A Pseudopolynomial Time $O(\log n)$-Approximation Algorithm for Art Gallery Problems

Ajay Deshpande[1], Taejung Kim[2], Erik D. Demaine[1], and Sanjay E. Sarma[1]

[1] Massachusetts Institute of Technology, Cambridge, MA 02139 USA
{ajayd,edemaine,sesarma}@MIT.EDU
[2] Dankook University, Hanam-Dong, Seoul, 140-714 Korea
taejungkim@dankook.ac.kr

Abstract. In this paper, we give a $O(\log c_{opt})$-approximation algorithm for the point guard problem where c_{opt} is the optimal number of guards. Our algorithm runs in time polynomial in n, the number of walls of the art gallery and the spread Δ, which is defined as the ratio between the longest and shortest pairwise distances. Our algorithm is pseudopolynomial in the sense that it is polynomial in the spread Δ as opposed to polylogarithmic in the spread Δ, which could be exponential in the number of bits required to represent the vertex positions. The special subdivision procedure in our algorithm finds a finite set of potential guard-locations such that the optimal solution to the art gallery problem where guards are restricted to this set is at most $3c_{opt}$. We use a set cover cum VC-dimension based algorithm to solve this restricted problem approximately.

1 Introduction

The art gallery problem addresses the following question [7]: How many guards are required to guard an art gallery with n walls? This problem was first posed by Victor Klee in 1973 [8]. Chvátal showed that $\lfloor \frac{n}{3} \rfloor$ guards are always sufficient and occasionally necessary [10]. Since then, numerous variations of this problem have been studied including mobile guards, guards with limited visibility, guarding rectilinear polygons, etc., see, e.g., [8,9,7]. In this paper, we study one version of the art gallery problem, also known as the *point-guard problem.* The point-guard problem involves finding the minimum number of points and their positions so that guards located at these points *cover* (i.e. *see*) every point in the interior of the art gallery.

Lee and Lin show that the point-guard problem is NP-hard [11]. Eidenbenz, Stamm and Widmayer prove that even finding a $(1+\epsilon)$-approximation for this problem for any $\epsilon > 0$ is *NP-hard* [4]. They also show that the problem of art gallery with holes can not be approximated by a polynomial time algorithm with ratio $(\frac{1-\epsilon}{12}) \ln n$ for any $\epsilon > 0$, unless $NP \subseteq TIME(n^{O(\log \log n)})$. Brodén, Hammar and Nisson prove that the point-guard problem even for a special class of art galleries, which are *2-link polygons*, is *APX-hard* [12].

F. Dehne, J.-R. Sack, and N. Zeh (Eds.): WADS 2007, LNCS 4619, pp. 163–174, 2007.

In [5], Ghosh proposes an $O(\log n)$-approximation algorithm for the minimum *vertex-guard problem* where guards can be be located only at the vertices of the art gallery. González-Banos and Latombe [3] consider another version of the art gallery problem in which guards have range and incidence constraints and are required to cover only the walls of the art gallery. They choose a set of uniformly randomly selected points from the art gallery as potential guard-locations and solve this new problem. They argue that their algorithm computes with high probability a solution whose size is at most a factor $O(\log n \cdot \log (c \log n))$ times the size of the optimal solution c. In [15], Efrat and Har-Peled consider another variant of the art gallery problem where guards are restricted to be placed on the points of a dense grid and propose a randomized algorithm which with high probability yields the approximation ratio within $O(\log c')$, where c' is the optimal solution size for the modified problem. In the same paper [15], Efrat and Har-Peled propose an exact algorithm for the point-guard problem with running time at most $O((nc)^{3(2c+1)})$, where is c is the size of the optimal solution. This is the first known exact solution to the problem, although the running time is exponential in the size of the optimal solution.

Our result. We give a pseudopolynomial time $O(\log c_{opt})$-approximation algorithm for the point-guard problem, where c_{opt} is the size of the optimal solution which can be as large as $\Theta(n)$ in some cases. Our algorithm is pseudopolynomial in the sense that it is polynomial in the number of walls n of the art gallery and the *spread* Δ of the vertices of the art gallery. The spread of a set of points is defined as the ratio of the longest and shortest pairwise distances [13,14]. In the worst case, the spread Δ could possibly be exponential in the number of bits required to represent positions of the vertices of the art gallery. To the best of our knowledge, this is the first pseudopolynomial time algorithm that yields a solution with a guaranteed approximation ratio.

Our basic approach involves using a special subdivision procedure to obtain a finite set of potential guard-locations. We then consider a new problem of choosing the minimum number of guards from this finite set. We devise our algorithm such that the new problem has an optimal solution at most three times the optimal solution to the original point-guard problem. We solve the new problem using a set cover cum VC-dimension-based algorithm. Our overall algorithm can be summarized in the following 3 steps:

- **Step 1:** Generate an initial triangulation of the art gallery based on the *visibility cell decomposition.*
- **Step 2:** Subdivide the initial triangulation such that each triangle in the final triangulation satisfies a special property – the region that is visible to any point in a triangle is always a subset of the region simultaneously visible to the three vertices of the triangle.
- **Step 3:** Formulate the set cover problem and solve it approximately using the VC-dimension-based algorithm of González-Banos and Latombe [3].

2 Basic Terminology

Most of the definitions and notation we present in this section have been borrowed from [1,2]; however, we reformulate some of these and define new ones for our convenience. Most of the notions we describe below are illustrated in Figure 1.

For the sake of simplicity, we consider the case of an art gallery without holes. At the end of the paper, we comment about the case of an art gallery with holes. An art galley without holes can be represented as a simple polygon. Here, we consider the boundary also as a part of the polygon. Let P be a simple polygon with n vertices. Some of these are *reflex* vertices that subtend an angle greater than 180^0 inside P. We say two points in P *see* each other if the line segment between them does not intersect with the exterior of P.

The *visibility polygon* $V(x)$ for any point $x \in P$, is the polygon consisting of all the points in P that are visible from x. Note that some of the edges of $V(x)$ coincide with those of the original polygon P and some are newly introduced as shown in Figure 1(a). A new edge is introduced at a reflex vertex of P that blocks the view of x. We call this reflex vertex a *blocking reflex vertex.* The other end-point of the new edge which lies on the boundary of P is referred to as an *image of x through the blocking reflex vertex.* To remove any ambiguities, we assume that for P and $V(x)$, no two consecutive edges are collinear.

For any point $x \in P$, we say that x *sees* an edge of P, if it sees a point on the edge. If x cannot see either of the end-points of a visible edge of P, we say that x sees the edge *partially.* We call the corresponding edge of P a *partial edge with respect to x*. We say that *x sees* a visible edge of P *non-partially*, if it sees at least one of its end-points. We call the corresponding edge of P a *non-partial edge with respect to x*. If we join every vertex of $V(x)$ to x, we get a triangulation of $V(x)$. We call each triangle as a *visibility sector* of x. The edge of a visibility sector that is a part of an edge of P is referred to as a *base* of the visibility sector. Depending upon the type of the edge of P corresponding to the base of a visibility sector, we classify the visibility sector into *non-partial-edge sector* or *partial-edge sector.*

3 Initial Triangulation Using Visibility Cell Decomposition

In this section, we define a particular subdivision of the polygon – the *visibility cell decomposition.* Then we show how to triangulate this subdivision to generate the initial triangulation in Step 1 of our algorithm.

The *visibility cell decomposition* of P is a subdivision induced by visibility polygons of all the vertices of P. We call each component of the subdivision a *visibility cell.* We state without proofs the following properties of the visibility cell decomposition that are useful both in the construction and analysis of our algorithm. We refer interested readers to the papers by Bose *et al.* [1] and Guibas *et al.* [2] for further details.

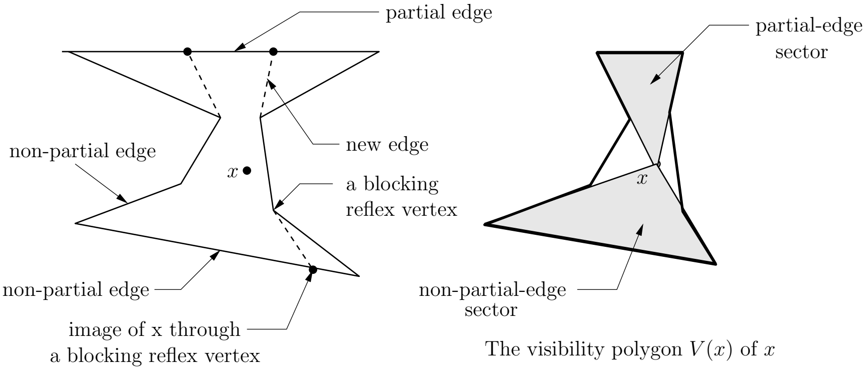

Fig. 1. Visibility polygon and visibility sectors

- Each visibility cell is a convex polygon.
- The total number of visibility cells in the visibility cell decomposition is $O(n^3)$.
- By definition, any two points in a visibility cell see the same set of vertices of P. Furthermore any two points in the same visibility cell see the same set of non-partial edges and the same set of partial edges of P.

Step 1 of our algorithm can be summarized as follows:

Construct the visibility cell decomposition of the polygon. Triangulate each visibility cell simply by joining its one particular vertex to every other vertex.

4 Further Subdivision of the Initial Triangulation

In this section, we describe Step 2 of our algorithm. We give a procedure to subdivide the initial triangulation in such a way that each triangle in the final triangulation satisfies a special visibility property – the region that is visible to any point in a triangle is covered by the visibility polygons of the three vertices of the triangle.

4.1 Vertex-Visibility Property and Vertex-Pair-Visibility Property

We first define the desirable property which each triangle in the final triangulation is required to satisfy.

Definition 1. *Let $\triangle abc$ be a triangle in the polygon P. We say $\triangle abc$ satisfies the* vertex visibility property, *if for any point $x \in \triangle abc$, $V(x) \subseteq V(a) \cup V(b) \cup V(c)$.*

Covering the visibility polygon of a point is equivalent to covering every visibility sector of the point. This motivates the following definition.

Definition 2. *A triangle in a visibility cell satisfies the* vertex-visibility property with respect to a particular edge *of the polygon, if the corresponding visibility sector of any point in the triangle is a subset of the union of the visibility polygons of the vertices of the triangle.*

The vertex-visibility property is not directly useful in the construction of our algorithm. We define a more convenient property.

Definition 3. *A triangle in a visibility cell satisfies the* vertex-pair-visibility property with respect to a particular edge *of the polygon, if the visibility sectors of any two vertices of the triangle overlap on the edge.*

Consider the images of two points in a visibility cell through a blocking reflex vertex on an edge of the polygon. We call the portion of the edge between the two images as a *span* of the two points corresponding to the blocking reflex vertex. Note that the image of a point on the segment joining these two points lies in the span by one-to-one mapping. Now consider the images of the three vertices of a triangle in a visibility cell through a blocking reflex vertex on an edge of the polygon. One of the three images lies between the other two. We call the portion of the edge between the two extreme images as a *span of the triangle* through the blocking reflex vertex.

Lemma 1. *For any point in a triangle in a visibility cell, its image through a blocking reflex vertex always lies in the span of the triangle through the blocking reflex vertex.*

Proof. The image of any point on a segment lies in the span of the two endpoints of the segment corresponding to a blocking vertex. Thus, the image of any point on the perimeter of a triangle lies in the span of the triangle. Now consider any point in the interior of the triangle. The image of this point is same as the image of the point on the perimeter of the triangle where the line segment joining this point, the blocking reflex vertex and its image intersects the perimeter. Hence the image of any point in the triangle is in its span. □

Theorem 1. *A triangle in a visibility cell satisfies the vertex-pair-visibility as well as the vertex-visibility property with respect to a non-partial edge.*

Proof. Let $\triangle abc$ be a triangle in a visibility cell C. Let e be a non-partial edge. As we have already seen, at least one of the end-points of a non-partial edge is visible from any point in a visibility cell. Depending on whether one or both the end-points of a non-partial edge are visible, we make two cases and deal with each case separately.

Case 1: Both the end-points of e are visible from any point in C. In this case, by definition, $\triangle abc$ satisfies the vertex-pair-visibility property. Let u and v be the end-points of e. Consider the convex hull of a, b, c, u and v. Since $\triangle abc$ is on one side of e, line segment uv must be one of the edges of the convex hull. Therefore, the convex hull can also be formed by considering the union of $\triangle abc$ and the visibility sectors of a, b and c. Note that the convex hull is a subset of $V(a) \cup V(b) \cup V(c)$ and the visibility sector of any point $x \in \triangle abc$ is a subset of this convex hull. Therefore, $\triangle abc$ also satisfies the vertex-visibility property with respect to e.

Case 2: In this case, only one end-point of e is visible from any point in C. Let u be the visible end-point. Let r be a blocking reflex vertex. Again by definition $\triangle abc$ satisfies the vertex-pair-visibility property because u is a common visible point. Now, consider any point x in $\triangle abc$. The visibility sector of x with respect to e consists of two triangles, $\triangle xur$ and $\triangle urx'$, where x' is the image of x through r. By similar arguments as in the first case, we can prove that $\triangle xur$ is a subset of $V(a) \cup V(b) \cup V(c)$. By Lemma 1, x' lies in the span of the image of $\triangle abc$ through r. Thus, at least one of a, b or c cover $\triangle rux'$. Therefore, $\triangle abc$ satisfies the vertex-visibility property with respect to e. □

Theorem 2. *If a triangle in a visibility cell satisfies the vertex-pair-visibility property with respect to a partial edge e, then it also satisfies the vertex-visibility property with respect to e.*

Proof. Let $\triangle abc$ be a triangle in a visibility cell C such that it satisfies the vertex-pair-visibility property with respect to the partial edge e. Let r_1 and r_2 be the two blocking reflex vertices. Consider vertices a and b. The visibility sectors of a and b overlap on e. Let a_1 and a_2 be images of a through r_1 and r_2 respectively. Let b_1 and b_2 be images of b through r_1 and r_2 respectively. Since a_1a_2 and b_1b_2 overlap on e, at least one of b_1 and b_2 lies in between a_1 and a_2. Since e is a partial edge, a_1b_1 and a_2b_2 do not overlap on e. In other words, the spans of a and b with respect to r_1 and r_2 do not overlap on e. By extending this argument to the three vertices, a, b and c, the spans of any two vertices with respect to r_1 and r_2 do not overlap. This implies that the spans of $\triangle abc$ also do not overlap on e because if they do, the previous condition of pairwise vertices having non-overlapping spans is violated for at least one pair. The portion of e that is simultaneously visible to a, b and c consists of the spans of $\triangle abc$ through r_1 and r_2 and the patch between the two spans. By Lemma 1, for any point x in $\triangle abc$, the two images of x through r_1 and r_2 lie in the spans of $\triangle abc$ through r_1 and r_2 respectively. Thus, the portion of e that is visible to x is contained in the portion that is visible to a, b and c. Therefore, the visibility sector of x is a subset of $V(a) \cup V(b) \cup V(c)$. □

The theorem we prove below is useful in the analysis of the algorithm. Let *subtriangle* be a triangle that is contained within a triangle.

Theorem 3. *If a triangle in a visibility cell satisfies the vertex-pair-visibility property with respect to a partial edge e, then any subtriangle also satisfies the vertex-pair-visibility property with respect to e.*

Proof. Let $\triangle abc$ be a triangle in a visibility cell C such that it satisfies the vertex-pair-visibility property with respect to the partial edge e. Let r_1 and r_2 be the two blocking reflex vertices. We already proved in the proof of Theorem 2 that the spans of $\triangle abc$ through r_1 and r_2 do not overlap on e because it satisfies the vertex-pair-visibility property. For any two points x and y in $\triangle abc$, the spans of x and y through r_1 and r_2 do not overlap on e because they are contained in the spans of $\triangle abc$ through r_1 and r_2. Therefore, the visibility sectors of x and y overlap on e. Therefore, any $\triangle xyz$ in $\triangle abc$ satisfies the vertex-pair-visibility property. □

The above theorem allows us to further subdivide the visibility cell without affecting already existent vertex-pair visibility property with respect to a partial-edge visibility sector.

4.2 Further Subdivision

In this subsection, we give a procedure to further subdivide the initial triangulation obtained in Step 1 of our algorithm. The subdivision procedure described below generates the final triangulation where every triangle satisfies the vertex-visibility property. This property is required so that we can reduce the art gallery problem to a problem with guaranteed approximation ratio. Using the results of Theorem 1 and Theorem 2, we achieve this by developing a subdivision procedure which is based on a stronger condition, the vertex-pair-visibility property.

First we define a notion that is useful in the description of our algorithm. Let a and b be two points in a visibility cell such that the visibility sectors of a and b do not overlap on a partial edge. Let r_1 and r_2 be the corresponding blocking reflex vertices. Consider the convex hull of a, b, r_1 and r_2. We call a triangle obtained by taking set difference between the convex hull and the union of the visibility sectors of a and b as a *dark triangle* of segment ab. An example of a dark triangle is shown in Figure 2(a).

Step 2 of our algorithm can be summarized as follows.

For every $\triangle abc$ in the initial triangulation obtained in Step 1, repeat the following procedure:

1. Construct a set S of partial edges for which $\triangle abc$ does not satisfy the vertex-pair-visibility property. Repeat the following procedure for every edge $e \in S$:
   (a) Construct a dark triangle of every edge of $\triangle abc$.
   (b) For each dark triangle whose interior is not disjoint with $\triangle abc$, invoke *SUBDIVIDE-DARK-TRIANGLE*.
   (c) Intersect with $\triangle abc$, the subdivisions of all such dark triangles on which the function *SUBDIVIDE-DARK-TRIANGLE* is invoked in the above step to generate a new subdivision of $\triangle abc$.
2. Intersect all the subdivisions of $\triangle abc$ corresponding to every edge $e \in S$ to generate the final subdivision. Triangulate the final subdivision in the similar way as in Step 1 of our algorithm and return the final triangulation of $\triangle abc$.

Function *SUBDIVIDE-DARK-TRIANGLE*:

Input: A dark triangle $\triangle aob$ corresponding to the two blocking reflex vertices r_1 and r_2

Procedure: Let a_1b_1 and a_2b_2 be the two spans of ab through r_1 and r_2 respectively on the partial edge. Construct a line joining the reflex vertex r_2 and the image a_1 of a through r_1 and another line joining the reflex vertex r_1 and the image b_2 of b through r_2. Depending upon whether the two lines intersect inside or outside $\triangle aob$, choose one of the following two steps.

(**Case 1**) The two lines meet outside $\triangle aob$: Return the new subdivision of $\triangle aob$ induced by the two lines (Figure 2(a)). Terminate the function.

(**Case 2**) The two lines meet in $\triangle aob$: Return the new subdivision of $\triangle aob$ without $\triangle a'o'b'$, where o' is the point of intersection of the two lines, and a' and b' are the points of intersection of the two lines with the segment ab. Check if $\triangle a'o'b'$ satisfies the vertex-pair-visibility property. If it does not, invoke *SUBDIVIDE-DARK-TRIANGLE* on $\triangle a'o'b'$. (Figure 2(b))

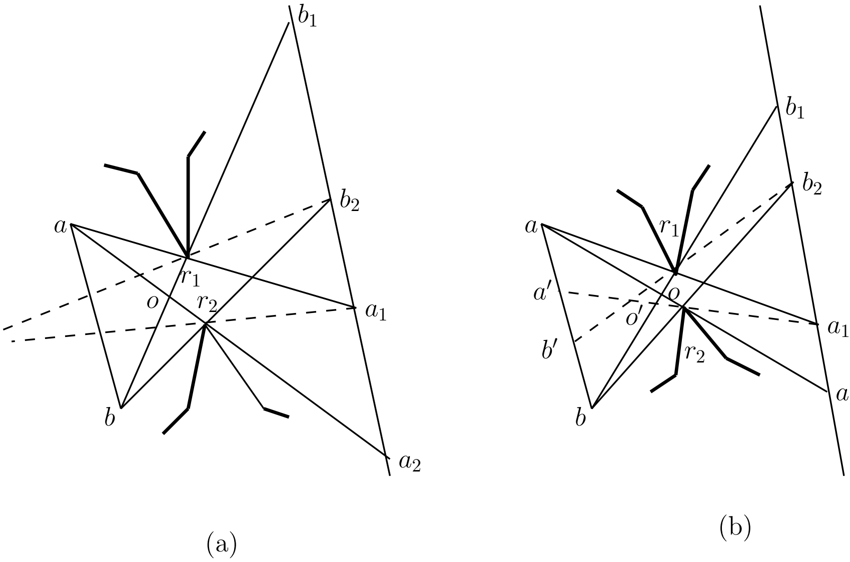

Fig. 2. $\triangle aob$ is a dark triangle. Two cases in *SUBDIVIDE-DARK-TRIANGLE*: (a)lines a_1r_2 and b_2r_1 meet outside $\triangle aob$ (b) lines a_1r_2 and b_2r_1 meet in $\triangle aob$.

As a result of Theorem 1 and Theorem 2, in our subdivision procedure, we need to subdivide a triangle only if it does not satisfy the vertex-pair-visibility property with respect to a partial edge. The result of our subdivision procedure is the final triangulation where every triangle satisfies the vertex-pair visibility property and in turn, the vertex-visibility property. Now we prove this result.

As we have already mentioned, using the results of Theorem 1 and Theorem 2, we check whether a triangle in the initial triangulation satisfies the vertex-pair-visibility with respect to partial edges only. As a result of Theorem 3, the subdivision procedure of a triangle with respect to one edge is 'independent' of the subdivision procedure with respect to another edge. This allows us to subdivide a triangle in the edge-by-edge fashion.

Lemma 2. *Consider the partial-edge visibility sector of a point in a visibility cell. Any triangle that lies in the visibility sector as well as the visibility cell always satisfies the vertex-pair-visibility property.*

Proof. Let x be a point in a visibility cell C. Let r_1 and r_2 be the blocking reflex vertices corresponding to the partial edge. Let x_1 and x_2 be the images of point x through r_1 and r_2 respectively. Any point a that lies in the visibility sectors of x as well as in the same visibility cell C, sees the line segment x_1x_2. Therefore, by definition, any triangle that lies in the visibility sector of x as well as in C satisfies the vertex-pair-visibility property. □

Let $\triangle abc$ be a triangle in the initial triangulation. Suppose that it does not satisfy the vertex-pair-visibility property with respect to a partial edge. Consider the convex hull of a, b, c, r_1 and r_2. The convex hull can also be obtained by taking union of the visibility sectors of a, b and c and the dark triangles of all the edges of $\triangle abc$. By Lemma 2, the portions of $\triangle abc$ that lie in the visibility sector of any of the vertices satisfies the vertex-pair-visibility property. The remaining part of $\triangle abc$ is a subset of the union of the dark triangles. Therefore, in our subdivision procedure in Step 2, we just subdivide the dark triangles.

Now we prove correctness of the function *SUBDIVIDE-DARK-TRIANGLE* with reference to Figure 2

Theorem 4. *In the first case, the subdivision of $\triangle aob$ satisfies the vertex-pair-visibility property.*

Proof. Consider line a_1r_2. It subdivides $\triangle aob$ into two part. a_1 is always visible from any point in one part. Therefore that always satisfies the vertex-pair visibility property. Similarly line b_2r_1 subdivides $\triangle aob$ in two parts out of which one part always satisfies the vertex-pair-visibility property because b_2 is the common visible point from that part. In the first case lines a_1r_2 and b_2r_1 meet outside $\triangle abc$. Both the parts of mentioned above that satisfy the vertex-pair-visibility property cover $\triangle aob$ in the first case. Therefore, the subdivision of $\triangle aob$ satisfies the vertex-pair-visibility property. □

Theorem 5. *In the second case, the subdivision of $\triangle aob$ except $\triangle a'o'b'$ satisfies the vertex-pair-visibility property.*

The proof of the above theorem is similar to the proof of Theorem 4. $\triangle a'o'b'$ may not satisfy the vertex-pair-visibility property. In that case, we subdivide $\triangle a'o'b'$ by again invoking the function *SUBDIVIDE-DARK-TRIANGLE*. The first case is the termination case for the recursion in *SUBDIVIDE-DARK-TRIANGLE*. In the next section, we show that *SUBDIVIDE-DARK-TRIANGLE* indeed terminates. Thus, subdivision generated by *SUBDIVIDE-DARK-TRIANGLE* always satisfies the vertex-pair-visibility property.

The function *SUBDIVIDE-DARK-TRIANGLE* in the subdivision procedure described above is recursive. Here, we address the question after how many steps this recursion ends. We define *spread Δ* of the vertices of the art gallery as the ratio of the longest and shortest pairwise distances [13,14]. Now we prove the following theorem.

Theorem 6. *The recursive function SUBDIVIDE-DARK-TRIANGLE ends in $O(\varDelta)$ steps.*

Proof. Let L be the longest and let ϵ be the shortest pairwise distances among the vertices of the art gallery. Thus, $\varDelta = L/\epsilon$. The length of each subdivision of the partial edge at the end of the recursive procedure is at most ϵ. Since the length of any partial edge can be at most L, the total number of subdivisions does not exceed $\varDelta$. □

5 Set Cover Formulation and Approximate Solution

In this section, we describe Step 3 of our algorithm. We choose all the vertices of the final triangulation obtained in Step 2 as the potential guard-locations and formulate the set cover problem. The set cover problem is then solved approximately using a VC-dimension-based algorithm.

Step 3 of our algorithm can be summarized in the following way:

1. Construct a set G consisting of all the vertices of the final triangulation obtained in Step 2 of our algorithm. Let $|G| = m$.
2. Construct the visibility polygon for every $g_i \in G$ and generate the new subdivision of the polygon. Enumerate all the cells in the new subdivision and group them in the set $X = \{1, 2, .., l\}$. For each $g_i \in G$, construct a set R_i of cells visible from g_i, that is, $R_i = \{x \in X | x \in V(g_i)\}$. Build the set family, $R = \{R_1, R_2 ..., R_m\}$. Group X and R together to form the set system (X, R).
3. Invoke the function *SET-COVER* on the set system (X, R) to obtain a near-optimal covering of X from the set family R.

The function *SET-COVER* used in the above procedure is based on the algorithm proposed by Brönnimann and Goodrich [6] for finding set covers for set systems with finite VC-dimension. Here, we do not give details of the function *SET-COVER*. Instead, we refer interested readers to [3] for further details.

6 Analysis of the Algorithm

In this section, we analyze the bound on the approximation ratio and running time of our algorithm.

6.1 Bound on the Approximation Ratio of Our Algorithm

Consider the set system (X, R) that we construct in Step 3 of our algorithm. Let T_x, where $x \in X$, be a set consisting of all the sets in R that contain x. We define the *dual set system* (Y, S) of (X, R) by setting $Y = R$ and $S = \{T_x | x \in X\}$ [6,3]. Y corresponds to the set of candidate locations for guards. An element in S corresponds to a cell and is a set of candidate guard-locations that are visible from every point in the cell. We can also write this dual set system as $(G, \{G \cap V(x) \mid x \in P\})$, where G consists of the set of all candidate guard-locations. Valtr showed that the VC-dimension of the more general set system $(P, \{P \cap V(x) \mid x \in P\})$ is bounded by 23 [16][1]. Using the definition of the VC-dimension it is easy to prove that the VC-dimension of the dual set system (Y, S) is also bounded by 23.

The result from [6] implies that it is possible to compute an approximate solution to the set cover problem with the approximation ratio $O(d \log(dc))$, where d is the VC-dimension and c is the size of the optimal solution. The constant bound on the VC-dimension in this case implies that we obtain $O(\log c_{opt})$-approximate solution, where c_{opt} is the size of the optimal solution. c_{opt} can be as large as $\Theta(n)$ in some cases.

6.2 Analysis of the Running Time of the Algorithm

Theorem 7. *The running time of our algorithm is polynomial in the number of walls, n and the spread Δ of the vertices of the art gallery.*

Proof. Here we provide only the sketch of the proof. In Step 1, the initial triangulation can be generated in $O(n^4)$ time and consists of $O(n^4)$ triangles [1,2]. In Step 2, for each triangle in the initial triangulation, we can check in $O(n)$ time whether it satisfies the vertex-pair visibility property. In the worst case, the recursive subdivision procedure for each triangle with respect to a partial edge may run in $O(\Delta)$ time as shown in Theorem 6 and may generate $O(\Delta)$ line segments to form the subdivision. This ensures that the number of triangles in the final subdivision is polynomial in n and Δ. In Step 3, *SET-COVER* runs in $O(|X|)$ time [6,3], where $|X|$ is the total number of cells. □

Δ can be at most exponential in the input size. Thus, our algorithm runs in pseudopolynomial time.

6.3 Art Gallery with Holes

When the art gallery has holes, our algorithm can still be used. Guibas et al. [2] extend the visibility cell decomposition to a polygon with holes; except that in this case, the vertices of the holes also act as the blocking vertices. The subdivision procedure of our algorithm is still valid in this case. Valtr prove that for this case of an art gallery with holes the VC-dimension is bounded by $O(\log h)$, where h is the number of holes [16]. Thus, in this case our algorithm yields a solution with the approximation ratio $O(\log h \cdot \log(c_{opt} \log h))$.

[1] In the earlier draft of this paper, we had used $O(\log n)$ bound on the VC-dimension. Csaba Toth pointed us to the constant VC-dimension bound in [16].

7 Conclusions

In this paper, we have presented a pseudopolynomial time algorithm for the point guard problem with guaranteed $O(\log n)$ approximation ratio. The imminent question is whether we can improve the running time of our algorithm. An interesting topic for future research is to investigate whether our subdivision procedure can be applied to other variants of the art gallery problems, particularly for the case when guards have limited range.

Acknowledgments

We thank Csaba Toth for pointing to us a better VC-dimension bound (see Footnote 1). We thank the anonymous referees for their useful comments and bringing to our attention the reference [15]. The first author would like to thank Prahladh Harsha for many useful discussions and feedback on the earlier drafts of the paper.

References

1. Bose, P., Lubiw, A., Munro, J.I.: Efficient visibility queries in simple polygons. In: Proc. 4th Canad. Conf. Comput. Geom. pp. 23–28 (1992)
2. Guibas, L.J., Motwani, R., Raghavan, P.: The robot localization problem. SIAM J. Comput. 26(4), 1120–1138 (1997)
3. González-Banos, H., Latombe, J.: A randomized art-gallery algorithm for sensor placement. In: Proc. 17th Symp. Comput. Goem, pp. 232–240 (2001)
4. Eidenbenz, S., Stamm, C., Widmayer, P.: Inapproximability Results for Guarding Polygons and Terrains. Algorithmica 31, 79–113 (2001)
5. Ghosh, S.: Approximation algorithm for art gallery problems. In: Canad. Information Processing Soc. Congress (1987)
6. Brönnimann, H., Goodrich, M.T.: Almost optimal set covers in finite VC-dimension. In: Proc. 10th Symp. Comp. Geom. pp. 293–302 (1994)
7. Urrutia, J.: Art Gallery and Illumination Problems. In: Sack, J. R., Urrutia, J. (eds.) Handbook of Computational Geometry (2000)
8. O'Rourke, J.: Art Gallery Theorems and Algorithms (1987)
9. Shermer, T.: Recent results in art galleries. Proc. IEEE. 80, 1384–1399 (1992)
10. Chvátal, V.: A combinatorial theorem in plane geometry. J. Combinat. Theory B 18, 39–41 (1975)
11. Lee, D.T., Lin, A.K.: Computational complexity of art gallery problems. IEEE Trans. Info. Theory. IT-32, 276–282 (1986)
12. Brodén, B., Hammar, M., Nilsson, B.J.: Guarding lines and 2-link polygons is APX-hard. In: Proc. 13th Canad. Conf. Comp. Geom. pp. 45–48 (2001)
13. Erickson, J.: Nice point sets can have nasty Delaunay triangulations. In: Proc. 17th Symp. Comp. Geom. pp. 96–105 (2001)
14. Erickson, J.: Dense point sets have sparse Delaunay triangulations: or "...but not too nasty". In: Proc. 13th Symp. Disc. Algo. pp. 125–134 (2002)
15. Efrat, A., Har-Peled, S.: Guarding galleries and terrains. Info. Proc. Lett. 100(6), 238–245 (2006)
16. Valtr, P.: Guarding galleries where no point sees a small area. Israel J. Math. 104, 1–16 (1998)