Cutting a Convex Polyhedron Out of a Sphere *

Syed Ishtiaque Ahmed, Masud Hasan, and Md. Ariful Islam

Department of Computer Science and Engineering Bangladesh University of Engineering and Technology Dhaka-1000, Bangladesh http://www.buet.ac.bd/cse ishtiaque@csebuet.org, masudhasan@cse.buet.ac.bd, arifulislam@csebuet.org

Abstract. Given a convex polyhedron P of n vertices inside a sphere Q, we give an $O(n^3)$ -time algorithm that cuts P out of Q by using guillotine cuts and has cutting cost $O(\log^2 n)$ times the optimal.

Keywords: Approximation algorithm, guillotine cut, polyhedra cutting

1 Introduction

The problem of cutting a convex polygon P out of a piece of planar material Q (P is already drawn on Q) with minimum total cutting length is a well studied problem in computational geometry. The problem was first introduced by Overmars and Welzl in 1985 [13] but has been extensively studied in the last decades [2–5,8,9,11,13–15] with several variations, such as P and Q are convex or non-convex polygons, Q is a circle, and the cuts are line cuts or ray cuts. This type of cutting problems have many industrial applications such as in metal sheet cutting, paper cutting, furniture manufacturing, ceramic industries, fabrication, ornaments, and leather industries. Some of their variations also fall under *stock cutting problems* [4].

If Q is another convex polygon with m edges, this problem with line cuts has been approached in various ways [3–6,9,10,13,14]. If the cuts are allowed only along the edges of P, Overmars and Welzl [13] proposed an $O(n^3 + m)$ -time algorithm for this problem with optimal cutting length, where n is the number of edges of P. The problem is more difficult if the cuts are more general, i.e., they are not restricted to touch only the edges of P. In that case, Bhadury and Chandrasekaran showed that the problem has optimal solutions that lie in the algebraic extension of the input data field [4], and due to this algebraic nature of this problem, an approximation scheme is the best that one can achieve [4]. They also gave an approximation scheme with pseudo-polynomial running time [4].

After the indication of Bhadury and Chandrasekaran [4] to the hardness of the problem, many people have given polynomial time approximation algorithms. Dumitrescu proposed an $O(\log n)$ -approximation algorithm with $O(mn+n\log n)$ running time [9,10]. Then, Daescu and Luo [6] gave the first constant factor approximation algorithm with ratio 2.5 + ||Q||/||P||, where ||P|| and ||Q|| are the

^{*} An earlier version appeared in Proc. WALCOM 2010, LNCS, Springer, 2010.

perimeters of P and the minimum area bounding rectangle of P respectively. Their algorithm has a running time of $O(n^3 + (n + m) \log (n + m))$. The best known constant factor approximation algorithm is due to Tan [14] with an approximation ratio of 7.9 and running time of $O(n^3 + m)$. In the same paper [14], the author also proposed an $O(\log n)$ -approximation algorithm with improved running time of O(n + m). As the best known result so far, very recently, Bereg, Daescu and Jiang [3] gave a polynomial time approximation scheme (PTAS) for this problem with running time $O(m + \frac{n^6}{\epsilon^{12}})$. Recently, Ahmed et.al. [2] have given similar constant factor and $O(\log n)$ -factor approximation algorithms where Qis a circle. As observed in [2], algorithms for Q being a convex polygon are not easily transferred for Q being a circle, as the running time of the formers depend upon the number of edges of Q.

For ray cuts, Demaine, Demaine and Kaplan [8] gave a linear time algorithm to decide whether a given polygon P is *ray-cuttable* or not. For optimally cutting P out of Q by ray cuts, if Q is convex and P is non-convex but ray-cuttable, then Daescu and Luo [6] gave an almost linear time $O(\log^2 n)$ -approximation algorithm. If P is convex, then they gave a linear time 18-approximation algorithm. Tan [14] improved the approximation ratio for both cases as $O(\log n)$ and 6, respectively, but with much higher running time of $O(n^3 + m)$. See Table 1 for a summary of these results.

Dim.	Cut Type	Q	P	Approx. Ratio	Running Time	Reference
2D	Line	Convex	Convex	$O(\log n)$	$O(mn + n \log n)$	[9, 10]
		Convex	Convex	2.5 + Q / P	$O(n^3 + (n+m)\log(n+m))$	[6]
		Convex	Convex	7.9	$O(n^3 + m)$	[14]
		Convex	Convex	$(1 + \epsilon)$	$O(m + \frac{n^6}{\epsilon^{12}})$	[3]
		Circle	Convex	$O(\log n)$	O(n)	[2]
		Circle	Convex	6.48	$O(n^3)$	[2]
	Ray	-	Non-convex	Ray-cuttable?	O(n)	[8]
		Convex	Convex	18	O(n)	[6]
		Convex	Non-convex	$O(\log^2 n)$	O(n)	[6]
		Convex	Convex	6	$O(n^3 + m)$	[14]
		Convex	Non-convex	$O(\log n)$	$O(n^3 + m)$	[14]
3D	Hot-wire	-	Non-convex	Cuttable?	$O(n^5)$	[11]
	Guillotine	Sphere	Convex	$O(\log^2 n)$	$O(n^3)$	This paper

 Table 1. Comparison of the results.

Our results The generalization of this problem in 3D is very little known. To the best of our knowledge, the only result is to decide whether a polyhedral object can be cut out form a larger block using continuous hot wire cuts [11]. In this paper we attempt to generalize the problem in 3D. We consider the problem of cutting a convex polyhedron P which is fixed inside a sphere Q by using only guillotine cuts with minimum total cutting cost. A guillotine cut, or simply a cut, is a plane that does not pass through P and partitions Q into two smaller convex pieces. After a cut is applied, Q is updated to the piece that contains P.

The *cutting cost* of a guillotine cut is the area of the newly created face of Q. We give an $O(n^3)$ -time algorithm that cuts P out of Q by using only guillotine cuts and has cutting cost no more than $O(\log^2 n)$ times the optimal cutting cost. Also see Table 1.

The rest of the paper is organized as follows. We give some preliminaries in Section 2, then Section 3 gives the algorithms, and finally Section 4 concludes the paper with some future work.

2 Preliminaries

A cut is a *vertex/edge/face cut* if it is tangent to P at a single vertex/a single edge/a face respectively. We call P to be *cornered* (within Q) if it does not contain the center o of Q, otherwise it is called *centered*. For cornered P, the *D*-separation of P is the minimum-cost (single) cut that separates P from o.

We represent an orthogonal projection of a convex polyhedron P by the corresponding projection direction coming towards the origin from a view point at infinity. A face f of P is visible in an orthogonal projection if the view point lie in the half space that is defined by the supporting plane of f and does not contain P. An orthogonal projection of P is called non-degenerate if the projection direction is not parallel to any face of P. An orthogonal projection of P is a convex polygon. If the projection is non-degenerate, then each edge of the projected convex polygon corresponds to an edge of P.

3 The algorithm

Let C^* be the optimal cutting cost. We shall have two phases in our algorithm: box cutting phase and carving phase. In the box cutting phase, we shall cut a minimum volume rectangular box B containing P out of Q with cutting cost no more than a constant factor of C^* . Then in the carving phase we shall cut P out of B with cutting cost bounded by $O(\log^2 n)$ times of C^* .

3.1 Box cutting phase

We first deal with cornered P. If P is cornered, we shall first apply the D-separation to Q. The following lemma gives a characterization of the D-separation, which will help finding it quickly.

Lemma 1. For cornered P, let x be the closest point of P from o. Then the D-separation of P is the plane that is perpendicular to the line segment \overline{ox} at x.

Proof. A D-separation is a tangent to P that separates P from o and is farthest form o. Let π be the plane that is perpendicular to the line segment \overline{ox} at x. To prove that π is the D-separation, we first prove that π is tangent to P. Suppose not. Then there exists some portion of P in the neighborhood of x that lies in

the half space of π containing o. Then there must be a point y in that portion that is closer to o than x, which is a contradiction that x is closest to o.

We next prove that π is the farthest tangent of P from o that separates x from o. To separate x from o, π must intersect \overline{ox} . Now, any other plane that intersects \overline{ox} and is not perpendicular to \overline{ox} at x is closer to o than π . Therefore, π is the farthest.

Observe that since P is convex, the closest point x of P from o is unique, and therefore, the D-separation of P is also unique. However, x can be a vertex, or a point of an edge or a face.

Lemma 1 The D-separation can be found in O(n) time.

Proof. By Lemma 1, we need to find the closest point x of P from o. We first find the closest vertex v from o in O(n) time. Then for each edge e, we find the closest point o_e of e from o as follows: Let l_e be the line passing through e. Draw a line segment $\overline{oo'}$ perpendicular to l_e . If o' is a point of e, then o_e is o', otherwise o_e is the end point of e that is closer to o'. Finding o_e can be done in constant time. For all edges of P, it takes O(n) time. Similarly, for each face f, we find the closest point o_f of f from o as follows: Let π_f be the supporting plane of f. Draw a line segment $\overline{oo'}$ perpendicular to π_f . If o' is a point of f, then o_f is o', otherwise o_f is among the closest point of the edges of f or among the vertices of f. Finding o_f can be found in $O(d_f)$ time, where d_f is the number of edges of f. For all faces of P, it takes $\sum_f O(d_f) = O(n)$ time. Finally, x is the closest among v, all o_e 's and o_f 's.

For cornered P, after the D-separation is applied, Q is a spherical segment and let r be the radius of the base circle of that segment. The following lemma relates for cornered P the cost of D-separation and C^* .

Lemma 2 For cornered P, cost of the D-separation, which is πr^2 , is at most C^* .

Proof. Consider an optimal cutting sequence C with cutting cost C^* . C must separate P from o. However, it may do that by using a single cut or by using more than one cut. If it uses a single cut, then it is in fact doing the the D-separation, since the D-separation is the minimum cost single cut that can separate P from o. Therefore, $C^* \geq \pi r^2$.

If C uses more than one cut, then let $C = C_1, C_2, \ldots, C_k, \ldots$ with C_k being the first cut that separates o from P. Observe that C_k can not be the very first cut of C, since otherwise it is doing a D-separation and we are in the previous case. Replace $C = C_1, C_2, \ldots, C_k$ by a single cut C_k' whose plane is the same as that of C_k . We will show that cost of C_k' is smaller than the total cost of C_1, C_2, \ldots, C_k .

Consider the first two cuts C_1 and C_2 . Replace C_1 and C_2 by a single cut C_2' whose plane is the same as that of C_2 . Since C_1 does not separate P from o, the portion of Q that is created due to C_1 and that does not contain P is no larger

than a half sphere of Q. It implies that the portion of C_2' that is not present in C_2 is smaller than C_1 (also see Fig. 1). That means the cost of C_2' is smaller than the total cost of C_1 and C_2 . Similarly, we can show that replacing C_2' and C_3 with a single cut C_3' in the plane of C_3 has smaller cost than the total cost of C_2' and C_3 . Repeating this for k-1 times would show that C_k' has smaller cost than the total cost of C_1, C_2, \ldots, C_k . Therefore, using more than one cut to separate P from o is even worse than using a single cut, and we already proved that an optimal way to use a single cut is to use the D-separation. Thus the lemma holds.



Fig. 1. 2D view of C_1, C_2 and C'_2 . Broken line represents the part of C'_2 that is not in C_2 ; This portion is smaller than C_1 .

We now deal with centered P. A lemma that is similar to the previous one and gives lower bound for centered P is the following.

Lemma 3 For centered P, it holds that $C^* \ge \pi R^2$, where R is the radius of Q.

Proof. For centered P, Q remains a sphere. Since P contains the center o of Q, any cutting sequence, starting from the boundary of Q, must "wrap" P and finally get out of Q by a plane different from the starting plane. That means the wrapping must enclose the center o. In the best case when P is simply a point that lies in the center o of P, the cutting sequence must traverse at least $\frac{1}{2}\pi R^2$ area to reach P and then to traverse another $\frac{1}{2}\pi R^2$ area to finish the cutting. In the worst case, when P is almost the sphere Q, the sequence must traverse the whole area of Q, which is $4\pi R^2$.

We next find a minimum volume rectangular bounding box B of P in $O(n^3)$ time by the algorithm of O'Rourke [12]. Then we cut out this box from Q by applying six cuts along the six faces of B.

Lemma 4 Cost of cutting B out of Q is at most $3C^*$ for cornered P and at most $4C^*$ for centered P.

Proof. Consider Q before B was cut out of it. We denote the area of Q by $|Q|^1$. For cornered P, since Q is no bigger than a half sphere, it holds that $|Q| \leq 3\pi r^2$, which by Lemma 2 becomes $|Q| \leq 3C^*$. For centered P, since we do not apply D-separation, we have $|Q| = 4\pi R^2$, which by Lemma 3 becomes $|Q| \leq 4C^*$. While cutting along the faces of B, for each cut c let Q' be the portion of Q that does not contain P. Let q' be the portion of the surface of Q' that is "inherited" from Q, i.e., that was a part of the surface of Q. One important observation is that the cost of c is no more than the area of q'. Moreover, over all six cuts, sum of these inherited surface area is |Q|. Therefore, the lemma holds.

Once the minimum area bounding box B has been cut, a lower bound on C^* can be given in terms of the area of B.

Lemma 5 $C^* \geq \frac{1}{6}|B|$, where |B| is the area of B.

Proof. Let h be a maximum area face of B. Project P orthogonally from the direction perpendicular to h. P projects to a convex polygon X. In this projection, h is the minimum area bounding rectangle of X, since otherwise we could rotate the four faces of B that are not perpendicular to h and would get a bounding rectangle smaller than h, which in turn would give a bounding box smaller than B, but that would be a contradiction that B is the smallest bounding box. It implies that the area of X is at least $\frac{1}{2}|h|$. Now, C^* is at least twice the area of X, and $|B| \leq 6|h|$. Therefore, $C^* \geq 2|X| \geq 2 \cdot \frac{1}{2}|h| \geq \frac{1}{6}|B|$.

3.2 Carving phase

Let $T = B \setminus P$ be the portion of B that we would achieve if P were removed from B. Then, T is a polyhedral object. T may be convex or non-convex and possibly disconnected. The *outer* surface of T is the surface of T that existed in B when P was not removed from B. Our idea is to apply an edge cut through each edge of P, and we shall do that in two types of rounds: *face rounds* and *edge rounds*. Face rounds will find polygonal chains that will partition the faces of P into smaller sets and edge rounds will apply edge cuts through the edges of those polygonal chains. There will be $O(\log n)$ face rounds, and within each face round there will be a number of edge rounds but their total cost will be $O(C^* \log n)$. Once we have applied edge cuts through all the edges of P, each face f of P will have a small "cap"-like portion of T over it, which we shall cut at a cost of the area of f to get P, giving a cost of $O(C^*)$ for all faces.

Face rounds Let F be a set of faces of P. From now on, we use the term *face* set to represent a set of faces of P. At the very first face round i = 0, F consists of all the faces of P. We find a chain of edges P' that will partition F into two smaller face sets F_1 and F_2 by the following lemma.

¹ In the subsequent text, we use $|\cdot|$ to denote the area of a 3D object.

Lemma 6 Let l be the number of faces in a face set F. It is always possible to find in $O(l \log l)$ time a non-degenerate orthogonal projection of P such that the two sets of visible and invisible faces of F contain at least $\lfloor \frac{l}{2} \rfloor$ faces each.

Proof. For this proof we shall move on to the surface of an origin-centered sphere s. For each face $f \in F$, its normal point is the intersection point of s and the outward normal vector of f when the vector is translated to the origin. Each point of s also represents an orthogonal projection direction of P. So, a nondegenerate orthogonal projection of P can be represented by a great circle of sthat does not pass through the normal points of the faces of P. We need one such great circle satisfying an additional criterion that its two hemispheres contain at least $\lfloor \frac{l}{2} \rfloor$ normal points each. There exists infinitely many such great circles and one of them can be found in $O(l \log l)$ time as follows. Take as *poles* any two antipodal points that are not normal points of the faces of P. Take a great circle q through these two poles and rotate it around these poles until the number of normal points in its two hemisphere differ by at most one. If it happens that some normal points fall on q when we stop, then slightly change the poles as well as q so that the normal points on q are distributed into two hemispheres as necessary. For running time, all we need to do is to sort the normal points according to their angular distance with the plane of initial position of q at the origin. The resulting projection is the one from the perpendicular direction of the plane of final position of q.

The projection direction achieved by the above lemma is called the *zone* direction of F. P' is the chain of edges in the boundary of the above projection whose corresponding edges in P have both adjacent faces (one is visible and another is invisible) in F. We call P' a separating chain of F. F_1 and F_2 are the two sets of faces separated by P'. In the next face round i + 1, we shall apply Lemma 6 for each of F_1 and F_2 recursively and thus get two separating chains and four face sets. We shall repeat the same procedure for each of these four face sets. We shall continue like this until each face set has only one face. Clearly, we need $O(\log n)$ face rounds.

Edge rounds Let $P' = e_1, e_2, \ldots, e_k$ be the separating chain of a particular face round with its two ends, which are two vertices of e_1 and e_k , touching the outer surface of T. Observe that for the very first face round i = 0, P' is a cycle and the two ends are the same. We shall apply edge cuts through the edges of P' such that all of them are parallel to a particular direction. Such a direction can be the corresponding zone direction. We shall call this set of k edge cuts a zone of cuts and their direction of cuts the zone cut direction. We shall apply these cuts in $O(\log k)$ edge rounds. At the very first edge round j = 0, we apply an edge cut through $e_{k/2}$ in the zone cut direction. This cut will partition the edges of P' into two sub chains of size at most $\lfloor \frac{k}{2} \rfloor$. In the next edge round j+1, we apply two edge cuts through the two middle edges of these two sub chains, which will result into four sub chains. Then in the next round we apply four

similar cuts to the four sub chains. We continue like this until each sub chain has only one edge. Clearly, we need $O(\log k)$ edge rounds for P'.

Lemma 7 After all the face rounds and their corresponding edge rounds are completed, all edges of P get an edge cut.

Proof. Let e be an edge that does not get an edge cut through it. Then the two adjacent faces of e are in the same face set. But that is a contradiction that each face set has only one face.

Analysis We are now ready to find the approximation ratio and the running time of our algorithm.

We define the *box area* of a face set F as follows. When F contains all faces of P, its box area is B—the whole surface area of B. Zone of cuts through the separating chain of F partitions F into F_1 and F_2 and T into two components, say T_1 and T_2 , respectively. Then the *box area* of F_1 (F_2) is the outer surface area of T_1 (T_2), which we denote by by B_1 (B_2). Observe that $|B_1| + |B_2| \le |B|$. Box area of any subsequent face set is similarly defined. Moreover, two face sets from the same face round have their box areas disjoint and in any face round sum of all box area is at most |B|.

The following lemma bounds the cutting cost of an edge rounds in a particular face round.

Lemma 8 Let P'_m be the separating chain with k edges of an arbitrary face set F_m to which we apply $O(\log k)$ edge rounds. Let B_m be the box area of F_m . At each edge round j, total cost of 2^j cuts is $O(|B_m|)$. Over all $O(\log k)$ edge rounds, total cost is $O(|B_m| \log n)$.

Proof. This proof is similar to that of Lemma 4. Consider a particular edge round j. For each cut c the cost of c is no more than the portion of B_m that is thrown away by c. Moreover, these cuts are pairwise disjoint, since they can at best intersect the cut which is in between them and was applied in (j - 1)-th round. It implies that the total cost of 2^j cuts is at most $|B_m|$. Since $k \leq n$, the second part of the lemma follows.

The next lemma bounds the total cutting cost over all face rounds.

Lemma 9 At each face round *i*, total cost of 2^i zones of cuts is $O(|B| \log n)$. Over all $O(\log n)$ face rounds, the total cost is $O(C^* \log^2 n)$.

Proof. At each face round *i*, we apply 2^i zones of cuts to 2^i face sets. By the previous lemma, for a particular face set F_m , $0 \le m \le 2^i$, cost of the zone of cuts applied to it is at most $O(|B_m| \log n)$. Since $\sum_{1}^{2^i} |B_m| \le |B|$, cost of all zone cuts is $\sum_{1}^{2^i} O(|B_m| \log n) = O(|B| \log n)$. Over all $O(\log n)$ face rounds, the total cost is $O(|B| \log^2 n)$, which by Lemma 5 is $O(C^* \log^2 n)$.

We now see the running time of our algorithm. Running time in face round i involves finding 2^i separating chains, each of size $\frac{n}{2^i}$, plus applying a zone of cuts to each of them. Each separating chain can be found in $O(\frac{n}{2^i} \log \frac{n}{2^i})$ time by Lemma 6. Each cut needs to update Q, which can be done in O(n) time assuming that Q is represented by suitable data structures [7]. It gives that a zone of cuts needs $O(\frac{n^2}{2^i})$ time. So, in round i total time is $O(2^i(\frac{n^2}{2^i} + \frac{n}{2^i}\log \frac{n}{2^i})) = O(n^2)$. Over all $O(\log n)$ rounds, it becomes $O(n^2 \log n)$.

We summarize the result in the following theorem.

Theorem 1 Given a convex polyhedron P fixed inside a sphere Q, P can be cut out of Q by using only guillotine cuts in $O(n^3)$ time with cutting cost $O(\log^2 n)$ times the optimal, where n is the number of vertices of P.

4 Conclusion

In this paper, we have given an $O(n^3)$ -time algorithm that cuts a convex polyhedron P with n vertices from a sphere Q, where P is fixed inside Q, by using guillotine cuts with cutting cost $O(\log^2 n)$ times the optimal.

This problem is well studied in 2D, where the series of results include several $O(\log n)$ and constant factor approximation algorithms and a PTAS. The key ingredients of the 2D algorithms involve three major steps: (1) take some approximate vertex cuts through the vertices of P, (2) use dynamic programming to find an optimal cutting sequence among the edge cuts, and the vertex cuts taken in step (1), and (3) show that the cutting cost of the sequence obtained in step (2) is within the desired factor of the optimal. Using the idea of 2D algorithms may be a way to improve the approximation ratio of our algorithm. Among the above three steps, it may not be difficult to generalize steps (1) and (3) for 3D, but the most difficult part we find is the applying a dynamic programming.

An immediate future work would be to find approximation algorithms when Q is another convex polyhedron. Recently, Ahmed et.al. [1] have studied a more generalized version of the problem in 2D where the polygon P is not fixed inside a circle Q. For that problem they have given several constant factor approximation algorithms. It would be interesting to study that version of the problem in 3D.

References

- S. I. Ahmed, M. M. A. Bhuiyan, M. Hasan, and I. K. Khan. On finding a better position of a convex polygon inside a circle to minimize the cutting cost. In WALCOM'10, LNCS. Springer, 2010. To appear.
- S. I. Ahmed, M. Hasan, and M. A. Islam. Cutting a cornered convex polygon out of a circle. *Journal of Computers*, 5(1):4–11, 2010.
- S. Bereg, O. Daescu, and M. Jiang. A PTAS for cutting out polygons with lines. In COCOON'06, volume 4112 of LNCS, pages 176–185. Springer, 2006.
- J. Bhadury and R. Chandrasekaran. Stock cutting to minimize cutting length. Euro. J. Oper. Res., 88:69–87, 1996.

- R. Chandrasekaran, O. Daescu, and J. Luo. Cutting out polygons. In CCCG'05, pages 183–186, 2005.
- 6. O. Daescu and J. Luo. Cutting out polygons with lines and rays. International Journal of Computational Geometry and Applications, 16:227–248, 2006.
- M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. Computational Geometry: Algorithms and Applications. Springer, 2nd edition, 2000.
- 8. E. D. Demaine, M. L. Demaine, and C. S. Kaplan. Polygons cuttable by a circular saw. *Computational Geometry: Theory and Algorithms*, 20:69–84, 2001.
- 9. A. Dumitrescu. An approximation algorithm for cutting out convex polygons. Computational Geometry: Theory and Algorithms, 29:223–231, 2004.
- A. Dumitrescu. The cost of cutting out convex n-gons. Discrete Applied Mathematics, 143:353–358, 2004.
- J. W. Jaromczyk and M. Kowaluk. Sets of lines and cutting out polyhedral objects. Computational Geometry: Theory and Algorithms, 25:67–95, 2003.
- J. O'Rourke. Finding minimal enclosing boxes. International Journal of Computer and Information Sciences, 14(3):183–199, 1985.
- M. H. Overmars and E. Welzl. The complexity of cutting paper. In SoCG'85, pages 316–321, 1985.
- X. Tan. Approximation algorithms for cutting out polygons with lines and rays. In COCOON'05, volume 3595 of LNCS, pages 534–543. Springer, 2005.
- 15. G. Toussaint. Solving geometric problems with the rotating calipers. In *MELE-CON'83*, Athens, Greece, 1983.