A Simple Probablistic Algorithm for Approximating Two and Three-dimensional Objects [1]

Binhai Zhu [2]

1 Introduction

Approximating complex geometric objects with simple ones is an important problem in geometric computing (i.e. in GIS, graphics, image processing). Usually, given an error bound d, to approximate a geometric object O we make two "copies" of O (O_1, O_2) such that the Hausdroff distance between them and O is bounded by d and then we simply compute the minimum size polyhedron (polygon in 2D) between O_1, O_2. (Recall that the *Hausdroff* distance between two objects is the maxmin-distance, or more formally the *sup inf*-distance, between all the points in the two objects.) In two dimension (2D), given a small error bound, optimal linear time algorithms are known to approximate simple polygonal objects [II86, II88, HS91]. All these algorithms are very similar to the result of Suri [Su86]. If the error bound is large such that O_1, O_2 become non-simple, Guibas et al. also present $O(n \log n)$ time algorithm to solve the problem [GHMS93]. In three dimension (3D), the situation is a little different. Even computing the minimum size convex polyhedron between a pair of convex polyhedra is NP-complete [DJ90, DJ92, DG97]. (We call a pair of nested convex polyhedra *convex annulus* throughout this paper.) Nevertheless, several approximation algorithms have been proposed to solve this problem and among them Mitchell and Suri first proposed an $O(n^3)$ time $O(\log n)$-factor approximate solution [MS95]. Later, Clarkson presented a simple randomized algorithm with the same approximate ratio [Cl93] and most recently Brönnimann and Goodrich obtained a constant factor approximate solution for this problem using a beautiful combination of set covers in finite VC-dimension and randomized "natural selection" algorithms [BG95]. However, their algorithm is very slow (O($n^4 \log n$) time) when the optimal solution has size $\geq n^\delta$ since it uses the algorithm of Matousěk et al. as a subroutine which spends $O(n^3)$ time to compute ϵ-nets in a set system [MSW90].

We note that in 2D the algorithms of [II86, II88, HS91] involve computing weakly visible polygon of a segment in a simple polygon and in practice this operation is tedious to implement and in 3D most of the solutions are too slow; therefore, the first motivation of this paper is to present a simple, fast probablistic algorithm to approximate the minimum size convex polyhedron (polygon in 2D) within a pair of nested convex polyhedra (convex annulus). Then we note that given a general simple object in 2D and 3D there exists good practical algorithm to decompose it into (minimum number of) convex pieces (in 2D it is trivial and in 3D although theoretically it is NP-hard the recent algorithm of Chazelle et al. reports exciting results [CDST97]). Consequently, we can approximate general simple objects in 2D and 3D by decomposing them into convex pieces and then use our above approximate algorithm as subroutines.

We use the following 2D example to illustrate our idea in 3D (see Figure 1). Given a 2D convex annulus, i.e., a pair of nested convex polygons P, Q with $Q \subset P$, we want to approximate the minimum size convex polygon contained in $P - Q$. We first generate a set R of r i.i.d. random points in $P - Q$, then we compute the convex hull of $R \cup Q$ and finally output the convex hull as the approximate solution A. Intuitively if $r \to +\infty$ A will have an infinitely large size and if r is very small (like 0 or 1) then A is of roughly the same size as P, Q. In both situations, we are not able to approximate the minimum size convex polygon in $P - Q$. In the next section we show that under a weak condition on P, Q, by choosing $r = O(\log n)$ we can bound the size of A by n (n is the maximum size of P, Q). The actual size of A should be much smaller but we cannot prove this as it involves complicated conditional probability in a geometric domain. Several different empirical results strongly support our claim.

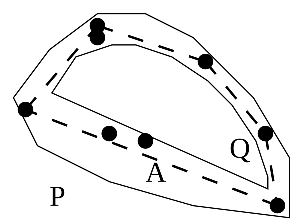

Figure 1: A simple example of approximating minimum size convex polygon in a convex annulus.

2 Theoretical Result

In this section we present our algorithm and then give a proof that under a mild condition the expected approximate solution has size $< n$. Without loss of generality we only study 3D objects in this section. We formally define a convex annulus as the difference between a pair of nested convex polyhedra P, Q, i.e., $P - Q$. P is called the outer polyhedron and Q is called the inner polyhedron. The optimal approximate convex polyhedron O^* is the one which lies totally within the annulus and its size (number of vertices) is minimized. Computing such an optimal minimum size polyhedron O^* is NP-complete. We present the following algorithm to compute an approximate solution for O^*. To make our theoretical result meaningful, we assume that P and Q has the same number of vertices (size) n. We call n the the size of the annulus. We point out that this assumption has no restriction on the algorithm itself.

Algorithm APP(P,Q)

1. Generate a set R of r i.i.d. points in $P - Q$ where $r = C \log n$.
2. Compute the convex hull of $R \cup Q$.
3. Output the computed convex hull A as the approximate solution.

---

[1] This work is partially supported by the research grant No. 9030617 from City University of Hong Kong.

[2] Dept. of Computer Science, City University of Hong Kong, Kowloon, Hong Kong and Dept. of Math and Computer Science, Laurentian University, Sudbury, Ont P3E 2C6, Canada. Email: bhz@cs.cityu.edu.hk.

The running time of Steps (2) and (3) is clearly $O(n \log n)$. The efficiency of Step (1) depends on the implementation (i.e., whether one is willing to do extra preprocessing on P and Q). In the worst case (as what we have done in our implementation), we perform no preprocessing on P, Q and we generate random points uniformly within the minimum volume axis-parallel bounding box B of P. To generate one random point with $P - Q$ it takes expected $O(n \frac{volume(B)}{volume(P-Q)})$ time (as we have to spend $O(n)$ time to check whether it is within $P - Q$ or not) and clearly to generate $O(\log n)$ number of random points in $P - Q$ it takes $O(\frac{volume(B)}{volume(P-Q)} n \log n)$ time. Therefore, if we do not perform any preprocessing on P, Q our algorithm takes expected $O(\frac{volume(B)}{volume(P-Q)} n \log n + n \log n) = O(n \log n)$ time.

We repeat that the performance of above algorithm is very intuitive: if we choose r to be very small (like 1 or 2) then the size of A is almost n so we have no approximate solution at all, but if r is too big then with the result of Rényi and Sulanke [RS63] the size of A goes to infinite. So r has to be something in between. In the proof of the following theorem we in fact prove why r must be bounded by $O(\log n)$. Without loss of generality, we can relax the condition for the points in R: they can be under the (α, β)-measure, i.e.,$\mathbf{P}[P-Q]=1$ and $\alpha\gamma(S) \leq \mathbf{P}[S] \leq \beta\gamma(S)$ for every measurable subset S of $P - Q$, where γ is the usual Lebesgue measure. We state the result as follows.

Theorem 1 *Given a well-shaped convex annulus with size n, by choosing $r = C \log n$ (C is a universal constant only related to the shape of P, Q) the expected size of the approximate convex polyhedron computed by Algorithm APP(P,Q) is less than n.*

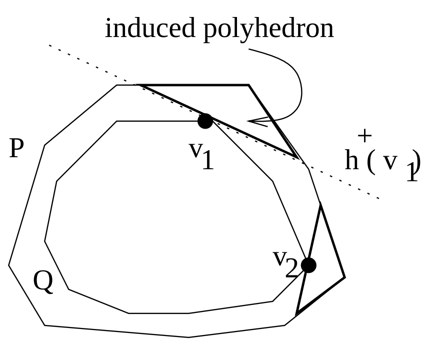

Figure 2: Illustration for the definition of "well-shaped" convex annulus.

The term of "well-shaped" is defined as follows. First of all we note that among the vertices of A there are some interesting ones: some of them are the vertices of Q. For each vertex $v \in Q$ we compute the plane $h(v)$ through v such that the intersection of the halfspace through $h(v)$ not containing Q, $h^+(v)$, and $P - Q$ has the minimum volume (see Figure 2). We call this volume, which must be larger than a constant (to be specified below), the *induced volume* of v over P, Q (induced volume of v, for short), $h^+(v) \cap (P - Q)$ is called the induced polyhedron of v and v is called an *induced vertex* of P, Q. Clearly, as the points in R are independently identically distributed (i.i.d.) if an induced vertex v_1 has a large induced polyhedron then it has a larger chance to contain some i.i.d. points in R hence has less chance of being a vertex of A. However, we note that the induced polyhedra of the vertices of Q might intersect each other and for easy probabilistic analysis we are only interested in the maximum set of *independent induced polyhedra*, i.e., the maximum set of induced polyhedra none of which intersect each other. (If we take each induced polyhedron as a vertex in a graph $\mathcal{G}$ and there is an edge between two vertices in $\mathcal{G}$ if and only if the two corresponding induced polyhedra intersect each other, then the maximum set of independent induced polyhedra corresponds exactly to the *maximum independent set* of $\mathcal{G}$.)

Definition. A convex annulus P, Q is *well-shaped* if Q contains a set of at least $c_1 \log n$ independent induced polyhedra with volume at least $a_1 d^3$ where d is the Hausdroff distance between P, Q and c_1, a_1 are universal constants only related to the shape of P, Q.

Now we proceed to prove the theorem.

Proof: Note that a vertex of Q is a vertex of A if and only if there exist two points p, q in R such that the intersection of the halfspace through the plane $\triangle(pqv)$ and not containing Q, with $P - Q$ contains no other points in R. Obviously this intersection region (polyhedron) has a volume at least as large as the induced volume of v. If P, Q is well-shaped then among the corresponding maximum set of induced vertices the expected number of A vertices is bounded by

$$(c_1 \log n) \cdot \beta \cdot \binom{r}{2} \cdot (1 - a_1 d^3)^{r-2} < (c_1 \log n) \cdot \beta \cdot n^2 \cdot (1 - a_1 d^3)^{r-2}.$$

There are some other points which can be the vertices of A and their number is bounded by the size of R, r, plus the number of non-induced vertices of Q, $n - c_1 \log n$. Therefore, the expected number of vertices of A is bounded by

$$(c_1 \log n) \cdot \beta \cdot n^2 \cdot (1 - a_1 d^3)^{r-2} + r + (n - c_1 \log n).$$

This is bounded by

$$(c_1 \log n) \cdot \beta \cdot n^2 \cdot (1 - a_2)^{r-2} + r + (n - c_1 \log n)$$

with $a_2 = a_1 d^3$ and this can further be bounded by

$$(c_1 \log n) \cdot \beta \cdot n^2 \cdot e^{-a_2(r-2)} + r + (n - c_1 \log n).$$

If we choose $\frac{2 \log n + \log \log n}{a_2 \log e} + 2 \leq r \leq (c_1 - \delta) \log n$ (δ is an arbitrary small constant) then this number is bounded by

$$c_1 \beta + (c_1 - \delta) \log n + (n - c_1 \log n),$$

which is

$$n - \delta \log n + c_1 \beta < n.$$

□

We note that the above theorem is closely related to the term of "well-shaped". In fact, if we set the condition that Q has $c'n$ ($c' < 1$) number of independent vertices then we can prove A has an expected number of at most cn vertices with $c < 1$. We believe that the actual expected number of vertices of A is much smaller but we are not able to prove this as when the induced polyhedra intersect each other it is difficult to compute the (conditional) probability that a vertex of Q is a vertex of A. But our empirical results, obtained over different classes of 3D convex polyhedra and 2D convex polygons, strongly support our claim.

3 Empirical Results

In this section we present some empirical results. We first test our approximate subroutine in 3D, i.e. testing the performance of the algorithm on 3D annulus. We then test the algorithm on approximating 2D monotone polygonal chains. Lastly we comment on how to use our algorithm to approximate (simplify) polyhedral terrains. We would like to mention that the running time of our algorithm is not a problem in our implementation, as long as the input is of reasonable size, and in fact the running time is likely to be dominated by generating random points in the annulus instead of computing convex hulls if the input size is huge and if we employ a brute force method to test whether a random point is within the annulus.

3.1 Empirical results on nested convex polyhedra

In testing the algorithm on 3D annulus (nested convex polyhedra) we try three classes of convex polyhedra: Delaunay polyhedra, s-polyhedra (obtained by generating random points in the sphere $x^2 + y^2 + z^2 = 1$ and then compute their convex hull) and q-polyhedra (obtained by generating random points in the qube bounded by (-1,-1,-1) and (1,1,1) and then compute their convex hull). In each of these situations, P, Q are isomorphic (i.e. we can obtain P from Q by a scaling and vice versa).

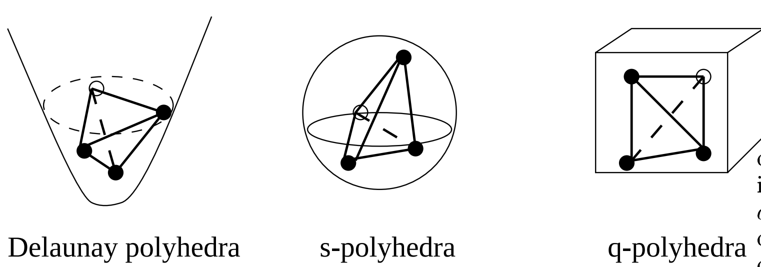

Figure 3: Illustration for the constructing of various polyhedra.

Delaunay polyhedra are obtained by projecting each point (x, y) in the plane to the parabola $z = x^2 + y^2$ and then computing the convex hull of all the projected 3D points. Because of convexity, all the projected points are on the Delaunay polyhedra. It is therefore easy to know the size of the Delaunay polyhedra P, Q. We mainly test our algorithm for Delaunay polyhedra with size 10,000. We start with generating 25 random points in $P - Q$ and each time we increase the number of random points generated by 25 and for each case we repeat 10 times and compute the mean (average) of the size of A. The 2D points are generated within square bounded by (-1.1,-1.1) and (1.1,1.1) then we make two "scaled copies", one bounded by (-1,-1) and (1,1), the other bounded by (-1.2,-1.2) and (1.2,1.2). P, Q are the convex hull of the 3D projected points form the latter two copies of points. In this case, the Hausdroff distance between P, Q is 0.2. The results are reported in the following table. It is clear that using our algorithm the optimal size of A is obtained when roughly 100 random points are generated.

r (size of R)	25	50	75	100	125	150	175	200
Size of A	41	34	33	29	32	36	36	37

Table 1. Empirical results for Delaunay polyhedra with size 10K, $d = 0.1$.

We also test the algorithm on Delaunay polyhedra under the same condition as above, except that the Hausdroff distance between P, Q is now 0.02. The results are reported in Table 2. In this situation, using our algorithm the optimal size of A is obtained when roughly 75 random points are generated.

r (size of R)	25	50	75	100	125	150	175	200
Size of A	28	28	26	27	31	32	34	34

Table 2. Empirical results for Delaunay polyhedra with size 10K, $d = 0.01$.

As described above, s-polyhedra are the convex hull of random points in a unit sphere. We generate 100K points in the unit sphere, but the number of vertices in the corresponding s-polyhedra are roughly 1400. We report in the following table the corresponding empirical results when the Hausdroff distance between P, Q is 0.2. Again, we take average over ten tries.

Size of P, Q	1441	1411	1404	1431
r (size of R)	50	100	150	200
Size of A	34	29	30	41

Table 3. Empirical results for s-polyhedra generated from 100K points.

We remark that we have tried larger size polyhedra occasionally and similar (usually better) results hold. For instance, for Delaunay polyhedra with 30K points and $d = 0.01$ if we add 500 random points the resulting size of A is only 37. In this case the approximate ratio is only about 0.0012. We also obtain some empirical results for q-polyhedra. But since when we choose 100K random points the resulting q-polyhedra has size of only about 200, the performance of the approximate algorithm is not easy to judge. When we choose more than 100K random points it takes longer and longer time to compute the subsequent q-polyhedra.

In this implementation, we use O'Rourke's program to compute 3D convex hulls [O'R94]. This program is very robust (no numerical error is ever encountered in our testing) but it is fairly slow when all the points are on their convex hull.

3.2 Empirical results on approximating monotone chains

In this section, we present some empirical results on approximating monotone chains (functions) given a fixed error bound. Unlike results in the previous subsection, we need to first decompose a given monotone chain (function) into convex or concave subchains and then approximate each subchain with APP(P,Q).

Because of the space constraint, we omit all the details in this subsection. But our empirical results show that for reasonably complex monotone chains, our algorithm can obtain approximate ratios between 22% and 32%.

3.3 Some comments on approximating polyhedral terrains

Approximating (simplifying) polyhedral terrains is a very important problem in GIS and spatial databases. Theoretically this problem is NP-hard [AS94] and some provably good approximations are known [AS94, AD97]. Most previous work are based on heuristics (see [AD97] for a complete list of references). We believe our algorithm, combined with the recent practical convex decomposition algorithm of Chazelle et al. [CDST97], will produce good practical results. As pointed out by Chazelle et al. in practice most of the surfaces only consist of a constant number of convex pieces. Therefore, this idea should work well and besides that, both the heuristic algorithm of Chazelle et al. and ours runs in $O(n \log n)$ time so the running time should not be causing any problem even if the data has a huge size. We note that the approximation algorithm of [AS94] has a time complexity of $O(n^8)$ and the one proposed in [AD97] runs in $O(n^{2+\delta})$ time.

4 Concluding Remarks

In this paper we present a very simple probablistic algorithm to approximate 2D and 3D geometric objects. This algorithm is very easy to implement and the empirical results are very promising especially for 3D objects. Nevertheless, this work also raises several problems. (1) What will this algorithm perform in approximating (simplifying) polyhedral terrains? Although we believe that the result will be excellent if we employ the practical algorithm of Chazelle et al. to decompose a polyhedral terrain into convex pieces, the actual performance will only be known after this work is performed. (2) Our Theorem 1 only proves a very weak theoretical result. Is it possible to apply deeper probablistic analysis to obtain stronger theoretical results?

5 Acknowledgements

I thank Cao An Wang, D.T. Lee for comments in the early stage of this work.

References

[AD97] P. Agarwal, and P. Desikan. An efficient algorithm for terrain simplification. In *Proc. 8th Sympos. Discrete Algorithms*, pages 139–147, 1997.

[AS94] P. Agarwal, and S. Suri. Surface approximation and geometric partitions. In *Proc. 5th Sympos. Discrete Algorithms*, pages 24–33, 1994.

[ABOSY89] A. Agarwal, H. Booth, J. O'Rourke, S. Suri and C.K. Yap. Finding minimal convex nested polygons. *Info. and Comput.*, 83:98–110, 1989.

[BG95] H. Brönnimann and M. Goodrich. Almost optimal set covers in finite VC-dimension. *Disc. Comput. Geom.*, 14:463–479, 1995.

[Cl93] K. Clarkson. Algorithms for polytope covering and approximation. In *Proc. 3rd WADS (LNCS 709)*, pages 246–252, 1993.

[CDST97] B. Chazelle, D. Dobkin, N. Shouraboura and A. Tal. Strategies for polyhedral surface decomposition–An experimental study. *Comput. Geom. Theory Appl.*, 7:327–342, 1997.

[DJ90] G. Das and D. Joseph. The complexity of minimum convex nested polyhedra. In *Proc. 2nd Canadian Conf. on Comput. Geom.*, pages 296–301, 1990.

[DG97] G. Das and M. Goodrich. On the complexity of optimization problems for 3-dimensional convex polyhedra and decision trees. *Comput. Geom. Theory Appl.*, 8:123–137, 1997.

[DJ92] G. Das and D. Joseph. Minimum vertex hulls for polyhedral domains. *Theo. Comp. Sci.*, 103:107–135, 1992.

[GHMS93] L. Guibas, J. Hershberger, J. Mitchell and J. Snoeyink. Approximating polygons and subdivision with minimum link paths. *Intl. J. Comput. Geom. and Appl.*, 3:383–415, 1993.

[HS91] S. Hakimi and E. Schmeichel. Fitting polygonal functions to a set of points in the plane. *CVGIP: Graph. Mod. Image Process.*, 53(2):132–136, 1991.

[II86] H. Imai and M. Iri. An optimal algorithm for approximating a piecewise linear function. *J. Inform. Process.*, 9(3):159–162, 1986.

[II88] H. Imai and M. Iri. Polygonal approximations of a curve-formulation and algorithms. In *Computational Morphology*, G. Toussaint ed., North Holland, Amsterdam, pages 71–86, 1988.

[MS95] J. Mitchell and S. Suri. Separation and approximation of polyhedral objects. *Comput. Geom. Theory. Appl.*, 5:95–114, 1995.

[MSW90] J. Matoušek, R. Seidel and E. Welzl. How to net a lot with little: small ϵ-nets for disks and halfspaces. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 16–19, 1990.

[O'R94] J. O'Rourke. Computational Geometry in C. Cambridge University Press, 1994.

[RS63] A. Rényi and R. Sulanke. Ueber die konvexe Hulle von n zufallig gewahlten Punkten, I. *Z. Wahrschein*,2:75–84, 1963.

[Su86] S. Suri. A linear time algorithm for minimum link paths inside a simple polygon. *Comput. Vision Graph. Image Process.*, 35:99–110, 1986.