Oriented Rotating Rays Voronoi Diagram in the Plane and in Simple Polygons

Ioannis Mantas[1], Marko Savić[2]*, and Martin Suderland[3]

[1] Endra, Zurich, Switzerland
ioanni.mantas@gmail.com
[2] Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Serbia
marko.savic@dmi.uns.ac.rs
[3] Independent Researcher, Lugano, Switzerland
martinsuderland.research@gmail.com

Abstract. We consider Voronoi diagrams where sites are rays and distance is given by an angular rotation of the ray around its apex. This structure, known as the *rotating rays Voronoi diagram*, has been shown to encode the critical common floodlight aperture needed to illuminate a target domain. In the oriented variant, where rotation is restricted to one direction, we show that the diagram of n rays in the plane has combinatorial complexity $O(n^2 \log^{6+\epsilon} n)$ and can be constructed in expected $O(n^2 \log^{7+\epsilon} n)$ time. We also study the diagram in simple polygons, extending the results for convex polygons. For edge-aligned rays on a simple n-gon with r reflex vertices, we prove an $O(n^3 + n^2 r\, 2^{\alpha(n)})$ upper bound and an $\Omega(nr)$ worst-case lower bound on the combinatorial complexity, and give an $O((n^3 + n^2 r\, 2^{\alpha(n)}) \log n)$-time construction algorithm.

Keywords: Rotating rays Voronoi diagram · Oriented angular distance · Brocard illumination · Floodlight illumination · Wedge coverage

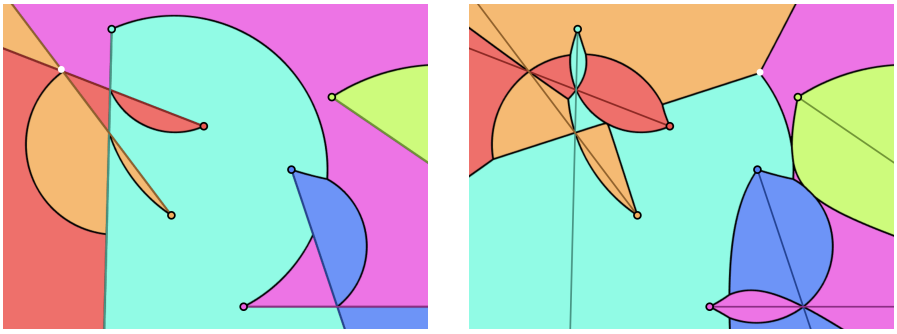
1 Introduction

A *rotating rays Voronoi diagram* takes a set of rays as input sites and measures distance by the smallest angle through which a ray must rotate around its apex until it hits a query point. Depending on whether that rotation is restricted to a single direction or allowed in both, one obtains the *oriented* or *unoriented* variants, respectively. See Fig. 1 for instances of the diagram in $\mathbb{R}^2$.

Motivation. The initial motivation came from a variant of floodlight illumination problems, known as *Brocard illumination* [5]. One floodlight is aligned with each ray; all floodlights open with a common aperture, increasing at the same speed, and the goal is to find the minimum aperture needed to illuminate the whole

* Partly supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia (Grants No. 451-03-33/2026-03/ 200125 & 451-03-34/2026-03/ 200125) and by the Science Fund of the Republic of Serbia, #7462, Graphs in Space and Time: Graph Embeddings for Machine Learning in Complex Dynamical Systems – TIGRA.

This version of the contribution has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: [DOI TBD]. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms



(a) Oriented variant. (b) Unoriented variant.

Fig. 1: Voronoi diagrams of a set of 6 rays in the plane.

domain. Alegría et al. [2] showed that the rotating rays Voronoi diagram encodes the minimum common aperture required to illuminate the domain. In Figs. 1 and 2, the last point to be illuminated is illustrated with a white dot.

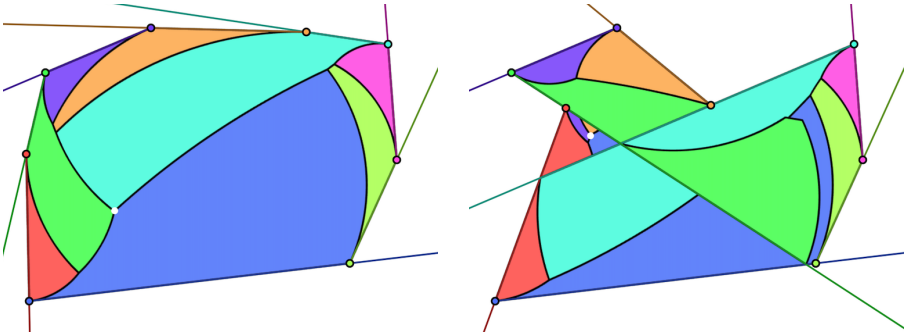
More generally, the diagram can model the angular component of directional coverage problems involving wedge-shaped sensing or transmission regions [21], as encountered in camera networks [8], antennas [12], and wireless charging [14].

Related work. The oriented rotating rays Voronoi diagram was introduced and analyzed by Alegría et al. [2] in different domains. For n rays in the plane, they established an $\Omega(n^2)$ combinatorial lower bound and an $O(n^{2+\epsilon})$ upper bound, together with a construction algorithm, whose running time matches the upper bound. For convex polygonal domains, where the rays are edge-aligned, they showed that the diagram inside the polygon can be computed in optimal $\Theta(n)$ time. They also analyzed the complexity of the diagram restricted to curves. The general simple-polygon setting has been largely left open. A first discussion was given by Mantas [17], who presented an $O(n^{4+\epsilon}r)$ combinatorial complexity bound, with a matching construction time, where r is the number of reflex vertices. An $\Omega(r^2)$ lower-bound construction was given by Alegría et al. [3]. Examples of the diagram in different polygonal domains are given in Fig. 2.

The unoriented rotating rays Voronoi diagram in $\mathbb{R}^2$ remains largely open. It appeared as a special case of a diagram defined by de Berg et al. [11], who showed an $\Omega(n^2)$ worst-case lower bound on its combinatorial complexity, and its bisectors were partially studied by Haverkort and Klein [16]. Alegría et al. [3] later reviewed the structure, together with an interactive visualization tool [4][1].

For the associated Brocard illumination problem, the plane and convex-polygon constructions of Alegría et al. [2] yield the minimum common aperture within the same time bounds. For both convex and general simple polygons, Alegría et al. [5] proposed direct algorithms.

[1] All images of Voronoi diagrams in this work were generated using this software.



(a) Convex polygon. (b) Non-convex polygon.

Fig. 2: Voronoi diagrams in polygonal domains, each with 8 edge-aligned rays.

For an overview of illumination problems we refer to [22,19]. For general results on Voronoi diagrams see the handbooks [10,18].

Contributions. We extend results on the oriented diagram in two settings.

- For the oriented diagram in $\mathbb{R}^2$, we give an upper bound of $O(n^2 \log^{6+\epsilon} n)$ on the combinatorial complexity. We also present an expected $O(n^2 \log^{7+\epsilon} n)$-time construction algorithm, and refine some related structural bounds.
- For edge-aligned rays on simple polygons, we prove an $O(n^3 + n^2 r\, 2^{\alpha(n)})$ upper bound on the combinatorial complexity and an $\Omega(nr)$ worst-case lower bound. We construct the diagram in $O((n^3 + n^2 r\, 2^{\alpha(n)}) \log n)$ time by a generalized collapse algorithm. We also compute the minimum common aperture α^* in the same time and show that its range is $(0, 2\pi)$.

The paper is organized as follows: Section 2 gives the preliminaries, Section 3 deals with the oriented diagram in $\mathbb{R}^2$, and Section 4 treats the edge-aligned diagram in polygons. Finally, Section 5 concludes with some open questions.

2 Preliminaries

Let $\mathcal{S}$ be a set of n rays in the plane. A *ray* s is determined by its apex $p(s) \in \mathbb{R}^2$ and its direction $\widehat{d}(s)$. We denote by $l(s)$ the supporting line of s. For two points $x, y \in \mathbb{R}^2$, $\overline{xy}$ denotes the line segment joining them. We denote by $\alpha(n)$ the inverse Ackermann function and by $\lambda_s(n)$ the maximum length of a Davenport-Schinzel sequence of order s on n symbols.

Definition 1. *Given a ray s and a point $x \in \mathbb{R}^2$, the* angular distance *from x to s, denoted by $d(x, s)$, is the counterclockwise angle by which s has to rotate around its apex so that it hits x; see Fig. 3.*

Thus, $d(x, s) \in [0, 2\pi)$ and $d(x, s) = 0$ if and only if $x \in s$. Given two rays s, t, the *dominance region* of s over t is $dreg(s, t) := \{\, x \in \mathbb{R}^2 \mid d(x, s) < d(x, t) \,\}$, and their *bisector* $b(s, t)$ is the common boundary of $dreg(s, t)$ and $dreg(t, s)$.

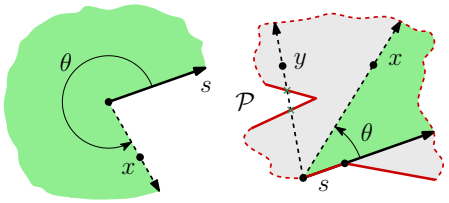


Fig. 3: Angular distances.
(Left) Domain $\mathbb{R}^2$: $d(x, s) = \theta$.
(Right) $\mathcal{P}$: $d(x, s) = \theta$, $d(y, s) = +\infty$.

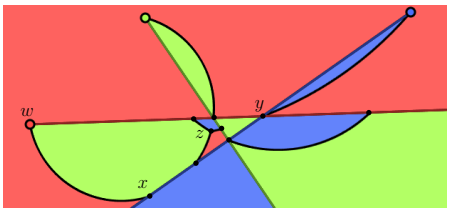


Fig. 4: Voronoi diagram of 3 rays in $\mathbb{R}^2$ with 12 vertices; x is mixed, y is intersection, z is proper, w is apex.

As shown in [2], $b(s, t)$ consists of the two rays s and t, together with a circular part connecting their apices; this part may degenerate into straight-line pieces in non-general position. Unless otherwise stated, we assume that no two rays share an apex and that no two rays are parallel or anti-parallel. Additional general-position assumptions are stated separately in the corresponding sections.

Definition 2. *Let $\mathcal{S}$ be a set of rays, called sites, and let $s \in \mathcal{S}$. The* Voronoi region *of s is*

$$vreg(s) := \bigcap_{t \in \mathcal{S} \setminus \{s\}} dreg(s, t).$$

The rotating rays Voronoi diagram *of $\mathcal{S}$, denoted by $RVD(\mathcal{S})$, is the subdivision of $\mathbb{R}^2$ induced by the Voronoi regions of the rays in $\mathcal{S}$; see Fig. 4.*

A connected component of a Voronoi region is called a *face*. We next define the corresponding diagram inside a simple polygon. Let $\mathcal{P}$ be a simple polygon, including its boundary, with vertices $v_1, \ldots, v_n$ in counterclockwise order, r of which are reflex; indices are taken modulo n. At every vertex v_i we place the *edge-aligned* ray s_i along the edge $\overrightarrow{v_i v_{i+1}}$, and write $\mathcal{S} = \{s_1, \ldots, s_n\}$.

Definition 3. *Let $s \in \mathcal{S}$ and $x \in \mathcal{P}$. A ray s* sees *a point x, or equivalently s is visible from x, if and only if $\overline{p(s)x} \subseteq \mathcal{P}$. The* visibility-restricted angular distance *from x to s in $\mathcal{P}$, denoted by $d(x, s)$, is the angular distance of Definition 1 if s sees x, and $+\infty$ otherwise; see Fig. 3.*

We use d for either distance, with the domain understood from context.

Definition 4. *Let $\mathcal{P}$ be a simple polygon and let $\mathcal{S}$ be its set of edge-aligned rays. For $s \in \mathcal{S}$, the* Voronoi region *of s in $\mathcal{P}$ is*

$$vreg(s) := \{\, x \in \mathcal{P} \mid d(x, s) < d(x, t) \text{ for every } t \in \mathcal{S} \setminus \{s\} \,\}.$$

The edge-aligned rotating rays Voronoi diagram *of $\mathcal{P}$, denoted by $PRVD(\mathcal{P})$, is the subdivision of $\mathcal{P}$ induced by these Voronoi regions; see Fig. 2.*

In both settings, portions of the circular parts of pairwise bisectors give rise to *circular edges* of the diagram. Other features arise on the domain boundary, and at loci where the distance is discontinuous or visibility changes. In the plane these are induced by the input rays; inside a polygon they are induced by visibility boundaries, and may also appear on $\partial\mathcal{P}$. Under general position, a vertex incident to three circular edges is called a *proper vertex*. The remaining vertex types depend on the setting and are introduced in the corresponding sections.

An *α-floodlight aligned with s* is the set of points at distance at most α from s. Finally, we use the following formulation of the associated illumination problem.

Definition 5. *Given a domain $\mathcal{D}$ and a set of rays $\mathcal{S}$, the* α^*-floodlight illumination problem *asks for the smallest common aperture α^* such that the α^*-floodlights aligned with the rays of $\mathcal{S}$ jointly cover $\mathcal{D}$. Equivalently,*

$$\alpha^* \;=\; \sup_{x\in\mathcal{D}} \min_{s\in\mathcal{S}} d(x,s).$$

3 Voronoi Diagram in the Plane

Throughout this section, we assume general position in addition to the assumptions of Section 2. No apex lies on the supporting line of another ray, and no three supporting lines are concurrent.

The non-circular edges of $\mathrm{RVD}(\mathcal{S})$ are portions of input rays, called *ray edges*. For every subset of at least three sites, the diagram is assumed to have only the nondegenerate vertex incidences described below. Besides proper vertices, defined in Section 2, the diagram has three further vertex types [2]: A *mixed vertex* is incident to one circular edge and two ray edges induced by a single ray. An *intersection vertex* is incident to one circular edge and four ray edges induced by two rays. An *apex vertex* is incident to one circular edge and one ray edge, where the ray inducing the ray edge is also one of the two rays inducing the circular edge. See Fig. 4 for an illustration.

We first look at the diagram of three rays, tightening the result of [2].

Proposition 6. *For any three rays s,t,u, $\mathrm{RVD}(\{s,t,u\})$ has at most* 1 *proper vertex, at most* 3 *intersection vertices, at most* 5 *mixed vertices, and exactly* 3 *apex vertices. Instances achieving all* 12 *vertices are realizable; see Fig. 4.*

Proof (sketch). All but the upper bound on the mixed vertices have been proved in [2]; it suffices to rule out six mixed vertices, the upper bound of [2]. To do so, we reduce to the cyclic and transitive orientations of the triangle formed by the rays' supporting lines and derive incompatible secant inequalities in each case.

For a line ℓ, let $\mathrm{RVD}_\ell(\mathcal{S})$ denote the one-dimensional minimization diagram of the angular-distance functions restricted to ℓ.

Proposition 7. *$\mathrm{RVD}_\ell(\mathcal{S})$ has $O(n\alpha(n))$ complexity and it can be constructed in $O(n\alpha(n)\log n)$ time. If no ray in $\mathcal{S}$ intersects ℓ, then $\mathrm{RVD}_\ell(\mathcal{S})$ has $O(n)$ complexity and it can be constructed in $O(n\log n)$ time.*

Proof. By viewing $\mathrm{RVD}_\ell(\mathcal{S})$ as the lower envelope of the distance functions induced by $\mathcal{S}$ along ℓ, an $O(n2^{\alpha(n)})$ bound was shown [2]. This envelope is contained in the boundary of the unbounded face below the graphs of these distance functions. Aronov and Drusvyatskiy [6] distinguish three categories of curves in $\mathbb{R}^2$, no two of which intersect more than s times: *bounded*, *semi-infinite*, and *bi-infinite* (with respectively two, one, and no endpoints), for which the complexity of a single face is $O(\lambda_{s+2}(n))$, $O(\lambda_{s+1}(n))$, and $O(\lambda_s(n))$, respectively.

For the improved bound, parametrize ℓ by a real coordinate. If an input ray intersects ℓ at a single point i, split its distance function into two continuous pieces with domains $(-\infty, i)$ and $(i, +\infty)$. If ℓ contains an input ray, split its distance function at its apex in the same way. Split each remaining bi-infinite graph at an arbitrary point. Taking closures gives $O(n)$ semi-infinite curves. Any two such curves intersect at most twice, by the pairwise bisector characterization. The semi-infinite case therefore gives $O(\lambda_3(n)) = O(n\alpha(n))$ complexity. Restoring the actual distance values at the split points adds only $O(n)$ features. If no input ray intersects ℓ, the original distance functions are continuous along ℓ and their graphs are bi-infinite, giving $O(\lambda_2(n)) = O(n)$ complexity.

$\mathrm{RVD}_\ell(\mathcal{S})$ can be constructed by divide and conquer. Split $\mathcal{S}$ into two balanced groups, recursively construct their lower envelopes, and merge them by a traversal along ℓ. Discontinuities are handled during the traversal, and each merge takes time linear in the total complexity of the two envelopes, since any two active pieces intersect at most twice. The bounds above apply to every subset, so each of the $O(\log n)$ levels takes $O(n\alpha(n))$ time, or $O(n)$ if no input ray intersects ℓ. The resulting times are $O(n\alpha(n)\log n)$ and $O(n\log n)$, respectively.

The above bounds also clearly hold for restrictions to connected subsets of ℓ.

Proposition 8. *RVD($\mathcal{S}$) has $O(n^2\alpha(n))$ mixed vertices. Further, if the rays in $\mathcal{S}$ are pairwise disjoint, then RVD($\mathcal{S}$) has $O(n^2)$ mixed vertices.*

Proof. Fix a ray $s \in \mathcal{S}$. Every mixed vertex on s appears on the lower envelope of the distance functions of $\mathcal{S}\backslash\{s\}$ restricted to s. By Proposition 7, this envelope has $O(n\alpha(n))$ complexity. If the input rays are pairwise disjoint, the remaining distance functions are continuous on the same halfline s. Any two intersect at most twice, so their lower-envelope sequence has order at most two and hence $O(n)$ complexity. Summing over the n input rays gives the claimed bounds.

Theorem 9. *RVD($\mathcal{S}$) has $O(n^2\log^{6+\epsilon} n)$ combinatorial complexity and it can be constructed in $O(n^2\log^{7+\epsilon} n)$ expected time, for any fixed $\epsilon > 0$.*

Proof. We apply to all angular distances the same strictly increasing transformation of [2]: $f(\theta) = 1 - \cos\theta$ for $0 \leqslant \theta \leqslant \pi$ and $f(\theta) = 3 + \cos\theta$ for $\pi < \theta < 2\pi$. The transformed functions are order-equivalent; except at the apex, each graph consists of two semi-algebraic surface patches of constant description complexity.

Let Σ be the resulting collection of patches; thus $|\Sigma| = 2n$. By order-equivalence, the minimization diagram of Σ, after identifying the two patches associated with each ray, coincides with $\mathrm{RVD}(\mathcal{S})$ except for the omitted apex features. It therefore suffices to bound the lower envelope of Σ.

Agarwal, Ezra, and Sharir [1] proved that the lower envelope of N semi-algebraic surface patches of constant description complexity has complexity $O(N^2 \log^{6+\epsilon} N)$ and can be constructed in expected $O(N^2 \log^{7+\epsilon} N)$ time, provided that the relative interiors of any three patches intersect in at most two points. We show that, in our case, they intersect in at most one point.

If two patches in a triple belong to the same ray, their relative interiors are disjoint. Otherwise, a common point of three patch interiors projects to a point x satisfying $d(x, s) = d(x, t) = d(x, u)$ for three distinct rays s, t, u. Since x lies in the interiors of the three patch domains, these equalities occur on the circular parts of the corresponding bisectors. Under our general-position assumptions, x is therefore a proper vertex of $\mathrm{RVD}(\{s, t, u\})$. By Proposition 6, there is at most one such vertex, proving the required triple-intersection bound.

Applying [1] with $N = 2n$ gives the claimed envelope bounds. The omitted $O(n)$ apex features are restored in $O(n^2)$ total comparisons by finding the closest other site at each apex; the remaining reconstruction is linear in the envelope complexity. Thus, the complexity and algorithmic bounds also hold for $\mathrm{RVD}(\mathcal{S})$.

4 Edge-Aligned Voronoi Diagram in Simple Polygons

For a vertex v_i, its *visibility polygon* is the set of points $x \in \mathcal{P}$ such that $\overline{v_i x} \subseteq \mathcal{P}$. An edge of the visibility polygon whose relative interior lies in the interior of $\mathcal{P}$ is a *visibility window* of v_i [7]. Each visibility polygon has at most r visibility windows, and hence there are $O(nr)$ visibility windows in total.

Let Γ be the union of $\partial\mathcal{P}$ and all visibility windows, and call the connected components of $\operatorname{int}\mathcal{P}\backslash\Gamma$ the *cells* of Γ. Within each cell C, the set of visible sites is constant. Let $\mathcal{S}_C \subseteq \mathcal{S}$ denote this set. Then $\mathrm{PRVD}(\mathcal{P}) \cap C = \mathrm{RVD}(\mathcal{S}_C) \cap C$. Hence, outside Γ, $\mathrm{PRVD}(\mathcal{P})$ locally coincides with a plane diagram. Moreover, the visible part of every edge-aligned ray consists of its polygon edge and, possibly, a visibility window, and is therefore contained in Γ. Consequently, every vertex of $\mathrm{PRVD}(\mathcal{P})$ outside Γ is a proper vertex.

For three sites $s, t, u \in \mathcal{S}$, a proper vertex of $\mathrm{RVD}(\{s, t, u\})$ is called a *proper candidate*. Each triple induces at most one, since two bisecting circles sharing an apex have at most one further isolated intersection. A proper candidate is *visibility-valid* if it lies in $\operatorname{int}\mathcal{P}$ and all three inducing sites see it. Notice that a visibility-valid proper candidate need not be a vertex of $\mathrm{PRVD}(\mathcal{P})$, since another visible site may have smaller distance at that point.

On Γ, visibility may change, and Voronoi edges may meet or terminate. A visibility window w of site s splits $\mathcal{P}$ into two (*left* and *right*) sides, as seen from $p(s)$. For a visibility window w and either of its two sides, the subdivision of w obtained by approaching it from that side is called the *one-sided trace* of $\mathrm{PRVD}(\mathcal{P})$ on w.

For the sake of clarity, we assume general position, although all results remain valid without this assumption: no three polygon vertices are collinear, no three sites have equal finite distance to a point on Γ, and no four sites have equal finite distance to a point of $\operatorname{int}\mathcal{P}\backslash\Gamma$.

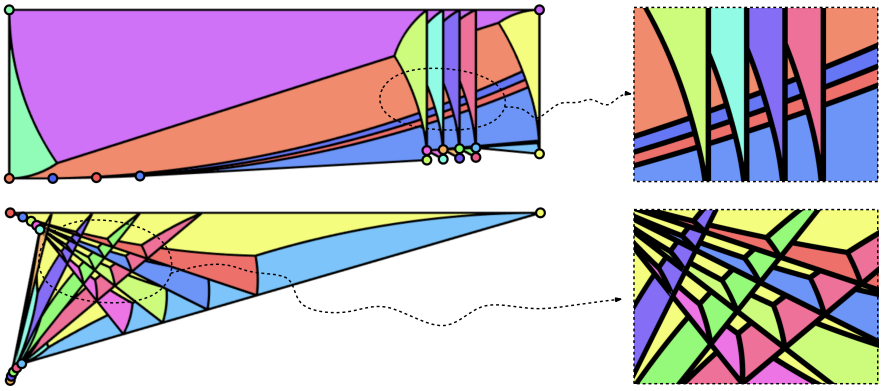


Fig. 5: Simple polygons whose diagrams have $\Omega(nr)$ proper vertices. Top: a convex chain and a chain of "reflex teeth". Bottom: two reflex chains.

4.1 Combinatorial Complexity

We give upper and lower bounds on the combinatorial complexity of PRVD($\mathcal{P}$).

Theorem 10. *PRVD($\mathcal{P}$) has combinatorial complexity $O(n^3 + n^2 r 2^{\alpha(n)})$.*

Proof. Assume that $r \geqslant 1$; if $r = 0$, the polygon is convex and the result follows from the $O(n)$ bound of [2]. We bound separately the vertices of PRVD($\mathcal{P}$) that lie on Γ and those that lie in the interior of the cells of Γ.

Vertices on Γ. Let w be a visibility window and fix one of its sides. For every site s_i, consider the one-sided limit of its visibility-restricted distance along w. Its finite domain is an interval, possibly empty, by the visibility-interval property applied to nearby parallel segments [7], and the limit is continuous in its interior, because the visible part of s_i lies on the boundary of the region that v_i sees. Any two such functions intersect at most twice. Therefore, by the bounded-curve bound of [6], each one-sided trace has complexity $O(\lambda_4(n)) = O(n2^{\alpha(n)})$ and can be computed in $O(n2^{\alpha(n)} \log n)$ time (as in the proof of Proposition 7). The trace on an edge $\overline{v_i v_{i+1}}$ has one piece, since s_i has distance 0 there. There are $O(nr)$ visibility windows, so the complexity along Γ is $O(nr\lambda_4(n) + n) = O(n^2 r 2^{\alpha(n)})$.

Vertices off Γ. Every diagram vertex outside Γ is a visibility-valid proper candidate. Since each triple of sites induces at most one such candidate, there are at most $\binom{n}{3} = O(n^3)$ vertices outside Γ.

Consequently, PRVD($\mathcal{P}$) has complexity $O(n^3 + n^2 r 2^{\alpha(n)})$.

Theorem 11. *For any n, r such that $r \leqslant n-3$ there exists a polygon $\mathcal{P}$ having a diagram PRVD($\mathcal{P}$) with $\Omega(nr)$ combinatorial complexity.*

Proof (sketch). We give two constructions, depending on the values of n and r; see Fig. 5. For $1 \leqslant r \leqslant n/4$, we adapt the two-family construction of [2],

combining a shallow convex chain of $\Theta(n)$ sites with r "reflex teeth". The lower envelope of the chain distances has $\Theta(n)$ transitions along each tooth window. Each transition produces a nearby proper vertex where the two chain sites and the upward site of the tooth are equally close.

For $r > n/4$, two short reflex chains yield $\lfloor r/2 \rfloor \lceil r/2 \rceil = \Omega(nr)$ cross-family ray intersections. A distant closing site creates a nearby proper vertex for every crossing, at a sufficiently small positive angle. The chain geometry ensures visibility of the three inducing sites, while all other sites remain farther away.

4.2 Construction Algorithm

We construct PRVD($\mathcal{P}$) by a generalized *collapse* algorithm: one can think of simultaneously growing floodlights of common aperture θ and tracing the boundary of the still uncovered region as θ increases. It can be viewed as a generalization of the convex collapse algorithm [2] to general polygons. The next theorem provides the result of this section, which we will explain afterwards.

Theorem 12. *PRVD($\mathcal{P}$) can be constructed in $O((n^3 + n^2 r 2^{\alpha(n)}) \log n)$ time.*

Let $d(x) = \min_{s \in \mathcal{S}} d(x, s)$ and $U_\theta = \{x \in \mathcal{P} : d(x) > \theta\}$. Thus, U_θ is the part of $\mathcal{P}$ that is still uncovered when every floodlight has aperture θ. As θ increases, U_θ shrinks until it becomes empty. For a given θ, every boundary cycle of U_θ is represented by a cyclic list of boundary pieces, each associated with a site.

Four types of events appear during the collapse process: *destruction*, *creation*, *extension*, and *shrinking*; see Fig. 6. The events are named according to the changes along the boundary of the uncovered region. For simplicity, we assume that no *event* values coincide for any $\theta > 0$; generic coincidences can be removed by a small perturbation of the polygon.

The **destruction** event occurs when 3 neighboring (along a cycle of the uncovered region) floodlights meet at a diagram vertex; see the red, blue and green floodlights in Fig. 6a. The boundary piece of the middle floodlight disappears, leaving the other two pieces. If an entire cycle of ∂U_θ consists of just 3 edges, then an entire uncovered region disappears at this event, as it is fully covered.

The **creation** event occurs at a vertex of the diagram, where only 2 floodlights are adjacent along ∂U_θ just before the event; see the teal and green floodlights in Fig. 6b. Along the cycle a new (red) piece appears between the two.

The **extension** event occurs when a floodlight illuminates a reflex vertex of the polygon and extends past it; see the green floodlight in Fig. 6c. The extension may pierce multiple uncovered regions, each such cut splitting a boundary cycle into two. A special case arises at aperture zero, which we handle during initialization as described later.

The **shrinking** event occurs when a floodlight illuminates a reflex vertex v of the polygon, after which its rotating boundary shortens; see the blue floodlight in Fig. 6d. The portion of the boundary along the visibility window behind v becomes stationary.

Unlike typical collapse algorithms [2,9,20], our algorithm precomputes the candidate events and uses the actual collapsing process only to assemble the

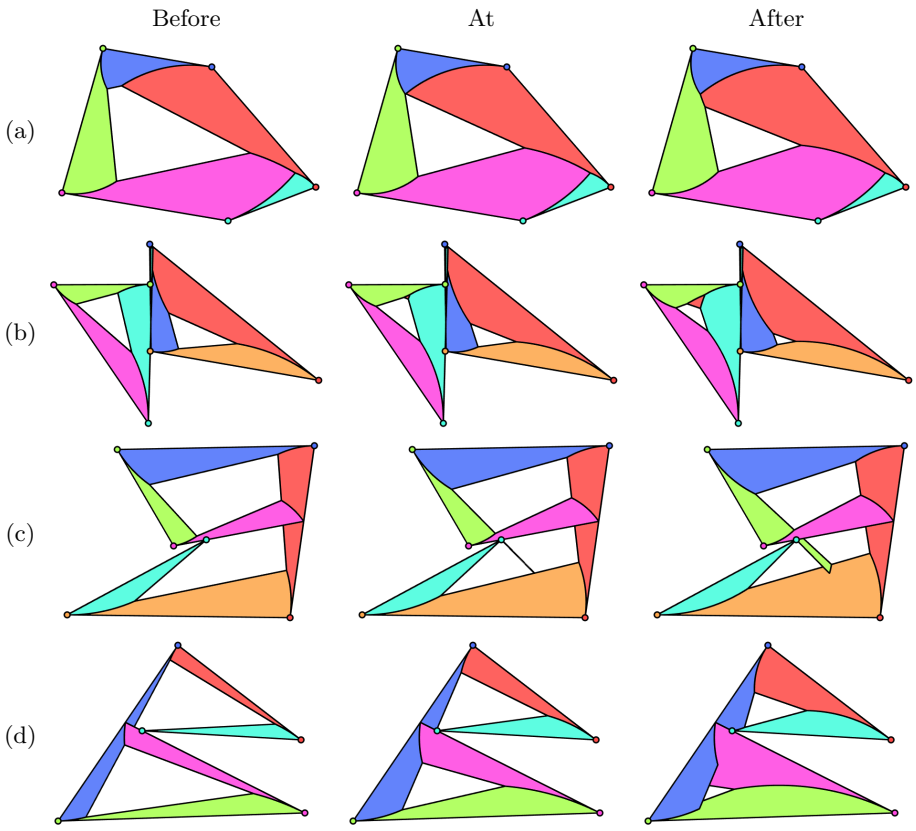


Fig. 6: Events on different polygons, shown before, at, and after the event: (a) destruction, (b) creation, (c) extension, and (d) shrinking.

diagram. This is because creation events cannot be predicted locally from the sites along the boundary of the uncovered region. The algorithm has the following outline:

1. Construct the visibility cell decomposition
2. For each visibility window w, construct the $\mathrm{PRVD}(\mathcal{S})\big|_w$ and $\mathrm{PRVD}(\mathcal{S}\backslash\{s\})\big|_w$, where s denotes the site defining w
3. Generate candidate vertices
4. Perform the collapsing procedure

In Step 2, we abuse notation: $\mathrm{PRVD}(\mathcal{S}\backslash\{s\})$ means the diagram restricted to w of the subset of sites with visibility still measured inside $\mathcal{P}$.

Preprocessing (Steps 1–3). In Step 1, the visibility cell decomposition of $\mathcal{P}$ can be constructed in $O(n^3 \log n)$ time and allows us to check whether a point $p \in \mathcal{P}$ is visible from an apex v in $O(\log n)$ time, see [7,13,15].

In Step 2, the 1-dimensional diagrams along the visibility windows can be constructed as in the proof of Theorem 10; this takes $O(n2^{\alpha(n)} \log n)$ time per window and so in total $O(n^2 r 2^{\alpha(n)} \log n)$ time. From these diagrams we can derive all the relevant information to process the extension and shrinking events.

In Step 3, we generate candidate vertices for all destruction and creation events involving no stationary edges. For each triplet of sites $s, t, u \in \mathcal{S}$, we extract the proper vertex of $\mathrm{RVD}(\{s, t, u\})$, if one exists. We retain the candidate only if it lies in the interior of $\mathcal{P}$ and outside Γ and is visible from all three sites, and save it in a set C of candidate vertices, which takes $O(\log n)$ time. Creating this candidate set C takes $O(n^3 \log n)$ time in total.

Finally, we associate with each visibility window and vertex candidate in C the aperture required for its defining sites to illuminate it, and store these in an initial event list L. Note that all visibility windows lying on input rays correspond to an aperture angle of 0; they are processed first, in any order, as extension events.

Collapse procedure (Step 4). Sort the list L by aperture angle in $O(n^3 \log n)$ time. We start with a single cycle following the polygon boundary and process all zero-aperture extension events first, obtaining the correct representation of ∂U_0. We then process the other event candidates in increasing aperture order.

If the next event candidate belongs to C, induced by sites $s, t, u \in \mathcal{S}$, we determine whether it is a **destruction** or **creation** event, or discard it. To do this, we determine which edges of $\mathrm{RVD}(\{s, t, u\})$ approach the candidate as θ increases and check whether each is traced by an active corner of two moving boundary pieces. If so, we process the corresponding destruction or creation event; otherwise, we discard the candidate. After detecting the correct event type we update ∂U_θ accordingly, i.e. remove (resp. add) a boundary piece along the cyclic list of boundary pieces representing ∂U_θ depending on whether this event is a destruction (resp. creation) event. In case of a destruction event with just 3 boundary pieces appearing along the boundary of a shrinking region, the corresponding cycle is removed completely. Checking for such active corners can be done in $O(\log n)$ time if we maintain a table of quadratic size, which records the active corners of pairs of sites together with their location in the cyclic lists. Creating this table initially takes $O(n^2)$ time, and maintaining it does not increase the asymptotic cost per event. In this table we only keep track of corners between "moving" boundary pieces, in contrast to "stationary" ones introduced later.

If the event is an **extension** event with visibility window w and corresponding site s, we need to update the cyclic lists of boundary pieces representing ∂U_θ. Several uncovered regions may be split by w, which can be read off the precomputed $\mathrm{PRVD}(\mathcal{S})\big|_w$, which corresponds to the left-sided trace of $\mathrm{PRVD}(\mathcal{P})$ on w. Note that an extension event induces stationary edges along ∂U_θ, more precisely, for the shrinking regions that are split by w and that are to the right of the rotated s. Creation and destruction events involving stationary edges can be found easily. These are exactly the vertices of the precomputed $\mathrm{PRVD}(\mathcal{S}\backslash\{s\})\big|_w$ (that is the right-sided trace of $\mathrm{PRVD}(\mathcal{P})$ on w), where we only consider the

portion that splits a shrinking region. We add these additional vertices with the corresponding priority to the sorted queue L. Given the precomputed diagrams along w, splitting the affected regions adds no further asymptotic cost. Identifying the additional events and adding them to a sorted list takes $O(n2^{\alpha(n)} \log n)$ time per extension event, of which there are $O(nr)$ many.

If the event is a **shrinking** event with visibility window w and corresponding site s, the portion of ∂U_θ corresponding to w and s becomes stationary. As with extension events, we need to add the events involving stationary edges on w to the event list L. The portion of $\mathrm{PRVD}(\mathcal{S})\big|_w$, where site s is dominant, corresponds exactly to the new stationary edges. Within this portion, the new events are the vertices of $\mathrm{PRVD}(\mathcal{S}\backslash\{s\})\big|_w$. Reading off the new events from $\mathrm{PRVD}(\mathcal{S}\backslash\{s\})\big|_w$ and adding them to the event queue L takes $O(n2^{\alpha(n)} \log n)$ time per shrinking event, of which there are $O(nr)$ many.

After having handled one event, we continue processing the next larger aperture event, and stop when the last uncovered boundary cycle disappears. The correctness of the algorithm relies on its inductive nature. After processing the zero-aperture events, we have a correct representation of ∂U_0. At each event candidate related to aperture θ, we assume that we have a correct representation of ∂U_θ just before the event. We then update ∂U_θ if that event candidate turns out to be a real event, or discard it otherwise, to maintain a correct representation of ∂U_θ after the event.

4.3 α^*-Floodlight Illumination

We return to the illumination problem of Definition 5 for $\mathcal{P}$. Using the notation of Section 4.2, we have $\alpha^* = \sup_{x\in\mathcal{P}} d(x)$. Note that the algorithm of [5] runs in $O(n^3 \log^2 n)$ time, but it is incomplete: it omits some cases in which α^* is approached at a visibility window; see Fig. 7.

Theorem 13. *Given a simple polygon $\mathcal{P}$, the angle α^* can be computed in $O((n^3 + n^2 r\, 2^{\alpha(n)}) \log n)$ time.*

Proof. Run the collapse algorithm of Theorem 12, and let $\bar{\theta}$ be the aperture at which the last uncovered boundary cycle disappears. Before $\bar{\theta}$, a boundary cycle exists, so $U_\theta \neq \varnothing$ for $\theta < \bar{\theta}$. After $\bar{\theta}$ no boundary cycle exists, so $U_\theta = \varnothing$ for $\theta > \bar{\theta}$, and $U_{\bar{\theta}} = \varnothing$, because a point with $d(x) > \bar{\theta}$ lies in U_θ for some $\theta > \bar{\theta}$. Since $\alpha^* = \sup_{x\in\mathcal{P}} d(x)$, we get $\alpha^* = \bar{\theta}$. The running time is that of Theorem 12.

Proposition 14. *The range of α^* over simple polygons is $(0, 2\pi)$.*

Proof (sketch). Fix $\alpha \in (0, 2\pi)$. Choose a counterclockwise triangle $T = A_1A_2A_3$ and an interior point M such that $\angle A_{i+1}A_iM < \alpha$ and $\angle MA_iA_{i-1} < 2\pi - \alpha$ for $i = 1, 2, 3$, with indices modulo 3. For example, take the appropriate Brocard point of a sufficiently flat T. Replace each edge A_iA_{i+1} by a small wing $A_iB_iC_iA_{i+1}$ outside T, directing A_iB_i so that its angular distance to M is α. By the assumed inequalities, this direction points out of T. Direct $\overrightarrow{A_iC_i}$ clockwise

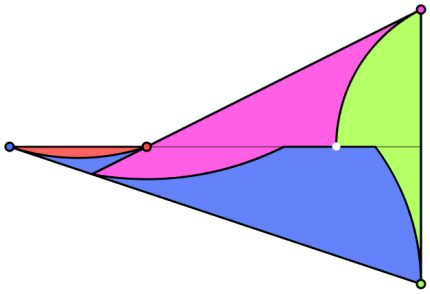


Fig. 7: α^* approached at a window.

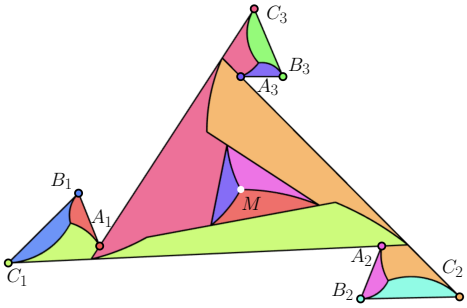


Fig. 8: Polygon with $\alpha^* = 3\pi/2$.

from $\overrightarrow{MA_i}$ by an angle smaller than $\angle A_{i+1}A_iM$ and, if $\alpha > \pi$, also smaller than $\alpha - \pi$; then place C_i sufficiently close to A_i and B_i much closer still. Each auxiliary site either does not see M or has angular distance to M greater than α; see Fig. 8 for $\alpha = 3\pi/2$. The sites at A_1, A_2, A_3 see M at distance α, so $\alpha^* \geqslant \alpha$.

By the first inequality, the site at A_i illuminates $A_iA_{i+1}M$, and these triangles cover T. The sites at A_i and C_i cover the wings, giving $\alpha^* \leqslant \alpha$. Since $0 < \alpha^* < 2\pi$ for every simple polygon, the claimed range follows.

5 Concluding Remarks

We studied the oriented rotating rays Voronoi diagram in two settings. For the diagram in the plane, we narrowed the remaining gap in both combinatorial complexity and construction time. For edge-aligned rays in simple polygons, we gave a systematic study of the diagram that improves all known bounds.

Several questions remain open. In $\mathbb{R}^2$, an $O(\text{polylog}\, n)$ factor still separates the new upper bounds from the worst-case $\Omega(n^2)$ lower bound. In polygons, there is a considerable gap between the $\Omega(nr)$ lower bound and the $O(n^3 + n^2 r\, 2^{\alpha(n)})$ upper bound. It would be interesting to identify polygon classes of intermediate complexity, given the jump from the $\Theta(n)$ bound for convex polygons. A systematic study of the unoriented diagram in the plane has been left largely open. Finally, lower bounds for directly computing α^* would also be of interest, as this would indicate whether constructing the diagram first remains the best option.

References

1. Agarwal, P.K., Ezra, E., Sharir, M.: Lower envelopes of surface patches in 3-space. In: 32nd Annual European Symposium on Algorithms (ESA 2024). pp. 6:1–6:17. Schloss Dagstuhl–Leibniz-Zentrum für Informatik (2024)
2. Alegría, C., Mantas, I., Papadopoulou, E., Savić, M., Seara, C., Suderland, M.: The Voronoi diagram of rotating rays with applications to floodlight illumination. Algorithmica **88**(2), 27 (2026)

3. Alegría, C., Mantas, I., Savić, M., Suderland, M.: Interactive uniform floodlight illumination and rotating rays Voronoi diagrams (media exposition). In: 42nd International Symposium on Computational Geometry (SoCG 2026). pp. 98:1–98:7. Schloss Dagstuhl–Leibniz-Zentrum für Informatik (2026)
4. Alegría, C., Mantas, I., Savić, M., Suderland, M.: RVD-explorer (Software). Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2026), `https://github.com/VD-collective/rvd-explorer`
5. Alegría, C., Orden, D., Seara, C., Urrutia, J.: Illuminating polygons by edge-aligned floodlights of uniform angle (Brocard illumination). In: Proceedings of the 33rd European Workshop on Computational Geometry (EuroCG 2017). pp. 281–284 (2017)
6. Aronov, B., Drusvyatskiy, D.: Complexity of a single face in an arrangement of s-intersecting curves. arXiv preprint arXiv:1108.4336 (2011)
7. Aronov, B., Guibas, L.J., Teichmann, M., Zhang, L.: Visibility queries and maintenance in simple polygons. Discrete & Computational Geometry **27**(4), 461–483 (2002)
8. Arslan, O., Min, H., Koditschek, D.E.: Voronoi-based coverage control of pan/tilt/zoom camera networks. In: 2018 IEEE International Conference on Robotics and Automation (ICRA 2018). pp. 5062–5069. IEEE (2018)
9. Aurenhammer, F., Drysdale, R.L.S., Krasser, H.: Farthest line segment Voronoi diagrams. Information Processing Letters **100**(6), 220–225 (2006)
10. Aurenhammer, F., Klein, R., Lee, D.T.: Voronoi Diagrams and Delaunay Triangulations. World Scientific (2013)
11. de Berg, M., Gudmundsson, J., Haverkort, H., Horton, M.: Voronoi diagrams with rotational distance costs. In: Abstracts of the Computational Geometry Week: Young Researchers Forum (CG:YRF) (2017)
12. Berman, P., Jeong, J., Kasiviswanathan, S.P., Urgaonkar, B.: Packing to angles and sectors. In: Proceedings of the 19th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'07). pp. 171–180 (2007)
13. Bose, P., Lubiw, A., Munro, J.I.: Efficient visibility queries in simple polygons. Computational Geometry **23**(3), 313–335 (2002)
14. Dai, H., Wang, X., Liu, A.X., Ma, H., Chen, G.: Optimizing wireless charger placement for directional charging. In: IEEE INFOCOM 2017-IEEE Conference on Computer Communications. pp. 1–9. IEEE (2017)
15. Guibas, L.J., Motwani, R., Raghavan, P.: The robot localization problem. SIAM Journal on Computing **26**(4), 1120–1138 (1997)
16. Haverkort, H., Klein, R.: Hyperbolae are the locus of constant angle difference. arXiv preprint arXiv:2112.00454 (2021)
17. Mantas, I.: Problems on Planar Voronoi Diagrams. Ph.D. thesis, Università della Svizzera Italiana (2022)
18. Okabe, A., Boots, B., Sugihara, K., Chiu, S.N.: Spatial tessellations: concepts and applications of Voronoi diagrams, vol. 501. John Wiley & Sons (2009)
19. O'Rourke, J.: Visibility. In: Handbook of Discrete and Computational Geometry, pp. 875–896. CRC Press (2017)
20. Papadopoulou, E., Suderland, M., Wang, Z.: A collapse process for farthest Voronoi diagrams of lines in three dimensions. arXiv preprint arXiv:2609.29865 (2026)
21. Sung, T.W., Yang, C.S.: Voronoi-based coverage improvement approach for wireless directional sensor networks. Journal of Network and Computer Applications **39**, 202–213 (2014)
22. Urrutia, J.: Art gallery and illumination problems. In: Handbook of Computational Geometry, pp. 973–1027. Elsevier (2000)