

# Approximating Geodesic Distances on 2-Manifolds in $\mathbb{R}^3$

Christian Scheffer\*

Jan Vahrenhold†

## Abstract

We present an algorithm for approximating geodesic distances on 2-manifolds in  $\mathbb{R}^3$ . Our algorithm works on an  $\varepsilon$ -sample of the underlying manifold and computes approximate geodesic distances between all pairs of points in this sample. The approximation error is *multiplicative* and depends on the density of the sample. For an  $\varepsilon$ -sample  $S$ , the algorithm has a near-optimal running time of  $\mathcal{O}(|S|^2 \log |S|)$ , an optimal space requirement of  $\mathcal{O}(|S|^2)$ , and approximates the geodesic distances up to a factor of  $1 - \mathcal{O}(\sqrt{\varepsilon})$  and  $(1 - \mathcal{O}(\varepsilon))^{-1}$ .

## 1 Introduction

The study of geodesic (paths and) distances on three-dimensional objects has a long history both in Differential and Discrete Geometry, and a recent survey [4] summarizes the main results and the variety of applications, e.g., in GIS and Robotics. If the underlying object is differentiable, methods from Differential Geometry, e.g. special classes of partial differential, can be applied. If, on the other hand, the underlying object is non-differentiable, e.g. polyhedral objects, discretized versions of algorithms from Differential Geometry or discrete shortest-paths algorithm have to be used.

The problem setting we focus on can be seen as a hybrid between these two extremes: We study the problem of computing geodesic distances on 2-manifolds in  $\mathbb{R}^3$  but assume that the input is a set of points sampled from the considered surface. The task then is to compute geodesic distances *on the manifold* between all pairs of points in the sample. Since the set of sample points can only approximate the manifold, the algorithm can only be expected to compute *approximate* geodesic distances. Our main contribution is to show that an approximation with an multiplicative approximation error that only depends on the quality of the point sample with respect to the manifold can be computed in near-optimal time.

### 1.1 Shortest Paths on Manifolds

Kimmel and Sethian [7] present the so-called *fast marching method* to compute geodesics on a discretized (ex-

PLICITLY given) manifold. For the case of computing distance functions and geodesics on an implicitly given manifold, we can resort to the theoretical framework of Mémoi and Sapiro [8]. Their theory is based upon continuous differential geometry methods, and a discrete version can be implemented using the fast marching method which results in a running time of  $\mathcal{O}(n^2 \log n)$  for a discretized manifold consisting of  $n$  points. The algorithm approximates geodesic distances up to an *additive* error term that depends on the granularity of the discretized manifold (and thus on the number  $n$  of points), also, an upper bound on the local curvature of the manifold has to be known to the algorithm. While this latter restriction can be removed using methods we developed in a companion paper [10], the additive error seems to be inherent to any approach using the fast marching method.

### 1.2 Shortest Paths on Polyhedral Objects

In contrast to the algorithms described in the previous section, our approach is to first to approximate the manifold by a polyhedral object and then to compute (exact) geodesics on this approximation. As for the case of explicitly given manifolds, the fast marching method of Kimmel and Sethian can be used, see, e.g., the work by Mémoi and Sapiro [9], but the analysis shows that it still results in an additive approximation error. Efficient exact shortest path computations on general polyhedra are considered complex and challenging, and recent surveys [1, 4] conclude that the general problem is still wide open. The currently best known results related to shortest path computations on polyhedra are due to Chen and Han [5] and Schreiber [11]. Chen and Han [5] present an algorithm based on a *continuous Dijkstra* technique that, after  $\mathcal{O}(n^2)$  preprocessing time, can answer distance queries to a fixed source in  $\mathcal{O}(q \cdot \log n / \log q)$  time where  $1 \leq q \leq n$  is a trade-off parameter. Since this source has to be known during preprocessing, using their algorithm as a building block in an all-pairs geodesics problem results in  $\mathcal{O}(n^3)$  running time. The algorithm by Schreiber [11] solves the single-source shortest path problem in optimal  $\mathcal{O}(n \log n)$  time but assumes that the underlying polyhedron belongs to one of three classes of polyhedra where the edge length of adjacent faces do not differ by more than a constant factor. In the general case we are considering, this assumption cannot be made.

\*Department of Computer Science, Technische Universität Dortmund, christian.scheffer@cs.tu-dortmund.de

†Department of Computer Science, Technische Universität Dortmund, jan.vahrenhold@cs.tu-dortmund.de.

Thus, one of the main contributions of this paper is to demonstrate that we can transform the polyhedral approximations of the manifolds we are working with such that we can both apply Schreiber’s algorithm and at the same time maintain the asymptotic approximation quality of the resulting shortest paths. We embed this algorithm in an approximation context and show that computing approximate geodesic distances between all pairs of points can be done in  $\mathcal{O}(n^2 \log n)$  time.

Due to the apparent difficulty of the exact shortest path problem on general polyhedra (possibly of genus  $g \geq 1$ ), Aleksandrov *et al.* [1] focus on answering approximate shortest path queries. They present an  $(1 \pm \delta)$ -approximation algorithm that, after  $\mathcal{O}\left(\frac{(g+1)n^2}{\delta^{3/2}q} \log \frac{n}{\delta} \log^4 \frac{1}{\delta}\right)$  preprocessing time, can answer a shortest-path query between an arbitrary pair of points in time  $\mathcal{O}(q)$  (again,  $q$  is a trade-off parameter, this time chosen such that  $\frac{1}{\sqrt{\delta}} \log^2 \frac{1}{\delta} < q < \frac{(g+1)^{2/3} n^{1/2}}{\sqrt{\delta}}$ ). We come back to this algorithm in Section 4.

## 2 Description of the Algorithm

As discussed in the introduction, our approach is to work on a point set sampled from the considered surface. In a nutshell, we (re-)construct a polyhedral object having the sample points as its vertices that approximates the manifold and then use Schreiber’s exact algorithm to compute geodesic distances between all pairs of points in the sample. As a consequence, the approximation quality of our algorithm only depends on the approximation quality of the point sample  $S$  with respect to the manifold  $\Gamma$  from which the points have been sampled. This quality can be characterized using a central concept introduced by Amenta and Bern [2] in the context of reconstructing smooth surfaces from point samples: For any point  $x$  on the manifold  $\Gamma$ , the *local feature size*  $lfs(x)$  is defined as the distance of  $x$  to the medial axis of  $\Gamma$ . Thus, the local feature size captures the curvature and the folding of  $\Gamma$ . It is known that the (topological) correctness of a reconstruction algorithm depends on the density of the set  $S \subset \Gamma$  of sample points used for the reconstruction relative to the local feature size.

**Definition 1** *A discrete subset  $S$  a smooth 2-manifold  $\Gamma \subset \mathbb{R}^3$  is an  $\varepsilon$ -sample of  $\Gamma$  if and only if for every point  $x \in \Gamma$  there is a sample point  $s \in S$  with  $|xs| \leq \varepsilon \cdot lfs(x)$ .*

In the light of the above definition, we present an algorithm that takes an  $\varepsilon$ -sample  $S$  of a manifold  $\Gamma$  and computes geodesic distances on a specific polyhedron  $\Pi$  whose vertex set is derived from  $S$ . For any two points  $s_1, s_2 \in S$ , the geodesic distance on  $\Pi$  between  $s_1$  and  $s_2$  is within a multiplicative factor of the geodesic distance between these points when measured on (the unknown) manifold  $\Gamma$ ; the approximation error depends only on  $\varepsilon$ .

## 2.1 Outline of Our Approach

We start out by giving an algorithm for converting the input  $\varepsilon$ -sample  $S$  into what we call a *self-conforming* sample  $S^{\text{conf}}$  (see Definition 5). We then compute the *restricted Delaunay tetrahedrization*  $\text{Del}_\Gamma(S^{\text{conf}})$  (see Definition 3) and prove that  $\text{Del}_\Gamma(S^{\text{conf}})$  is a *self-conforming* polyhedron (see Definition 2), i.e., can be handled by Schreiber’s algorithm in optimal time.

**Definition 2 (Schreiber [11], p. 40)** *A polyhedron  $P \subset \mathbb{R}^3$  (possibly non-convex) is self-conforming if for each edge  $e$  of  $\partial P$ , there is a connected region  $R(e)$ , which is the union of  $\mathcal{O}(1)$  facets of  $\partial P$  and whose interior contains  $e$ , so that the shortest path distance from  $e$  to any edge  $e'$  of  $\partial R(e)$  is at least  $2c \cdot \max\{|e|, |e'|\}$ , where  $c$  is some positive constant.*

We emphasize that it is sufficient for the algorithm (and its correctness proof) to know that for each edge  $e$ , some region  $R(e)$  with the desired properties exists; it is not necessary to have an explicit description of this region available. Thus, to prove that  $\text{Del}_\Gamma(S^{\text{conf}})$  is self-conforming (see Lemma 16), we only need to show that  $\text{Del}_\Gamma(S^{\text{conf}})$  has the following properties:

- The degree of each vertex of  $\text{Del}_\Gamma(S^{\text{conf}})$  is upper-bounded by a constant.
- The minimum inner angle of each facet of  $\text{Del}_\Gamma(S^{\text{conf}})$  is lower-bounded by a constant.

The above is summarized in the following lemma:

**Lemma 1** *Let  $\Gamma$  be a smooth 2-manifold in  $\mathbb{R}^3$  and let  $S$  be an  $\varepsilon$ -sample of  $\Gamma$ . We can decimate  $S$  such that the restricted Delaunay tetrahedrization  $\text{Del}_\Gamma(S^{\text{conf}})$  of the decimated set  $S^{\text{conf}}$  is a self-conforming polyhedron.*

Finally, we show that the (exact) geodesic distances computed on  $\text{Del}_\Gamma(S^{\text{conf}})$  are within a multiplicative factor (depending on  $\varepsilon$ ) of the geodesic distances on  $\Gamma$ .

## 2.2 Construction of a Self-Conforming Sample

Our algorithm computes a self-conforming subset of the input  $\varepsilon$ -sample and then constructs a restricted Delaunay tetrahedrization. To make the definition of a self-conforming sample more transparent, we first present the definition of a restricted Delaunay tetrahedrization:

**Definition 3 (Funke and Ramos [6], p. 782)** *Let  $S$  be a set of points sampled from a manifold  $\Gamma \in \mathbb{R}^3$ .*

1. *The restricted Voronoi diagram  $\text{Vor}_\Gamma(S)$  consists of cells  $\text{Vor}(s) \cap \Gamma$ ,  $s \in S$ .*
2. *The restricted Delaunay tetrahedrization  $\text{Del}_\Gamma(S)$  is the dual to  $\text{Vor}_\Gamma(S)$ .*

Since the manifold  $\Gamma$  is unknown to the algorithm, it is impossible to exactly compute the intersection of a Voronoi cell with the manifold. Amenta and Bern [2] and (building upon their work) Funke and Ramos [6] presented the following approach to approximating this intersection: For each point  $s \in S$ , compute its restricted Voronoi cell by intersecting the full-dimensional cell with an approximation of the plane tangent to  $\Gamma$  in  $s$ . Computing of all approximate tangent planes takes  $\mathcal{O}(|S| \log |S|)$  time [6], but the approximation quality has been analyzed in an asymptotic sense only; this is mainly due to the fact that – to achieve a near-linear running time – several trade-offs need to be made.

In the situation of our algorithm, we need an exact bound on the approximation quality (to derive the bounds for vertex degrees and inner angles mentioned above). What we do *not* need, however, is a running time better than  $\Theta(|S|^2 \log |S|)$ , and thus we can use exact textbook algorithms for computing the Voronoi diagram and computing the intersection of the faces of a cell with the respective (approximation of the) tangent planes. With some technical effort, but using elementary trigonometry only, we then can prove the following non-asymptotic bound on the approximation quality:

**Lemma 2** *Let  $\varepsilon \leq \frac{1}{22}$  be a constant and let  $s$  be an arbitrary sample point in an  $\varepsilon$ -sample of a 2-manifold  $\Gamma$ . Let  $x'$  be the furthest point (from  $s$ ) in  $\text{Vor}_{|\Gamma}(s)$ , and let  $v$  be to the furthest vertex (from  $s$ ) in the intersection of  $\text{Vor}(s)$  and the approximate tangent plane computed by the above algorithm. Then  $|sx'| \leq 1.0005 \cdot |sv|$  holds.*

Similarly, since neither  $\Gamma$  nor its medial axis are known, an exact computation of the local feature size is impossible. Instead, Funke and Ramos [6] discuss how to compute a (pointwise) lower bound for  $\varepsilon \cdot \text{lfs}(\cdot)$ . This lower bound is derived from the distance of a point  $s \in S$  to the furthest vertex of its restricted Voronoi cell. Again, this bound is given in an asymptotic sense only. To be able to prove the above-mentioned bounds on the degree of each vertex in  $\text{Del}_{|\Gamma}(S^{\text{conf}})$  and on the minimum inner angle of each facet of  $\text{Del}_{|\Gamma}(S^{\text{conf}})$ , we need to know the constants hidden in the Big-Oh notation. In a companion paper [10], we prove the following result:

**Theorem 3** *For an  $\varepsilon$ -sample  $S$  of a 2-manifold  $\Gamma$  in  $\mathbb{R}^3$  we can compute, in quadratic time, a function  $\phi(s)$  for each  $s \in S$  as the distance of  $s$  to the furthest Voronoi vertex  $v$  of the intersection of the approximation of the plane tangent to  $\Gamma$  in  $s$  and  $\text{Vor}(s)$ . For this function holds that  $\phi(s) \leq 1.135 \cdot \frac{\varepsilon}{1-\varepsilon} \cdot \text{lfs}(s)$ .*

If we define  $\phi'(s) := 1.0005 \cdot \phi(s)$  and assume that  $\varepsilon \leq \frac{1}{22}$  holds, we have the following result:

**Corollary 4** *Let  $s$  be a point in an  $\varepsilon$ -sample  $S$  of a 2-manifold  $\Gamma$  in  $\mathbb{R}^3$ . Then,  $\phi'(s) \leq 1.0005 \cdot 1.135 \cdot \frac{\varepsilon}{1-\varepsilon}$ .*

*$\text{lfs}(s) < 1.19 \cdot \varepsilon \cdot \text{lfs}(s)$  holds. Furthermore,  $\text{Vor}_{|\Gamma}(s)$  is contained in  $B_{\phi'(s)}(s)$ , i.e. the ball centered at  $s$  with radius  $\phi'(s)$ .*

### 2.2.1 Definition of a Self-Conforming Point Sample

We are now almost ready to define what constitutes a self-conforming point sample. One property of the local feature size that is crucial for most proofs is that the local feature size is a 1-Lipschitz function [2]:

**Definition 4** *For  $\alpha \in \mathbb{R}^+$ , a non-negative, real-valued function  $f$  is  $\alpha$ -Lipschitz if  $f(x) \leq f(x') + \alpha \cdot |xx'|$  for all  $x, x' \in \text{dom}(f)$ .*

Incorporating this property into the requirement that the point sample should be locally not too dense yields the following definition:

**Definition 5** *A subset  $S' \subset S$  is self-conforming, if there is a control function  $f : S \rightarrow \mathbb{R}^+$  and constants  $0 < \alpha, \beta < 1$  such that the following holds:*

1. *The function  $f$  is  $\alpha$ -Lipschitz.*
2. *For each  $s \in S'$ , we have  $B_{\beta \cdot f(s)}(s) \cap S' = \{s\}$ .*

We note that a self-conforming point sample is defined similarly to a *locally uniform* point sample (see Funke and Ramos [6]). The latter definition requires more properties for the point set but allows for an *approximate* control function. Since our proofs require a subset of the properties of a locally uniform point sample and since we can afford to compute an exact control function, we consider it instructive to (introduce and) use this new definition.

### 2.2.2 Construction of a Lipschitz Control Function

As noted by Funke and Ramos [6, p. 785], a “natural” way to make a function  $\alpha$ -Lipschitz is to encode the maximum distance to any other point in the domain. Using the constants derived in the previous paragraph, we define the function  $\psi$  as follows:

$$\psi(s) : S \rightarrow \mathbb{R}, s \mapsto \max_{s' \in S} \left\{ \phi'(s') - 1.19 \cdot \frac{1}{22} \cdot |ss'| \right\} \quad (1)$$

We note that the domain of  $\psi$  could be extended to  $\Gamma$  in the following way: For each Voronoi cell  $\text{Vor}(s)$ ,  $s \in S$ , and each point  $x \in \text{Vor}_{|\Gamma}(s)$ , define  $\psi(x) := \psi(s)$  (breaking ties for the boundary of  $\text{Vor}_{|\Gamma}(s)$  arbitrarily). Doing so allows us to prove that  $2.056 \cdot \psi$  is an approximate control function in the sense of Funke and Ramos’ definition (*local uniformity*), but increases all constants in the following by a factor of roughly two.

**Lemma 5** *The function  $\psi$  is a  $\frac{1}{18}$ -Lipschitz function.*

### 2.2.3 Decimation of the $\varepsilon$ -Sample

To derive a sample  $S^{\text{conf}}$  that leads to a self-conforming polyhedron, we use the ‘‘Decimation Step’’ algorithm from Funke and Ramos [6, p. 789] (Algorithm 1). In contrast to their setting, we can afford to spend quadratic time during preprocessing, and thus this step can be implemented straightforward, i.e. without the need for approximate range-reporting structures.

---

**Algorithm 1** Coarsening an input  $\varepsilon$ -sample [6].

---

```

1: function DECIMATE(Points  $S$ , Function  $\psi$ )
2:    $S^{\text{conf}} = \emptyset$ ,  $S^{\text{dense}} = S$ .
3:   while  $S^{\text{dense}} \neq \emptyset$  do
4:     Let  $s$  be an arbitrary point in  $S^{\text{dense}}$ .
5:      $S^{\text{conf}} = S^{\text{conf}} \cup \{s\}$ .
6:      $S^{\text{dense}} = S^{\text{dense}} \setminus (S^{\text{dense}} \cap B_{\psi(s)}(s))$ .
7:   Return  $S^{\text{conf}}$ .
```

---

This approach allows to prove the following lemma:

**Lemma 6** For  $s \in S^{\text{conf}}$ ,  $B_{\frac{17}{18}\psi(s)}(s) \cap S^{\text{conf}} = \{s\}$ .

Lemmas 5 and 6 together imply that  $\psi$  is a control function in the sense of Definition 5:

**Corollary 7** For an  $\varepsilon$ -sample  $S$  of a 2-manifold  $\Gamma$ , we can derive a self-conforming sample  $S^{\text{conf}}$  with control function  $\psi$  and  $\alpha = \frac{1}{18}$ ,  $\beta = \frac{17}{18}$  in  $\mathcal{O}(|S|^2)$  time.

### 2.3 Construction of $\text{Del}_{|\Gamma}(S_{s_{\text{src}}}^{\text{conf}})$

We first observe that no point  $x \in \Gamma$  is ‘‘too far’’ away from a point  $s$  of the self-conforming sample  $S^{\text{conf}}$ .

**Lemma 8** For each  $x \in \Gamma$ , there is some  $s \in S^{\text{conf}}$  such that  $|xs| \leq 2.056 \cdot \psi(s)$ .

Using this property, we can prove that in a restricted Delaunay tetrahedrization of a self-conforming sample the edge lengths of a single face are bounded relative to each other and that all faces have a minimum inner angle lower-bounded by a constant. First, we prove:

**Lemma 9** For each edge  $\overline{s_1s_2}$  in  $\text{Del}_{|\Gamma}(S^{\text{conf}})$ ,  $|s_1s_2|$  is upper-bounded by  $4.112 \cdot \min\{\psi(s_1), \psi(s_2)\}$ .

For the estimation of the approximation quality of our algorithm, we will also need to relate the length of the Delaunay edges to the local feature size.

**Lemma 10** For each point  $s$  in the input  $\varepsilon$ -sample  $S$ ,  $\psi(s) \leq 1.19 \cdot \varepsilon \cdot lfs(s)$ .

Since  $\psi$  is fixed before the decimation, Lemma 10 also holds for each  $s \in S^{\text{conf}}$ .

**Corollary 11** For each edge  $\overline{s_1s_2}$  in  $\text{Del}_{|\Gamma}(S^{\text{conf}})$ ,  $|s_1s_2|$  is upper-bounded by  $5 \cdot \varepsilon \cdot \min\{lfs(s_1), lfs(s_2)\}$ .

An immediate implication is that the degree of each vertex in  $\text{Del}_{|\Gamma}(S^{\text{conf}})$  is bounded by a constant.

**Lemma 12** The degree of each vertex in  $\text{Del}_{|\Gamma}(S^{\text{conf}})$  is bounded by 3925.

If, for each point  $s \in S^{\text{conf}}$ , we have its 3925 nearest neighbors at hand, Lemma 12 ensures that we can compute  $\text{Vor}_{|\Gamma}(S^{\text{conf}})$  and thus  $\text{Del}_{|\Gamma}(S^{\text{conf}})$  in  $\mathcal{O}(|S| \log |S|)$  time (the first step of the algorithm provides us with the approximations of the planes tangent to  $\Gamma$  needed for restricting the Voronoi diagram).

In comparison, Funke and Ramos [6] also prove that a *locally uniform* sample admits a constant-degree tetrahedrization. Again, however, the bound is only given asymptotically, i.e. depending on  $1/\varepsilon$ . Since our algorithm needs an *upper* bound on the number of Voronoi neighbors (to ensure that no relevant neighbor is missed), the *upper* bound on  $\varepsilon$ , i.e. the *lower* bound on  $1/\varepsilon$  is not sufficient for our purpose.

### 2.4 Intermediate Summary

As mentioned in Section 2.1, our approach is to compute a self-conforming polyhedron such that we can use Schreiber’s algorithm to efficiently compute shortest-path maps with respect to each point in the input sample. The above description indicates that a crucial step for doing this is to decimate the input sample. On the other hand, when computing a shortest path map with respect to some point  $s_{\text{src}}$ , this point needs to be present in the decimated sample. Since the decimation step is computationally expensive, we cannot afford to repeat it a linear number of times. Thus, we decimate the sample once and then, in each iteration, ensure that the point  $s_{\text{src}}$  with respect to which the shortest-path map is computed is present in the set for which the restricted Delaunay tetrahedrization is computed.

It should be pointed out that the set  $S_{s_{\text{src}}}^{\text{conf}} := S^{\text{conf}} \cup \{s_{\text{src}}\}$  for which the restricted Delaunay tetrahedrization is computed may no longer be a self-conforming set if  $s_{\text{src}} \notin S^{\text{conf}}$ , since  $s_{\text{src}}$  may violate Property (2) of Definition 5. Except for Lemma 12, however, no Lemma in the previous section relies on this property, so all other Lemmas are still valid. For Lemma 12, we observe that the introduction of  $s_{\text{src}}$  may increase each other vertex’s degree by one (thus, the constant needs to be adjusted accordingly). A close look at the proof of Lemma 12, however, shows that Property (2) of Definition 5 is *not* needed for the point  $s$  whose degree is to be bounded but only for the points inside  $B_{4.112 \cdot \psi(s)}(s)$ . Thus, the statement of Lemma 12 also holds for the point  $s_{\text{src}}$ . For ease of exposition we thus assume that the degree of *all* vertices in  $S_{s_{\text{src}}}^{\text{conf}}$  is bounded by 3925. Also, we assume that the computation of  $\text{Del}_{|\Gamma}(s)$  for each  $s \in S_{s_{\text{src}}}^{\text{conf}}$  takes  $s_{\text{src}}$  into account when computing  $\text{Vor}_{|\Gamma}(s)$  based upon its nearest neighbors. This results in Algorithm 2.

**Algorithm 2** Approximating geodesic distances between points of an  $\varepsilon$ -sample  $S$ .

---

```

1: function APPROXIMATEGEODESICDISTANCES(Points  $S$ )
2:   Let  $Distance$  be an  $|S| \times |S|$ -matrix, each entry of which is initialized with  $+\infty$ .
3:   Compute  $\phi(s)$  and tangent plane for each  $s \in S$ . ▷ Use the algorithm by Scheffer and Vahrenhold [10].
4:   Compute  $\psi$  from  $\phi$  (Lipschitziation). ▷ See Section 2.2.2.
5:    $S^{\text{conf}} = \text{DECIMATE}(S, \psi)$ . ▷ Use Algorithm 1.
6:   Compute for each point in  $S^{\text{conf}}$  its 3925 nearest neighbors. ▷ Use the brute-force algorithm.
7:   Compute for each point in  $S \setminus S^{\text{conf}}$  its nearest neighbor in  $S^{\text{conf}}$ . ▷ Use the brute-force algorithm.
8:   for each  $s_{\text{src}} \in S$  do
9:      $S_{\text{src}}^{\text{conf}} := S^{\text{conf}} \cup \{s_{\text{src}}\}$ .
10:    Compute  $\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$ . ▷ See Section 2.3.
11:     $SPMap = \text{SCHREIBERSPREPROCESSING}(\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}}), s_{\text{src}})$ . ▷ Use the algorithm by Schreiber [11].
12:    for each  $s \in S$  do
13:      if  $s \in S_{\text{src}}^{\text{conf}}$  then
14:         $Distance[s_{\text{src}}, s] = \text{SCHREIBERSQUERY}(SPMap, s)$ . ▷ Use the algorithm by Schreiber [11].
15:      else
16:        Let  $s'$  be  $s$ 's nearest neighbor in  $S_{\text{src}}^{\text{conf}}$ .
17:        for each face  $f$  of the  $\mathcal{O}(1)$  faces adjacent to  $s'$  do
18:          Let  $s''$  be the point on  $f$  closest to  $s$ .
19:           $Distance[s_{\text{src}}, s] = \min\{Distance[s_{\text{src}}, s], \text{SCHREIBERSQUERY}(SPMap, s'')\}$ .
▷ Use the algorithm by Schreiber [11].
20:   Return  $Distance$ .
    
```

---

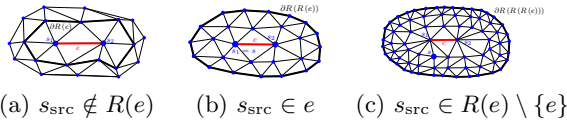


Figure 1: Region  $R^{\text{strip}}(e)$  for an edge  $e \in \text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$ .

**Lemma 13** Assuming that  $\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$  is self-conforming, Algorithm 2 runs in  $\mathcal{O}(|S|^2 \log |S|)$  time.

## 2.5 Properties of $\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$

To show that  $\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$  is self-conforming in accordance to Definition 2, we need to show that for each edge  $e$  of a facet of  $\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$ , there is a constant-size region  $R^{\text{strip}}(e)$  containing  $e$  so that the shortest-path distance from  $e$  to any edge  $e'$  of  $\partial R^{\text{strip}}(e)$  is at least  $2c \cdot \max\{|e|, |e'|\}$ , where  $c$  is a positive constant.

Intuitively, we define  $R^{\text{strip}}(e)$  as a (constant-size, yet not necessary minimal) triangle patch surrounding  $e$  such that if  $s_{\text{src}}$  lies inside  $R^{\text{strip}}(e)$ , it is neither a vertex of  $\partial R^{\text{strip}}(e)$  nor adjacent to a vertex of  $\partial R^{\text{strip}}(e)$ .

**Definition 6** Let  $e$  be an edge of a facet of  $\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$  and define  $R(e)$  as the set of facets adjacent to  $e$ . If  $s_{\text{src}} \notin R(e)$ , define  $R^{\text{strip}}(e) := R(e)$ —see Figure 1(a). Otherwise, depending on whether  $s_{\text{src}} \in e$  or  $s_{\text{src}} \in R(e) \setminus \{e\}$ , define  $R^{\text{strip}}(e) := R(R(e))$  (see Figure 1(b)) or  $R^{\text{strip}}(e) := R(R(R(e)))$  (see Figure 1(c)), where  $R(R(e))$  consists of  $R(e)$  and all facets adjacent to  $R(e)$  and where  $R(R(R(e)))$  is defined analogously.

By Lemma 12, each vertex of  $R^{\text{strip}}(e)$  has constant degree, and thus  $R^{\text{strip}}(e)$  consists of  $\mathcal{O}(1)$  facets. To bound the shortest-path distance in  $R^{\text{strip}}(e)$ , we need the following technical lemma that relates the edge lengths on the boundary of  $\partial R^{\text{strip}}(e)$  and the boundary of the “next inner” strip, i.e.,  $\partial R(R(e))$ ,  $\partial R(e)$ , or  $e$ .

**Lemma 14** Let  $\Delta$  be a facet of  $\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$  such that  $s_{\text{src}}$  is no vertex of  $\Delta$ . Then, for any two edges  $e, e'$  of  $\Delta$ , we have  $\frac{1}{6} \leq \frac{|e|}{|e'|} \leq 6$  and for any vertex  $s$  and any edge  $e$  of  $\Delta$ , we have  $\frac{2}{6} \cdot |e| \leq 2.056 \cdot \psi(s) \leq \frac{6}{2} \cdot |e|$ .

Using Lemma 14 and elementary trigonometry, we can prove the following:

**Lemma 15** Let  $\Delta$  be a facet of  $\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$  such that  $s_{\text{src}}$  is no vertex of  $\Delta$ . Then, each angle of  $\Delta$  is of size at least  $\pi/20$ .

The constant factor of 4.112 in the upper bound for the edge length in  $\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$  (Lemma 9), the “edge-length” ratio of 1:6 (Lemma 14), and the minimum angle of  $\pi/20$  (Lemma 15) allow to determine the constant  $c$  needed in the definition of a self-conforming polyhedron.

**Lemma 16** Let  $e$  be an edge of a facet of  $\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$ . Then the shortest path distance from  $e$  to any edge  $e'$  of  $\partial R^{\text{strip}}(e)$  is at least  $c \cdot \max\{|e|, |e'|\}$  for  $c := \left(1 - \frac{1}{18} \cdot 4.112 \cdot \left(2 + \frac{1}{18} \cdot 4.112 + \left(1 + \frac{1}{18} \cdot 4.112\right)^2\right)\right) \cdot \sin\left(\frac{\pi}{20}\right) \cdot \frac{1}{6^3}$ .

**Corollary 17**  $\text{Del}_{|\Gamma}(S_{\text{src}}^{\text{conf}})$  is self-conforming.

**Corollary 18** Algorithm 2 takes  $\mathcal{O}(|S|^2 \log |S|)$  time.

### 3 Analysis of the Approximation Quality

We fix two points  $s_1, s_2$  in the original  $\varepsilon$ -sample  $S$  and use  $L_{\text{Del}} = L_{\text{Del}}(s_1, s_2)$  to denote their geodesic distance as returned by Algorithm 2 and  $L_\Gamma = L_\Gamma(s_1, s_2)$  to denote their (unknown) geodesic distance on  $\Gamma$ .

The following lemma needed to bound the approximation quality is derived from Bernstein *et al.* [3, Cor. 4].

**Lemma 19** *Let  $x_1$  and  $x_2$  be two arbitrary points on  $\Gamma$  with  $|x_1 x_2| \leq \sqrt{\varepsilon^{-1}} \cdot \max\{\text{ifs}(x_1), \text{ifs}(x_2)\}$ . Then we have  $L_{\text{Del}}(x_1, x_2) \geq (1 - \mathcal{O}(\varepsilon)) \cdot L_\Gamma(x_1, x_2)$ .*

For the lower bound for  $L_{\text{Del}}$  relative to  $L_\Gamma$  we iteratively partition the geodesic path on  $\text{Del}_\Gamma(S_{\text{src}}^{\text{conf}})$  between  $s_1$  and  $s_2$  into subpaths whose endpoints fulfill the assumptions of Lemma 19. With rather technical computations synchronized over all subpaths we obtain the following lemma.

**Lemma 20**  $(1 - \mathcal{O}(\varepsilon^{\frac{1}{2}})) \cdot L_\Gamma \leq L_{\text{Del}}$ .

For the upper bound for  $L_{\text{Del}}$  we first establish a bijection between the facets of  $\text{Del}_\Gamma(S_{\text{src}}^{\text{conf}})$  and patches on  $\Gamma$ . We then replace each subpath of the (unknown) geodesic path on  $\Gamma$  between  $s_1$  and  $s_2$  that crosses a patch in  $\Gamma$  by a path along the boundary of the corresponding facet of  $\text{Del}_\Gamma(S_{\text{src}}^{\text{conf}})$ . Exploiting the lower bound on the minimum inner angle of the facets (Lemma 15) and the approximation quality of the facet w.r.t. the patch on  $\Gamma$ , we obtain the following result.

**Lemma 21**  $(1 - \mathcal{O}(\varepsilon))^{-1} \cdot L_\Gamma \geq L_{\text{Del}}$ .

### 4 Using Aleksandrov et al.'s Algorithm

The algorithm of Aleksandrov *et al.* [1] is a  $(1 \pm \delta)$ -approximation algorithm for all-pair shortest distance queries which could also be applied instead of repeatedly using Schreiber's (involved) algorithm. Lemma 22 shows that this leads to either the same running time and strictly worse approximation quality or to the same approximation quality and strictly worse running time.

Since an  $\varepsilon$ -sample  $S$  can be augmented by arbitrarily many points while still remaining an  $\varepsilon$ -sample, we need to restrict ourselves to *tight*  $\varepsilon$ -samples when comparing the dependence between  $|S|$ , the running time, and the approximation quality for the two algorithms. A tight  $\varepsilon$ -sample is an  $\varepsilon$ -sample  $S$  for which there is a positive constant  $\eta$  such that  $B_{\eta \cdot \text{ifs}(s)}(s) \cap S = \{s\}$  for  $s \in S$ .

**Lemma 22** *Let  $S$  be a tight  $\varepsilon$ -sample of a 2-manifold  $\Gamma$  and assume that we use the Algorithm of Alexandrov et al. as part of Algorithm 2 to compute approximate shortest paths on  $\text{Del}_\Gamma(S_{\text{src}}^{\text{conf}})$ .*

1. *If the running time of Algorithm 2 is to remain in  $\mathcal{O}(|S|^2 \log |S|)$ ,  $L_\Gamma$  and  $L_{\text{Del}}$  relate as follows:*

$$(a) (1 - \Omega(\varepsilon^{0.427})) \cdot L_\Gamma \leq L_{\text{Del}}.$$

$$(b) \left(1 - \mathcal{O}\left(\varepsilon^{\frac{1}{1.08}}\right)\right)^{-1} \cdot L_\Gamma \geq L_{\text{Del}}.$$

2. *If the approximation quality of Algorithm 2 is to remain as given in Lemmas 20 and 21, a tight sample of a higher density than  $S$  is required. The size of this sample implies a running time of  $\omega(|S|^{\frac{19}{8}})$ .*

### References

- [1] L. Aleksandrov, H. N. Djidjev, H. Guo, A. Maheshwari, D. Nussbaum, and J.-R. Sack. Algorithms for approximate shortest path queries on weighted polyhedral surfaces. *Discrete & Computational Geometry*, 44:762–801, 2010.
- [2] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete & Computational Geometry*, 22(4):481–504, Dec. 1999.
- [3] M. Bernstein, V. de Silva, J. C. Langford, and J. B. Tenenbaum. Graph approximations to geodesics on embedded manifolds. Unpublished, <http://isomap.stanford.edu/BdSLT.pdf>, Dec. 2000.
- [4] P. Bose, A. Maheshwari, C. Shu, and S. Wührer. A survey of geodesic paths on 3D surfaces. Submitted, a technical report version is available from <http://arxiv.org/abs/0904.2550>, Aug. 2009.
- [5] J. Chen and Y. Han. Shortest paths on a polyhedron. *International Journal of Computational Geometry and Applications*, 6:127–144, 1996.
- [6] S. Funke and E. A. Ramos. Smooth-surface reconstruction in near-linear time. In *Proc. Thirteenth Symp. on Discrete Algorithms*, pp. 781–790, 2002.
- [7] R. Kimmel and J. Sethian. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Science*, 95(15):8431–8435, July 1998.
- [8] F. Mévoli and G. Sapiro. Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. *Journal of Computational Physics*, 173(2):730–764, 2001.
- [9] F. Mévoli and G. Sapiro. Distance functions and geodesics on submanifolds of  $\mathbb{R}^d$  and point clouds. *SIAM Journal of Applied Mathematics*, 65(4):1227–1260, 2005.
- [10] C. Scheffer and J. Vahrenhold. Learning a 2-manifold with a boundary in  $\mathbb{R}^3$ . In M. Hoffmann, editor, *Proceedings of the 27th European Workshop on Computational Geometry*, pp. 213–216, 2011.
- [11] Y. Schreiber. An optimal-time algorithm for shortest paths on realistic polyhedra. *Discrete & Computational Geometry*, 43:21–53, 2010.