had be shown to be a NP-hard problem [3].

In opposition to the 2D space problem where the shortest path calculation is reduced to a discrete graph search, in 3D space, the problem is not even discrete since there seems to be a continuum of potential paths to be considered.

Moreover, knowing the sequence of obstacle edges through which the desired shortest path passes is not sufficient. Furthermore, the calculation of the contact points requires high-degree algebraic equation solutions. For this purpose either numerical approximate methods or exact but inefficient symbolic computations are used.

2 The discrete geodesic problem

We are interested in a special 3D case of the three-dimensional shortest path problem called "the Discrete Geodesic Problem" [17, 13, 12, 4, 8]. In this case, two points lying on the surface of a known polyhedron are given. The shortest path on the polyhedral surface joining these points is asked for. A widely used approach for the problem follows the so called "continuous Dijkstra" method introduced by [12] and inspired by the algorithm of Dijkstra for finding shortest paths in a graph. The essence of this method is the propagation of a "wave front" of signals from the source to the rest of the surface. A point of the surface is labeled with the time at which it receives the signals and corresponds to the minimum distance from the source to it. Furthermore, the point propagates the signal wave front by analogy with a polyhedral sweep outwards from the point to increasing distances. An illustration is given in Fig 1(a,b). Beginning at point s a shortest path to the point t is constructed following Dijkstra's algorithm. The integer digits correspond to the number of edges spanned by the shortest path from s to any vertex of the given mesh starting with the source and expanding monotonically with respect to the distance from the source. Besides, as shown in Fig 1(c,d), solution is not unique, but paths in Fig 1(c,d) are better than the one in Fig 1(b) as long as they are closer to the geodesic curve joining s and t. In fact, the shortest path has local optimality feature to unfold into a straight line. This feature could be used to calculate "conform" subdivisions of the polyhedron faces and to optimize approximation solutions [6].

Uniform subdivisions do not guarantee the shortest path problem optimization. See for example the uniform "1:2" subdivision[1] in Fig 2(c,d). The subdivision in Fig 2(c) succeeds to produce an optimal solution while in Fig 2(d) it fails. In some cases, an edge flip operation is sufficient to transform the surface mesh in the appropriated form as in Fig 2(b).

See also the examples in Fig 3 and Fig 4. The surface mesh in Fig 3 is transformed into the surface mesh in Fig 4 by the means of an uniform "1:6" subdivision.[2] Graphically, the path is represented along with two triangle strips differing in colour, dark gray for the right side and light gray for the left one. Each triangle in the strips contains a shortest path vertex or edge. As one can

[1] Each triangle from the initial mesh model is split into two triangles along a line joining a vertex and a point lying on the opposite side.

[2] Each triangle from the initial mesh model is split into six triangles along the three medians.

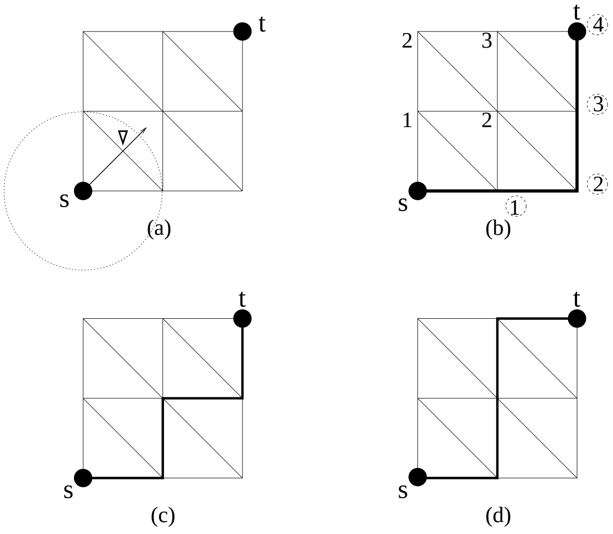


Figure 1: Shortest path computation: (a) Circular front wave propagation. (b) "Continuous" Dijkstra algorithm. (c,d) Dijkstra's algorithm shortest path solutions.

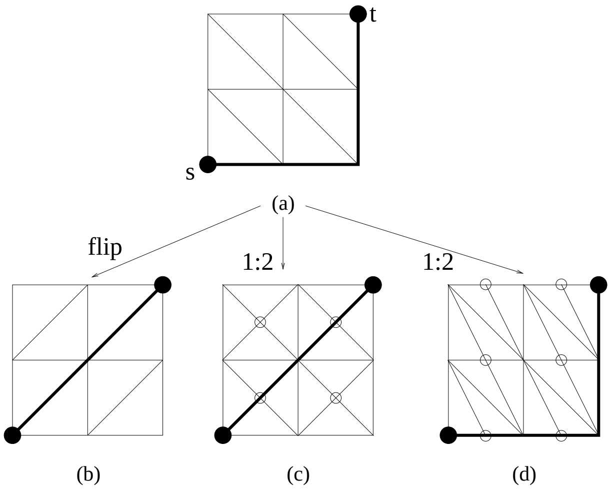


Figure 2: Shortest path and surface subdivision: (a) Initial mesh. (b) Shortest path optimization after an uniform edge flip operation. (c) Shortest path optimization after an uniform "1:2" subdivision. (d) The uniform "1:2" subdivision has no effect on the shortest path optimization.

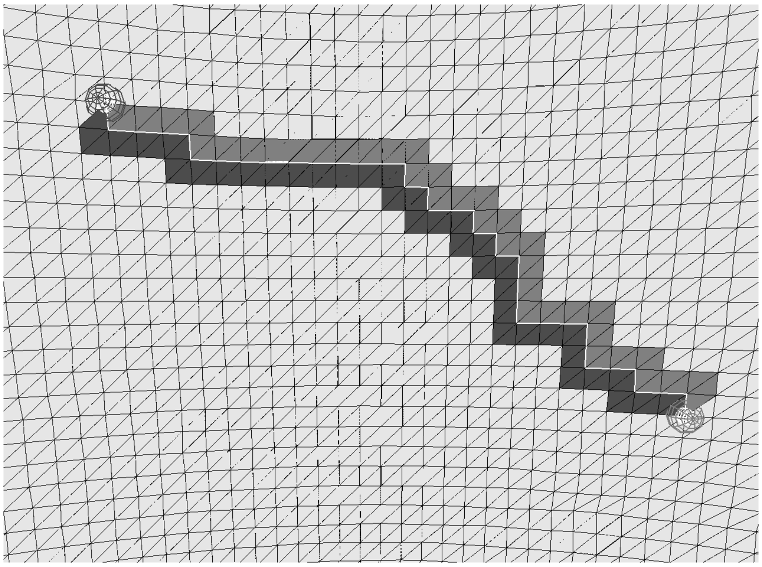


Figure 3: Geodesic path.

see, in spite of the subdivision the shortest path is still far from the geodesic curve.

Another concept relevant to the current research is the concept of discrete geodesic curvature introduced by [16] and that allows to move on polyhedral surfaces in a given direction along a discrete "straightest" curve. Inspired from the Polthier's straightest geodesic theory, an interactive algorithm for approximated geodesic computation is proposed by [10]. Starting with an initial shortest path approximation, at each step a newly computed geodesic path locally reduces path length at each point. As stated by the authors, the convergence rate strongly depends on the initial approximation. Reported results are obtained using an initial shortest path calculated by the Fast Marching Method as in [9]. Algorithm convergence is proved for planar surfaces while the general case remains an open issue. Besides, reported problem solutions are handling mainly time and memory complexity optimization.

In the present paper, our interest is directed towards the shortest path approximation solution and its optimization in terms of geodesic distance metric and shape evaluation. The motivation underlying our research are applications as computer assisted paleoanthropology where polyhedral surfaces are issued from fossil reconstruction and where morphometry shape evaluation involves computations with a precision in the range of 0.1mm.[3]

Our goal is to define and to measure the geodesic curve connecting landmarks

[3] Anatomical structures showing significant details below this threshold, are processed with microCT scanners yielding voxel resolutions in the micrometer range.

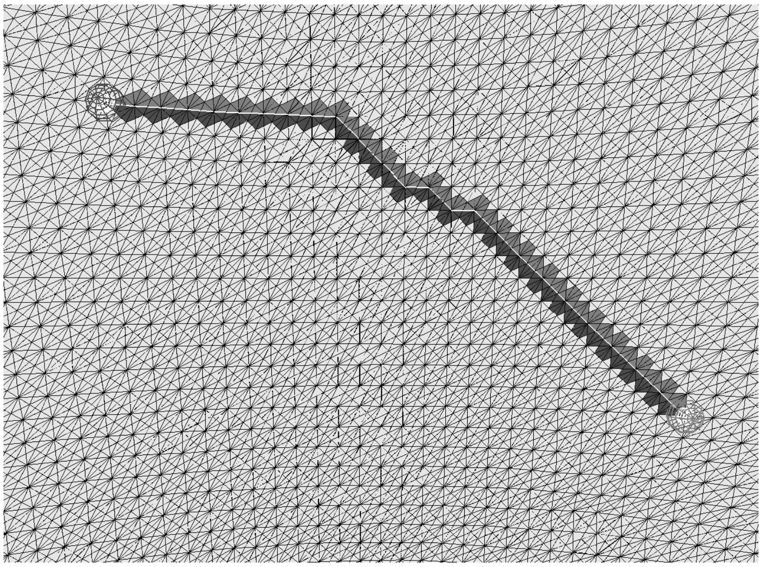


Figure 4: Geodesic path after uniform "1:6" subdivision.

on object boundary surface model as near as possible to the curve spanned by a millimeter ruler used "in situ" as illustrated in Fig 9. For this purpose shortest path should fulfill additional constraints.

The intuition is to limit path curvature and path length variation to a desired specified threshold. The basic idea is to find out a plane fitting to a major portion of a shortest geodesic path and to subdivide the polyhedral surface along its intersection with this plane. We call this plane a "consensus plane", and the subdivision, "consensus plane constrained subdivision", abbreviated in the text as respectively "CP" and "CPCS". In the newly calculated subdivision, a shortest geodesic path is confined to a bounding volume delimited by an offset cylinder aligned along an axe through the source and the destination points, and by thin planar strip.

3 Shortest path computation

We propose to compute a shortest geodesic path between a source point s and a destination point t in four main steps:

1. Get a shortest path approximation.
   Our algorithm starts with an initial approximation of the shortest path. In our experimentation, Dijkstra's algorithm is used for this purpose. One should remark that despite the fact that Dijkstra's algorithm may produce a minimum path that is far from the geodesic as illustrated in Fig 1(b), our

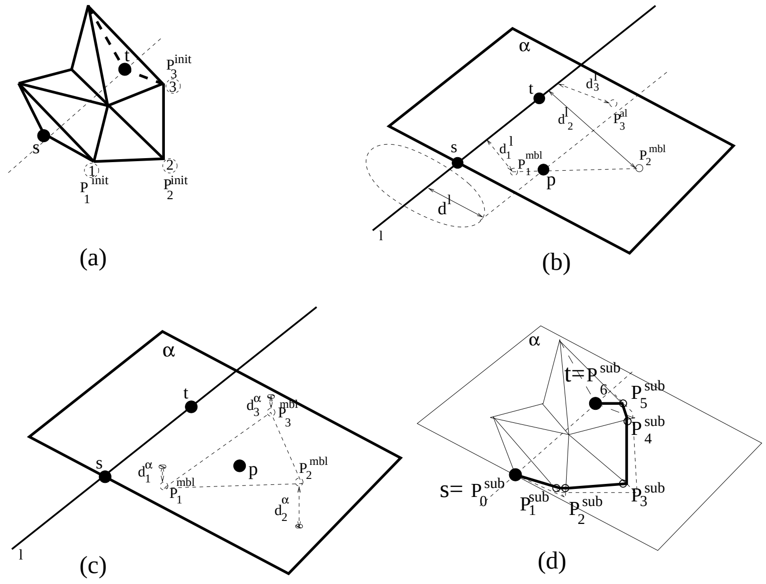


Figure 5: Consensus plane constrained subdivision: (a) Initial mesh. (b) CP initialization. (c) CP estimation. (d) CPCS.

algorithm can correct this deflection. With no restriction other algorithms could be also exploited. Let us denote this approximation as $P^{init} = \{P_i^{init}\}_{i=0}^{n}$ with $s = P_0^{init}$ and $t = P_n^{init}$.

2. Construct the consensus plane.
   At this step we try to find out a plane α , called "consensus plane", that passes through the points s and t, and that fits "best" to the rest of points in P^{init}. For this purpose we iterate a RANSAC like algorithm [5].

   (a) Let P^{mbl} such as $P^{mbl} \subseteq P^{init}$ be a subset of $m+1$ points ("maybeinliers"). At the first iteration, points from P^{mbl} are randomly chosen among the points from P^{init}.

   (b) Define a point p:

$$p = \sum_{j=0}^{m} \alpha_j P_j^{mbl}, 0 < m \leq n \tag{1}$$

$$\alpha_j = d_j^l / \sum_{k=0}^{m} d_k^l \tag{2}$$

   where d_j^l is the euclidean distance from P_j^{mbl} to the line l supporting s and t. Let us denote the distance from p to l with d^l.

(c) Construct the consensus plane α as a plane incident to the points s, t and p.

(d) Calculate the euclidean distances d_j^α from P_j^{mbl} to α, $0 \leq j \leq m$ and the average distance d^{alpha}:

$$d^{alpha} = \sum_{j=0}^{m} d_j^\alpha/(m+1) \tag{3}$$

(e) Construct the set of points P^{al} ("alsoinliers")

$$P^{al} = \{P_j \in P^{init} \wedge P_j \notin P^{mbl} \wedge d_j^l \leq d^l \wedge d_j^\alpha \leq d^{alpha}\} \tag{4}$$

where d_j^l and d_j^α are the euclidean distances from P_j to respectively l and α.
The condition $d_j^l \leq d^l$ supports an offset cylinder aligned along l.
The condition $d_j^\alpha \leq d^{alpha}$ provides a planar strip enclosing α.

(f) Test the stop criterion if either the maximum number of iterations is achieved or all points in P^{init} are proceeded.

(g) Update the set of maybeinliers.

$$P^{mbl} = P^{mbl} \cup P^{al} \tag{5}$$

Go to (a).

This step is illustrated in Fig 5. For the initial mesh in Fig 5(a) the set P^{mbl} is constructed as shown in Fig 5(b). On a second iteration the construction comes to end as long as all points in P^{init} are proceeded, see Fig 5(c).

3. Compute the consensus plane constrained subdivision.

   At this stage the polyhedral surface is subdivided along the consensus plane α. Each triangle adjacent to the path P^{init} has at most a line segment intersection with α and is subdivided as illustrated in Fig 6.

4. Compute shortest path approximation in the refined subdivision and output the computed shortest path as the resulting geodesic P^{result}.

4 Shortest path evaluation

In order to evaluate the computed geodesic curves, several path characteristics are investigated.

First, the euclidean geodesic length is calculated as the sum of the edge lengths in the final geodesic path.

Second, the shortest path linearity is estimated with the average and the standard deviation of the angles between consecutive edges as shown in Fig 7(b).

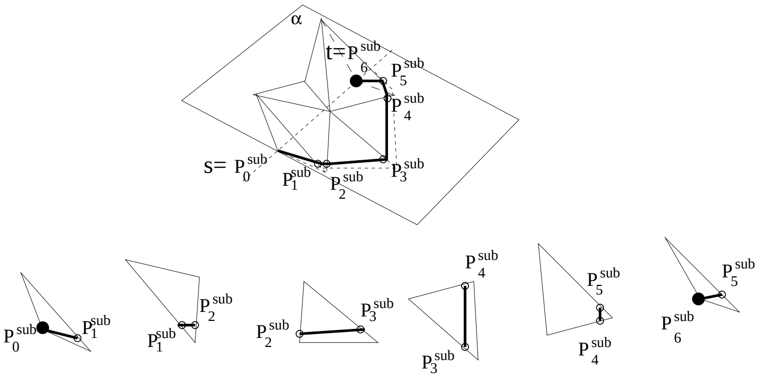


Figure 6: Consensus plane constrained subdivision (CPCS).

The value of the angle β_{012} between the vectors $\overrightarrow{P_1^{sub}P_0^{sub}}$ and $\overrightarrow{P_1^{sub}P_2^{sub}}$ in the plane α for collinear segments should be 180^0.

Third, the path discrete geodesic curvature is assessed following the definition given in [10]. Average and standard deviation are supplied in table 2 and in table 3. The illustration is given in Fig 7(c). Let us denote the left polyhedral angles at P_1^{res} as θ_{10}^l, θ_{11}^l, and $\theta_1^l = \theta_{10}^l + \theta_{11}^l$. By analogy, let the right polyhedral angles be θ_{10}^r, θ_{11}^r, and $\theta_1^r = \theta_{10}^r + \theta_{11}^r$. The discrete geodesic curvature at P_1^{res} is calculated as follows

$$\kappa(P_1^{res}) = \frac{2\pi}{\theta_1^l + \theta_1^r}\left(\frac{\theta_1^l + \theta_1^r}{2} - \theta_1^r\right) \tag{6}$$

and its value for a "straightest geodesic" should be zero.

Finally, execution times are also provided. The elaborated algorithm and the appropriated computer graphics interface are written in C and run on an Intel Pentium4, 3GHz, 1Go RAM.

It should be observed that comparison with other algorithms is very difficult since algorithm performances are reported mainly with respect to time and memory complexity [7, 14, 10, 18].

5 Results and discussion

In this section, we show some results of our algorithm. As our long term objective is the resolution and the error control along a complete 3D laser acquisition pipeline, all experiments deal with real physical objects. At a preprocessing stage the polyhedral surfaces are reconstructed from 3D laser scans as illustrated in Fig 8. A description of our experimental 3D acquisition pipeline could be found in [19].

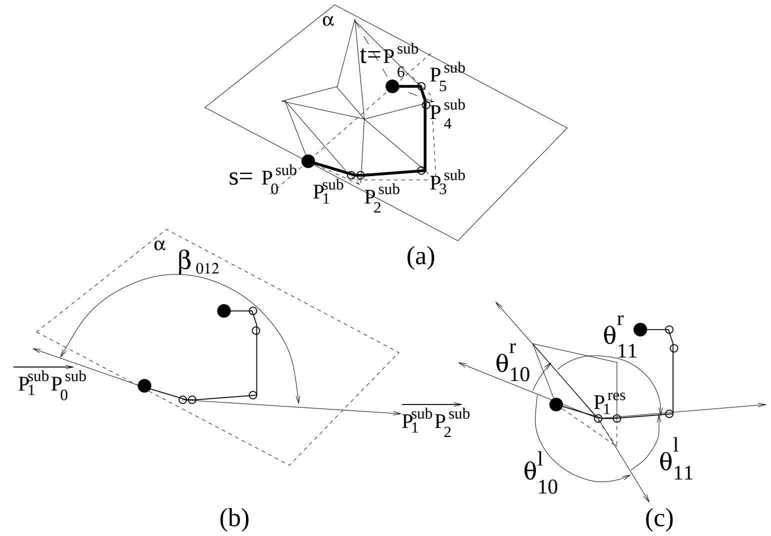


Figure 7: Shortest path evaluation: (a) Shortest path. (b) Angle deviation in the consensus plane. (c) Discrete curvature evaluation.

Figure 8: 3D laser scanner acquisition.

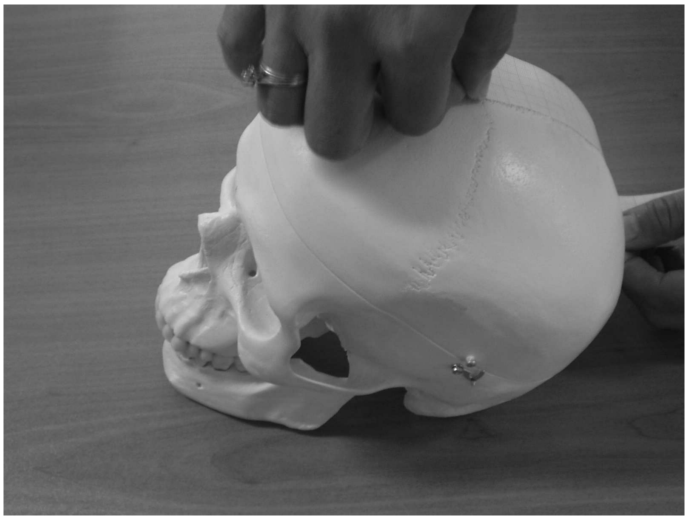


Figure 9: Morphometric measurement.

Next, shortest geodesic paths between chosen landmarks on the reconstructed object boundary are computed and finally the real and the numerically measured distances are compared.

In the first example, the shortest geodesic path is supported by a regular cone surface illustrated in Fig 10. In the second example, the path extremities are on the bottom cone face that presents two concentric circular engravings shown in Fig 12. The third example given in Fig 14, concerns morphometric measurement illustrated in Fig 9 for the shortest path between the bregma and the lambda landmarks on a skull[4]. The pics on the polyhedral surfaces define the landmark positions.

Quantitative results are summarized in table 1, 2 and 3. Each shortest path example is described in the corresponding table line.

In table 1 the shortest path length is evaluated. All lengths are measured along geodesic curves and are given in millimeters. The first column denominates the polyhedral model in use. The second column contains the physic measures L , the third consists of path lengths L_{Dij} using Dijkstra's algorithm, and the last column describes path lengths L_{CPCS} according the CPCS method. As one can see, our algorithm produces shortest paths with geodesic length closer to the real measures. Especially, in case of irregularities, as the cone bottom ridge contours, CPCS method remains stable.

The evaluation of shortest path shape for the Dijkstra's algorithm in terms of the discrete curvature variation is given in table 2. The review of the same characteristics for the proposed CPCS method is given in table 3. According to

[4]The skull landmark distances are measured by an expert and according to a standard protocol

Model	L [mm]	L_{Dij} [mm]	L_{CPCS} [mm]
cone	62	64.797	63.239
cone bot.	129	165.253	138.681
skull	121	132.236	122.219

Table 1: Shortest path length evaluation.

Model	β		$\|\theta^l - \theta^r\|$		κ	
	ave	var	ave	var	ave	var
cone	173.674	10.545	8.064	21.170	0.005	0.197
cone bot.	136.603	41.388	83.919	83.492	0.009	1.051
skull	166.502	27.568	24.086	57.416	0.013	0.543

Table 2: Dijkstra's shortest path curvature evaluation.

the both criteria, the planar angular variation β and the discrete curvature κ, CPCS method has better performances in particular for angular shapes.

Execution times are recapitulated in table 4. As expected, CPCS algorithm is slower than Dijkstra's but still compatible with acquisition routine. Indeed, CPCS algorithm includes Dijkstra's path computation on step one and step four. The CPCS calculation takes in average 20.16% of the total time.

Finally, visual quality of Dijkstra's and CPCS method are compared for the side cone surface in Fig 11, for the bottom cone surface in Fig 13 and for the skull polyhedral surface in Fig 15. It is seen that CPCS method produces more precise and thin trace for the geodesic paths.

6 Future work

Owing to our validation tests three directions for further algorithm performance improvement have been identified with respect to:

- Getting a better initialization

  Global optimality of the shortest geodesic problem as in many optimization problems depends on the initial approximation. Improper initial-

Model	β		$\|\theta^l - \theta^r\|$		κ	
	ave	var	ave	var	ave	var
cone	177.361	2.147	2.087	2.098	0.006	0.024
cone bot.	160.306	19.677	36.508	37.896	0.010	0.459
skull	177.128	2.184	0.729	1.048	0.000	0.010

Table 3: CPCS shortest path curvature evaluation.

Model	V	$time_{Dij}$	$time_{CPCS}$
cone	13097	0.85	2.11
cone bot.	5842	0.25	0.66
skull	18616	1.54	3.87

Table 4: Shortest path execution time evaluation.

ization could converge to a local minimizer. We can expect that for a good initial approximation the final path length is very close to a global minimizer of the length functional. For this purpose alternatives to the Dijkstra's method as the Fast Marching Method, the MMP [5] method, and their variations should be investigated.

- Constraint optimization

  In the presented algorithm, the construction of the constraint plane depends on three points: s and t, the starting and the ending path points and a third point p defined following equations (1) and (2) and a RANSAC like algorithm. We expect that other point distribution models could be experienced in order to calculate "the mean geometry" of the consensus plane.

- Subdivision refinement

  Lastly but not least we examine the surface subdivision issued from the cut of the initial mesh along the consensus plane. This operation could deteriorate the quality of the final mesh model, as for example producing a badly shape or degenerate triangle face. This problem is not addressed in the present research but is an important open outcome.

7 Conclusion

A new algorithm for the shortest path computation on polyhedral surfaces is proposed. The elaborated method, called consensus plane constrained subdivision method, computes an optimal solution with respect to both metric and shape criteria. The algorithm is implemented as a part of a geodesic computation utility of a 3D laser scanner acquisition and reconstruction pipeline. More experiments on algorithm performances with respect to robustness and time complexity are in progress.

[5] Mitchell, Mount and Papadimitrou method

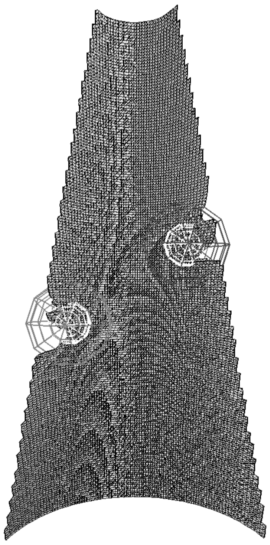
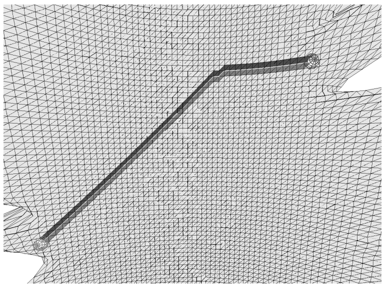
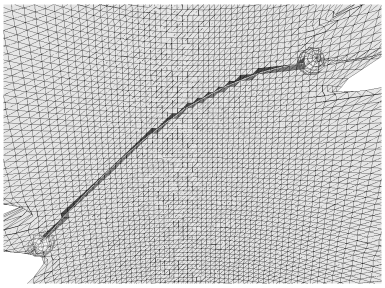


Figure 10: Polyhedral cone surface.



(a) Dijkstra's path



(b) CPCS path

Figure 11: Shortest path calculation on cone polyhedral surface.

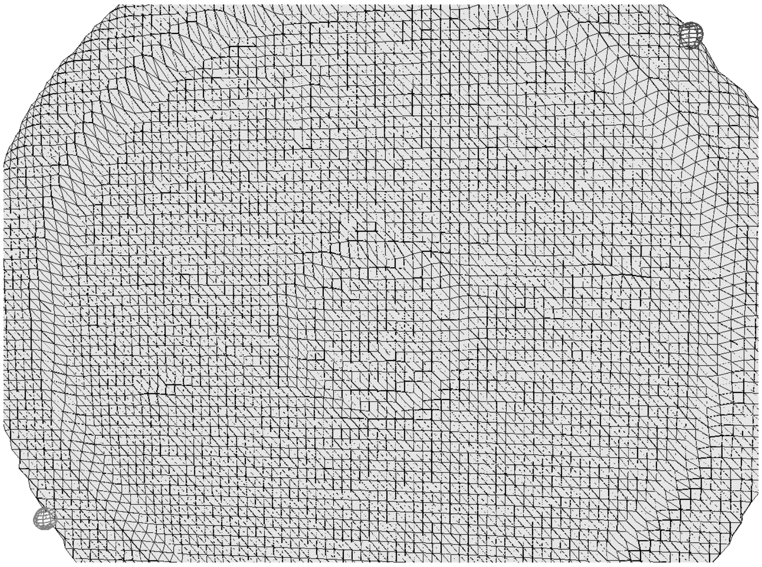
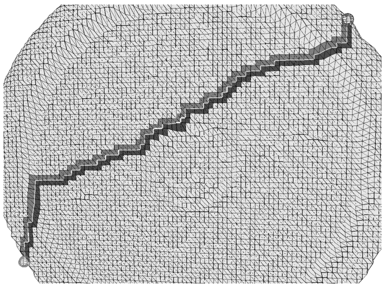
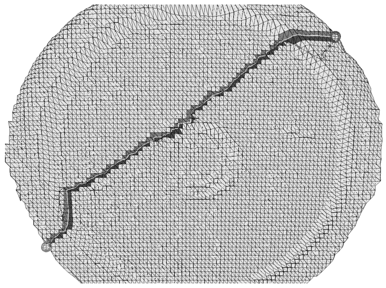


Figure 12: Polyhedral cone surface: bottom.



(a) Dijkstra's path



(b) CPCS path

Figure 13: Shortest path calculation on cone polyhedral surface: bottom.

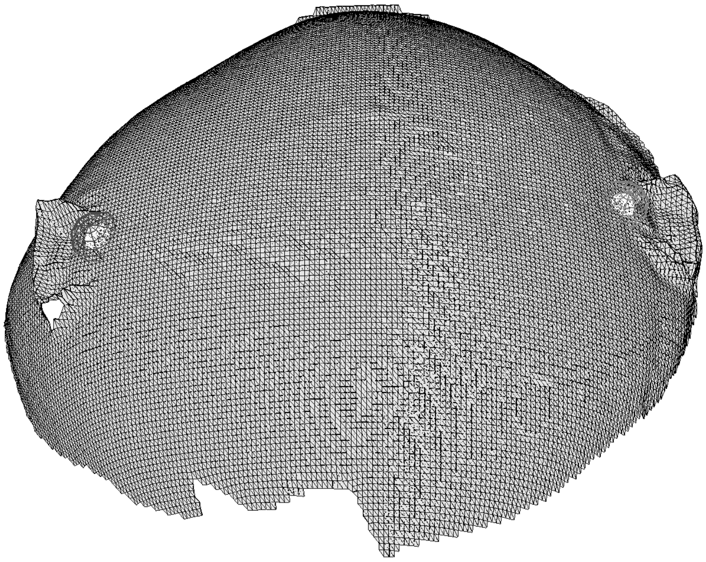
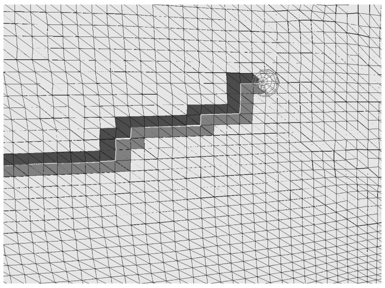
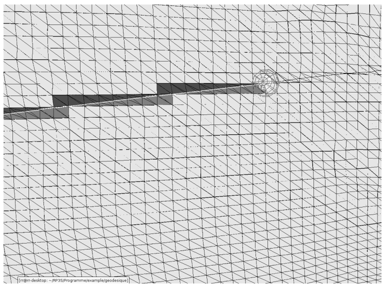


Figure 14: Polyhedral skull surface.



(a) Dijkstra's path



(b) CPCS path

Figure 15: Shortest path calculation on skull polyhedral surface.

References

[1] Pankaj K. Agarwal, Sariel Har-Peled, Micha Sharir, and Kasturi R. Varadarajan. Approximating shortest paths on a convex polytope in three dimensions. *Journal of the ACM*, 44(4):567–584, 1997.

[2] Marshall Bern and David Eppstein. Approximation algorithms for geometric problems. *8th ACM-SIAM Symposium on Discrete Algorithms*, pages 296–345, 1997.

[3] John F. Canny and John H. Reif. New lower bound techniques for robot motion planning problems. In *FOCS*, pages 49–60, 1987.

[4] J. Chen and Y. Han. Shortest paths on a polyhedron; part i: computing shortest paths, 1996.

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[6] Hershberger and Suri. Practical methods for approximating shortest paths on a convex polytope in R^3. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1995.

[7] Biliana Kaneva and Joseph O'Rourke. An implementation of chen & han's shortest paths algorithm. In *CCCG*, 2000.

[8] Sanjiv Kapoor. Efficient computation of geodesic shortest paths. In *STOC '99: Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 770–779, New York, NY, USA, 1999. ACM.

[9] R. Kimmel and J. Sethian. Computing geodesic paths on manifolds, 1998.

[10] D. Martinez, L. Velho, and P.C. Carvalho. Computing geodesics on triangular meshes. *Computers&Graphics*, 29(5):667–675, 2005.

[11] J. Mitchell. Geometric shortest paths and network optimization, 1998.

[12] Joseph S. B. Mitchell, David M. Mount, and Christos H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16(4):647–668, 1987.

[13] David Mount. On finding shorthest paths on convex polyhedra, 1985.

[14] Marcin Novotni and Reinhard Klein. Computing geodesic paths on triangular meshes. In *WSCG*, pages 341–348, 2002.

[15] Joseph O'Rourke. Computational geometry column 35. *International Journal of Computational Geometry and Applications*, 9(4/5):513–, 1999.

[16] Konrad Polthier and Markus Schmies. Straightest geodesics on polyhedral surfaces. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, pages 30–38, New York, NY, USA, 2006. ACM.

[17] Micha Sharir(and Amir Schorr. On shortest paths in polyhedral spaces. In *STOC '84: Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 144–153, New York, NY, USA, 1984. ACM.

[18] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J. Gortler, and Hugues Hoppe. Fast exact and approximate geodesics on meshes. *ACM Trans. Graph.*, 24(3):553–560, 2005.

[19] Remi Synave, Pascal Desbarats, and Stefka Gueorguieva. Toolkit for registration and evaluation for 3d laser scanner acquisition. In *WSCG'2008*, pages 199–204, Plzen: University of West Behemia, February 2008.