Surface Reconstruction in Almost Linear Time under Locally Uniform Sampling

Tamal K. Dey * Stefan Funke † Edgar A. Ramos ‡

Abstract

We describe an implementation of the COCONE algorithm for smooth surface reconstruction which runs in $O(n \log n)$ time if the sample is "locally uniform" in addition to being good in the sense required for the Cocone algorithm. If the local uniformity condition does not hold, the algorithm still produces a correct result and its worst case running time is $O(n^2 \log n)$. In contrast, the original Cocone algorithm requires $\Omega(n^2)$ time in the worst case, even if the sample is locally uniform.

1 Introduction

We consider the problem of computing a piecewise linear approximation $\tilde{S}$ to a smooth surface S from a set $P \subseteq S$ of n sample points (*samples*). Recently, an algorithm with a correctness guarantee under certain *sampling condition* was proposed by Amenta and Bern [1], and a refined version of it by Amenta *et al.* [2], which is refered as the COCONE algorithm. We review this algorithm in Section 2. Given a "good" sample set P from S, this algorithm outputs a set of triangles in the Delaunay tetrahedrization of P, that form a surface that approximates and is also topologically equivalent to S. Since computing the Delaunay tetrahedrization of S requires $O(n^2)$ time in the worst case [7], the surface reconstruction algorithm has also this quadratic worst case time behavior. On the other hand, since the size of $\tilde{S}$ is linear in n, it remains the question of whether the surface reconstruction can be computed in almost linear time, say $O(n \log n)$. This is important in practice because current scanning techniques produce a number of sample points in the order of several hundred thousands or even up to a million. A linear running time is observed experimentally for the "ball-pivoting" algorithm of Bernardini *et al.* [4]; however, there are neither a correctness guarantee nor a theoretical running time analysis for this algorithm (furthermore, it essentially requires globally uniform sampling). In Section 3, we describe a new implementation of the Cocone algorithm that runs in time $O(n \log n)$ under a "locally uniform" sampling condition, in addition to the original sampling condition. Such a condition seems reasonable for the output of current scanning techniques. Recent work of Erickson [8] shows that there are smooth surfaces with uniform sets of samples that have a Delaunay tetrahedrization of quadratic complexity. Therefore, even if the original Cocone algorithm uses an output sensitive algorithm for computing the Delaunay tetrahedrization [5], it could not match the running time of the new implementation.

2 CoCone Algorithm

2.1 Sampling Condition

The *medial axis* of a surface S in $\mathbb{R}^3$ is the closure of the set of points which have more than one closest point on S. The *local feature size* $\text{lfs}(p)$ at a point $p \in S$ is the least distance from p to the medial axis of S. A set P of sample points from S is said to be an ϵ-*sample* from S if every point $p \in S$ has a sample in P within distance $\epsilon \cdot \text{lfs}(p)$.

2.2 Algorithm

For points o and p, let $\vec{op}$ denote the vector from o to p. Let θ be a parameter with $0 < \theta < \pi/8$. The algorithm of Amenta *et al.* [2] proceeds in four steps:

1. Compute the Voronoi diagram V of P.

2. For each sample $p \in P$ do the following: Let $\tilde{n}_p$, the *estimated normal* at p, be $\vec{pv}$ where v is the furthest Voronoi vertex of the Voronoi cell V_p containing

*Dept. of Computer and Information Science, Ohio State University, Columbus, OH. E-mail: tamaldey@cis.ohio-state.edu

†Max-Planck-Institut für Informatik, Saarbrücken, Germany. E-mail: funke@mpi-sb.mpg.de. Supported by ESPRIT LTR project 28155 (GALIA) and a scholarship 'Graduiertenkolleg' by the Deutsche Forschungsgesellschaft.

‡Max-Planck-Institut für Informatik, Saarbrücken, Germany. E-mail: ramos@mpi-sb.mpg.de. Supported by ESPRIT LTR project 28155 (GALIA).

p (some vector to infinity inside the Voronoi cell if it is unbounded).

3. Put a (Delaunay) triangle t, whose dual Voronoi edge is e_t, in the set T of *candidate triangles*, if for each endpoint p of t, there is a point q in e_t such that the angle between $\tilde{n}_p$ and $\overrightarrow{pq}$ is in $(\pi/2 - \theta, \pi/2 + \theta)$.

4. Extract from T the approximating surface $\tilde{S}$.

Intuitively, Step 3 selects a Delaunay triangle if (i) its normal is "close" to the estimated normal of its vertices, and (ii) it has a circumsphere that is "not far" from being diametral.

For Step 4, time linear in the number of candidate triangles suffices. Therefore, we do not elaborate on it (see [1]) and concern ourselves with a new implementation of Steps 2 and 3.

2.3 Correctness

Amenta *et al.* show that if P is an ϵ-sample from a smooth surface S, where $\epsilon > 0$ is sufficiently small, $\tilde{S}$ approximates S within error ϵ (for $p \in S$, a point in $\tilde{S}$ is within distance $\epsilon \cdot \mathrm{lfs}(p)$) and, furthermore, $\tilde{S}$ is homeomorphic to S.

How small one can choose the parameter θ in the algorithm, depends on ϵ. For our algorithm which we will present in the following, we need a slightly smaller θ, hence enforcing a smaller ϵ to guarantee correctness.

3 The New Algorithm

In addition to the sampling condition mentioned above, which gives a lower bound on how sparse the sample is depending on the local feature size, we add another condition which states that any oversampling must be relatively uniform. With this extra condition, we show that the new algorithm has a running time $O(n \log n)$. We emphasize that if this uniformity condition does not hold, the algorithm still produces a correct output.

3.1 Locally Uniform ϵ-Sample

Let $\alpha \in \mathbb{R}^+$ and $m_0 \in \mathbb{N}^+$ be fixed parameters. We say that $P \subseteq S$ is a *uniform ϵ-sample* if it is an ϵ-sample and, furthermore, it satisfies the following:

Local Uniformity Condition: Let B be the ball with center at $c \in S$ and radius $r \leq \epsilon \cdot \mathrm{lfs}(c)$, and suppose B contains $m \geq m_0$ samples. Then the sphere B' with center at c and radius $2r$ contains at most $4\alpha m$ samples.

Intuitively, this conditions describes that the sampling density must not change too rapidly within a short distance on the surface. It does *not* imply an absolute upper bound on the sampling density anywhere, nor a uniform sampling as defined in [8] (but a sample set that is uniform in [8], is also locally uniform).

Observation 1 *Let $c \in S$ and $r \leq \epsilon \cdot \mathrm{lfs}(c)$. If the ball B centered at c with radius r is empty of samples, then the sphere B' centered at c with radius $b \cdot r$, $b \geq 2$, contains at most $m_0 \cdot (4\alpha)^{2\lceil \log_2 b \rceil}$ samples.*

3.2 Approximate Nearest Neighbors

A *k-th nearest neighbor* (NN) of a point q in P is a point $p \in P$ such that at most $k-1$ other points in P are closer to q than p, and at least $k-1$ other points are no farther from q than p. A *k-th δ-approximate nearest neighbor* (δ-ANN) of q is a point $p \in P$ for which $d(q,p) \leq (1+\delta) \cdot d(q,p')$ where p' is a k-th NN of q; k different points $p_1, p_2, \ldots, p_k \in P$ are *k δ-ANNs* of q if $d(q,p_i) \leq (q, p_{i+1})$, and p_i is an i-th δ-ANN of q.

Observation 2 *Let $p_1, \ldots, p_k$ be k δ-ANNs of q in P with p_k a k-th δ-ANN. Then all $p \in P$ with $d(q,p) \leq d(q,p_k)/(1+\delta)$ are included in $p_1, \ldots, p_{k-1}$.*

In our algorithm we need a data structure that for a query point q reports k δ-ANNs of q in P efficiently. For constant δ, the data structures presented in [3] and [6] can be constructed in time $O(n \log n)$ and report a set of k δ-ANNs in time $O(\log n + k)$. The constant factor in the query time depends on δ, but since we do not need to choose δ too small, this dependency is not important. In fact, $\delta = 1$ suffices.

3.3 Normal Estimation

We cannot afford to estimate the normals via the poles. Rather we estimate the normal at a sample p as follows:

1. Let p_1 be a 1-st δ-ANN of p in $P - \{p\}$.

2. Let p_2 be a 1-st δ-ANN of p among the points in $q \in P$ with $\angle p_1 p q$ between 45^o and 135^o.

3. The estimated normal $\tilde{n}_p$ is equal to the normal of the triangle $\triangle p p_1 p_2$.

Correctness. The two vectors $\overrightarrow{pp_1}$ and $\overrightarrow{pp_2}$ are approximately orthogonal to the normal at p, as guaranteed by the following lemma from [2]:

Lemma 1 *A line segment connecting points $x, x' \in S$ with $|xx'| \leq c\epsilon \cdot \mathrm{lfs}(x)$, $c \leq \sqrt{2}$, makes an angle with the surface normal within $\sin^{-1}(c\epsilon/2)$ from $\pi/2$.*

Therefore, since the angle between these vectors is not too small, the normal to the plane they determine is an approximation of the normal at p:

Lemma 2 *Assuming that P is an ϵ-sample from S,the angle between the real normal n_p at $p \in S$ and the normal $\tilde{n}_p$ estimated by the procedure above is $O(\epsilon)$.*

3.4 Algorithm

In the following, θ_0 is a constant (angle) whose value follows from the analysis. The algorithm uses a parameter $\theta \leq \theta_0$. For a sample p with estimated normal $\tilde{n}_p$, its *cocone region* C_p is the set of points q for which the angle between $\overrightarrow{pq}$ and $\tilde{n}_p$ is in $(\pi/2 - \theta, \pi/2 + \theta)$. We describe an alternative implementation of steps 2 and 3 in the COCONE algorithm.

1. Construct a data structure for reporting δ-ANNs in P.
2. For every sample $p \in P$:
   (a) Let $c \leftarrow 6$ and $p_2 \leftarrow$ nil.
   (b) Repeat while $p_2 =$ nil
   - $N_p \leftarrow$ set of c δ-ANNs of p
   - Let p_1 be the sample nearest to p in N_p
   - Let p_2 be the sample nearest to p in N_p with $\angle p_1 p p_2$ between 45^o and 135^o
   - $c \leftarrow 2 \cdot c$

   (c) $\tilde{n}_p \leftarrow$ normal of $\triangle p p_1 p_2$
   (d) Repeat
   - $N_p \leftarrow$ set of c δ-ANNs of p
   - Compute the Voronoi cell V_p of p with respect to all samples in $N_p \cap C_p$.
   - Let $d_{\max}$ be the maximal distance of a point on a Voronoi edge of V_p within C_p.
   - if $V_p \cap C_p$ is bounded and $\max_{q \in N_p} d(p,q) > 3(1+\delta) d_{\max}$ then exit
   - $c \leftarrow 2 \cdot c$
3. Output a triangle t as candidate if its dual edge appears in V_p within C_p for each vertex p of t.

3.5 Correctness

The algorithm makes sure that the cocone region of each Voronoi cell (with respect to the estimated normal) is computed exactly: further nearest neighbors are added until one can be sure that the cocone region is not affected anymore. Therefore, the same correctness argument as in [1] can be applied.

3.6 Worst-Case Running Time

Note that for each sample p, the Voronoi cell of s with respect to N_p can be computed as the intersection of $|N_p|$ halfspaces which can be done in time O($|N_p| \log |N_p|$). The running time for each cell computation is clearly dominated by the 'last' computation of the Voronoi cell which can be as bad as O($n \log n$). Hence the overall worst-case running time is O($n^2 \log n$).

3.7 Running Time under Local Uniformity Condition

We claim that under the Local Uniformity Condition, the work that has to be done for each sample p is proportional to the number of triangles adjacent to p in the correct reconstruction (apart from a log-factor), hence obtaining a total running time of O($n \cdot \log n$).

Lemma 3 *Let E be the set of Voronoi edges inside the cocone region of a sample p which are dual to triangles of the correct reconstruction of S. If the local uniformity condition is fulfilled, the number of Voronoi vertices inside the cocone region is at most O($m_0 \cdot |E|$).*

Proof. Consider the maximal empty balls centered at the intersection points of the surface and the Voronoi edges of the Voronoi cell of p. For $\theta \leq \theta_0$, clearly all Voronoi vertices within the cocone region must be contained in the union of these empty balls. Consider one of these balls. Note that all Voronoi vertices within this ball have distance at most $2r$ from p. Therefore when growing the ball to a radius of $3r$ definitely all defining samples of these Voronoi vertices are contained in the grown ball. But by Observation 1 this ball can contain at most $m_0 \cdot (\alpha \cdot 4)^{\lceil \log_2 3 \rceil}$ samples. Hence summing over all the maximum empty balls one obtains at most O($m_0 \cdot |E|$) if α is considered as constant.

This lemma implies that computing the Voronoi cell for the O($m_0 \cdot |V_p|$) nearest neighbors of each sample p suffices to be sure to have the *exact* restricted Voronoi cell V_p. Actually, as the following lemma shows, $O(m_0)$ nearest neighbors suffice.

Lemma 4 *Let V_p be the Voronoi cell of p restricted to its cocone and $d_{\max}$ the furthest distance of a voronoi edge of V_p within the cocone region. Then the number of samples which have distance less than $3 \cdot d_{\max} \cdot (1+\delta)$ is O(m_0).*

Proof. Consider a point q on some Voronoi edge of V_p which has distance $d_{\max}$ from p. Let $d'_{\max}$ be the maximal distance of a point q' on some Voronoi edge of V_p intersected with the surface. If $\theta \leq \theta_0$, $d'_{\max} \geq d_{\max}/2$. Clearly the ball centered at q' with radius $d'_{\max}$ must be empty of samples. The same ball grown to a radius of $8 \cdot d'_{\max} \cdot (1+\delta)$ cannot contain more than $m_0 \cdot (\alpha \cdot 4)^{\lceil \log_2 8(1+\delta) \rceil}$ but definitely contains all samples which have distance at most $3 \cdot d_{\max} \cdot (1+\delta)$ from p. [1]

Theorem 1 *Assuming P is a locally uniform $\epsilon-$sample of the surface S, all candidate triangles can be determined in time O($n \cdot \log n$).*

[1] Actually this lemma implies the previous one as it also bounds the number of adjacent samples of p in the Delaunay tetrahedrization within the cocone region.

References

[1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Disc.Comput. Geom.* **22** (1999), 481–504. Earlier version appeared in the 14th Annual ACM Symposium on Computational Geometry, pages 39-48, (1998).

[2] N. Amenta, S. Choi, T. K. Dey and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. In *Proceedings ACM Sympos. Comput. Geom.*, 2000, 213–222. To appear in *International Journal of Computational Geometry and its Applications.*

[3] S. Arya, D. M. Mount, N. S. Netanyahu and R. Silverman. An optimal algorithm for approximate nearest neighbor searching in fixed dimension. *J. ACM* 45(6):891–923, (1998).

[4] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva and G. Taubin. The ball-pivoting algorithm for surface reconstruction. In *IEEE Trans. Visualization and Comput. Graphics* **5** (1999), 349–359.

[5] T.M. Chan, J. Snoeyink, and C.-K. Yap. Output sensitive construction of polytopes in four dimensions and clipped Voronoi diagrams in three. In *Proc. 6th Annu. ACM-SIAM Sympos. Discrete Algorithms*, 282-291, 1995. Journal version in *Disc. Comput. Geom.*

[6] C. Duncan and M. T. Goodrich. Balanced aspect ratio trees: Combining the advanatages of k-d trees and octrees. In *Proceedings of Symp. Discrete Algorithms (SODA 98),* 1998.

[7] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer Verlag, Heidelberg, 1987.

[8] J. Erickson. Nice point sets can have nasty Delaunay triangulations. Available from the author's web site at `www.cs.uiuc.edu`. Submitted to the 17th Annual ACM Symposium on Computational Geometry 2001