A 1/4-Approximation Algorithm for the Maximum Hidden Vertex Set Problem in Simple Polygons*

Carlos Alegría[†1], Pritam Bhattacharya[‡2], and Subir Kumar Ghosh[3]

1 Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México
calegria@uxmcc2.iimas.unam.mx

2 Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur
pritam.bhattacharya@cse.iitkgp.ernet.in

3 Department of Computer Science, School of Mathematical Sciences, Ramakrishna Mission Vivekananda University
subir.ghosh@rkmvu.ac.in

Abstract

Given a simple polygon, two points in its interior are said to be *hidden* to each other if the straight line segment connecting them intersects the exterior of the polygon. We study the *Maximum Hidden Vertex Set* problem, where given a simple polygon, we are required to find a subset of vertices of maximum cardinality such that every pair of them are hidden to each other. This problem is known to be NP-hard, and in fact also APX-hard. In this paper we present a $O(n^2)$ time algorithm to compute a 1/4-approximation to the maximum hidden vertex set of a simple polygon. Although exact algorithms are known for some special classes of polygons (such as polygons that are weakly visible from a convex edge), to the best of our knowledge this is the first deterministic polynomial-time algorithm to compute a constant-factor approximation to the optimal solution for general simple polygons without holes.

Lines 170

1 Introduction

Visibility problems are some of the most prominent and intensively studied problems in Computational Geometry. Since the classic Art Gallery Problem was proposed in 1973 by Victor Klee, many extensions have been studied [11, 15, 9], and combinatorial as well as computational results have been applied to practical problems that can commonly be found in computer generated graphics [4], computer vision [6], and robotics [10].

A well known class of visibility problems are those related to hiding. Given a simple polygon, we say that two points in its interior are *visible* to each other if the line segment connecting the points does not intersect the exterior of the polygon. Conversely, the points are said to be *hidden* to each other if they are not mutually visible. In this paper we study the so called Maximum Hidden Vertex Set (MHVS) problem, where given a simple polygon

* Preliminary results were obtained during the working and interacting sessions of the Intensive Research Program in Discrete, Combinatorial and Computational Geometry. We thank the Centre de Recerca Matemàtica, Universitat Autónoma de Barcelona, for hosting this event and the organizers for providing us with the platform to meet and collaborate. We also thank Jayson Lynch from Massachusetts Institute of Technology and Bodhayan Roy from Masaryk University for useful discussions.

† Partially supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734922.

‡ Supported by cycle 11 of the Tata Consultancy Services (TCS) Research Scholarship.

35th European Workshop on Computational Geometry, Utrecht, The Netherlands, March 19–20, 2019.

P, we want to find a subset of vertices of maximum cardinality such that every pair of them are hidden to each other.

The MHVS problem, which can also be looked upon as the problem of computing a maximum independent set in the vertex visibility graph of P, is known to be NP-hard [13]. It is in fact not even easy to find an approximate solution. It was shown to be APX-hard by Eidenbenz [5] even in polygons with no holes. Nevertheless, an exact solution can be computed in polynomial time for special classes of simple polygons. A maximum hidden vertex set can be computed in $O(n^2)$ time in a polygon weakly visible from a convex edge [7] (we describe weakly visible polygons in Section 2), and in $O(ne)$ time in the class of so called convex fans, where e is the number of edges of their vertex visibility graph [8]. Heuristic-based algorithms have also been explored which seem to work well in practice, as evidenced by experimental results showing that they provide solutions that are usually quite close to the exact solution for input polygons without holes [1]. Recently, there has also been a study on gender-aware facility location problems [12], which are closely related to the MHVS problem.

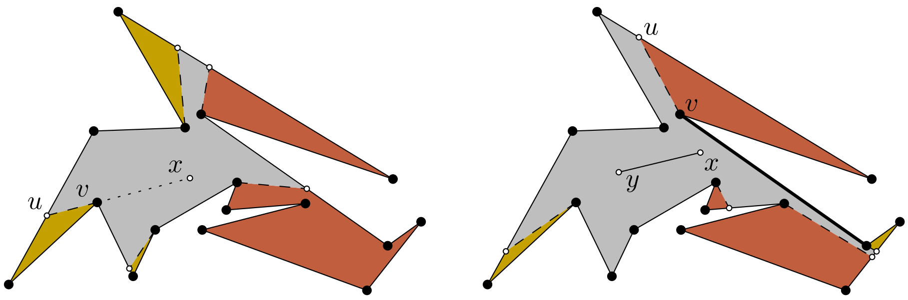
In this paper, we describe a deterministic $O(n^2)$ time algorithm to compute a 1/4-approximation to the maximum hidden vertex set of an n-sided simple polygon with no holes. As far as we are aware, this is the first deterministic algorithm to compute a constant-factor approximation to the optimal solution of the MHVS problem for general simple polygons.

2 Preliminaries

Hereafter, let P be a simple polygon with no holes. For the sake of simplicity, we assume that no three vertices of P are collinear. Given a point x inside P, we denote with $\mathcal{VP}(x)$ the visibility polygon of x. The boundary of $\mathcal{VP}(x)$ is a closed polygonal chain formed by polygonal edges and non-polygonal edges called *constructed edges*. A constructed edge connects a reflex vertex v with a point u lying on an edge of P (see Figure 1a), where the points x, v, and u are collinear.

Let x and y be two points inside P that are visible to each other. A point inside P is said to be *weakly visible* from the line segment $\overline{xy}$, if it is visible from at least one point of $\overline{xy}$. The set of points inside P that are weakly visible from $\overline{xy}$ is called the *weak visibility polygon* of $\overline{xy}$. We denote this polygon with $\mathcal{VP}(\overline{xy})$. Like the visibility polygon of a point, the boundary of $\mathcal{VP}(\overline{xy})$ is a closed polygonal chain formed by polygonal edges and constructed edges. If the boundary of $\mathcal{VP}(\overline{xy})$ contains no constructed edges, then $\mathcal{VP}(\overline{xy}) = P$ and the polygon is said to be *weakly visible* from $\overline{xy}$ (see Figure 1b).

Let ∂P denote the boundary of P. Given two points $a, b \in \partial P$, let $bd(a, b)$ denote the clockwise boundary of P from a to b, so we have $\partial P = bd(a, a) = bd(a, b) \cup bd(b, a)$. Consider a point x inside P and a constructed edge $\overline{vu}$ of $\mathcal{VP}(x)$, where v is a vertex and u is a point on an edge of P. The segment $\overline{vu}$ divides P into two subpolygons: one bounded by $bd(v, u) \cup \overline{vu}$ and the second one bounded by $bd(u, v) \cup \overline{vu}$. Out of these two, the subpolygon that does not contain x is called a *pocket* of $\mathcal{VP}(x)$. We denote this pocket with $P(v, u)$. If x is not contained in the polygon bounded by $bd(v, u) \cup \overline{vu}$, then $\overline{vu}$ is called a *left constructed edge* and $P(v, u)$ is called a *left pocket*. Otherwise, $\overline{vu}$ is called a *right constructed edge* and $P(v, u)$ is called a *right pocket*. The constructed edges and left and right pockets of a weak visibility polygon are defined in a similar way. Examples of these definitions are shown in Figure 1. In particular, in Figure 1a the segment $\overline{vu}$ is a left constructed edge and $P(v, u)$ is a left pocket of $\mathcal{VP}(x)$. On the other hand, in Figure 1b the segment $\overline{vu}$ is a right constructed edge and $P(v, u)$ is a right pocket of $\mathcal{VP}(\overline{xy})$.



(a) The visibility polygon of x. **(b)** The weak visibility polygon of $\overline{xy}$.

Figure 1 A simple polygon along with (a) the visibility polygon of a point, and (b) the weak visibility polygon of a line segment, both in gray. The constructed edges of the visibility polygons are shown in dashed lines, the left pockets in yellow, and the right pockets in red. The polygon is weakly visible from the thick edge.

▶ **Lemma 2.1** (Lemma 2 from Bärtschi et al. [2]). *Let $\overline{v_1u_1}$ and $\overline{v_2u_2}$ be two left constructed edges (or similarly, two right constructed edges) of $\mathcal{VP}(\overline{xy})$, where v_1 and v_2 are reflex vertices of P. Possibly excluding v_1 and v_2, the vertices of P inside the left (or right) pocket $P(v_1, u_1)$ are hidden from the vertices of P inside the left (or right) pocket $P(v_2, u_2)$.*

3 The polygon partition

Our algorithm is based on a link-distance-based partition from Bhattacharya et al. [3] (which is itself adapted from the partitioning method used by Suri [14]) that decomposes the polygon P into a set of disjoint visibility windows. We next outline how this partition is constructed and describe properties that are relevant to our algorithm.

Given two points x and y inside P, the *link distance* from x to y is the minimum number of line segments required in a polygonal chain inside P to connect x to y. The visibility window decomposition is a hierarchical partition of P into visibility polygons, where polygons on the same level contain points at the same link distance from a given vertex p. The first level of the hierarchy is formed by the set $V_1 = \{\mathcal{VP}(p)\}$ that contains the points of P at link distance one from p. Let $\overline{v_1u_1}, \ldots, \overline{v_cu_c}$ be the constructed edges of $\mathcal{VP}(p)$ in clockwise order around p, where v_i is a vertex and u_i is a point lying on an edge of P. The region $P \setminus \mathcal{VP}(p)$ consists of c disjoint polygons we denote with $P_1, \ldots, P_c$. Let $V_{2,i} = \mathcal{VP}(\overline{v_iu_i}) \cap P_i$ be the weak visibility polygon of $\overline{v_iu_i}$ inside P_i. The second level of the hierarchy is formed by the set $V_2 = \{V_{2,1}, \ldots, V_{2,c}\}$ of disjoint weak visibility polygons. The remaining levels are formed by the sets $V_3, V_4, \ldots$ obtained by repeating the previous process until we have a set V_d of disjoint weak visibility polygons with no constructed edges. We thus have that $P = V_1 \cup \cdots \cup V_d = \mathcal{VP}(p) \cup V_{2,1} \cup V_{2,2} \cup \cdots \cup V_{d,1} \cup V_{d,2} \cup \cdots$. The set V_i is formed by the disjoint regions containing the points at link distance i from p, and d is the maximum link distance from p to any point inside P (see Figure 2).

▶ **Lemma 3.1.** *The following statements hold true for the visibility window partition of P:*

i) The vertices of P lying in any subpolygon belonging to V_i are hidden from the vertices of P lying in any subpolygon belonging to V_j, unless $|j - i| \leq 1$.

ii) Let $\overline{uv}$ be a constructed edge of the weak visibility polygon $V_{i,j}$. Then, the constructed edge $\overline{uv}$ is actually a convex edge with respect to every subpolygon $V_{i+1,j} = \mathcal{VP}(\overline{uv}) \cup P_j$.

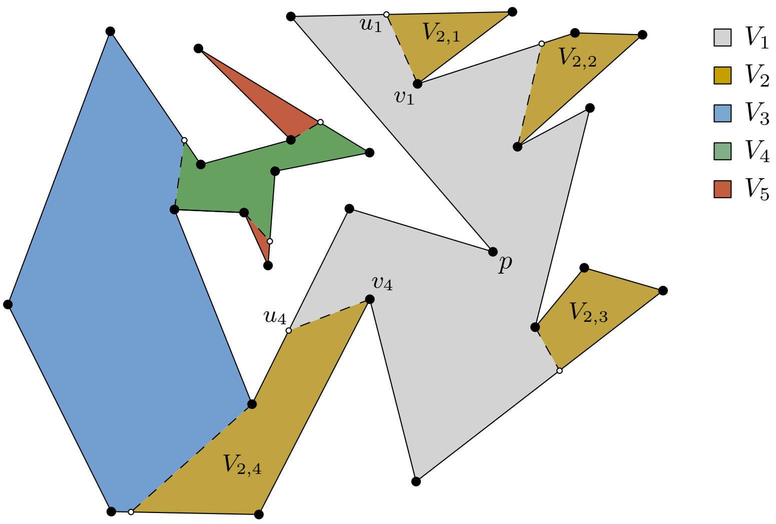


Figure 2 The partition of a simple polygon into weak visibility subpolygons, where each subpolygon consists of points at the same link distance from the vertex p.

Proof. The lemma follows directly from the construction of the partition. For details, please refer to the proof of Lemma 2 from Bhattacharya et al. [3]. ◀

4 The algorithm

We now describe an $O(n^2)$ time 1/4-approximation algorithm for the Maximum Hidden Vertex Set problem in simple polygons. The overall strategy of our algorithm is to decompose P into regions as we described in Section 3, and classify the regions of the partition into four disjoint sets such that vertices of regions belonging to different sets are hidden to each other. We then compute the (exact) maximum hidden set of every region using the algorithm from Ghosh et al. [7], and keep the hidden set with more guards in the same group. Hereafter, we denote by n the number of vertices of P.

1. **Partition the polygon P using link distance**
   Create the decomposition of P based on the link-distance from an arbitrary vertex that we described in Section 3. This partition can be created in $O(n)$ time [3].
2. **Classify visibility windows**
   As P is a simple polygon without any holes, the dual graph of the partition is a tree. Each node of this tree represents a visibility polygon of the partition, and the children of a node are the regions inside the pockets formed by constructed edges belonging to their parent's visibility polygon. Using this tree, we separate the nodes into four disjoint sets in the following manner. First we separate the nodes into two sets: those appearing at odd levels in the tree, and those appearing at even levels in the tree. Then, we further separate the nodes in each of the above sets into two subsets: those created due to a left constructed edge, and those created due to a right constructed edge. At the end of this separation process, we obtain four disjoint subsets, which are as follows:
   - R_1, containing regions at odd levels in the tree created by a left constructed edge
   - R_2, containing regions at odd levels in the tree created by a right constructed edge
   - R_3, containing regions at even levels in the tree created by a left constructed edge
   - R_4, containing regions at even levels in the tree created by a right constructed edge

Observe that the vertices of P inside different regions of the same set are hidden from each other (see Figure 3), either because they belong to non-consecutive levels of the tree (see Lemma 3.1), or because both of them belong to the same level in the tree and are both created by a left (or right) constructed edge (see Lemma 2.1). Also note that the separation process can be completed in $O(n)$ time.

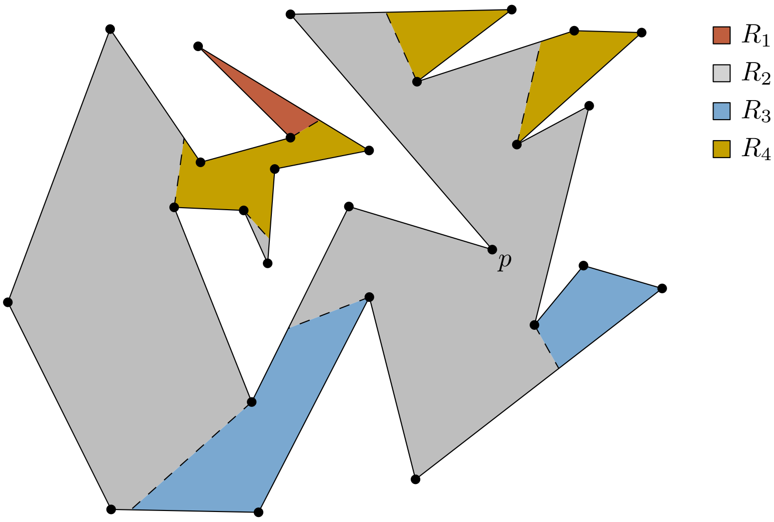


Figure 3 The separation of the regions into four independent sets.

3. Compute an approximate maximum hidden vertex set

Observe that each region of the partition is a polygon which is weakly visible from the constructed edge (of its parent's visibility polygon) that created it. So, within each region, we can compute the (exact) maximum hidden set of vertices using the algorithm by Ghosh et al. [7], which computes the maximum hidden set of a weak visibility polygon with n vertices in $O(n^2)$ time.

Since the vertices of P inside two regions belonging to the same set (from among R_1, R_2, R_3, R_4) are hidden from each other, the union of the maximum hidden sets of the regions in a particular set is a valid hidden set for P. Thus, we can compute four valid hidden sets S_1, S_2, S_3, S_4, that correspond to the sets R_1, R_2, R_3, R_4 respectively, in $O(n^2)$ time. Out of these four valid hidden vertex sets of P, we choose as our approximation of the maximum hidden set the one containing the most number of vertices.

Let $\mathcal{S}^{\text{opt}}$ denote an exact maximum hidden vertex set of P. Also, let $\mathcal{S}_{i,j}^{\text{opt}} \subseteq \mathcal{S}^{\text{opt}}$ denote the subset of vertices that lie within the weak visibility subpolygon $V_{i,j}$ in the partitioning of P. If we denote the exact maximum hidden set computed for each subpolygon $V_{i,j}$ by $\mathcal{S}_{i,j}^*$, then observe that $|S_{i,j}^*| \geq |S_{i,j}^{\text{opt}}|$. Therefore, we have:

$$|S_1| + |S_2| + |S_3| + |S_4| = \sum_{i,j} |S_{i,j}^*| \geq \sum_{i,j} |S_{i,j}^{\text{opt}}| = |\mathcal{S}_{\text{opt}}|$$

$$\max\left(|S_1|, |S_2|, |S_3|, |S_4|\right) \geq \frac{|S_1| + |S_2| + |S_3| + |S_4|}{4} \geq \frac{|\mathcal{S}_{\text{opt}}|}{4}$$

Therefore, by choosing from among $S_1, \ldots, S_4$ the set containing the maximum number of vertices, we obtain a $\frac{1}{4}$-approximation of $\mathcal{S}_{\max}$. Note that Step 3 is the most expensive step of the algorithm described above, so the algorithm runs in $O(n^2)$ time. This leads us to our main result, which we summarize below.

▶ **Theorem 4.1.** *Given a simple polygon P with n vertices, there exists a $\frac{1}{4}$-approximation algorithm for computing the maximum hidden vertex set in P, which runs in $O(n^2)$ time.*

5 Concluding remarks

We present a $O(n^2)$ time algorithm to compute a 1/4-approximation to the maximum hidden vertex set of a simple polygon with n vertices and no holes. To the best of our knowledge, this is the first constant-factor approximation algorithm for general simple polygons without holes. Observe that our current algorithm cannot be applied when the input polygon P has holes, since then the dual graph of the visibility window partitioning of P is no longer guaranteed to be a tree. However, we are currently investigating possible improvements to our algorithm which could make it work even for input polygons containing holes. Another future research direction is to explore variants of the problem involving restricted-orientation models of visibility, such as rectangular, periscope, or $\mathcal{O}$-visibility.

Acknowledgments

We thank the reviewers for their careful reading and many useful comments, which really helped us to improve the quality and readability of this paper. In particular we thank the reviewer who pointed out to us some interesting references that might motivate future research directions.

References

1 Antonio L. Bajuelos, Samtiago Canales, Gregorio Hernández, and A. Mafalda Martins. Estimating the maximum hidden vertex set in polygons. In *International Conference on Computational Sciences and Its Applications, ICCSA 2018*, pages 421–432, 2008. doi:10.1109/ICCSA.2008.19.

2 Andreas Bärtschi, Subir Kumar Ghosh, Matúš Mihalák, Thomas Tschager, and Peter Widmayer. Improved bounds for the conflict-free chromatic art gallery problem. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, SOCG'14, pages 144:144–144:153. ACM, 2014. doi:10.1145/2582112.2582117.

3 Pritam Bhattacharya, Subir Kumar Ghosh, and Sudebkumar Pal. Constant approximation algorithms for guarding simple polygons using vertex guards. *ArXiv e-prints*, 2017. arXiv:1712.05492v2.

4 Michael F. Cohen and John R. Wallace. *Radiosity and realistic image syntheses*. Academic Press Professional, 1993.

5 Stephan Eidenbenz. Inapproximability of finding maximum hidden sets on polygons and terrains. *Computational Geometry*, 21(3):139–153, 2002. doi:10.1016/S0925-7721(01)00029-3.

6 Olivier Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, 1993.

7 Subir Kumar Ghosh, Anil Maheshwari, Sudebkumar Prasant Pal, Sanjeev Saluja, and C.E. Veni Madhavan. Characterizing and recognizing weak visibility polygons. *Computational Geometry*, 3(4):213–233, 1993. doi:10.1016/0925-7721(93)90010-4.

8 Subir Kumar Ghosh, Thomas Caton Shermer, Binay Kumar Bhattacharya, and Partha Pratim Goswami. Computing the maximum clique in the visibility graph of a simple polygon. *Journal of Discrete Algorithms*, 5(3):524–532, 2007. doi:10.1016/j.jda.2006.09.004.

9 Sumir Kubar Ghosh. *Visibility algorithms in the plane*. Cambridge University Press, 2007.

10 Jean-Claude Latombe. *Robot motion planning*. Kluwer Academic Publishers, 1991.

11 Joseph O'Rourke. *Art gallery theorems and algorithms*. The international series of monographs on Computer Science. Oxford University Press, 1987.

12 Valentin Polishchuk and Leonid Sedov. Gender-Aware Facility Location in Multi-Gender World. In Hiro Ito, Stefano Leonardi, Linda Pagli, and Giuseppe Prencipe, editors, *9th International Conference on Fun with Algorithms (FUN 2018)*, volume 100 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.FUN.2018.28`.

13 T. Shermer. Hiding people in polygons. *Computing*, 42(2):109–131, 1989. `doi:10.1007/BF02239742`.

14 Subhash Suri. *Minimum link paths in polygons and related problems*. PhD thesis, The Johns Hopkins University, Baltimore, Maryland, 1987.

15 Jorge Urrutia. *Handbook of computational geometry*, chapter Art Gallery and illumination problems, pages 973–1027. Elsevier, 2000.