Vertex Guarding in Weak Visibility Polygons

Pritam Bhattacharya, Subir Kumar Ghosh, and Bodhayan Roy

Tata Institute of Fundamental Research, Mumbai-400005, India
pritamb@tcs.tifr.res.in, {ghosh,bodhayan}@tifr.res.in

Abstract. The art gallery problem enquires about the least number of guards that are sufficient to ensure that an art gallery, represented by a polygon P, is fully guarded. In 1998, the problems of finding the minimum number of point guards, vertex guards, and edge guards required to guard P were shown to be APX-hard by Eidenbenz, Widmayer and Stamm. In 1987, Ghosh presented approximation algorithms for vertex guards and edge guards that achieved a ratio of $\mathcal{O}(\log n)$, which was improved upto $\mathcal{O}(\log\log OPT)$ by King and Kirkpatrick in 2011. It has been conjectured that constant-factor approximation algorithms exist for these problems. We settle the conjecture for the special class of polygons that are weakly visible from an edge and contain no holes by presenting a 6-approximation algorithm for finding the minimum number of vertex guards that runs in $\mathcal{O}(n^2)$ time. On the other hand, for weak visibility polygons with holes, we present a reduction from the Set Cover problem to show that there cannot exist a polynomial time algorithm for the vertex guard problem with an approximation ratio better than $((1-\epsilon)/12)\ln n$ for any $\epsilon > 0$, unless NP = P.

1 Introduction

1.1 The Art Gallery Problem and Its Variants

The art gallery problem enquires about the least number of guards that are sufficient to ensure that an art gallery (represented by a polygon P) is fully guarded, assuming that a guard's field of view covers 360° as well as an unbounded distance. This problem was first posed by Victor Klee in a conference in 1973, and in the course of time, it has turned into one of the most investigated problems in computational geometry.

A *polygon* P is defined to be a closed region in the plane bounded by a finite set of line segments, called edges of P, such that, between any two points of P, there exists a path which does not intersect any edge of P. If the boundary of a polygon P consists of two or more cycles, then P is called a *polygon with holes*. Otherwise, P is called a *simple polygon* or a *polygon without holes*. An art gallery can be viewed as an n-sided polygon P (with or without holes) and guards as points inside P. Any point $z \in P$ is said to be *visible* from a guard g if the line segment zg does not intersect the exterior of P. In general, guards may be placed anywhere inside P. In 1975, Chvátal [4] showed that $\lfloor \frac{n}{3} \rfloor$ stationary guards are sufficient and sometimes necessary for guarding a simple polygon. In 1978, Fisk [9] presented a simpler and more elegant proof of this result.

S. Ganguly and R. Krishnamurti (Eds.): CALDAM 2015, LNCS 8959, pp. 45–57, 2015.

1.2 Related Hardness and Approximation Results

The decision version of the art gallery problem is to determine, given a polygon P and a number k as input, whether the polygon P can be guarded with k or fewer guards. The problem was first proved to be NP-complete for polygons with holes by O'Rourke and Supowit [19]. For guarding simple polygons, it was proved to be NP-complete for vertex guards by Lee and Lin [18], and their proof was generalized to work for point guards by Aggarwal [1]. The problem is NP-hard even for simple orthogonal polygons as shown by Katz and Roisman [16] and Schuchardt and Hecker [20]. Each one of these hardness results hold irrespective of whether we are dealing with vertex guards, edge guards, or point guards.

In 1987, Ghosh [10,12] provided an $\mathcal{O}(\log n)$-approximation algorithm for the case of vertex and edge guards by discretizing the input polygon and treating it as an instance of the Set Cover problem. In fact, applying methods for the Set Cover problem developed after Ghosh's algorithm, it is easy to obtain an approximation factor of $\mathcal{O}(\log OPT)$ for vertex guarding simple polygons or $\mathcal{O}(\log h \log OPT)$ for vertex guarding a polygon with h holes. Deshpande et al. [5] obtained an approximation factor of $\mathcal{O}(\log OPT)$ for point guards or perimeter guards by developing a sophisticated discretization method that runs in pseudopolynomial time. Efrat and Har-Peled [6] provided a randomized algorithm with the same approximation ratio that runs in fully polynomial expected time. For guarding simple polygons using vertex guards and perimeter guards, King and Kirkpatrick [17] obtained $\mathcal{O}(\log \log OPT)$ approximation ratio in 2011.

In 1998, Eidenbenz, Stamm and Widmayer [7,8] proved that the problem is APX-complete, implying that an approximation ratio better than a fixed constant cannot be achieved unless P=NP. They also proved that if the input polygon is allowed to contain holes, then there cannot exist a polynomial time algorithm for the problem with an approximation ratio better than $((1-\epsilon)/12)\ln n$ for any $\epsilon > 0$, unless NP $\subseteq$ TIME($n^{\mathcal{O}(\log \log n)}$). Contrastingly, in the case of simple polygons without holes, the existence of a constant-factor approximation algorithm for vertex guards and edge guards has been conjectured by Ghosh [10,13] since 1987. However, this conjecture has not yet been settled even for special classes of polygons such as weak visibility polygons, monotone polygons, orthogonal polygons etc.

1.3 Our Contributions

A polygon P is said to be a *weak visibility polygon* if every point in P is visible from some point of an edge [11]. In Section 2, we present a 6-approximation algorithm, which has running time $\mathcal{O}(n^2)$, for vertex guarding polygons that are weakly visible from an edge and contain no holes. This result can be viewed as a step forward towards solving Ghosh's conjecture for a special class of polygons. Following the construction of Eidenbenz, Stamm and Widmayer [7], we establish a reduction from Set Cover and show that, for the special class of polygons containing holes that are weakly visible from an edge, there cannot exist a polynomial time algorithm for the vertex guard problem with an approximation ratio better than $((1-\epsilon)/12)\ln n$ for any $\epsilon > 0$, unless NP = P. For details of this reduction, refer to the full version of the paper [3].

2 Placement of Guards in Weak Visibility Polygons

Let P be a simple polygon. If there exists an edge uv in P (where u is the next clockwise vertex of v) such that P is weakly visible from uv, then it can be located in $\mathcal{O}(n^2)$ time [2,14]. Henceforth, we assume that such an edge uv has been located. Let $bd_c(p,q)$ (or, $bd_{cc}(p,q)$) denote the clockwise (respectively, counterclockwise) boundary of P from a vertex p to another vertex q. Note that, by definition, $bd_c(p,q) = bd_{cc}(q,p)$. The *visibility polygon* of P from a point z, denoted by $VP(z)$, is defined to be the set of all points in P that are visible from z. In other words, $VP(z) = \{q \in P : q \text{ is visible from } z\}$.

The *shortest path tree* of P rooted at a vertex r of P, denoted by $SPT(r)$, is the union of Euclidean shortest paths from r to all the vertices of P. This union of paths is a planar tree, rooted at r, which has n nodes, namely the vertices of P. For every vertex x of P, let $p_u(x)$ and $p_v(x)$ denote the parent of x in $SPT(u)$ and $SPT(v)$ respectively. In the same way, for every interior point y of P, let $p_u(y)$ and $p_v(y)$ denote the vertex of P next to y in the Euclidean shortest path to y from u and v respectively.

2.1 Guarding All Vertices of a Polygon

Suppose a guard is placed on each non-leaf vertex of $SPT(u)$ and $SPT(v)$. It is obvious that these guards see all points of P. However, the number of guards required may be very large compared to the size of an optimal guarding set. In order to reduce the number of guards, placing guards on every non-leaf vertex should be avoided. Let A be a subset of vertices of P. Let S_A denote the set which consists of the parents $p_u(z)$ and $p_v(z)$ of every vertex $z \in A$. Then, A should be chosen such that all vertices of P are visible from guards placed at vertices of S_A. We present a method for choosing A and S_A as follows:-

Algorithm 2.1. An $\mathcal{O}(n^2)$-algorithm for computing a guard set S_A for all vertices of P

```
Compute SPT(u) and SPT(v)
Initialize all the vertices of P as unmarked
Initialize A ← ∅, S_A ← ∅ and z ← u
while z ≠ v do
    z ← the vertex next to z in clockwise order on bd_c(u, v)
    if z is unmarked then
        A ← A ∪ {z} and S_A ← S_A ∪ {p_u(z), p_v(z)}
        Place guards on p_u(z) and p_v(z)
        Mark all vertices of P that become visible from p_u(z) or p_v(z)
    end if
end while
return the guard set S_A
```

Now, assume a special condition such that for every vertex $z \in A$, all vertices of $bd_c(p_u(z), p_v(z))$ are visible from $p_u(z)$ or $p_v(z)$. We prove that, in such a situation, $|S_A| \leq 2.|S_{opt}|$, where S_{opt} denotes an optimal vertex guard set.

Lemma 1. *Any guard $g \in S_{opt}$ that sees vertex z of P must lie on $bd_c(p_u(z), p_v(z))$.*

Proof. Since $p_u(z)$ is the parent of z in $SPT(u)$, z cannot be visible from any vertex of $bd_c(u, p_u(z))$. Similarly, since $p_v(z)$ is the parent of z in $SPT(v)$, z cannot be visible from any vertex of $bd_{cc}(v, p_v(z))$. Hence, any guard $g \in S_{opt}$ that sees z must lie on $bd_c(p_u(z), p_v(z))$.

Lemma 2. *Let z be a vertex of P such that all vertices of $bd_c(p_u(z), p_v(z))$ are visible from $p_u(z)$ or $p_v(z)$. For every vertex x lying on $bd_c(p_u(z), p_v(z))$, if x sees a vertex q of P, then q must also be visible from $p_u(z)$ or $p_v(z)$.*

Proof. If q lies on $bd_c(p_u(z), p_v(z))$, then it is visible from from $p_u(z)$ or $p_v(z)$ by assumption. So, consider the case where q lies on $bd_{cc}(p_u(z), p_v(z))$. Now, either q lies on $bd_c(u, p_u(z))$ or q lies on $bd_{cc}(v, p_v(z))$. In the former case, if $bd_{cc}(q, p_v(z))$ intersects the segment $qp_v(z)$, then q or $p_v(z)$ is not weakly visible from uv (see Fig. 1). Moreover, no other portion of the boundary can intersect $qp_v(z)$ since qx and $zp_v(z)$ are internal segments. Hence, q must be visible from $p_v(z)$. Analogously, if q lies on $bd_{cc}(v, p_v(z))$, q must be visible from $p_u(z)$.

Lemma 3. *Assume that every vertex $z \in A$ is such that every vertex of $bd_c(p_u(z), p_v(z))$ is visible from $p_u(z)$ or $p_v(z)$. Then, $|A| \leq |S_{opt}|$.*

Proof. Assume on the contrary that $|A| > |S_{opt}|$. This implies that Algorithm 2.1 includes two distinct vertices z_1 and z_2 belonging to A which are both visible from a single guard $g \in S_{opt}$. Moreover, it follows from Lemma 1 that g must lie on $bd_c(p_u(z_1), p_v(z_1))$. Without loss of generality, let us assume that vertex z_1 is added to A before z_2 by Algorithm 2.1. In that case, Algorithm 2.1 places guards at $p_u(z_1)$ and $p_v(z_1)$. Now, as vertex z_2 is visible from g, it follows from Lemma 2 that z_2 is also visible from $p_u(z_1)$ or $p_v(z_1)$. Therefore, z_2 is already marked, and hence, Algorithm 2.1 does not include z_2 in A, which is a contradiction.

Lemma 4. $|S_A| = 2|A|$.

Proof. For every $z \in A$, since Algorithm 2.1 includes both the parents $p_u(z)$ and $p_v(z)$ of z in S_A, it is clear that $|S_A| \leq 2|A|$. If both the parents of every $z \in A$ are distinct, then $|S_A| = 2|A|$. Otherwise, there exists two distinct vertices z_1 and z_2 in A that share a common parent, say p. Without loss of generality, let us assume that vertex z_1 is added to A before z_2 by Algorithm 2.1. In that case, Algorithm 2.1 places a guard at p, which results in z_2 getting marked. Thus, Algorithm 2.1 cannot include z_2 in A, which is a contradiction. Hence, $|S_A| = 2|A|$ must be true.

Theorem 1. *If every vertex $z \in A$ is such that all vertices of $bd_c(p_u(z), p_v(z))$ are visible from $p_u(z)$ or $p_v(z)$, then $|S_A| \leq 2|S_{opt}|$.*

Proof. By Lemma 4, $|S_A| = 2|A|$. Also, by Lemma 3, $|A| \leq |S_{opt}|$.
So, $|S_A| = 2|A| \leq 2|S_{opt}|$.

The above bound does not hold if there exists $z \in A$ such that some vertices of $bd_c(p_u(z), p_v(z))$ are not visible from $p_u(z)$ or $p_v(z)$. Now, consider Fig. 2. For each $i \in \{1, 2, \ldots, k-1\}$, z_{i+1} is not visible from $p_u(z_i)$ or $p_v(z_i)$, which forces Algorithm 2.1 to place guards at $p_u(z_{i+1})$ and $p_v(z_{i+1})$. Therefore, Algorithm 2.1 includes $z_1, z_2, z_3, \ldots, z_k$ in A and places a total of $2k$ guards at vertices $u, p_{v1}, p_{u2}, p_{v2}, \ldots, p_{uk}, p_{vk}$. However, all vertices of P are visible from just two guards placed at u and g. Hence, $|S_A| = 2k$ whereas $|S_{opt}| = 2$. Since the construction in Fig. 2 can be extended for any arbitrary integer k, $|S_A|$ can be arbitrarily large compared to $|S_{opt}|$. So, we present a new and better algorithm which gives us a 4-approximation of $|S_{opt}|$.

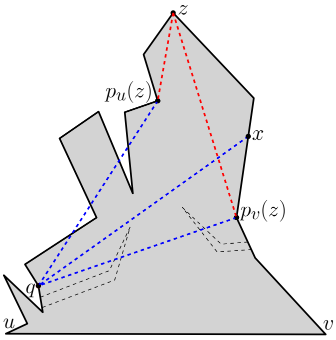

Fig. 1. Case in Lemma 2 where the segment $qp_v(z)$ is intersected by $bd_c(u, p_u(z))$

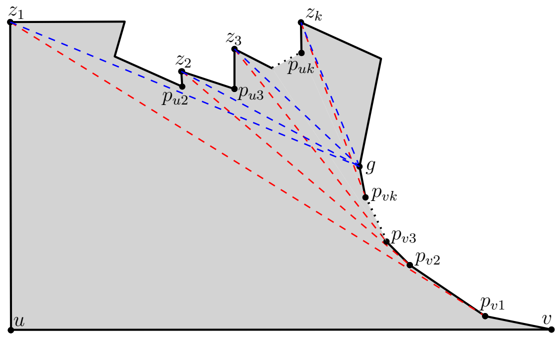

Fig. 2. An instance where the guard set S_A computed by Algorithm 2.1 is arbitrarily large compared to an optimal guard set S_{opt}

In the new algorithm, $bd_c(u, v)$ is scanned to identify a set of unmarked vertices, denoted as B, such that all vertices of P are visible from guards in $S_B = \{p_u(z) | z \in B\} \cup \{p_v(z) | z \in B\}$. During the scan, let z denote the current unmarked vertex under consideration. At every step, the algorithm maintains the invariance that, for every unmarked vertex y of $bd_c(u, z)$ (excluding z), $p_u(y)$ and $p_v(y)$ see all unmarked vertices of $bd_c(p_u(y), y)$. Let z' denote the next unmarked vertex of $bd_c(z, p_v(z))$ in clockwise order from z such that z' is not visible from either $p_u(z)$ or $p_v(z)$. Depending on whether z' exists, the current vertex z must satisfy one of the following properties.

(A) All vertices of $bd_c(z, p_v(z))$ are already marked due to the guards currently included in S_B (see Fig. 3).

(B) Every unmarked vertex of $bd_c(z, p_v(z))$ is visible from $p_u(z)$ or $p_v(z)$ (see Fig. 4).

(C) Not all unmarked vertices of $bd_c(p_u(z'), z')$ are visible from $p_u(z')$ or $p_v(z')$ (see Fig. 5).

(D) Every unmarked vertex of $bd_c(p_u(z'), z')$ is visible from $p_u(z')$ or $p_v(z')$ (see Fig. 6).

If z satisfies property (A) or (B), then z is included in B and the first unmarked vertex of $bd_c(p_v(z), v)$ in clockwise order from $p_v(z)$ becomes the new z. If z satisfies property (C), then z is included in B and z' becomes the new z. If z satisfies property (D), then z' becomes the new z. Whenever z is included in B, $p_u(z)$ and $p_v(z)$ are included in S_B and all unmarked vertices that become visible from $p_u(z)$ or $p_v(z)$ are marked. After doing so, if there remain unmarked vertices on $bd_{cc}(z, u)$, then $bd_{cc}(z, u)$ is scanned from z in counterclockwise order and more guards are included in S_B according to the following strategy:-

- (i) $y \leftarrow p_u(z)$
- (ii) Scan $bd_{cc}(y, u)$ from y in counterclockwise till an unmarked vertex x is located.
- (iii) Add x to B. Add $p_u(x)$ and $p_v(x)$ to S_B.
- (iv) Mark every vertex visible from $p_u(x)$ or $p_v(x)$.
- (v) $y \leftarrow p_u(x)$
- (vi) Repeat steps (ii)-(v) until all vertices of $bd_{cc}(z, u)$ are marked.

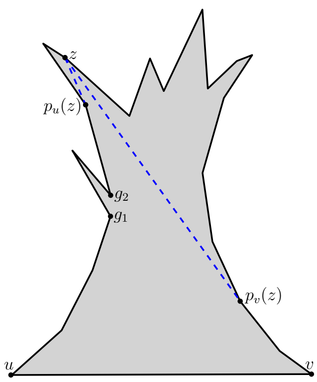

Fig. 3. All vertices of $bd_c(z, p_v(z))$ are already marked due to guards at g_1 and g_2

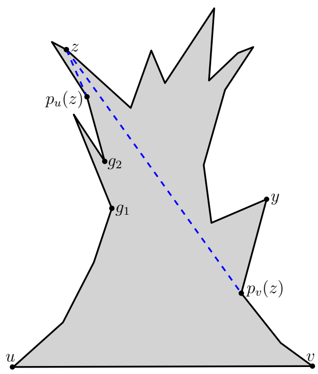

Fig. 4. The only unmarked vertex y of $bd_c(z, p_v(z))$ is visible from $p_v(z)$

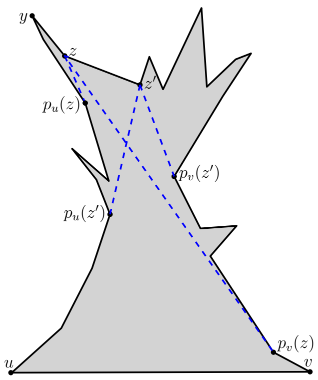

Fig. 5. Guards at $p_u(z')$ and $p_v(z')$ do not see the unmarked vertex y of $bd_c(p_u(z'), z')$

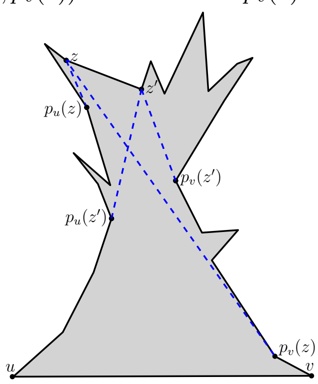

Fig. 6. Guards at $p_u(z')$ and $p_v(z')$ see all unmarked vertices of $bd_c(p_u(z'), z')$

Initially, z is chosen to be the first unmarked vertex of $bd_c(u, v)$ in clockwise order from u. Then, for each z under consideration along the clockwise scan of $bd_c(u, v)$, the appropriate action is performed corresponding to the property of z. Then, z is updated and the process is repeated till v is reached. The set of vertices S_B is returned by the algorithm as a guard set. The entire process is described in pseudocode as Algorithm 2.2.

Algorithm 2.2. An $\mathcal{O}(n^2)$-algorithm for computing a guard set S for all vertices of P

Compute $SPT(u)$ and $SPT(v)$
Initialize all the vertices of P as unmarked
Initialize $B \leftarrow \emptyset$, $S_B \leftarrow \emptyset$ and $z \leftarrow u$
while there exists an unmarked vertex in P **do**
  $z \leftarrow$ the first unmarked vertex on $bd_c(u, v)$ in clockwise order from z
  if every unmarked vertex of $bd_c(z, p_v(z))$ is visible from $p_u(z)$ or $p_v(z)$ **then**
    $B \leftarrow B \cup \{z\}$ and $S_B \leftarrow S_B \cup \{p_u(z), p_v(z)\}$
    Mark all vertices of P that become visible from $p_u(z)$ or $p_v(z)$
    $z \leftarrow p_v(z)$
  else
    $z' \leftarrow$ the first unmarked vertex on $bd_c(z, v)$ in clockwise order
    while every unmarked vertex of $bd_c(p_u(z'), z')$ is visible from $p_u(z')$ or $p_v(z')$
do
      $z \leftarrow z'$ and $z' \leftarrow$ the first unmarked vertex on $bd_c(z', v)$ in clockwise order
    end while
    $w \leftarrow z$
    while there exists an unmarked vertex on $bd_c(u, z)$ **do**
      $B \leftarrow B \cup \{w\}$ and $S_B \leftarrow S_B \cup \{p_u(w), p_v(w)\}$
      Mark all vertices of P that become visible from $p_u(w)$ or $p_v(w)$
      $w \leftarrow$ the first unmarked vertex on $bd_{cc}(w, u)$ in counterclockwise order
    end while
  end if
end while
return the guard set $S = S_B$

For showing an upper bound on S, a bipartite graph $G = (B \cup S_{opt}, E)$ is constructed such that the degree of each vertex in B is exactly 1 and the degree of each vertex in S_{opt} is at most 2.

The graph G is constructed as follows. For every vertex $b_i \in B$, there exists a $g \in S_{opt}$ such that g sees b_i. By Lemma 1, g must lie on $bd_c(p_u(b_i), p_v(b_i))$. If g lies on $bd_c(p_u(b_i), b_i)$, then the edge gb_i is added to E. Observe that any vertex q lying on $bd_{cc}(p_u(b_i), b_i)$ that is visible from g is also visible from $p_u(b_i)$ or $p_v(b_i)$ (see the proof of Lemma 2). So, q is marked on inclusion of b_i in B, and therefore q cannot be included in B. Hence, for $k > i$, no vertex $b_k \in B$ exists that can add an edge gb_k.

If g lies on $bd_c(b_i, b'_i)$ and sees another $b_j \in B$, then the edges gb_i and gb_j are added to G. Similar arguments as above show that, for $k > j$, no vertex $b_k \in B$ exists that can add an edge gb_k.

If g lies on $bd_c(b'_i, p_v(b'_i)$ (see Fig. 7), there exists a vertex x_i on $bd_c(p_u(b_i), b_i)$ such that x_i is visible from $p_u(b_i)$ or $p_v(b_i)$, but not from $p_u(b'_i)$ or $p_v(b'_i)$. So, in order to see x_i, there must exist another guard $g' \in S_{opt}$ lying on $bd_c(p_v(b'_i), p_v(x_i))$. The edge $g'b_i$ is added to G. Let $V_{g'}$ denote the set of vertices of P lying on $bd_c(p_v(b'_i), p_v(x_i))$ that are visible from g'. If $V_{g'}$ does not contain any vertex of B, then the degree of g' is 1 in G. Otherwise, the first vertex $b_j \in B$ of $V_{g'}$ in clockwise order from $p_v(b'_i)$ is located and the edge $g'b_j$ is added to G. Now, every vertex belonging to $V_{g'}$ must be visible from $p_u(b_j)$ or $p_v(b_j)$, which means that no other vertex of $V_{g'}$ can be included in B. Hence, for $k > j$, no vertex $b_k \in B$ exists which can have an edge $g'b_k$ incident on g', ensuring that the degree of g' is at most 2 in G.

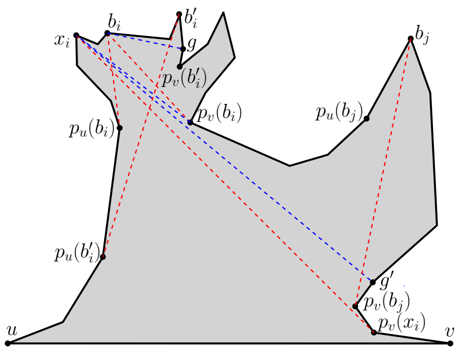

Fig. 7. The guard $g \in S_{opt}$ is located on $bd_c(b'_i, p_v(b'_i)$

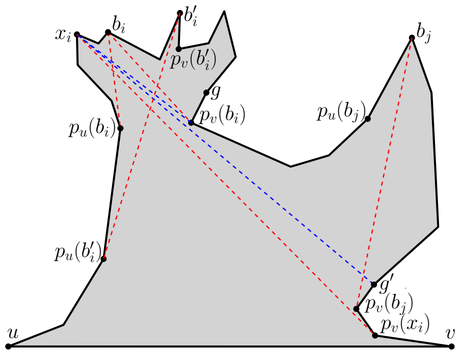

Fig. 8. The guard $g \in S_{opt}$ is located on $bd_c(p_v(b'_i), p_v(b_i))$

If g lies on $bd_c(p_v(b'_i), p_v(b_i)$ (see Fig. 8), then add the edge gb_i to G. Observe that no vertex lying on $bd_c(b'_i, p_v(b'_i))$ can be visible from g. Moreover, at most one other vertex $b_j \in B$ lying on $bd_c(p_v(b'_i), p_v(b_i))$ is visible from g, as explained earlier for the case of $g' \in S_{opt}$ seeing x_i. If b_j exists, then the edge gb_j is added to G, ensuring the degree of g is at most 2 in G. As a direct consequence of the above construction, we have the following results.

Lemma 5. *In the bipartite graph G, the degree of each vertex in B is exactly 1 and degree of each vertex in S_{opt} is at most 2.*

Corollary 1. $|B| \leq 2|S_{opt}|$.

Theorem 2. $|S| \leq 4|S_{opt}|$.

Proof. By arguments similar to those in the proof of Lemma 4, it can be shown that $|S_B| = 2|B|$. Also, by Corollary 1, $|B| \leq 2|S_{opt}|$. Therefore, $|S| = |S_B| = 2|B| \leq 4|S_{opt}|$.

2.2 Guarding All Interior Points of a Polygon

In the previous subsection, we presented an algorithm (see Algorithm 2.2) which returns a guard set S such that all vertices of P are visible from guards in S. However, it may not always be true that all interior points of P are also visible from guards in S. Consider the polygon shown in Fig. 9. While scanning $bd_c(u, v)$, our algorithm places guards at $p_u(z)$ and $p_v(z)$ as all vertices of $bd_c(p_u(z), p_v(z))$ become visible from $p_u(z)$ or $p_v(z)$. Observe that in fact all vertices of P become visible from these two guards. However, the triangular region $P \setminus (VP(p_u(z)) \cup VP(p_v(z)))$, bounded by the segments x_1x_2, x_2x_3 and x_3x_1, is not visible from $p_u(z)$ or $p_v(z)$. Also, one of the sides x_1x_2 of the triangle $x_1x_2x_3$ is a part of the polygonal edge a_1a_2. In fact, for any such region invisible from guards in S, one of the sides must always be a part of a polygonal edge. Otherwise, there should exist another guard g (see Fig. 9) from which the entire polygonal side (x_1x_2) of the region is visible and yet some portion of the region (including x_3) is not visible. However, such a vertex g cannot be weakly visible from the edge uv, which is a contradiction. Henceforth, any such region invisible from guards in S is referred to as an *invisible cell*, and the polygonal edge which contributes as a side to the invisible cell is referred to as its corresponding *partially invisible edge*. One additional guard is required in order to see each invisible cell entirely. For example, in Fig. 9, an extra guard is required at a vertex of $bd_c(z, w)$, since none of the vertices outside this boundary can see all points of the triangular invisible cell $x_1x_2x_3$.

The boundary of the visibility polygon $VP(s)$ of any vertex s consists of polygonal edges and constructed edges. A *constructed edge* yx is an edge formed by extending the segment sy (which could be either an edge of P or an internal segment), where y is some other vertex of P, till it touches the boundary of P at a point x. If y lies on $bd_c(s, x)$, the region of P bounded by $bd_c(y, x)$ and xy is referred to as the *left pocket* of $VP(z)$. Similarly, if y lies on $bd_{cc}(s, x)$, then the region of P bounded by $bd_{cc}(y, x)$ and xy is referred to as the *right pocket* of $VP(z)$. In both these cases, we refer to the vertex y as the *lid vertex* and the point x as the *lid point* of the corresponding left or right pocket.

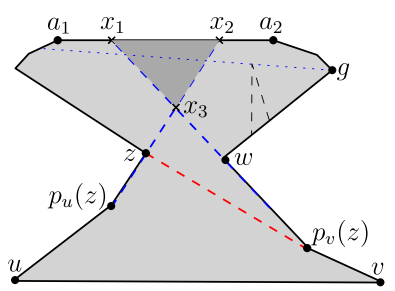

Fig. 9. All vertices are visible from $p_u(z)$ or $p_v(z)$, but triangle $x_1x_2x_3$ is invisible

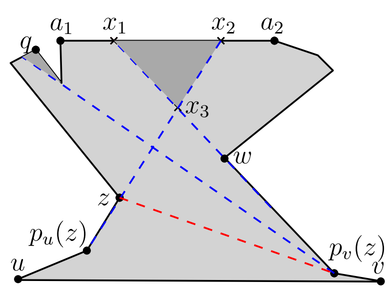

Fig. 10. The left pocket of $VP(p_u(z))$ can contain only one invisible cell

Observe that each invisible cell must be wholly contained within the intersection region (which is a triangle) of a left pocket and a right pocket. For example, in Fig. 9, the invisible cell $x_1x_2x_3$ is actually the entire intersection region of the left pocket of $VP(p_u(z))$ and the right pocket of $VP(p_v(z))$. Also, z is the lid vertex and x_2 is the lid point of the left pocket of $VP(p_u(z))$. Similarly, w is the lid vertex and x_1 is the lid point of the right pocket of $VP(p_v(z))$.

Suppose $bd_c(z, x_2)$ contains reflex vertices (see Fig. 10). In that case, in addition to the invisible cell $x_1x_2x_3$, the left pocket of $VP(p_u(z))$ may contain several regions that are not visible from $p_v(z)$. However, in each such region there exists a vertex, say q, that is not visible from $p_v(z)$, which contradicts the fact that all vertices of $bd_c(p_u(z), p_v(z))$ are visible from $p_u(z)$ or $p_v(z)$. So, the left pocket of $VP(p_u(z))$ can contain only one invisible cell. Analogously, the right pocket of $VP(p_v(z))$ can contain only one invisible cell.

Now consider the situation (as shown in Fig. 11) where $VP(p_u(z))$ has several left pockets and $VP(p_v(z))$ has several right pockets which intersect pairwise to create multiple invisible cells. In order to guard these invisible cells, additional guards are placed as follows. Let c_1 be the lid point of the left pocket containing the first invisible cell in clockwise order. Then, guards are placed at $p_u(c_1)$ and $p_v(c_1)$. Now, for every invisible cell T, the portions of T are removed that are visible from $p_u(c_1)$ or $p_v(c_1)$. Note that some of these cells may turn out to be totally visible and hence may be eliminated altogether. This process is repeated until all invisible cells become totally visible.

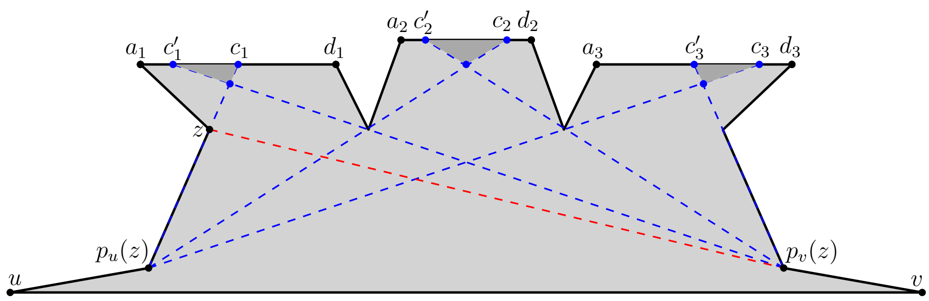

Fig. 11. Multiple invisible cells exist within the polygon that are not visible from the guards placed at $p_u(z)$ and $p_v(z)$

In general, we may have a situation where multiple invisible cells are created by the intersection of the left and right pockets of arbitrary pairs of guards belonging to S (see Fig. 12). In this scenario, all invisible cells are guarded by introducing a set of additional guards S' as follows. Initially, both C and S' are empty. Scan $bd_c(u, v)$ from u in clockwise order to locate the first edge a_id_i that is not totally visible from guards in $S \cup S'$, where d_i is the next clockwise vertex of a_i. Let c'_ic_i be the portion of a_id_i that is not visible from guards in $S \cup S'$, where $c'_i \in bd_c(a_i, c_i)$ and $c_i \in bd_c(c'_i, d_i)$. In other words, c'_ic_i is the polygonal side of the first invisible cell. Add $p_u(c_i)$ and $p_v(c_i)$ to S'. Also, add c_i to C. Repeat this process until all the edges of P are totally visible from guards in $S \cup S'$. At its termination, let us assume that $C = \{c_1, c_2, \ldots, c_k\}$. The entire procedure is described in pseudocode as follows.

Algorithm 2.3. An $\mathcal{O}(n^2)$-algorithm for computing a guard set $S \cup S'$ for guarding P entirely

---

Compute $SPT(u)$ and $SPT(v)$
Compute the set of guards S using Algorithm 2.2.
Initialize $C \leftarrow \emptyset$, $S' \leftarrow \emptyset$ and $z \leftarrow u$
while there exists an edge in P that is partially visible from guards in $S \cup S'$ **do**
    $z' \leftarrow$ the vertex next to z in clockwise order on on $bd_c(u, v)$
    if if the edge zz' is partially visible from guards in $S \cup S'$ **then**
        $c_i \leftarrow$ the lid point of the left pocket on zz'
        $C \leftarrow C \cup \{c_i\}$ and $S' \leftarrow S' \cup \{p_u(c_i), p_v(c_i)\}$
    end if
    $z \leftarrow z'$
end while
return the guard set $S \cup S'$

Theorem 3. *The running time of Algorithm 2.3 is $\mathcal{O}(n^2)$.*

Proof. $SPT(u)$ and $SPT(v)$ can be computed in $\mathcal{O}(n)$ time [15]. Then, the computation of the guard set S takes $\mathcal{O}(n^2)$ time, since it involves scanning the boundary of P and identifying vertices to be marked whenever new guards are placed. The number of lid points on an edge can be at most $\mathcal{O}(n)$. Therefore, each time a new vertex is added to S', the invisible portion of the first partially visible edge in clockwise order can be determined in $\mathcal{O}(n)$ time. Hence, the overall running time of Algorithm 2.3 is $\mathcal{O}(n^2)$.

Theorem 4. $2|C| = |S'| \leq 2|S_{opt}|$.

Proof. For every $c_i \in C$, there exists an invisible cell T_i. For every such invisible cell T_i, let l_i and r_i respectively denote the lid vertices of the left and right

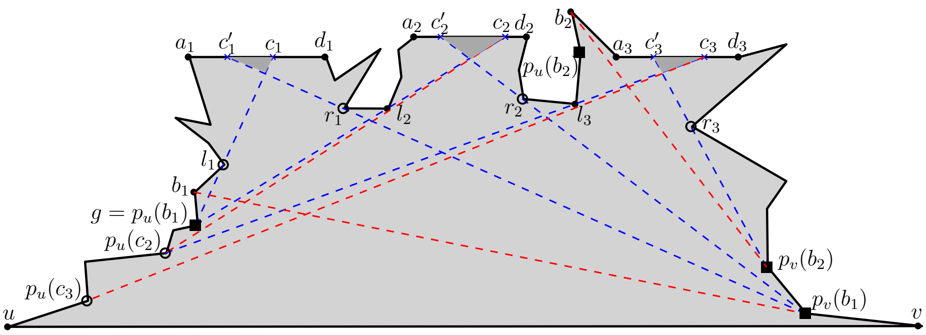

Fig. 12. Placement of guards to in order to see all invisible cells

pockets intersecting to form T_i (see Fig. 12). Let $g \in S$ be the guard such that l_i is the lid vertex of a left pocket of $VP(g)$. Similarly, let $g' \in S$ be the guard such that r_i is the lid vertex of a right pocket of $VP(g')$.

Assume that, for every T_i, there exists at least one guard in S_{opt} that sees all points of T_i. Now, consider any guard $g_{opt} \in S_{opt}$ that sees all points of T_i. Then, g_{opt} can lie on $bd_c(l_i, r_i)$. Also, g_{opt} can lie on $bd_c(p_u(c_i), g)$, but only when $p_u(c_i) \neq l_i$ and $p_u(c_i)$ lies on $bd_c(u, g)$. Now, let z be the vertex such that $p_v(z) = g'$. Then, no vertex of $bd_c(z, g')$ is visible from any vertex of $bd_c(g', v)$. Further, if z is such that $p_u(z) = g$, then z has to lie on $bd_c(g, l_i)$. Otherwise, z has to lie on $bd_c(l_i, c'_i)$. In either case, g_{opt} cannot lie on $bd_c(g', v)$ since c'_i lies on $bd_c(z, g')$.

Since the guard set S' includes $p_u(x)$ and $p_v(x)$ for every $z \in C$, clearly $|S'| = 2|C|$. If for every i, there exists an unique vertex belonging to S_{opt} that sees all points of T_i, then obviously $|S'| \leq 2|S_{opt}|$. Consider the special situation where $l_{i+1} = r_i$ for some i (see Fig. 11) so that both T_i and T_{i+1} are totally visible from r_i. Since all points of T_i are visible from r_i, it must be the case that $p_v(c_i) = r_i$. Moreover, r_i can be a vertex of S_{opt}. Therefore, no additional guards are chosen for T_{i+1} because all points of T_{i+1} become visible from the guard already placed at r_i.

If no vertex of $bd_c(l_i, r_i)$ belongs to S_{opt}, then there must be a vertex of S_{opt} lying on $bd_c(p_u(c_i), g)$ and $p_u(c_i)$ must belong to $bd_c(u, g)$. If $p_u(c_{i-1})$ also belongs to $bd_c(u, g)$, then S_{opt} must have a vertex on the boundary $bd_c(p_u(c_i), p_v(c_{i-1}))$ in order to see T_{i-1} because l_{i-1} is the lid vertex of a left pocket of $VP(p_u(c_{i-1}))$. Hence, $2|C| = |S'| \leq 2|S_{opt}|$.

Finally, if we remove the assumption that there exists at least one guard in S_{opt} that sees all points of T_i, then the size of S_{opt} increases but the size of our guard set S' remains the same. Therefore, the bound $|S'| \leq 2|S_{opt}|$ is still preserved.

Theorem 5. *$|S \cup S'| \leq 6|S_{opt}|$.*

Proof. By Theorem 2, $|S| \leq 4|S_{opt}|$ and by Theorem 4, $|S'| \leq 2|S_{opt}|$.
Therefore, $|S \cup S'| \leq |S| + |S'| \leq 4|S_{opt}| + 2|S_{opt}| \leq 6|S_{opt}|$.

References

1. Aggarwal, A.: The art gallery theorem: its variations, applications and algorithmic aspects. PhD thesis, The Johns Hopkins University, Baltimore, MD (1984)
2. Avis, D., Toussaint, G.: An optimal algorithm for determining the visibility of a polygon from an edge. IEEE Transactions on Computers 30, 910–914 (1981)
3. Bhattacharya, P., Ghosh, S.K., Roy, B.: Vertex guarding in weak visibility polygons. CoRR, abs/1409.4621 (2014)
4. Chtal, V.: A combinatorial theorem in plane geometry. Journal of Combinatorial Theory, Series B 18(1), 39–41 (1975)
5. Deshpande, A., Kim, T.-J., Demaine, E.D., Sarma, S.E.: A pseudopolynomial time $o(\log n)$-approximation algorithm for art gallery problems. In: Dehne, F., Sack, J.-R., Zeh, N. (eds.) WADS 2007. LNCS, vol. 4619, pp. 163–174. Springer, Heidelberg (2007)

6. Efrat, A., Har-Peled, S.: Guarding galleries and terrains. Information Processing Letters 100(6), 238–245 (2006)
7. Eidenbenz, S., Stamm, C., Widmayer, P.: Inapproximability of some art gallery problems. In: Canadian Conference on Computational Geometry, pp. 1–11 (1998)
8. Eidenbenz, S., Stamm, C., Widmayer, P.: Inapproximability results for guarding polygons and terrains. Algorithmica 31(1), 79–113 (2001)
9. Fisk, S.: A short proof of Chvátal's watchman theorem. Journal of Combinatorial Theory, Series B 24(3), 374 (1978)
10. Ghosh, S.K.: Approximation algorithms for art gallery problems. In: Proc. of Canadian Information Processing Society Congress, pp. 429–434 (1987)
11. Ghosh, S.K.: Visibility Algorithms in the Plane. Cambridge University Press (2007)
12. Ghosh, S.K.: Approximation algorithms for art gallery problems in polygons. Discrete Applied Mathematics 158(6), 718–722 (2010)
13. Ghosh, S.K.: Approximation algorithms for art gallery problems in polygons and terrains. In: Rahman, M. S., Fujita, S. (eds.) WALCOM 2010. LNCS, vol. 5942, pp. 21–34. Springer, Heidelberg (2010)
14. Ghosh, S.K., Maheshwari, A., Pal, S., Saluja, S., Veni Madhavan, C.E.: Characterizing and recognizing weak visibility polygons. Computational Geometry 3(4), 213–233 (1993)
15. Guibas, L.J., Hershberger, J., Leven, D., Sharir, M., Tarjan, R.E.: Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. Algorithmica 2, 209–233 (1987)
16. Katz, M.J., Roisman, G.S.: On guarding the vertices of rectilinear domains. Computational Geometry 39(3), 219–228 (2008)
17. King, J., Kirkpatrick, D.G.: Improved approximation for guarding simple galleries from the perimeter. Discrete & Computational Geometry 46(2), 252–269 (2011)
18. Lee, D.T., Lin, A.: Computational complexity of art gallery problems. IEEE Transactions on Information Theory 32(2), 276–282 (1986)
19. O'Rourke, J., Supowit, K.J.: Some NP-hard polygon decomposition problems. IEEE Transactions on Information Theory 29(2), 181–189 (1983)
20. Schuchardt, D., Hecker, H.-D.: Two NP-Hard Art-Gallery Problems for Ortho-Polygons. Mathematical Logic Quarterly 41, 261–267 (1995)