

Finite Guarding of Weakly Visible Segments via Line Aspect Ratio in Simple Polygons

Arash Vaezi^{1,*}

¹Department of Computer Science, Institute for Research in Fundamental Sciences (IPM)

Abstract

We address the problem of covering a target segment \overline{uv} using a finite set of guards \mathcal{S} placed on a source segment \overline{xy} within a simple polygon \mathcal{P} , assuming weak visibility between the target and source. Without geometric constraints, \mathcal{S} may be infinite, as shown by prior hardness results. To overcome this, we introduce the *line aspect ratio* (AR), defined as the ratio of the *long width* (LW) to the *short width* (SW) of \mathcal{P} . These widths are determined by parallel lines tangent to convex vertices outside \mathcal{P} (LW) and reflex vertices inside \mathcal{P} (SW), respectively.

Under the assumption that AR is constant or polynomial in n (the polygon's complexity), we prove that a finite guard set \mathcal{S} always exists, with size bounded by $\mathcal{O}(\text{AR})$. This AR-based framework generalizes some previous assumptions, encompassing a broader class of polygons.

Our result establishes a framework guaranteeing finite solutions for segment guarding under practical and intuitive geometric constraints.

Keywords

Simple polygon, Segment's visibility, Line aspect ratio, Weakly visible.

1. Introduction

Let \mathcal{P} be a simple polygon, and let $\text{int}(\mathcal{P})$ denote its interior. Two points x and y in \mathcal{P} are *visible* to each other if and only if the relatively open line segment \overline{xy} lies entirely within $\text{int}(\mathcal{P})$. The *visibility polygon* of a point $q \in \mathcal{P}$, denoted by $\text{VP}(q)$, is the set of all points in \mathcal{P} visible to q .

The *weak visibility polygon* of a line segment \overline{pq} , denoted $\text{WVP}(\overline{pq})$, is the maximal sub-polygon of \mathcal{P} visible to at least one interior point of \overline{pq} . A polygon \mathcal{P} is *completely visible* from a segment \overline{pq} (denoted CVP) if every point $z \in \mathcal{P}$ is visible from every point $w \in \overline{pq}$. Algorithms to compute WVP and CVP in linear time are known [1, 2].

1.1. Literature Review

Avis et. al. [1] developed a linear-time method for computing weak visibility polygons, enabling the characterization of regions visible from a given segment. Similarly, Guibas et. al. [3] introduced trapezoidal decomposition methods to support efficient segment-to-segment visibility queries. These techniques are crucial for testing the visibility conditions in different scenarios. Complementing these, Hershberger et. al. [4] proposed an optimal algorithm for visibility graph construction, effectively determining the visibility relationships between points on two segments.

The combinatorial conditions for mutual visibility between segments have been extensively studied. Sack et. al. [5] established necessary and sufficient conditions for weak mutual visibility, providing a duality-based framework that explains cases where partial visibility permits finite covering sets. Later, Ghodsi et. al. [6] developed decision algorithms for testing weak visibility between disjoint segments, showing that determining whether $\overline{uv} \subseteq \text{WVP}(\overline{xy})$ is solvable in $\mathcal{O}(n \log n)$ time.

Guard placement optimization has been explored in related contexts. Tóth [7] demonstrated that $\lfloor n/4 \rfloor$ edge guards suffice to cover simple polygons. King et. al. [8] investigated mobile guards along

ICTCS 2025: Italian Conference on Theoretical Computer Science, September 10–12, 2025, Pescara, Italy

*Corresponding author.

✉ avaezi@sharif.edu (A. Vaezi)

ORCID 0000-0003-4798-0029 (A. Vaezi)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

segments. Additionally, Bhattacharya et. al. [9] provided an $\mathcal{O}(\log n)$ -approximation for guarding weakly visible polygons.

The study of visibility within simple polygons has advanced significantly, particularly with specialized cases like sliding cameras, reflections, and structured segment visibility. Biedl et al. explored sliding-camera guards, providing approximation algorithms and demonstrating the NP-hardness of sliding-camera coverage in polygons with holes [10]. Vaezi et. al. addressed reflection-extended visibility, where polygon edges act as mirrors, enabling previously invisible segments to become visible; their work covers weak, strong, and complete visibility settings [11, 12, 13]. Lee and Chwa [14] focused on chain visibility, investigating the visibility of polygonal chains and providing efficient algorithms for both convex and reflex chains. Recent research has introduced k-transmitters, extending visibility to cases where light rays may cross polygon boundaries multiple times [15]. Furthermore, structured visibility profiles and efficient data structures for segment-to-segment queries have been studied extensively, offering solutions for visibility tracking and analysis in dynamic environments [16].

The inherent difficulty of unrestricted guarding has motivated the introduction of geometric constraints. Bonnet et. al. [17] proved the APX-hardness of guarding problems, highlighting the necessity of assumptions like integer coordinates for the given simple polygon. Notably, their results demonstrate that without such constraints, the set of guards \mathcal{S} may be infinite—a key motivation for our approach. Unlike previous work relying on integer-coordinate assumptions, our framework accommodates real-coordinate polygons with exponential complexity, generalizing these results.

1.2. Positioning of Our Contribution

Existing research has laid a strong foundation in:

- Efficient computation of weak visibility polygons [1]
- Visibility testing between segments [6]
- Hardness results for general guarding problems [17]

Our work extends this body of knowledge by providing:

- A guarantee of finite guard sets \mathcal{S} for segment coverage under the line aspect ratio assumption
- Explicit bounds on guard set size, $|\mathcal{S}| = \mathcal{O}(\text{AR})$ (Theorem 1)

2. Problem Definition

Consider two line segments: a *target* segment \overline{uv} and a *source* segment \overline{xy} . The visibility between these segments may fall into one of three cases:

1. \overline{uv} and \overline{xy} are completely visible (CVP).
2. At least one point of \overline{xy} or \overline{uv} is invisible to the other segment.
3. \overline{uv} and \overline{xy} are partially visible (i.e., every point on one segment is visible to at least one point on the other, but not necessarily all points). From now on we refer to this case as the target is *weakly visible* from the source.

This work focuses on the third case. We aim to find a finite and polynomial set \mathcal{S} of points on the source segment \overline{xy} such that:

$$\overline{uv} \subseteq \bigcup_{s \in \mathcal{S}} \text{VP}(s)$$

where $\text{VP}(s)$ denotes the visibility polygon of s . We assume \overline{uv} is weakly visible from \overline{xy} .

3. Assumptions

To ensure a finite size for \mathcal{S} , certain assumptions about \mathcal{P} are necessary. This section introduces these assumptions. Specifically, we present an algorithm for determining the points of \mathcal{S} and demonstrate that, under Assumption 1, the algorithm yields a finite set \mathcal{S} whose size is polynomial in n .

Assumption 1 (Line Aspect Ratio (Our assumption)). *For a simple polygon \mathcal{P} , the long width (LW) is the maximum distance between two parallel lines tangent to convex vertices of \mathcal{P} , on the outside of \mathcal{P} without intersecting its interior. The short width (SW) is the minimum distance between two such parallel line segments tangent to the reflex vertices in the interior of \mathcal{P} and constrained by the polygon's boundary. The line aspect ratio is:*

$$AR_{line} = \frac{LW}{SW}$$

One may consider two cases:

- Constant line aspect ratio: $AR_{line} = O(1)$
- Polynomial line aspect ratio: $AR_{line} = \text{poly}(n)$

where n is the complexity of \mathcal{P} .

Assumption 2 (Disk Aspect Ratio). *For a simple polygon \mathcal{P} , the long diameter (LD) is the diameter of the smallest enclosing circle tangent to the boundary. The short diameter (SD) is the diameter of the largest inscribed circle tangent to the boundary. The disk aspect ratio is:*

$$AR_{disk} = \frac{LD}{SD}$$

One may consider:

- Constant disk aspect ratio: $AR_{disk} = O(1)$
- Polynomial disk aspect ratio: $AR_{disk} = \text{poly}(n)$

For consistency, we use $AR = AR_{line}$ to denote the line aspect ratio in subsequent discussions.

4. Our Contribution

Theorem 1. *Under the line aspect ratio assumption (Definition 1), there exists a finite set \mathcal{S} on \overline{xy} such that:*

$|\mathcal{S}|$ is bounded by AR

$$\overline{uv} \subseteq \bigcup_{s \in \mathcal{S}} VP(s)$$

Proof. The size of \mathcal{S} is determined by $AR = LW/SW$:

1. The slicing approach (Section 4.1) decomposes \overline{uv} into visibility intervals.
2. By Lemma 1, the points in \mathcal{S} cover the target.
3. Observation 3 establishes that each interval has length $\geq SW$.
4. Since \overline{uv} has maximum length LW , the number of intervals is $\leq \frac{LW}{SW} = AR$.

Thus $|\mathcal{S}|$ is finite and bounded by AR . □

In the continue we present the slicing algorithm in Subsection 4.1, and Subsection 4.2 covers the lemmas and observations and their proofs. Section 5 provides a final discussion.

4.1. Slicing Algorithm

This subsection covers the slicing algorithm that splits a given source segment (\overline{xy}) by some middle points so that the union visibility of the set of all these points including the endpoints of the source segment covers an entire given target segment (\overline{uv}). Without loss of generality, we already suppose that the given source and target segments are weakly visible.

We start the slicing algorithm by defining two specific reflex vertices and their computing approach.

Since the target is weakly visible from the source, consider the visibility of those points on the source whose view of the target is obstructed by some reflex vertices of \mathcal{P} . For each point on the source, its visibility can be blocked by at most two reflex vertices. However, these two reflex vertices may differ for different points on the source. For a precise definition of these reflex vertices, refer to Definition 1.

Definition 1. Consider two reflex vertices: LBV , denoting the Left Blocking Vertex, and RBV , representing the Right Blocking Vertex. These reflex vertices are defined with respect to a specific point on the source. For a point q on the source, imagine standing at q , positioned between x and y , while observing \overline{uv} . Assume that x lies to the left and y lies to the right of q . There exists a **single** reflex vertex on each side of q such that LBV_q and RBV_q are uniquely determined by q (see Observation 2).

LBV_q (if it exists) is the reflex vertex where the line segment $\overline{qLBV_q}$ intersects with \overline{uv} and lies entirely inside \mathcal{P} , passes through at least one reflex vertex (LBV_q), and has the exterior of \mathcal{P} on the left side of $\overline{qLBV_q}$. If multiple reflex vertices lie on a single line crossing $\overline{qLBV_q}$, the closest reflex vertex to q along that line defines LBV_q .

The same strategy defines RBV_q , except that the exterior of \mathcal{P} lies on the right side of $\overline{qRBV_q}$.

4.1.1. Computing LBV_q and RBV_q for a point q on \overline{xy}

We already know that \overline{xy} and \overline{uv} are weakly visible. Consider the line \overline{qu} and run a sweeping algorithm on the reflex vertices of \mathcal{P} to obtain a line that meets the requirements of Definition 1. That is a line that lies on at least one reflex vertex passing \overline{uv} and holds other reflex vertices of \mathcal{P} on its left side. Note that if multiple reflex vertices lie on this line, the closest reflex vertex to q along that line defines LBV_q .

Using the same sweeping algorithm on the other side with an opposite direction obtains RBV_q .

4.1.2. Computing points on \overline{xy}

Denoting x as x_0 and y as y_0 , we will perform the iterations described below to compute a sequence of points x_i and y_i on \overline{xy} . This process continues until an iteration $j \geq 1$ is reached where x_j lies to the right of y_j . We will demonstrate that, assuming \mathcal{P} has a bounded line aspect ratio, the number of iterations has a polynomial upper bound (Lemma 2). Furthermore, when x_j lies to the right of y_j , the target will be covered by the set of points $\{x_i, y_i \mid 0 \leq i \leq j\}$ (denoted as \mathcal{S}) (see Lemma 1).

Iterations of the algorithm after computing LBV and RBV vertices

Consider two lines: the line intersecting x_i and LBV_{x_i} , and the line intersection y_i and RBV_{y_i} , $i \geq 0$.

The line crossing $\overline{x_iLBV_{x_i}}$ intersect the target on t_{x_i} . The line crossing $\overline{y_iRBV_{y_i}}$ intersects the target on t_{y_i} .

In each iteration $0 \leq i < j$, compute t_{x_i} and t_{y_i} .

Consider t_{x_i}/t_{y_i} as a middle point on \overline{uv} where v places at the left side of t_{x_i} . Assuming \overline{uv} as the source compute LBV and RBV vertices for t_{x_i} and t_{y_i} points.

Draw the line crossing t_{x_i} and $LBV_{t_{x_i}}$. The intersection of this line with \overline{xy} creates a point denoted as x_{i+1} .

Draw the line crossing t_{y_i} and $RBV_{t_{y_i}}$. The intersection of this line with \overline{xy} creates a point denoted as y_{i+1} .

If x_{i+1} lies to the left of y_{i+1} , set $i = i + 1$ and repeat the iteration procedure. Otherwise (that x_{i+1} lies to the right of (or if they lie on one point) y_{i+1}), we reach the j th iteration and the slicing algorithm stops since the target is covered (Lemma 3 reveals that in the j^{th} iteration the target gets covered successfully).

In case one of the points LBV , RBV does not exist, the corresponding lines do not exist as well. If it is an x or y points the point in that iteration can see the rest of the target. If it is a t point it can see rest of the source so the next point on the next point on the source is x or y itself. So, the algorithm has already reached a position where the points can see the entire target and the slicing algorithm terminates.

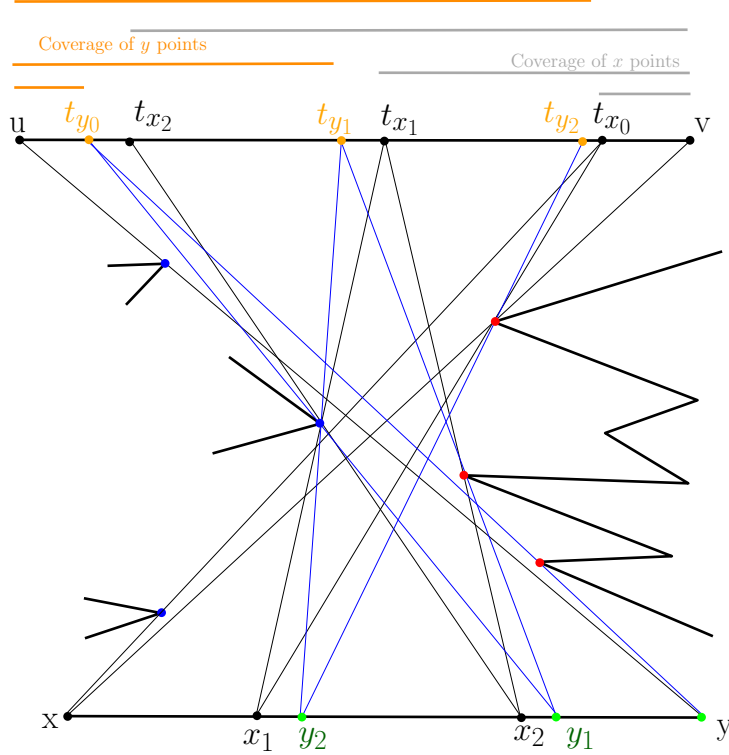


Figure 1: The coverage from both the left and right sides is demonstrated for each iteration. In the second iteration ($j = 2$), the slicing algorithm concludes, having successfully covered the entire target segment.

Figure 5 provides an example that illustrates the iterative process of the slicing algorithm.

End of the iteration.

The set of all x_i and y_i points determines \mathcal{S} .

End of the slicing algorithm.

4.1.3. Results of the slicing algorithm

Lemma 1 indicates that the set \mathcal{S} obtained by the slicing algorithm covers the target. Lemma 2 we know that under the cases of Assumption 1 $|\mathcal{S}|$ remains polynomial in n .

4.2. Observations, Lemmas, Theorems, and their proofs

Observation 1. Given two segments a source \overline{xy} and \overline{uv} inside a simple polygon \mathcal{P} , Assumption 2 cannot guarantee a finite set \mathcal{S} of points on the source to cover the target.

Proof. See Figure.2. Based on Lemma 4 of [18] mentioned previously, we cannot find an finite set of points around p (including the sub-segment of the source around p) that the union visibility of the points in the set can cover the visibility of p . Figure.2 illustrates a counter example for Assumption 2, where we can set the ratio of $\frac{LD}{SD}$ to be large enough without modifying the size of SW . Still the position of p and ℓ can be set so that p sees I_p as an interval on the target. For enlarging the ratio $\frac{LD}{SD}$, we can enlarge the minimal circle by taking the reflex vertices away, in fact we can move the reflex vertices on the parallel lines and provide a large polygon without changing SW . So, Assumption 2 cannot guarantee of finding a finite set of points on the source to cover the target.

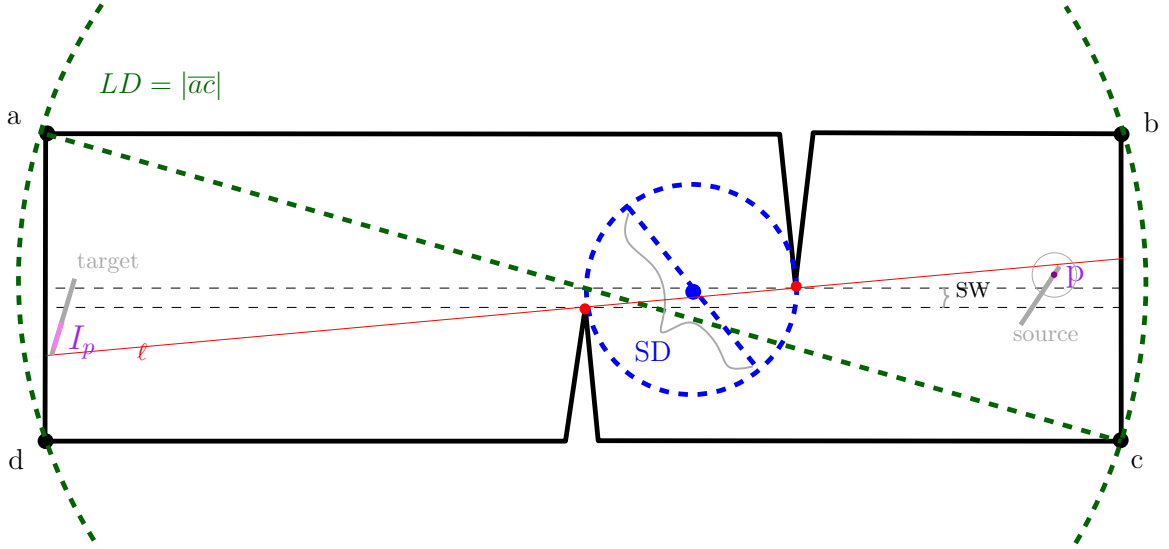


Figure 2: SW based on Assumption 1 is more effective than SD of Assumption 2.

Observation 2. Given a segment \overline{xy} as the source and a target segment \overline{uv} that is partially visible to \overline{xy} , consider a midpoint on the source, denoted as q . There exists a unique reflex vertex obstructing the visibility of q , denoted by LBV_q . This vertex is unique for q . Specifically, if we stand on \overline{xy} with x to our left, LBV blocks the visibility of q from the left side (if it exists). Similarly, RBV is unique (if it exists) and blocks the visibility of q from the right side of \overline{xy} .

Proof. We have to prove that LBV_q and RBV_q are unique reflex vertices on each side for a specific point q on \overline{xy} . Suppose considering the condition of the lemma both of these reflex vertices exist.

Without loss of generality consider LBV_q . On the contrary, suppose it is not unique. For proof, let's consider another reflex vertex denoted by $lrv \neq LBV_q$, which could potentially obstruct the visibility of q (a point on \overline{xy}) not to see some part of the target from the left side.

See Figure.3. To begin, we show that the visibility of q cannot be obstructed by any other reflex vertex aside from LBV_q . Suppose, for the sake of contradiction, that there exists a reflex vertex lrv on the left side of $\overline{qLBV_q}$, which obstructs the visibility of q , preventing it from seeing a portion of the target. In such a scenario, the line crossing \overline{qlrv} should hold LBV on its left side. Otherwise, lrv defines LBV_q itself.

The same analysis reveals that RBV is unique for q on the other side.

Observation 3. Any point p on the source sees an interval I_p which $I_p < SW$, where SW comes from Assumption 1.

Proof. A point p moving from x to y on the source, the half-lines through \overline{pLBV} and \overline{pRBV} are divergent. So, the interval created on the target are larger than distance between the parallel lines crossing LBV and RBV . \square

Lemma 1. Given two weakly visible segments, \overline{xy} (the source) and \overline{uv} (the target), inside a simple polygon \mathcal{P} , the slicing algorithm described in Subsection 4.1 generates a set of points on \overline{xy} , denoted by \mathcal{S} , that collectively cover the entire target.

Proof. Without loss of generality, consider x_i . Any point x_i sees the target between two lines: one crossing $\overline{x_iLBV_{x_i}}$ and the other crossing $\overline{x_iRBV_{x_i}}$. Note that the target is weakly visible to the source. Thus, $x = x_0$ must see v , and the visibility of the x points progressively aims to cover the target from v to u . The reflex vertices that block the visibility of the x series from seeing a part of the target are the

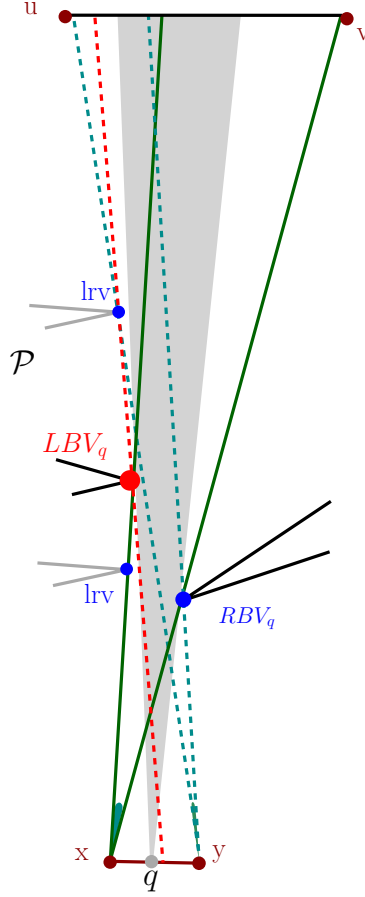


Figure 3: Considering a source and a target and a point q on \overline{xy} , LBV_q and RBV_q vertices (if exist) are unique.

LBV vertices. In each iteration i , the line crossing $\overline{x_i LBV_{x_i}}$ determines $t_{x_{i+1}}$ (see Figure.4), and x_{i+1} can see the target starting from $t_{x_{i+1}}$. Therefore, the visibility of the x_i points on the target are connected. Thus, the target is visible to x_i from t_{x_i} to $t_{x_{i+1}}$. This process continues until the iteration stops, either when x_i sees u or when reaching a point y_i where the remaining portion of the target has already been covered by y points from the previous iterations.

Lemma 2. *The output of the slicing algorithm (as detailed in Subsection 4.1) produces a finite set of points, \mathcal{S} , on the source. Under Assumption 1, the size of \mathcal{S} is polynomial in n , where n represents the complexity of \mathcal{P} .*

Proof. Without loss of generality, we present the proof considering only the points labeled x .

First, observe that $RBV_{x_{i+1}}$ and $LBV_{t_{x_i}}$ must be the same reflex vertex. If they were different, x_{i+1} would be able to see a point closer to v on the target, and $LBV_{t_{x_i}}$ would not be obstructing its visibility.

Thus, in each iteration, x_i sees the target between two lines: one crossing $\overline{x_i LBV_{x_i}}$ and the other crossing $\overline{x_i RBV_{x_i}}$. From Observation 3, we know that the number of points on the source obtained by the slicing algorithm is upper bounded by the ratio of LW to SW . \square

Lemma 3. *Given two weakly visible segments, the source \overline{xy} and the target \overline{uv} , inside a simple polygon \mathcal{P} , the slicing algorithm presented in Subsection 4.1 completes its execution in the j^{th} iteration, where y_j lies to the left of x_j . In this iteration, the target segment is guaranteed to be fully covered.*

Proof. We only have to prove that t_{y_j} lies to the left of (or on) t_{x_i} . This means that the coverage of the target from the left meets the coverage from the right side.

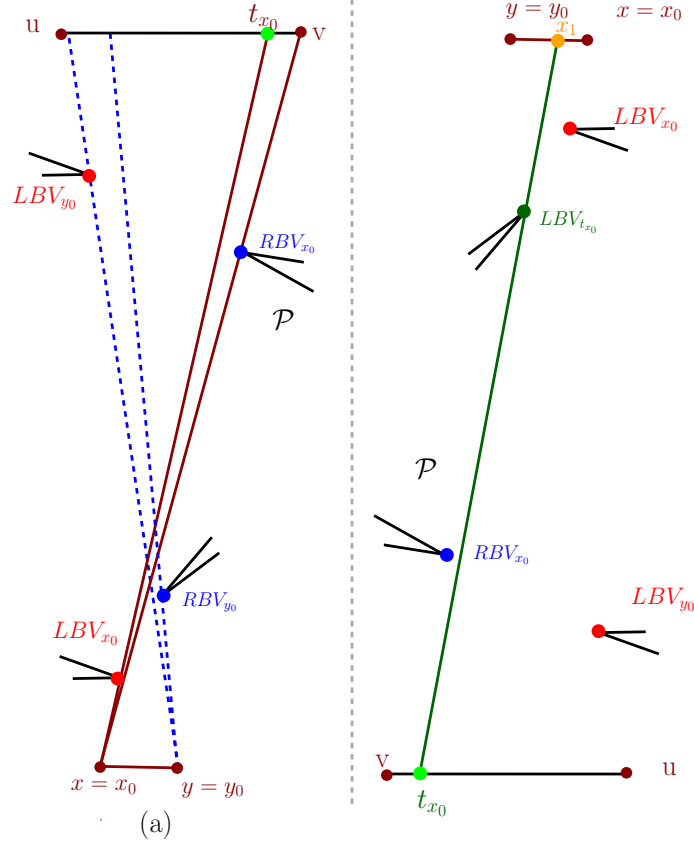


Figure 4: The first iteration of the slicing algorithm.

Without loss of generality, suppose the visibility of both x_j and y_j is obstructed by reflex vertices. Consider the triangle formed by the points x_j , y_j , and LBV_{x_j} . Since y_j is to the left of x_j and has visibility of some part of the target, y_j must lie to the left of the line passing through $\overline{x_j LBV_{x_j}}$. Similarly, analyzing the triangle formed by x_j , y_j , and RBV_{y_j} reveals that x_j lies to the right of the line passing through $\overline{y_j RBV_{y_j}}$. Given that LBV_{x_j} lies to the left of the interior of \mathcal{P} and RBV_{y_j} lies to the right of the interior of \mathcal{P} , the lines passing through $\overline{x_j LBV_{x_j}}$ and $\overline{y_j RBV_{y_j}}$ must intersect, placing their intersection on the target. Consequently, t_{x_j} lies to the left of t_{y_j} on the target.

In case x_j and y_j coincide at a single point, the LBV and RBV of that point are distinct. The lines passing through x_j (or y_j) and LBV , and through x_j (or y_j) and RBV , intersect the target at different points such that t_{x_j} lies to the left of t_{y_j} . \square

Lemma 4. For two parallel lines L_1 and L_2 at distance d tangent to reflex vertices in a simple polygon \mathcal{P} , the strip between them contains a region of area $\Omega(d^2)$.

Proof. Let L_1 and L_2 be two parallel lines at distance d , tangent to reflex vertices of a simple polygon \mathcal{P} . Since L_1 and L_2 are tangent, the polygon must touch both lines, ensuring that \mathcal{P} spans the strip between them.

The region of \mathcal{P} within this strip must occupy an area dictated by the strip's width d . The minimal configuration occurs when \mathcal{P} forms a parallelogram spanning the strip, with base and height both equal to d , yielding an area of d^2 . In degenerate cases, the polygon may form a triangular region within the strip, with area $\frac{d^2}{2}$.

Since both cases satisfy the lower bound of $\Omega(d^2)$, the result follows.

To be more precise:

1. The tangent lines create a “corridor” of width d .

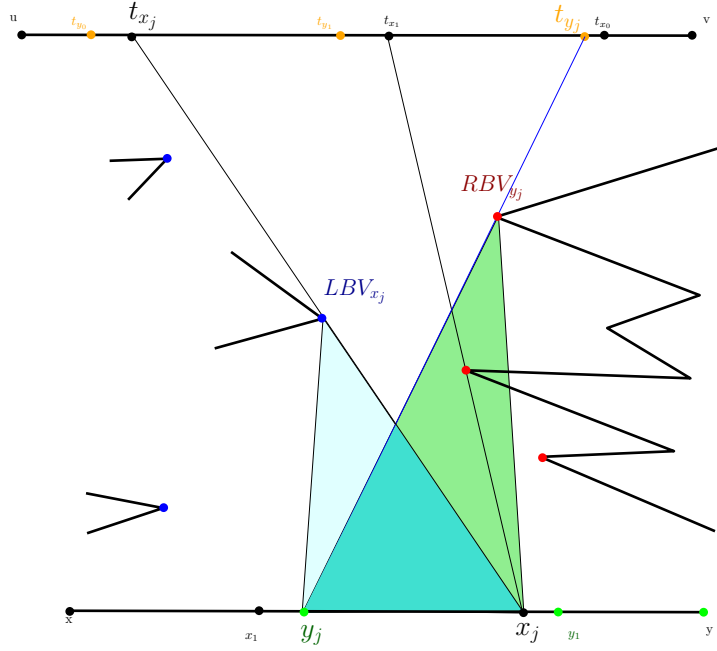


Figure 5: In this example, $j = 2$. It demonstrates the j^{th} iteration, where y_j is positioned to the left of x_j .

2. By the isoperimetric inequality, the minimal-area shape fitting this corridor is a rectangle or parallelogram.
3. The polygon must occupy at least the area of a parallelogram with: Base = d , and Height = d (ensuring area d^2)
4. In degenerate cases, the region may be a triangle with area $d^2/2$, preserving $\Omega(d^2)$.

5. Discussion

Our work addresses a core problem in segment guarding by ensuring finite solutions without relying on restrictive stability assumptions. We introduce the novel concept of the *line aspect ratio* (AR), a geometric parameter quantifying the anisotropy of reflex vertices. This framework guarantees finite guard sets (\mathcal{S}) with a size bounded by $\mathcal{O}(\text{AR})$.

5.1. A few open problems

1. Optimality of $|\mathcal{S}|$: Is $|\mathcal{S}| = \Theta(\text{AR})$ optimal?
2. Dynamic Guarding: Can the guard set \mathcal{S} adapt to polygon deformation while maintaining bounded size?
3. Extension to 3D: How can the AR framework be generalized for guarding problems in three-dimensional environments?

The AR assumption aligns with realistic geometric constraints observed in various domains:

1. *Architectural Layouts*: Reflex vertices often form corridors with bounded anisotropy, such as in building floor plans [19].
2. *Geographic Meshing*: Terrain models exhibit moderate variations in width and feature distribution [20].
3. *Sensor Networks*: Efficient sensor deployment frequently leverages regions with bounded aspect ratios [21].

4. *Robotic Navigation*: Structured environments simplify visibility reasoning for autonomous systems [22].

This framework bridges theoretical advances and practical applications, providing a robust foundation for future research in visibility and guarding problems

Declaration on Generative AI

The author has not employed any Generative AI tools.

References

- [1] D. Avis, G. T. Toussaint, An optimal algorithm for determining the visibility of a polygon from an edge., *IEEE Transactions on Computers* 30 (1981) 910–1014.
- [2] L. J. Guibas, J. Hershberger, D. Leven, M. Sharir, R. E. Tarjan., Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons., *Algorithmica* (1987) 209–233.
- [3] L. J. Guibas, R. Motwani, P. Raghavan, Visibility of disjoint polygons, *Algorithmica* 1 (1987) 49–63.
- [4] J. Hershberger, S. Suri, An optimal algorithm for euclidean shortest paths in the plane, *SIAM Journal on Computing* 28 (1989) 2215–2256. doi:10.1137/0728167.
- [5] J.-R. Sack, S. Suri, Weak visibility of two simple polygons, *Computational Geometry: Theory and Applications* 1 (1991) 43–58.
- [6] M. Ghods, A. Mirzaian, Weak visibility queries between disjoint line segments in polygons, *Computational Geometry: Theory and Applications* 39 (2008) 239–249. doi:10.1016/j.comgeo.2007.12.001.
- [7] C. D. Tóth, Art gallery problem with guards on edges, *SIAM Journal on Discrete Mathematics* 24 (2010) 1–11.
- [8] V. King, J. Snoeyink, Guarding polygons with mobile guards, *Computational Geometry: Theory and Applications* 26 (2004) 209–219.
- [9] B. Bhattacharya, S. K. Ghosh, Approximability of guarding weakly visible polygons, in: *Proceedings of the 31st International Symposium on Computational Geometry (SoCG)*, Springer, 2015, pp. 201–215. doi:10.1007/978-3-319-20086-6_15.
- [10] T. C. Biedl, T. M. Chan, F. Montecchiani, Sliding cameras in orthogonal polygons: Approximation, hardness, and art gallery bounds, in: *Proceedings of the Symposium on Computational Geometry (SoCG)*, 2017, pp. 45:1–45:16.
- [11] A. Vaezi, B. Roy, M. Ghodsi., Reflection helps guarding an art gallery., *The 38th European Workshop on Computational Geometry*, Mar 2022, Perugia, Italy. hal-03674221 (2022) 3:1–3:7. URL: <https://eurocg2022.unipg.it/booklet/EuroCG2022-Booklet.pdf>.
- [12] A. Vaezi, M. Ghodsi, How to extend visibility polygons by mirrors to cover invisible segments, in: S.-H. Poon, M. S. Rahman, H.-C. Yen (Eds.), *WALCOM: Algorithms and Computation*, Springer International Publishing, Cham, 2017, pp. 42–53.
- [13] A. Vaezi, M. Ghodsi, Visibility extension via reflection-edges to cover invisible segments., *Theoretical Computer Science* (2019). doi:10.1016/j.tcs.2019.02.011.
- [14] D. T. Lee, K.-Y. Chwa, Visibility of a simple chain in a polygon, in: *Proceedings of the 6th Annual Symposium on Computational Geometry (SoCG)*, 1990, pp. 211–220.
- [15] M. Authors, k-transmitters for segment visibility: Algorithms and applications, *ALGOWIN* 10 (2023) 100–115.
- [16] D. Chen, Structured visibility profiles in polygons, in: *Proceedings of the Symposium on Computational Geometry (SoCG)*, 1998, pp. 23–30.
- [17] É. Bonnet, T. Miltzow, Approximating the art gallery problem, *Leibniz International Proceedings in Informatics (LIPIcs)* 51 (2016) 17:1–17:15.
- [18] E. Bonnet, T. Miltzow, An approximation algorithm for the art gallery problem, *The 33rd*

International Symposium on Computational Geometry (SoCG'17) 77 (2017) 20:1–20:15. URL: <https://hal.science/hal-01994349v1>. doi:10.4230/LIPIcs.SoCG.2017.20.

- [19] Y. Schwartzburg, M. Pauly, High-resolution topological tools for immersive architectural design, *Computer-Aided Design* 55 (2014) 45–57. Demonstrates bounded aspect ratios in architectural feature decomposition.
- [20] E. Puppo, Variable resolution terrain surfaces, *Proc. CG International* (1997) 81–90. Shows natural terrains have bounded width ratios in mesh simplification.
- [21] B. Wang, K. C. Chua, Coverage in hybrid mobile sensor networks, in: *IEEE MASS*, 2007, pp. 1–8. Uses aspect-ratio constraints for sensor placement optimization.
- [22] S. M. Lavalle, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transactions on Robotics* 12 (2004) 566–580. Assumes bounded environment anisotropy for efficient visibility sampling.