On the Visibility Graphs of Pseudo-Polygons: Recognition and Reconstruction

Safwa Ameer
Department of Computer Science, The University of Texas at San Antonio, TX, USA
Matt Gibson-Lopez
Department of Computer Science, The University of Texas at San Antonio, TX, USA
Erik Krohn
Department of Computer Science, The University of Wisconsin – Oshkosh, WI, USA
Qing Wang
Department of Computer Science, University of Tennessee at Martin, TN, USA

Abstract

We give polynomial-time algorithms that solve the pseudo-polygon visibility graph recognition and reconstruction problems. Our algorithms are based on a new characterization of the visibility graphs of pseudo-polygons.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Pseudo-Polygons, Visibility Graph Recognition, Visibility Graph Reconstruction

Digital Object Identifier 10.4230/LIPIcs.SWAT.2022.7

Funding Supported by the National Science Foundation under Grant No. 1733874.

1 Introduction

Geometric covering problems have been a focus of research for decades. Here we are given a set of points P and a set S where each $s \in S$ can cover some subsets of P. The subset of P is generally induced by some geometric object. For example, P might be a set of points in the plane, and s consists of the points contained within some disk in the plane. For most variants, the problem is NP-hard and can easily be reduced to an instance of the combinatorial set cover problem which has a polynomial-time $O(\log n)$-approximation algorithm, which is the best possible approximation under standard complexity assumptions [5]. The main question therefore is to determine for which variants of geometric set cover we can obtain polynomial-time approximation algorithms with approximation ratio $o(\log n)$, as any such algorithm must exploit the geometry of the problem to achieve the result. This area has been studied extensively, see for example [2, 14, 1], and much progress has been made utilizing algorithms that are based on solving the standard linear programming relaxation.

Unfortunately this technique has severe limitations for some variants of geometric set cover, and new ideas are needed to make progress on these variants. In particular, the techniques are lacking when the points P we wish to cover is a simple polygon, and we wish to place the smallest number of points in P that collectively "see" the polygon. This problem is classically referred to as the *art gallery problem* as an art gallery can be modeled as a polygon and the points placed by an algorithm represent cameras that can "guard" the art gallery. This has been one of the most well-known problems in computational geometry for many years, yet still to this date the best polynomial-time approximation algorithms for this problem have approximation ratios that are $\omega(1)$. The key issue is a fundamental lack of understanding of the combinatorial structure of visibility inside simple polygons. It seems that in order to develop powerful approximation algorithms for this problem, the community first needs to better understand the underlying structure of such visibility.

18th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2022).
Editors: Artur Czumaj and Qin Xin; Article No. 7; pp. 7:1–7:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Visibility Graphs. A very closely related issue which has received a lot of attention in the community is the *visibility graph* (VG) of a simple polygon. Given a simple polygon P, the VG $G = (V, E)$ of P has the following structure. For each vertex $p \in P$, there is a vertex in V, and there is an edge connecting two vertices in G if and only if the corresponding vertices in P "see" each other (i.e., the line segment connecting the points does not go outside the polygon). The VG of a simple polygon must contain a Hamiltonian cycle that corresponds with the boundary of the P, and it generally is assumed that the input G comes with a labeled Hamiltonian cycle. Three major open problems regarding VGs of simple polygons are the VG characterization problem, the VG recognition problem, and the VG reconstruction problem. The *VG characterization* problem seeks to define a set of properties that all VGs satisfy. The *VG recognition* problem is the following. Given a graph G, determine if there exists a simple polygon P such that G is the VG of P in polynomial time. The *VG reconstruction* problem seeks to construct a simple polygon P such that a given VG G is the VG of P.

The problems of characterizing and recognizing the VGs of simple polygons have had partial results given dating back to over 25 years ago [6] and remain open to this day with only a few special cases being solved. Characterization and recognition results have been given in the special cases of "spiral polygons" [4] and "tower polygons" [3]. There have been several results [7, 4, 12] that collectively have led to four necessary conditions (NCs) that a simple polygon VG must satisfy. That is, if the graph G does not satisfy all four of the conditions then we know that G is not the VG for *any* simple polygon, and moreover it can be determined if a graph G satisfies all of the NCs in polynomial time. Streinu, however, has given an example of graph that satisfies all of the NCs but is not a VG for any simple polygon [13], implying that the set of conditions is not sufficient and therefore a strengthening of the NCs is needed. Unfortunately it is not even known if simple polygon VG recognition is in NP. See [8] for a nice survey on these problems and other related visibility problems.

Pseudo-polygons. Given the difficulty of understanding simple polygon VGs, O'Rourke and Streinu [10] considered the VGs for a special case of polygons called *pseudo-polygons* which we will now define. An arrangement of *pseudo-lines* $\mathcal{L}$ is a collection of simple curves, each of which separates the plane, such that each pair of pseudo-lines of $\mathcal{L}$ intersects at exactly one point, where they cross. Let $P = \{p_0, p_2, \ldots, p_{n-1}\}$ be a set of points in $\mathbb{R}^2$, and let $\mathcal{L}$ be an arrangement of $\binom{n}{2}$ pseudo-lines such that every pair of points p_i and p_j lie on exactly one pseudo-line in $\mathcal{L}$, and each pseudo-line in $\mathcal{L}$ contains exactly two points of P. The pair $(P, \mathcal{L})$ is called a *pseudo-configuration of points* (pcp) in general position.

Intuitively a pseudo-polygon is determined similarly to a standard Euclidean simple polygon except using pseudo-lines instead of straight line segments. Let $L_{i,j}$ denote the pseudo-line through the points p_i and p_j. We view $L_{i,j}$ as having three different components. The subsegment of $L_{i,j}$ connecting p_i and p_j is called the *segment*, and we denote it p_ip_j. Removing p_ip_j from $L_{i,j}$ leaves two disjoint *rays*. Let $r_{i,j}$ denote the ray starting from p_i and moving away from p_j, and we let $r_{j,i}$ denote the ray starting at p_j and moving away from p_i. Consider the pseudo-line $L_{i,i+1}$ in a pcp (indices taken modulo n and are increasing in counterclockwise order throughout the paper). We let e_i denote the segment of this line. A *pseudo-polygon* is obtained by taking the segments e_i for $i \in \{0, \ldots, n-1\}$ if (1) the intersection of e_i and e_{i+1} is only the point p_{i+1} for all i, and (2) for any i and j such that $j > i+1$, the segments e_i and e_j do not intersect. We call the segments e_i the *boundary edges*. A pseudo-polygon separates the plane into two regions: "inside" the pseudo-polygon and "outside" the pseudo-polygon, and any two points p_i and p_j see each other if the segment p_ip_j does not go outside of the pseudo-polygon. See Fig. 1 for an illustration. Pseudo-polygons

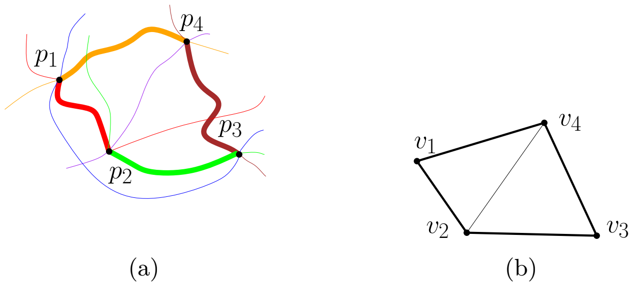


Figure 1 (a) A pcp and pseudo-polygon. (b) The corresponding VG.

can be viewed as a combinatorial abstraction of simple polygons. Note that every simple polygon is a pseudo-polygon (simply allow each $L_{i,j}$ to be the straight line through p_i and p_j), and Streinu showed that there are pseudo-polygons that cannot be "stretched" into a simple polygon [13].

O'Rourke and Streinu [10] give a characterization of *vertex-edge* VGs of pseudo-polygons. In this setting, for any vertex v we are told which *edges* v sees rather than which vertices it sees. Unfortunately, O'Rourke and Streinu showed that vertex-edge VGs encode more information about a pseudo-polygon than a regular VG [11]. Gibson, Krohn, and Wang [9] gave a characterization of the VGs of pseudo-polygons. Unfortunately this characterization did not directly lead to a polynomial-time recognition or reconstruction algorithm.

Our Results. In this paper, we give results for the VGs of both pseudo-polygons and simple polygons. First, we settle the remaining two open questions for the VGs of pseudo-polygons: recognition and reconstruction. First, we present a polynomial-time algorithm that can decide if a given graph G (with a labeled Hamiltonian cycle) is the VG for some pseudo-polygon, settling the recognition problem for pseudo-polygons. To obtain the result, we give a slightly different characterization of pseudo-polygon VGs than the one given in [9]. We then show that we can extend the recognition algorithm to obtain a polynomial-time reconstruction algorithm for pseudo-polygons. Our algorithm computes a vertex-edge VG that can then be reconstructed into a pseudo-polygon using the technique described in [10].

2 Preliminaries

We begin with some definitions that were relied upon heavily in the characterization of [9] that will be used in this paper as well. Note that the visibility graph G of a pseudo-polygon P must contain a Hamiltonian cycle because each p_i must see p_{i-1} and p_{i+1}. Since determining if a graph contains a Hamiltonian cycle is NP-hard, previous research has assumed that G does have such a cycle C and the vertices are labeled in counterclockwise order according to this cycle. So now suppose we are given an arbitrary graph $G = (V, E)$ with the vertices labeled p_0 to p_{n-1} such that G contains a Hamiltonian cycle $C = (p_0, p_2, \ldots, p_{n-1})$ in order according to their indices. We are interested in determining if G is the visibility graph for some pseudo-polygon P where C corresponds with the boundary of P. For any two vertices p_i and p_j, we let $\partial(p_i, p_j)$ denote the vertices and boundary edges encountered when walking counterclockwise around C from p_i to p_j (inclusive). For any edge $\{p_i, p_j\}$ in G, we say that $\{p_i, p_j\}$ is a *visible pair*, as their points in P must see one another. If $\{p_i, p_j\}$ is not an edge in G, then we call (p_i, p_j) and (p_j, p_i) *invisible pairs*. Note that visible pairs are unordered, and invisible pairs are ordered (for reasons described below).

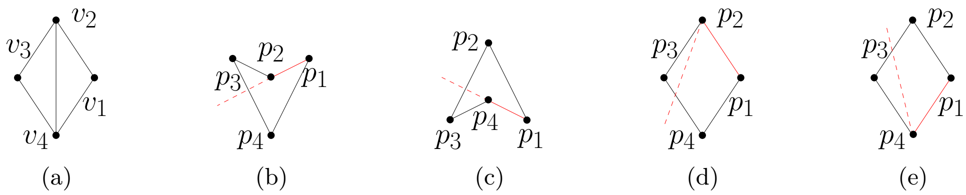


Figure 2 (a) A visibility graph G. (b) A simple polygon using p_2 to block p_1 and p_3. (c) A simple polygon using p_4 to block p_1 and p_3. (d) A pseudo-polygon using p_2 to block p_1 and p_3. Note that if the segment p_1p_3 does not exit the polygon then it would have to intersect $L_{1,2}$ at least twice (once at p_1 and once in the dashed ray $r_{2,1}$. (e) A pseudo-polygon using p_4 to block p_1 and p_3.

Consider any invisible pair (p_i, p_j). If G is the visibility graph for a pseudo-polygon P, the segment p_ip_j must exit P. For example, suppose we want to construct a polygon P such that the graph in Fig. 2 (a) is the visibility graph of P. Note that p_1 should not see p_3, and thus if there exists such a polygon, it must satisfy that p_1p_3 exits the polygon. In the case of a simple polygon, we view this process as placing the vertices of P in convex position and then contorting the boundary of P to block p_1 from seeing p_3. We can choose p_2 or p_4 to block p_1 from seeing p_3 (see (b) and (c)). Note that as in Fig. 2 (b) when using $p_2 \in \partial(p_1, p_3)$ as the blocker in a simple polygon, the line segment p_1p_2 does not go outside P and the ray $r_{2,1}$ first exits P through a boundary edge in $\partial(p_3, p_1)$. Similarly as in Fig. 2 (c) when using $p_4 \in \partial(p_3, p_1)$ as the blocker, the line segment p_1p_4 does not go outside of the polygon and the ray $r_{4,1}$ first exits the polygon through a boundary edge in $\partial(p_1, p_3)$. The situation is similar in the case of pseudo-polygons, but since we do not have to use straight lines to determine visibility, instead of bending the boundary of P to block the invisible pair we can instead bend the pseudo-line. See Fig. 2 (d) and (e). Note that the combinatorial structure of the pseudo-line shown in part (d) (resp. part (e)) is the same as the straight line in part (b) (resp. in part (c)). The following definition plays an important role in our characterization. Consider a pseudo-polygon P, and let p_i and p_j be two vertices of P that do not see each other. We say a vertex $p_k \in \partial(p_i, p_j)$ of P is a *designated blocker* for the invisible pair (p_i, p_j) if p_i sees p_k (i.e. the segment p_ip_k is inside the polygon) and the ray $r_{k,i}$ first exits the polygon through an edge in $\partial(p_j, p_i)$. The definition for $p_k \in \partial(p_j, p_i)$ to be a designated blocker for (p_i, p_j) is defined similarly. Intuitively, a designated blocker is a canonical vertex that prevents the points in an invisible pair from seeing each other.

The key structural lemma proved in [9] that led to their characterization was to show that every invisible pair (p_i, p_j) in a pseudo-polygon must have exactly one designated blocker. Moreover, the designated blocker must be one of at most two *candidate blockers*. There is at most one candidate blocker in $\partial(p_i, p_j)$ and there is at most one candidate blocker in $\partial(p_j, p_i)$. We will define the candidate blocker in $\partial(p_i, p_j)$, and the other case is handled symmetrically. Starting from p_j, walk clockwise towards p_i until we reach the first point p_k such that $\{p_i, p_k\}$ is a visible pair (clearly there must be such a point since $\{p_i, p_{i+1}\}$ is a visible pair). We say that p_k is a *candidate blocker* for (p_i, p_j) if there are no visible pairs $\{p_s, p_t\}$ such that $p_s \in \partial(p_i, p_{k-1})$ and $p_t \in \partial(p_{k+1}, p_j)$. If there is such a visible pair $\{p_s, p_t\}$, then there is no candidate blocker (and therefore no designated blocker) for (p_i, p_j) in $\partial(p_i, p_j)$. Note that a vertex may be a candidate blocker for (p_i, p_j) but not for (p_j, p_i), and we view invisible pairs as ordered pairs for this reason. The formal statement of the lemma proved in [9] is as follows.

▶ **Lemma 1.** *For any invisible pair (p_i, p_j) in a pseudo-polygon P, there is exactly one designated blocker p_k. Moreover, p_k is a candidate blocker for the invisible pair (p_i, p_j) in the visibility graph of P.*

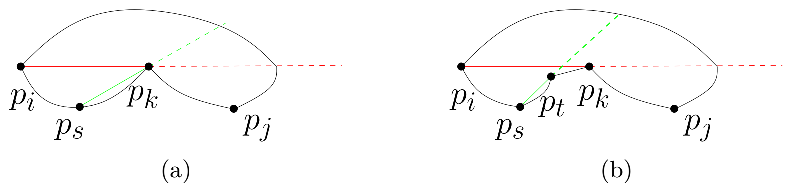


Figure 3 (a) If p_k is the designated blocker for (p_i, p_j) and p_s sees p_k then p_k is the designated blocker for (p_s, p_j). (b) If p_s does not see p_k, and p_t is the designated blocker for (p_s, p_k) then p_t is also the designated blocker for (p_s, p_j).

2.1 The Characterization of [9]

We now state the characterization of pseudo-polygon VGs given in [9]. The main idea is that if G is the VG of some pseudo-polygon P, then each invisible pair must be able to be assigned exactly one of its candidate blockers to serve as the designated blocker in P because of Lemma 1. However, we cannot simply arbitrarily pick a candidate blocker to serve as the designated blocker, as some choices may cause pseudo-lines to violate the pseudo-line properties. That is, some assignments may force a pair of pseudo-lines to intersect more than once and/or they may intersect at a vertex but not cross. The proof for each of the following assignment properties (APs) from the characterization in [9] showed that if the AP was violated then we would violate such a pseudo-line property. They later proved that if one can assign a designated blocker to each invisible pair that satisfies all of these properties, then G is in fact the VG for some pseudo-polygon (i.e., the properties are necessary and sufficient). We remark that the first four properties will also be used in the characterization in this paper, but the fifth property will be replaced with a new property to obtain a different characterization that will better fit within the framework of our reconstruction algorithm.

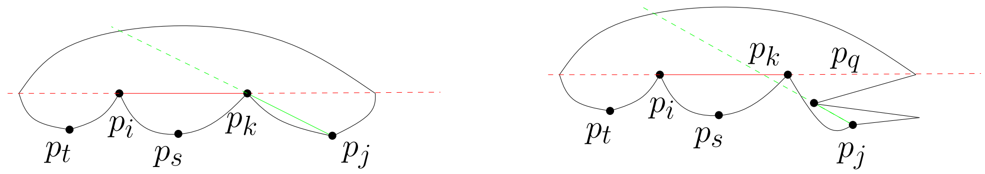
Let (p_i, p_j) be an invisible pair, and let p_k be the candidate blocker assigned to it. The first AP uses the definition of pseudo-lines and designated blockers to provide additional constraints on p_i and p_k. Note that while the condition is stated for $p_k \in \partial(p_i, p_j)$, a symmetric condition for when $p_k \in \partial(p_j, p_i)$ clearly holds.

▶ **Assignment Property 1.** *If $p_k \in \partial(p_i, p_j)$ is the candidate blocker assigned to invisible pair (p_i, p_j) then both of the following must be satisfied: (1) p_k is assigned to the invisible pair (p_i, p_t) for every $p_t \in \partial(p_{k+1}, p_j)$ and (2) if (p_k, p_j) is an invisible pair then p_i is not the candidate blocker assigned to it.*

Again let p_k be the candidate blocker assigned to an invisible pair (p_i, p_j) such that $p_k \in \partial(p_i, p_j)$. Since p_k is a candidate blocker, we have that (p_s, p_j) is an invisible pair for every $p_s \in \partial(p_i, p_{k-1})$. The next AP is a constraint on the location of designated blockers for (p_s, p_j). In particular, if $\{p_s, p_k\}$ is a visible pair, then p_k must be the designated blocker for (p_s, p_j). See Fig. 3 (a). If (p_s, p_k) is an invisible pair, then it must be assigned a designated blocker p_t. In this case, p_t must also be the designated blocker for (p_s, p_j). See Fig. 3 (b).

▶ **Assignment Property 2.** *Let (p_i, p_j) denote an invisible pair, and suppose p_k is the candidate blocker assigned to this invisible pair. Without loss of generality, suppose $p_k \in \partial(p_i, p_j)$, and let p_s be any vertex in $\partial(p_i, p_{k-1})$. Then exactly one of the following two cases holds: (1) $\{p_s, p_k\}$ is a visible pair, and the candidate blocker assigned to the invisible pair (p_s, p_j) is p_k, or (2) (p_s, p_k) is an invisible pair. If the candidate blocker assigned to (p_s, p_k) is p_t, then (p_s, p_j) is assigned the candidate blocker p_t.*

The next AP is somewhat similar to AP 2, except instead of introducing constraints on



(a) (b)

Figure 4 (a) If p_k is the designated blocker for (p_i, p_j) and p_j sees p_k then p_k is the designated blocker for $(p_j, p_s), (p_j, p_i)$, and (p_j, p_t). (b) If p_j does not see p_k, and p_q is the designated blocker for (p_j, p_k) then p_q is the designated blocker for $(p_j, p_s), (p_j, p_i)$, and (p_j, p_t). Moreover, (p_i, p_q) is an invisible pair and p_k is its designated blocker.

the designated blockers for (p_s, p_j), it introduces constraints on the designated blockers for (p_j, p_s) (where the order is reversed). Similar to the previous case, if p_j sees p_k then p_k must block p_j from seeing every $p_s \in \partial(p_i, p_{k-1})$, but we can also see that p_k must block p_j from any point p_t such that p_i is the designated blocker for (p_k, p_t). See Fig. 4 (a). If p_j does not see p_k, then there must be a designated blocker p_q for (p_j, p_k). See Fig. 4 (b). We show that in this case, p_q must be the designated blocker for all (p_j, p_s) and (p_j, p_t). Also, (p_i, p_q) must be an invisible pair with designated blocker p_k.

▶ **Assignment Property 3.** *Let (p_i, p_j) denote an invisible pair, and suppose p_k is the candidate blocker assigned to this invisible pair. Without loss of generality, suppose $p_k \in \partial(p_i, p_j)$. Then exactly one of the following two cases holds:*

1. *(a) $\{p_j, p_k\}$ is a visible pair. (b) For all $p_s \in \partial(p_i, p_{k-1})$, the candidate blocker assigned to the invisible pair (p_j, p_s) is p_k. (c) If p_t is such that p_i is the candidate blocker assigned to the invisible pair (p_k, p_t), then (p_j, p_t) is an invisible pair and is assigned the candidate blocker p_k.*
2. *(a) (p_j, p_k) is an invisible pair. Let p_q denote the candidate blocker assigned to (p_j, p_k). (b) (p_i, p_q) is an invisible pair, and p_k is the candidate blocker assigned to it. (c) For all $p_s \in \partial(p_i, p_k)$, the candidate blocker assigned to the invisible pair (p_j, p_s) is p_q. (d) If p_t is such that p_i is the candidate blocker assigned to the invisible pair (p_k, p_t), then (p_j, p_t) is an invisible pair and is assigned the candidate blocker p_q.*

Suppose p_k is a candidate blocker for an invisible pair (p_i, p_j) (or (p_j, p_i)), and suppose without loss of generality that $p_i \in \partial(p_j, p_k)$. If p_k is also a candidate blocker for an invisible pair (p_s, p_t) such that $p_s, p_t \in \partial(p_k, p_j)$ then we say that the two invisible pairs are a *separable invisible pair*. We have the following condition which is the same as Necessary Condition 3 for simple polygons in [8]. See Fig. 5 (a).

▶ **Assignment Property 4.** *Suppose (p_i, p_j) and (p_s, p_t) are a separable invisible pair with respect to a candidate blocker p_k. If p_k is assigned to (p_i, p_j) then it is not assigned to (p_s, p_t).*

We now give the final AP. Let p_j, p_i, p_t, and p_s be four vertices of G in “counter-clockwise order” around the Hamiltonian cycle C. We say that they are $\{p_i, p_t\}$-*pinched* if there is a $p_m \in \partial(p_i, p_t)$ such that p_i is the designated blocker for the invisible pair (p_j, p_m) and p_t is the designated blocker for the invisible pair (p_s, p_m). See Fig. 5 (b). The notion of $\{p_j, p_s\}$-pinched is defined symmetrically.

▶ **Assignment Property 5.** *Let p_i, p_j, p_s, and p_t be four vertices of G in counter-clockwise order around the Hamiltonian cycle C that are $\{p_i, p_t\}$-pinched. Then they are not $\{p_j, p_s\}$-pinched.*

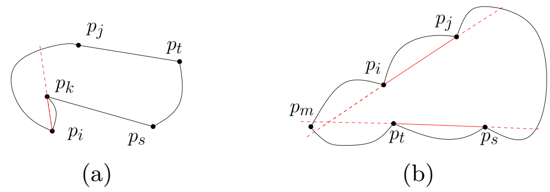


Figure 5 (a) If p_k blocks one invisible pair of a separable invisible pair then it cannot block the other one as well. (b) p_i, p_j, p_s, and p_t are $\{p_i, p_t\}$-pinched. If p_j blocks p_i from seeing some point, then p_s cannot also block p_t from seeing that point.

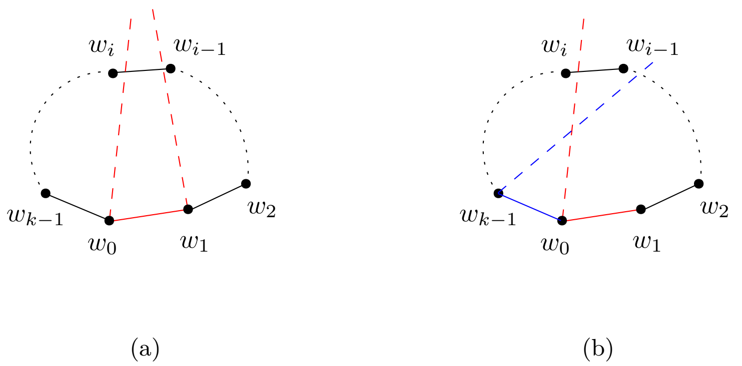


Figure 6 Illustrations for the proof of NC 1. (a) If w_0 is the designated blocker for (w_1, w_i) and w_1 is the designated blocker for (w_0, w_{i-1}) then $L_{0,1}$ will intersect the segment $w_{i-1}w_i$ twice. (b) If w_0 is the designated blocker for (w_1, w_i) and w_{k-1} is the designated blocker for (w_0, w_{i-1}) then $L_{0,1}$ intersects $L_{0,k-1}$ twice (once at w_0 and again in the dashed rays).

3 A New Characterization of Pseudo-Polygon VGs

In this section, we prove a different characterization of the VGs of pseudo-polygons. This characterization proves a property that must be satisfied by all VGs of pseudo-polygons. We then show that if G satisfies this property, then AP 5 is not needed. That is, if G satisfies this property and we can find an assignment of candidate blockers to invisible pairs that satisfies APs 1-4, then AP 5 must also be satisfied.

The property we prove is similar to the NC given by Ghosh [6] for simple polygon VGs; however, the proof uses geometric arguments that do not apply to general pseudo-polygons and therefore a new proof is needed. To state and prove the property, we first need some definitions. Suppose $w_0, w_1, \ldots, w_{k-1}$ form a cycle in G such that the vertices $w_0, w_1, \ldots, w_{k-1}$ follow the order in the Hamiltonian cycle C. Then, we say that $w_0, w_1, \ldots, w_{k-1}$ are an *ordered cycle*. Note that the Hamiltonian cycle C is an ordered cycle of all n vertices in G. An edge in G connecting two non-adjacent vertices of an ordered cycle is called a *chord*.

▶ **Necessary Condition 1.** *If G is the VG of a pseudo-polygon then any ordered cycle in G of length at least 4 must have at least one chord.*

Proof. Suppose G has an ordered cycle O of length $k \geq 4$, and let $w_0, w_1, \ldots, w_{k-1}$ denote the vertices around O in counterclockwise order, and for the sake of contradiction assume that O does not have any chords. This implies that any $w_i \in O$ sees no other vertices in O other than w_{i-1} and w_{i+1} (indices taken modulo k). Moreover, since $k \geq 4$, this implies that there is at least one vertex on O that w_i does not see. Let w_j be such a vertex. Note that any candidate blocker for (w_i, w_j) must be a vertex that is on O, because any vertex v not on O is in $\partial(w_a, w_{a+1})$ for some a and w_a sees w_{a+1} since they are consecutive points on O.

So now consider w_1. There is at least one vertex on O that w_1 does not see, and it must be that either w_0 and w_2 is the designated blocker for any such invisible pairs. Without loss of generality, assume that w_0 is a designated blocker for (w_1, w_j) for at least one $w_j \in O$. Walking counterclockwise around O starting at w_1, let w_i be the first point we encounter such that w_0 is the designated blocker for (w_1, w_i). Now note that $i - 1 \geq 2$, in particular i cannot be 2 because w_1 sees w_2. Therefore (w_0, w_{i-1}) is an invisible pair and either w_{k-1} or w_1 must be its designated blocker. But if w_1 is its designated blocker, then $L_{0,1}$ will intersect $w_{i-1}w_i$ twice (see Figure 6 (a)). If w_{k-1} is its designated blocker then $L_{k-1,0}$ will intersect $L_{0,1}$ twice (see Figure 6 (b)). ◀

Ghosh [7] showed it can be determined if G satisfies this property in $O(n^2)$ time. Now suppose that G does indeed satisfy this property. We will show then that it suffices to pick an assignment of candidate blockers to invisible pairs that satisfies only APs 1-4. In order to prove this, we prove two lemmas which will be used to prove the new characterization.

▶ **Lemma 2.** *Let p_i, p_j, p_k be three vertices of a pseudo-polygon VG in counterclockwise order. If p_j is a candidate blocker for invisible pair (p_i, p_k), then $(p_k, p_{k'})$ is an invisible pair for any $p_{k'} \in \partial(p_k, p_i)$ such that p_i is a candidate blocker for $(p_j, p_{k'})$.*

Proof. We will show that if p_k sees $p_{k'}$ then G violates NC 1. Suppose p_k sees $p_{k'}$. If p_k sees p_j and $p_{k'}$ sees p_i then $p_i, p_j, p_k, p_{k'}$ is an ordered cycle of length four with zero chords, a violation of NC 1. So now suppose that p_k does not see p_j. We will pick a "chain" C_1 of vertices $(c_1, c_2, \ldots c_m)$ in $\partial(p_j, p_k)$ such that: 1) the order of the vertices in C_1 is in clockwise order, and 2) a vertex c_i only sees vertices c_{i-1} and c_{i+1} (if they exist) in C_1. Initially we let $c_1 = p_k$. Now suppose c_i is the last vertex in C_1 we have found and we wish to find c_{i+1}. We start at p_j and walk counterclockwise until we find the first vertex that sees c_i (it may be p_j itself). If this vertex sees $p_{k'}$, then we throw out the current chain restart a new chain with this vertex as c_1, and otherwise we let this vertex be c_{i+1} in the current chain. We continue this process until p_j is added to C_1. Note that p_j cannot be c_1 since it does not see $p_{k'}$, and therefore the length of C_1 is at least 2.

We repeat a symmetric process to obtain a chain $C_2 = (c'_1, c'_2, \ldots, c'_{m'})$ in $\partial(p_{k'}, p_i)$, except we compute C_2 with respect to the vertices of C_1. That is, when we have computed c'_i and wish to compute c'_{i+1}, we consider if it sees any of the vertices in C_1. If it doesn't, then we add it to C_2 as vertex c'_{i+1}. If it does see some vertex of C_1, then we update *both* chains. We restart C_2 with this vertex as c'_1, and we will remove a "prefix" of C_1 depending on what c'_1 sees. Let z be the maximum integer such that c'_1 sees c_z of C_1. Then we remove all vertices from C_1 with index less than z. Note that c'_1 cannot see p_j or else p_i would not be a candidate blocker for $(p_j, p_{k'})$, and therefore C_1 still has length at least 2. We continue this until p_i gets added to C_2. Note that when p_i is added to C_2, C_1 is not reduced because if p_i saw any vertex in C_1 other than p_j then p_j would not be a candidate blocker for (p_i, p_k). Therefore C_1 and C_2 both have length at least 2.

The following is an ordered cycle of length at least four that does not have any chords: $c'_1, c'_2 \ldots, c'_{m'}, c_m, c_{m-1}, \ldots c_1$. Indeed we have that the only visible pairs that has one vertex in C_1 and the other in C_2 are $\{c_1, c'_1\}$ and $\{c'_{m'}, c_m\}$ by construction. Note $c'_{m'} = p_i$ and $c_m = p_j$. $\{p_i, p_j\}$ must be a visible pair because they are candidate blockers for each other. And finally there are no chords connecting two vertices of C_1 or two vertices of C_2 by construction of the chains. It follows that if p_k sees $p_{k'}$, then G violates NC 1. ◀

▶ **Lemma 3.** *Let p_i, p_j, p_s, p_t be four vertices of a pseudo-polygon VG G in counterclockwise order such that $\{p_i, p_j\}, \{p_i, p_s\}$, $\{p_j, p_t\}$, and $\{p_s, p_t\}$ are visible pairs. Suppose we have an assignment of candidate blockers to invisible pairs that satisfies AP 1-4. Let B be the set of all vertices in $\partial(p_j, p_s)$ such that for each $p_z \in B$, p_j is the designated blocker for (p_i, p_z) and p_s is the designated blocker for (p_t, p_z). If $|B| \geq 1$, then there is at least one vertex $p_k \in B$ such that satisfies one property of set 1 and one property of set 2:*

1. **a.** *$\{p_k, p_j\}$ is a visible pair, or*
   b. *(p_k, p_j) is an invisible pair blocked by a designated blocker in $\partial(p_j, p_k)$*
2. **a.** *$\{p_k, p_s\}$ is a visible pair, or*
   b. *(p_k, p_s) is an invisible pair blocked by a designated blocker in $\partial(p_k, p_s)$.*

Proof. We first show a fact about any point p_z in B that does not satisfy the conditions of the lemma. Suppose for p_z at least one of the two sets of properties has both properties not satisfied. Without loss of generality, assume it is the first set. Then (p_z, p_j) is an invisible pair blocked by a candidate blocker $p_b \in \partial(p_z, p_j)$. We will show that p_b must be in B. To prove this, we first show that p_b must be in $\partial(p_{z+1}, p_{s-1})$. p_z cannot see any point in $\partial(p_i, p_{j-1})$ or else p_j would not be a candidate blocker for (p_i, p_z). If p_b were in $\partial(p_s, p_i)$, AP 3 case 2 (b) implies that p_j must the designated blocker for (p_i, p_b). But if this is true then AP 1 case 1 implies that p_j must be the designated blocker for (p_i, p_s), but $\{p_i, p_s\}$ is a visible pair. Therefore it must be $p_b \in \partial(p_{z+1}, p_{s-1})$. We now show that $p_b \in B$. That is, it must be that (p_b, p_i) and (p_b, p_t) are both invisible pairs. If $\{p_b, p_t\}$ were a visible pair then p_s would not be a candidate blocker for (p_t, p_z). Also (p_i, p_b) must be an invisible pair with designated blocker p_j by AP 3 case 2 (b). Therefore p_b is indeed in B.

Now we show how to find a point in B that satisfies the conditions of the lemma. Walk counterclockwise starting from p_j until we find the first vertex that is in B, and let us call this vertex p_{x_1}. If this vertex does not satisfy a property from each set, then it must be that there is a blocker p_{x_2} that blocks p_{x_1} from p_j (it must also block it from p_s by AP 1). From the above analysis $p_{x_2} \in B$, and therefore p_{x_2} must be in $\partial(p_{x_1+1}, p_{s-1})$ since p_{x_1} is the first point encountered in B. Likewise, if p_{x_2} does not satisfy a property from each set, then there must be a blocker $p_{x_3} \in B$ that blocks p_{x_2} from both p_s and p_j. Again we can see that $p_{x_3} \in \partial(p_{x_2+1}, p_{s-1})$. Indeed, it cannot be in $\partial(p_j, p_{x_1-1})$ since none of these points are in B, and it cannot be in $\partial(p_{x_1}, p_{x_2-1})$ as that would contradict that p_{x_2} is a candidate blocker for (p_{x_1}, p_s). Inductively we repeat this until we find a point that satisfies the condition. This process must terminate since if p_{x_i} does not satisfy a property from each set, it must be blocked by a point in $\partial(p_{x_i+1}, p_{s-1})$. Eventually we will run out of points in B and there will be no more points to be the blocker, and therefore we will find the desired point. ◀

We are now ready to prove the our new characterization that removes the need to satisfy AP 5 if G satisfies NC 1.

▶ **Theorem 4.** *A graph G with a labeled Hamiltonian cycle is the visibility graph of some pseudo-polygon if and only if it satisfies NC 1 and there is an assignment of candidate blockers to the invisible pairs of G that satisfies APs 1-4.*

Proof. Assuming that G satisfies NC 1, we will prove that if an assignment violates AP 5, then it also violates one of the other APs. Assume AP 5 is violated. That is, let p_i, p_j, p_s, p_t be four vertices of G in counterclockwise order around C such that there is a $p_k \in \partial(p_j, p_s)$ and a $p_{k'} \in \partial(p_t, p_i)$ satisfying: 1) (p_i, p_k) is an invisible pair that has been assigned p_j, 2)

(p_t, p_k) is an invisible pair that has been assigned p_s, 3) $(p_j, p_{k'})$ is an invisible pair that has been assigned p_i, and 4) $(p_s, p_{k'})$ is an invisible pair that has been assigned p_t. Here, we pick p_k and $p_{k'}$ to be points that satisfy the conditions of Lemma 3.

Now note that $(p_k, p_{k'})$ must be an invisible pair by Lemma 2, and therefore there must be a candidate blocker assigned to this invisible pair. Since p_j is the candidate blocker assigned to (p_i, p_k) and p_i is the candidate blocker assigned to $(p_j, p_{k'})$, AP 3 implies that the candidate blocker assigned to $(p_k, p_{k'})$ must be in $\partial(p_{k'}, p_k)$ (i.e., the blocker is p_j if p_k sees p_j or whatever point is blocking p_k from p_j which must be in $\partial(p_j, p_k)$ since p_k satisfies Lemma 3). However, we can apply the same argument symmetrically to p_s and p_t to see that $(p_k, p_{k'})$ must be assigned a blocker in $\partial(p_k, p_{k'})$. Therefore no matter which candidate blocker we assign to $(p_k, p_{k'})$, it must be that AP 3 is violated. ◀

4 Recognition and Reconstruction Algorithms

In this section, we give a polynomial-time algorithm to determine whether or not a given graph G with a labeled Hamiltonian cycle C is the VG for some pseudo-polygon. We then extend the algorithm to obtain a polynomial-time algorithm that can reconstruct a pseudo-polygon P such that G is the VG of P if the recognition algorithm returns YES.

Algorithm 1 Recognition Algorithm.

if G does not satisfy NC 1 **then**
  Return NO
for all invisible pairs (p_i, p_j) **do**
  Compute their (at most 2) candidate blockers.
  Add (p_i, p_j) to the set of all invisible pairs I.
while there is an invisible pair (IP) $(p_i, p_j) \in I$ such that (p_i, p_j) has < 2 remaining feasible candidate blockers **do**
  if some IP has 0 remaining candidate blockers **then**
    Return NO
  else
    if some IP has exactly 1 feasible candidate blocker **then**
      Assign this candidate blocker to the IP.
      Remove the IP from I.
      For every other IP in I, remove any remaining candidate blocker if its selection would violate APs 1-4.
Return YES

The recognition algorithm is stated formally in Algorithm 1. The algorithm itself is fairly simple. We first check that G satisfies NC 1, and if it does not then we return NO. If it satisfies this property, then for each invisible pair, we compute the set of at most two candidate blockers for this invisible pair. If some invisible pair has 0 candidate blockers, we return NO. If some invisible pair has only 1 candidate blocker, then it must be the designated blocker so we assign it to the candidate blocker. We then remove any candidate blocker for any other invisible pair if that candidate blocker would violate one of the APs. We then repeat this until every invisible pair has been assigned a candidate blocker, or until every remaining invisible pair has two candidate blockers remaining. In either case, we return YES.

The algorithm runs in polynomial time. Checking NC 1 can be done in $O(n^2)$ time [7]. There are at most $O(n^2)$ invisible pairs of G. Checking a violation of an AP involves only 2 invisible pairs, and this check can be done in constant time. The algorithm is clearly correct if we return NO or if we return YES because every invisible pair was assigned a candidate blocker. We must prove that the algorithm is correct when we return YES because every remaining invisible pair has two candidate blockers. We prove this constructively by assigning a candidate blocker to each remaining invisible pair in I. The algorithm for this is formally stated in Algorithm 2.

Algorithm 2 Candidate Blocker Assignment.

1: Let I denote the invisible pairs of G that are not assigned a candidate blocker by Algorithm 1.
2: Let p_x be any arbitrarily chosen vertex of G.
3: **for all** invisible pairs $(p_i, p_j) \in I$ **do**
4: Walk counterclockwise around the Hamiltonian cycle C starting at p_x. Let p_a denote which vertex of (p_i, p_j) is encountered first and let p_b denote the other vertex.
5: Assign to (p_i, p_j) the candidate blocker in $\partial(p_a, p_b)$.

We now prove that the combination of candidate blocker assignments in Algorithm 1 and in Algorithm 2 satisfies APs 1-4, thereby proving the correctness of Algorithm 1.

▶ **Lemma 5.** *The combination of candidate blocker assignments in Algorithm 1 and in Algorithm 2 assigns a valid candidate blocker to every invisible pair of G.*

Proof. Each of the APs regards the feasibility of a pair of assigned candidate blockers. Since we removed any candidate blocker that would have created a violation with a choice made in Algorithm 1, we only need to consider candidate blockers assigned in Algorithm 2. We go through each AP and show that any assignment we make will not violate the AP, which completes the proof of the lemma.

AP 1. Case 1 states that if $p_k \in \partial(p_i, p_j)$ is the candidate blocker assigned to invisible pair, (p_i, p_j) then it must also be assigned (p_i, p_t) for every $p_t \in \partial(p_{k+1}, p_j)$. Since $p_k \in \partial(p_i, p_j)$, it must be that $p_i = p_a$ and $p_j = p_b$ in Algorithm 2 implying that $p_x \in \partial(p_{j+1}, p_i)$. Then it also must mean that $p_i = p_a$ and $p_t = p_b$ when considering (p_i, p_t), and therefore p_k will be assigned to (p_i, p_t). Case 2 states that if (p_k, p_j) is an invisible pair then p_i is not assigned to it, but since $p_x \in \partial(p_{j+1}, p_i)$ it must be that $p_k = p_a$ and $p_j = p_b$ when considering (p_k, p_j) and therefore we will assign to (p_k, p_j) the candidate blocker in $\partial(p_k, p_j)$ which is not p_i.

AP 2. This AP applies when $p_k \in \partial(p_i, p_j)$ is assigned to invisible pair (p_i, p_j) and then considers the invisible pairs (p_s, p_j) for each $p_s \in \partial(p_i, p_{k-1})$. If we choose $p_k \in \partial(p_i, p_j)$, then it must be that $p_x \in \partial(p_{j+1}, p_i)$, and therefore we will assign to (p_s, p_j) the candidate blocker in $\partial(p_s, p_j)$. Case 1 of AP 2 states that if $\{p_s, p_k\}$ is a visible pair, then p_k must be assigned to (p_s, p_j), and indeed this is what we do if $\{p_s, p_k\}$ is a visible pair because no point in $\partial(p_s, p_{k-1})$ can see any point in $\partial(p_{k+1}, p_j)$ or else p_k would not be a candidate blocker for (p_i, p_j). Case 2 says that if (p_s, p_k) is an invisible pair, then the candidate blocker assigned to (p_s, p_j) must be the same as the candidate blocker assigned to (p_s, p_k). But given the location of p_x, it must be that for both invisible pairs we have $p_s = p_a$ in Algorithm 2, and therefore we will assign the same candidate blocker to both invisible pairs.

AP 3. This AP applies when $p_k \in \partial(p_i, p_j)$ is assigned to invisible pair (p_i, p_j) and then considers the invisible pairs (p_j, p_i) as well as (p_j, p_t) for any p_t such that p_i is the candidate blocker assigned to (p_k, p_t). First we consider (p_j, p_i). If $p_k \in \partial(p_i, p_j)$ is assigned to invisible pair (p_i, p_j), then it must be that $p_x \in \partial(p_{j+1}, p_i)$. This implies

that when considering (p_j, p_i), we will assign the candidate blocker in $\partial(p_i, p_j)$ to (p_j, p_i). Case 1 states that if $\{p_j, p_k\}$ is a visible pair that this blocker must be p_k, and in fact this the candidate blocker in $\partial(p_i, p_j)$ because no point in $\partial(p_i, p_{k-1})$ can see any point in $\partial(p_{k+1}, p_j)$ or else p_k would not be a candidate blocker for (p_i, p_j). Case 2 states that if (p_j, p_k) is an invisible pair, then whatever blocker is assigned to (p_j, p_k) must be assigned to (p_j, p_i), but here we have $p_j = p_b$ in both scenarios, and therefore we will assign the same blocker to both (p_j, p_k) and (p_j, p_i).
Now consider (p_j, p_t) for any p_t such that p_i is the candidate blocker assigned to (p_k, p_t). The analysis is very similar to the previous case. If $p_k \in \partial(p_i, p_j)$ is assigned to invisible pair (p_i, p_j) and $p_i \in \partial(p_t, p_k)$ is assigned to invisible pair (p_k, p_t), then it must be that $p_x \in \partial(p_{j+1}, p_t)$. This implies that $p_t = p_a$ and $p_j = p_b$ when considering (p_j, p_t). For the same reasons as in Case 1 for AP 3, we will assign the correct candidate blocker to (p_j, p_t).

AP 4. Here we have (p_i, p_j) and (p_s, p_t) which are a separable invisible pair with respect to candidate blocker p_k. If p_k is assigned to (p_i, p_j), then that implies that $p_j = p_a$ and $p_i = p_b$ which means that $p_x \in \partial(p_{i+1}, p_j)$. This means that when we consider (p_s, p_t) we will have $p_s = p_a$ and $p_t = p_b$, and therefore we will assign it the candidate blocker in $\partial(p_s, p_t)$ which is not p_k. ◀

This gives us the following theorem.

▶ **Theorem 6.** *There is a polynomial-time algorithm that can determine whether a given graph G with a labeled Hamiltonian cycle C is the visibility graph for some pseudo-polygon P such that C corresponds with the boundary of P.*

We can then combine our Algorithms 1 and 2 into a reconstruction algorithm by building a vertex-edge VG based on our computed assignment (details in [9]) and then reconstruct the pseudo-polygon from this graph using the technique described in [10].

▶ **Theorem 7.** *There is a polynomial-time algorithm that can construct a pseudo-polygon P such that a given visibility graph G of a pseudo-polygon with a labeled Hamiltonian cycle C is the visibility graph of P where C corresponds to the boundary of P.*

References

1 Greg Aloupis, Jean Cardinal, Sébastien Collette, Stefan Langerman, David Orden, and Pedro Ramos. Decomposition of multiple coverings into more parts. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 302–310, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=1496770.1496804.

2 Boris Aronov, Esther Ezra, and Micha Sharir. Small-size epsilon-nets for axis-parallel rectangles and boxes. *SIAM J. Comput.*, 39(7):3248–3282, July 2010. doi:10.1137/090762968.

3 Seung-Hak Choi, Sung Yong Shin, and Kyung-Yong Chwa. Characterizing and recognizing the visibility graph of a funnel-shaped polygon. *Algorithmica*, 14(1):27–51, 1995. doi:10.1007/BF01300372.

4 Hazel Everett and Derek G. Corneil. Negative results on characterizing visibility graphs. *Comput. Geom.*, 5:51–63, 1995. doi:10.1016/0925-7721(95)00021-Z.

5 Uriel Feige, Magnús M. Halldórsson, Guy Kortsarz, and Aravind Srinivasan. Approximating the domatic number. *SIAM J. Comput.*, 32(1):172–195, January 2003. doi:10.1137/S0097539700380754.

6 Subir Kumar Ghosh. On recognizing and characterizing visibility graphs of simple polygons. In *SWAT*, pages 96–104, 1988.

7 Subir Kumar Ghosh. On recognizing and characterizing visibility graphs of simple polygons. *Discrete & Computational Geometry*, 17(2):143–162, 1997. `doi:10.1007/BF02770871`.

8 Subir Kumar Ghosh and Partha P. Goswami. Unsolved problems in visibility graphs of points, segments, and polygons. *ACM Comput. Surv.*, 46(2):22, 2013. `doi:10.1145/2543581.2543589`.

9 Matt Gibson, Erik Krohn, and Qing Wang. A characterization of visibility graphs for pseudo-polygons. In *ESA*, pages 607–618, 2015.

10 Joseph O'Rourke and Ileana Streinu. Vertex-edge pseudo-visibility graphs: Characterization and recognition. In *Symposium on Computational Geometry*, pages 119–128, 1997. `doi:10.1145/262839.262915`.

11 Joseph O'Rourke and Ileana Streinu. The vertex-edge visibility graph of a polygon. *Computational Geometry*, 10(2):105–120, 1998. `doi:10.1016/S0925-7721(97)00011-4`.

12 G. Srinivasaraghavan and Asish Mukhopadhyay. A new necessary condition for the vertex visibility graphs of simple polygons. *Discrete & Computational Geometry*, 12:65–82, 1994. `doi:10.1007/BF02574366`.

13 Ileana Streinu. Non-stretchable pseudo-visibility graphs. *Comput. Geom.*, 31(3):195–206, 2005. `doi:10.1016/j.comgeo.2004.12.003`.

14 Kasturi R. Varadarajan. Epsilon nets and union complexity. In *Symposium on Computational Geometry*, pages 11–16, 2009. `doi:10.1145/1542362.1542366`.