Farthest-polygon Voronoi diagrams

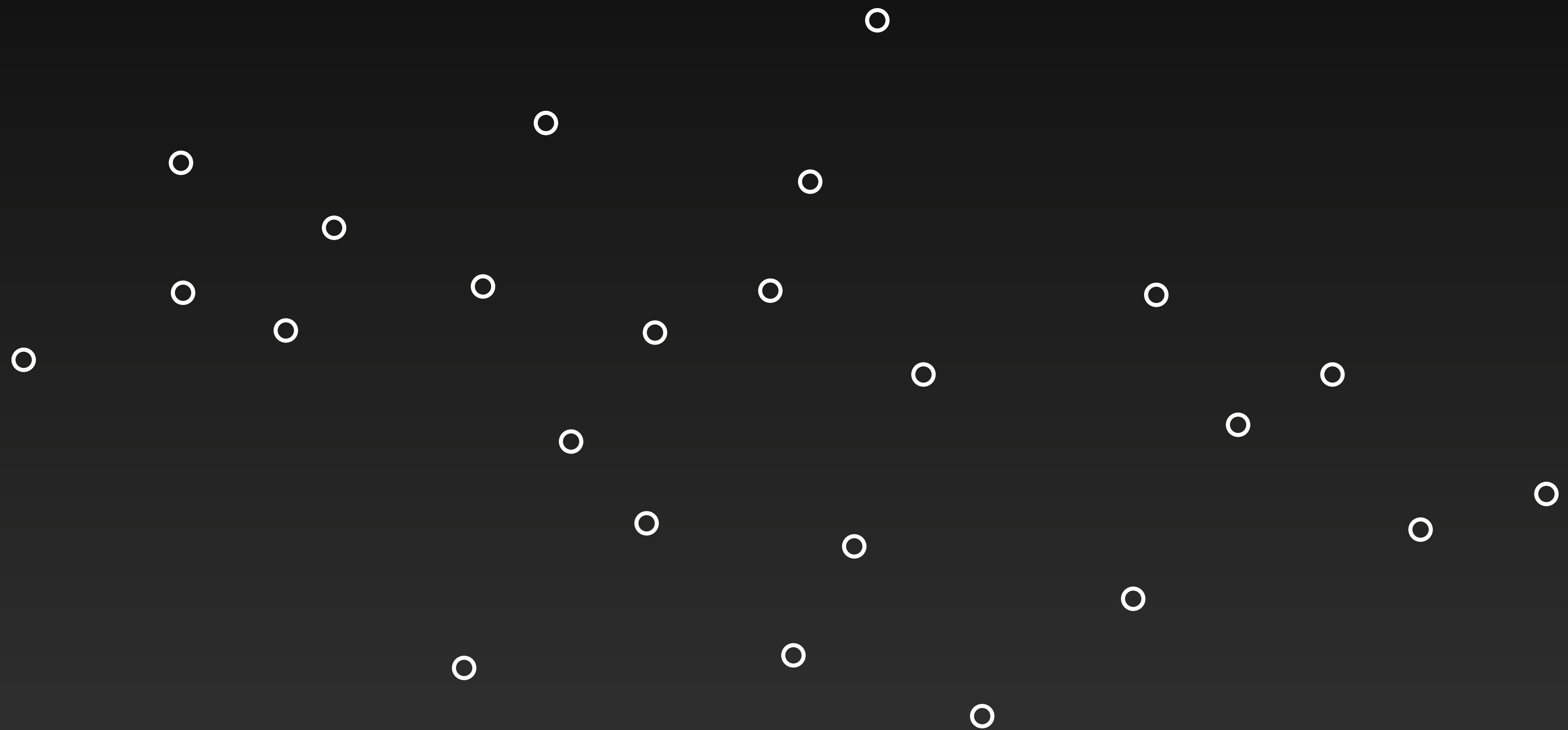
Otfried Cheong, Hazel Everett, Marc Glisse,
Joachim Gudmundsson, Samuel Hornus,
Sylvain Lazard, Mira Lee and Hyeon-Suk Na

ESA – October 2007

KAIST, INRIA, NICTA, Soongsil U.

Voronoi diagrams

Given some sites (points) in $\mathbb{R}^2$, the closest-point Voronoi diagram partitions the plane in convex regions, in each of which the closest site is the same.



Voronoi diagrams

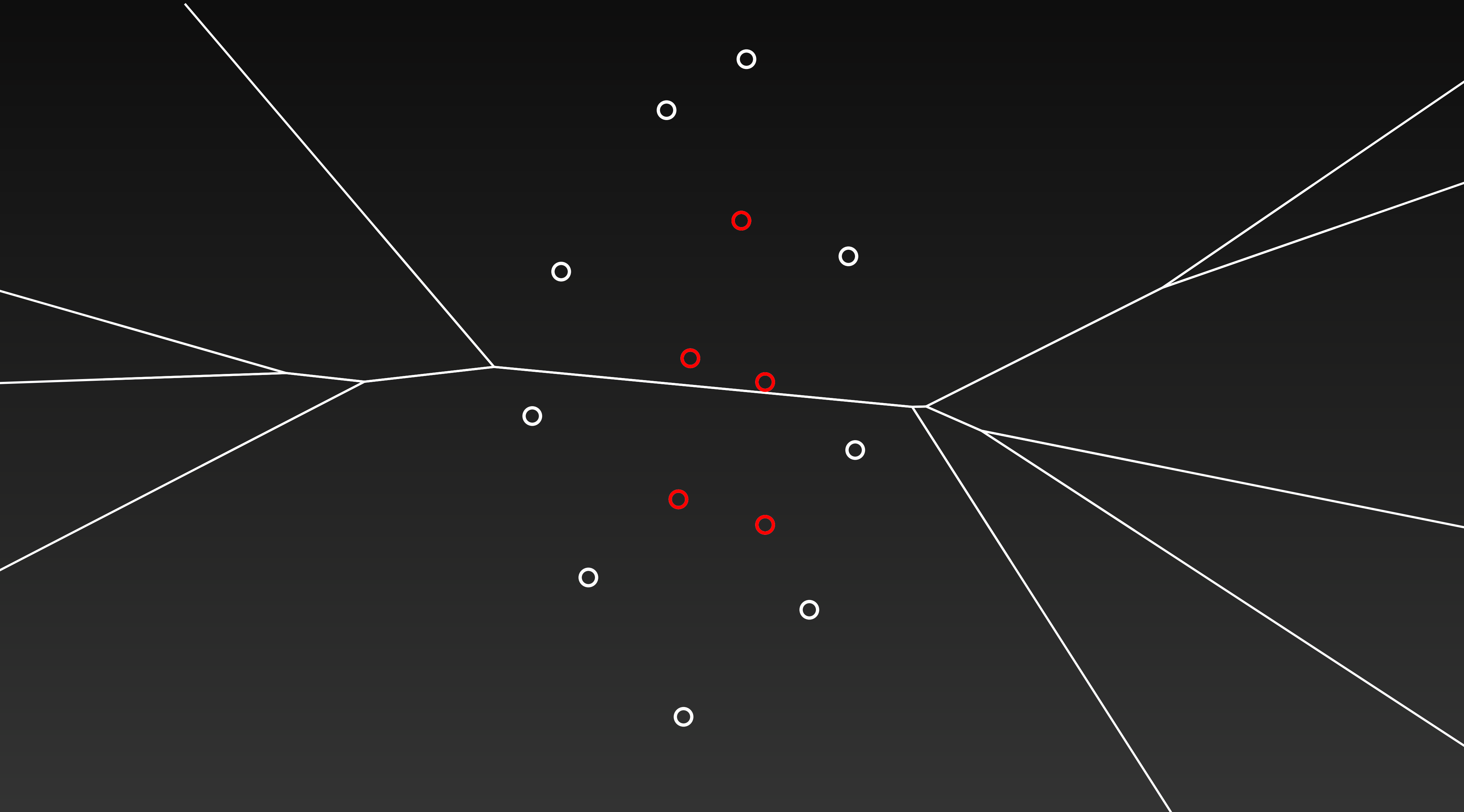
Given some sites (points) in $\mathbb{R}^2$, the closest-point Voronoi diagram partitions the plane in convex regions, in each of which the closest site is the same.

Voronoi diagrams

The farthest-point Voronoi diagram partitions the plane in convex regions, in each of which the farthest site is the same.

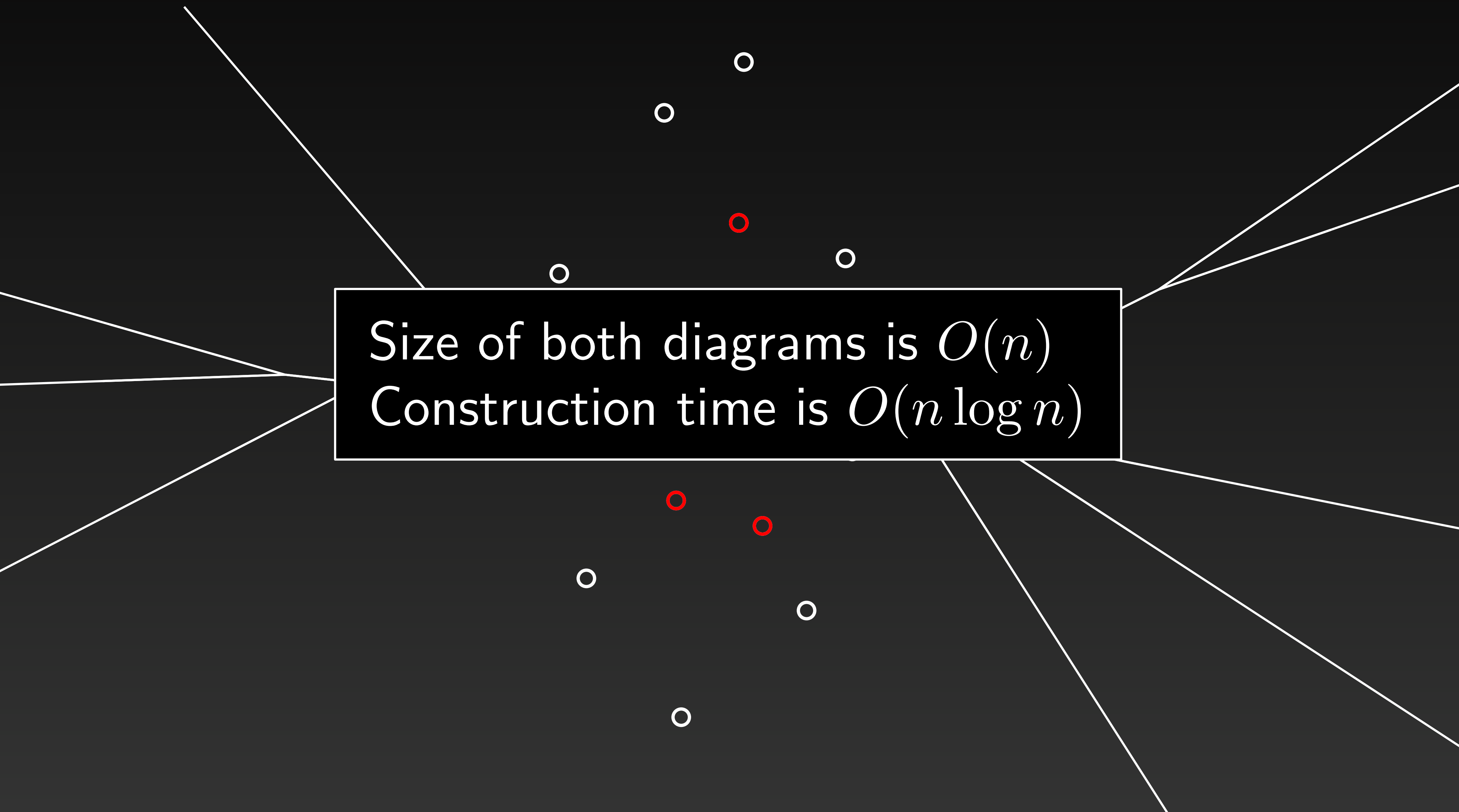
Voronoi diagrams

The farthest-point Voronoi diagram partitions the plane in convex regions, in each of which the farthest site is the same.



Voronoi diagrams

The farthest-point Voronoi diagram partitions the plane in convex regions, in each of which the farthest site is the same.



Size of both diagrams is $O(n)$
Construction time is $O(n \log n)$

Voronoi diagrams

Closest- Voronoi diagrams have been extended to different type of sites, including

- weighted points
- line segments
- couloured points
- polygons
- etc. . .

What about farthest-site Voronoi diagrams ?

Farthest-polygon Voronoi diagrams

k sets of disjoint line segments (n total):

Farthest-polygon Voronoi diagrams

k sets of disjoint line segments (n total):

Farthest-site Voronoi diagram

$\approx$

upper envelope of *(closest-site) Voronoi surfaces*,

Farthest-polygon Voronoi diagrams

k sets of disjoint line segments (n total):

Farthest-site Voronoi diagram

$\approx$

upper envelope of *(closest-site) Voronoi surfaces*,

which is know to have complexity $\Theta(nk)$

[Huttenlocher *et al.* 93].

New: when the line segments form k disjoint polygons, the complexity drops to $O(n)$.

Farthest-polygon Voronoi diagrams

. . . or FPolyVD, for short

Contribution:

Given k pairwise disjoint, connected simplicial complexes with total complexity n:

1. The FPolyVD has complexity $O(n)$.
2. It can be constructed in $O(n \log^3 n)$ expected time.

Applications

$O(\log n)$-time *farthest polygon* query for points
With additional $O(n \log n)$ preprocessing.

"Optimal" antenna placement
After the farthest-polygon Voronoi diagram is built, we can find, in linear time, the optimal placement of an antenna with minimum power reaching a given set of sites (e.g. cities, districts).

Applications

$O(\log n)$-time *farthest polygon* query for points
With additional $O(n \log n)$ preprocessing.

"Optimal" antenna placement
After the farthest-polygon Voronoi diagram is built, we can find, in linear time, the optimal placement of an antenna with minimum power reaching a given set of sites (e.g. cities, districts).

Definitions

Input: n line segments forming a family $\mathcal{P}$ of k 1D simplicial complexes ("polygons"). $|\mathcal{P}| = k$ and $\sum_{P \in \mathcal{P}} |P| = n$.

Definitions

Input: n line segments forming a family $\mathcal{P}$ of k 1D simplicial complexes ("polygons"). $|\mathcal{P}| = k$ and $\sum_{P \in \mathcal{P}} |P| = n$.

The point-polygon distance is the usual euclidean one.

Definitions

Input: n line segments forming a family $\mathcal{P}$ of k 1D simplicial complexes ("polygons"). $|\mathcal{P}| = k$ and $\sum_{P \in \mathcal{P}} |P| = n$.

The point-polygon distance is the usual euclidean one.

The region $R(P)$ of polygon P is the set of points farther from P than from any other polygon in $\mathcal{P}$.

Definitions

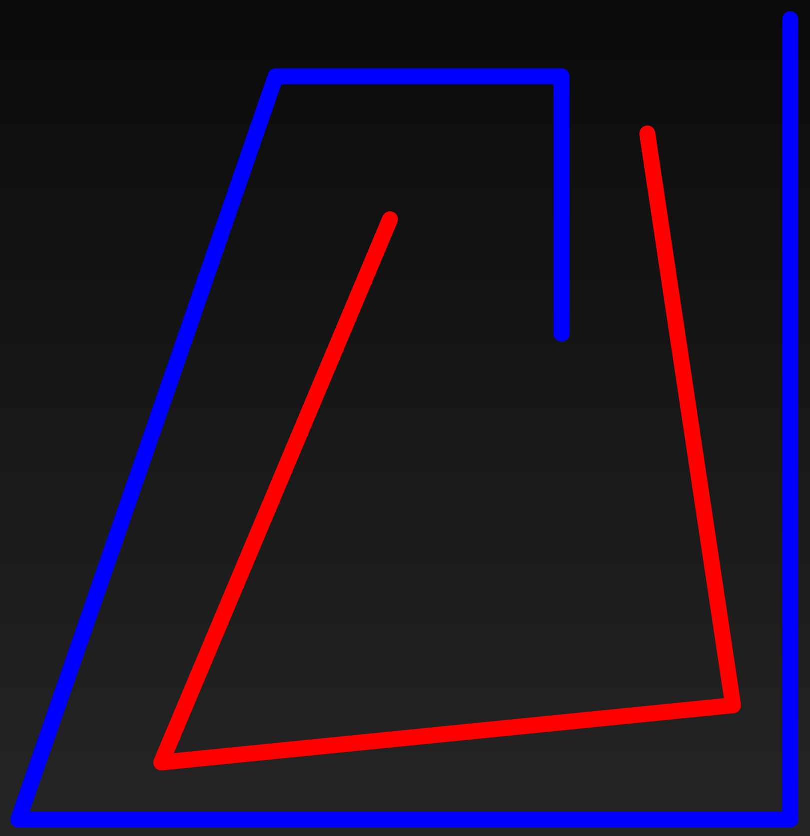
Input: n line segments forming a family $\mathcal{P}$ of k 1D simplicial complexes ("polygons"). $|\mathcal{P}| = k$ and $\sum_{P \in \mathcal{P}} |P| = n$.

The point-polygon distance is the usual euclidean one.

The region $R(P)$ of polygon P is the set of points farther from P than from any other polygon in $\mathcal{P}$.

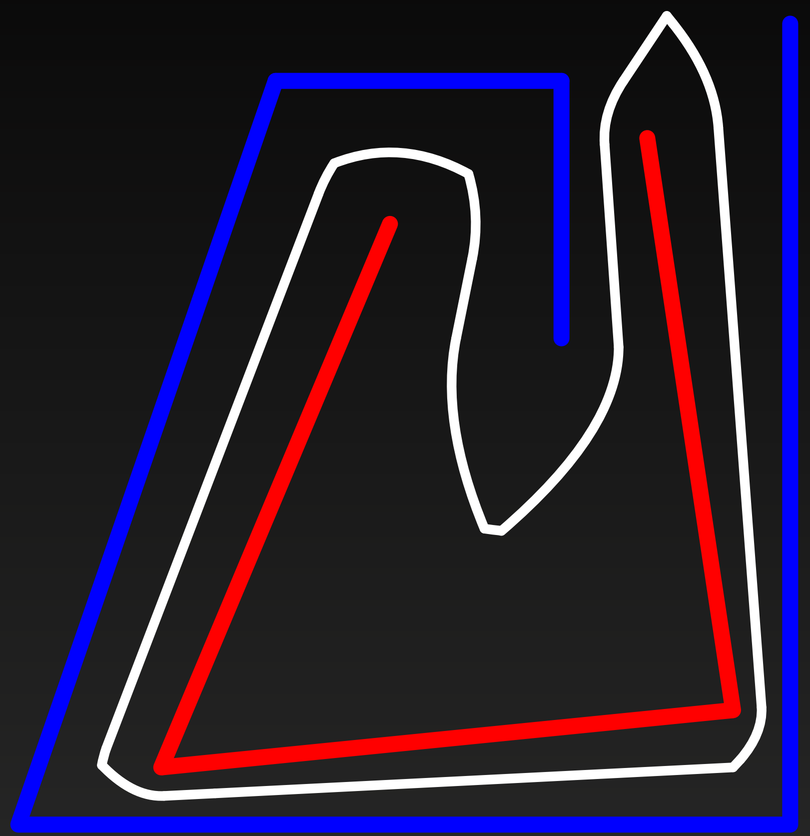
Further subdivide $R(P)$ into cells by cutting $R(P)$ along the medial axis of P.

Example



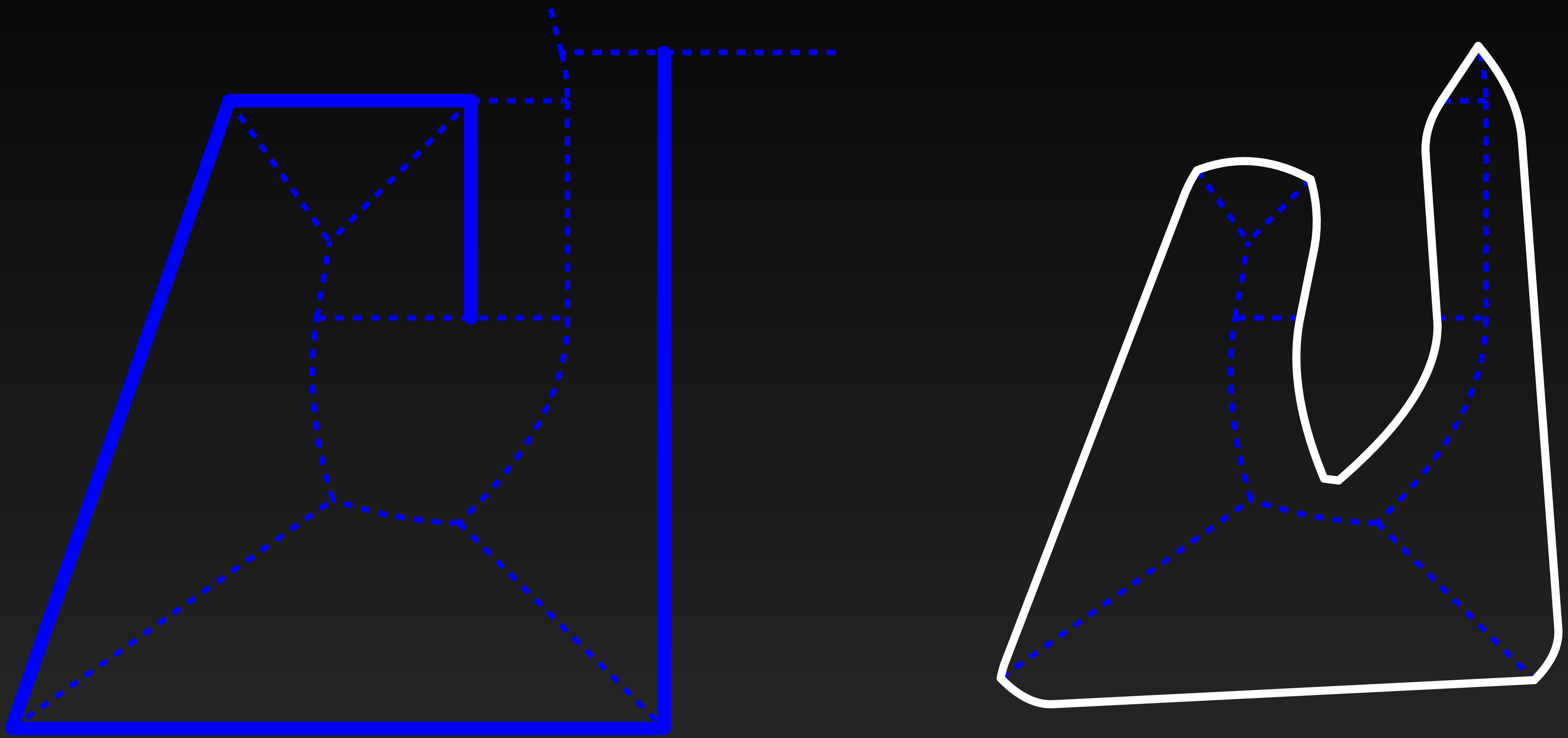
two polygons

Example



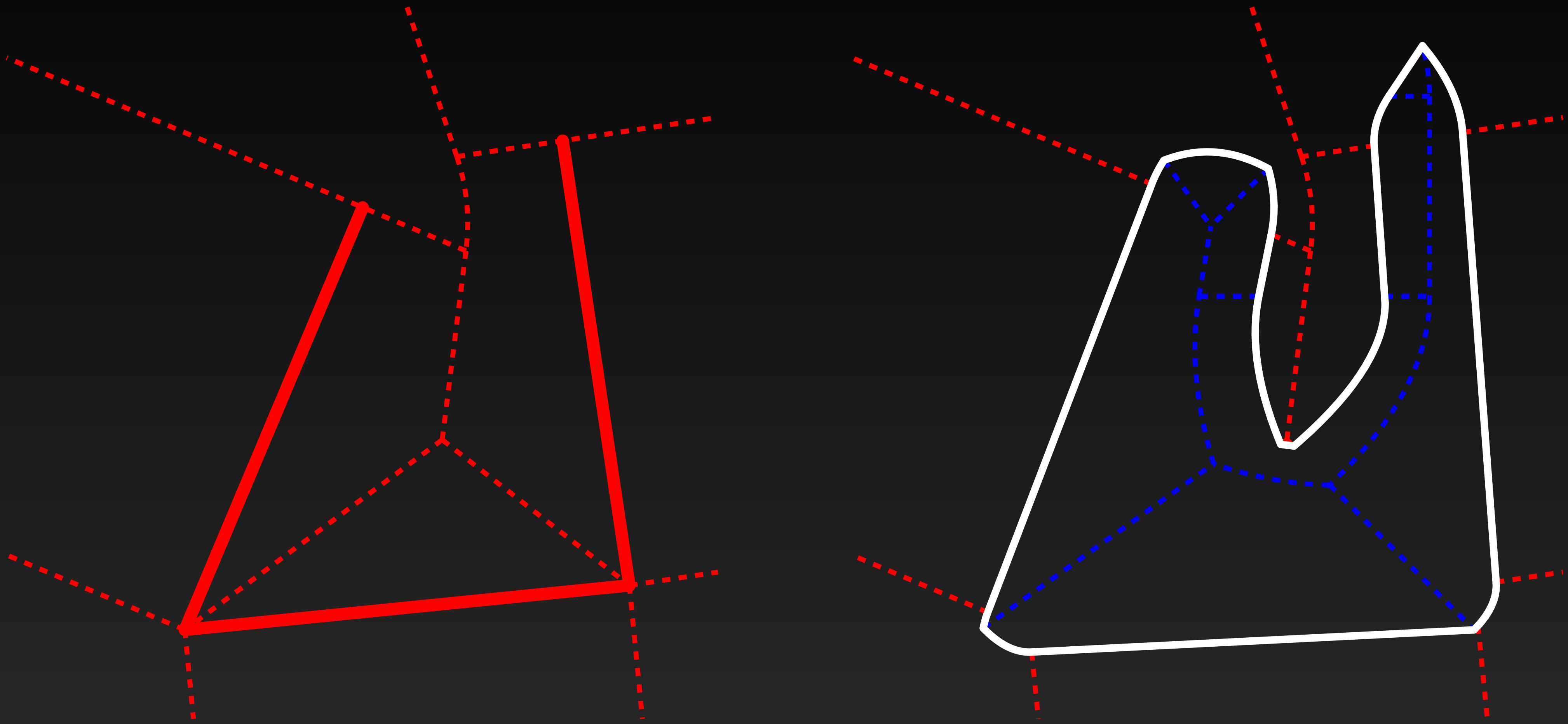
bisector

Example



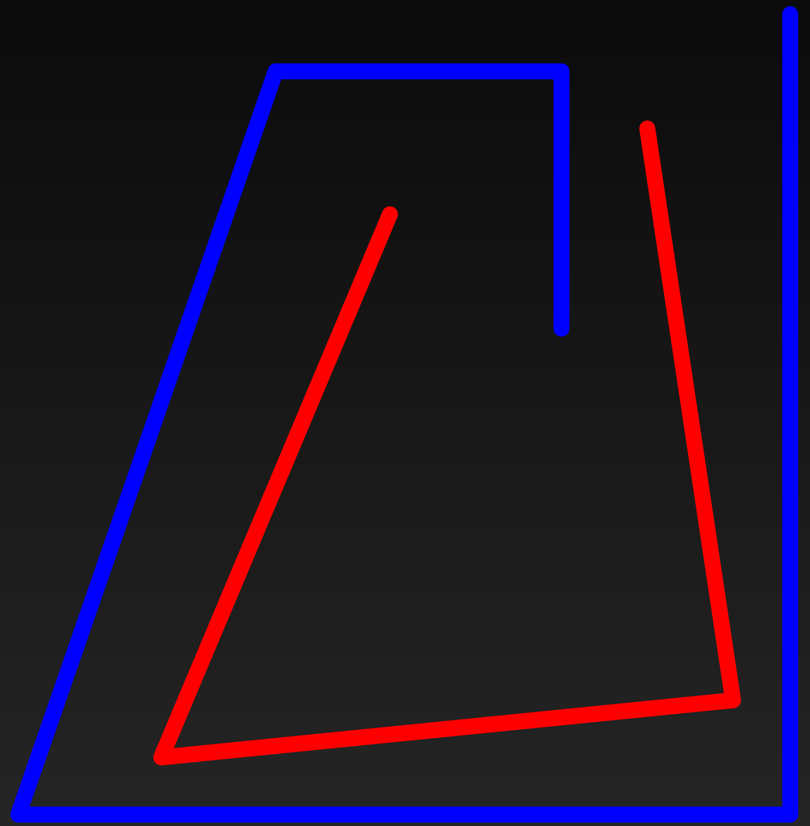
cutting the blue region with the blue medial axis

Example

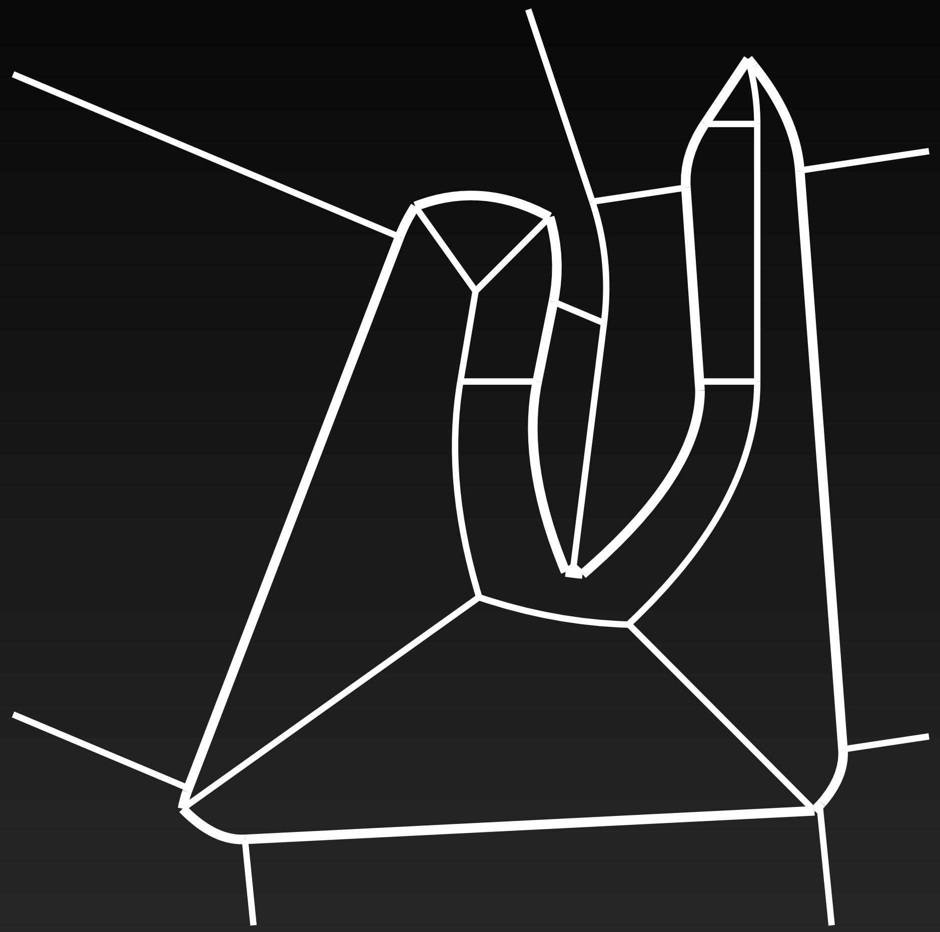


cutting the red region with the red medial axis

Example

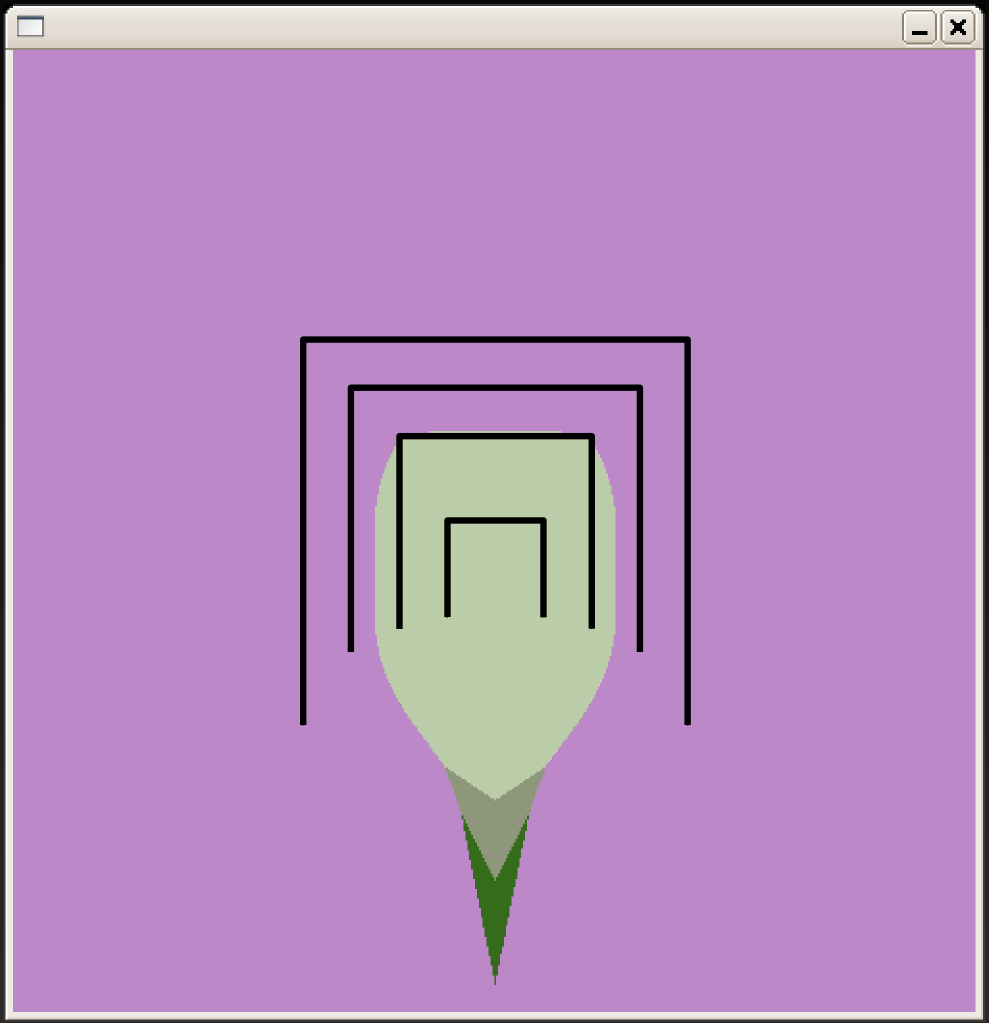


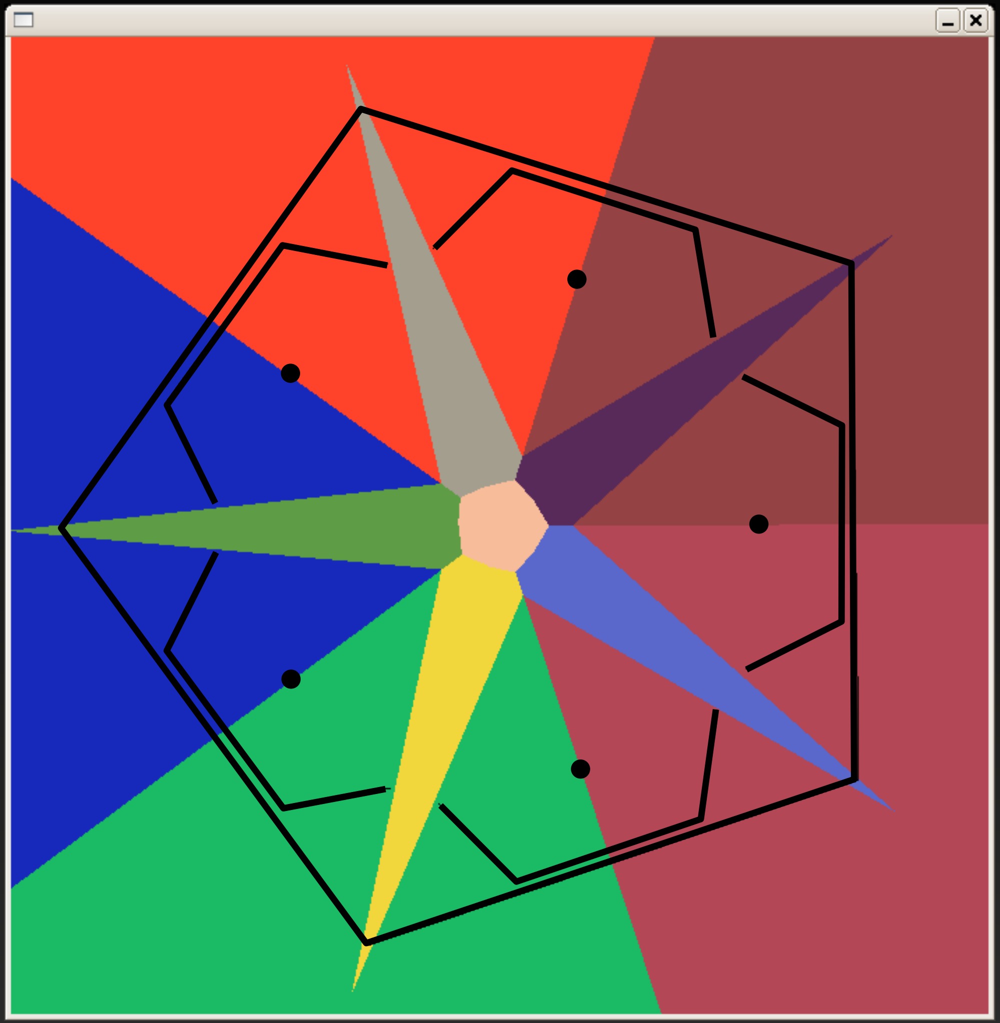
two polygons and. . .



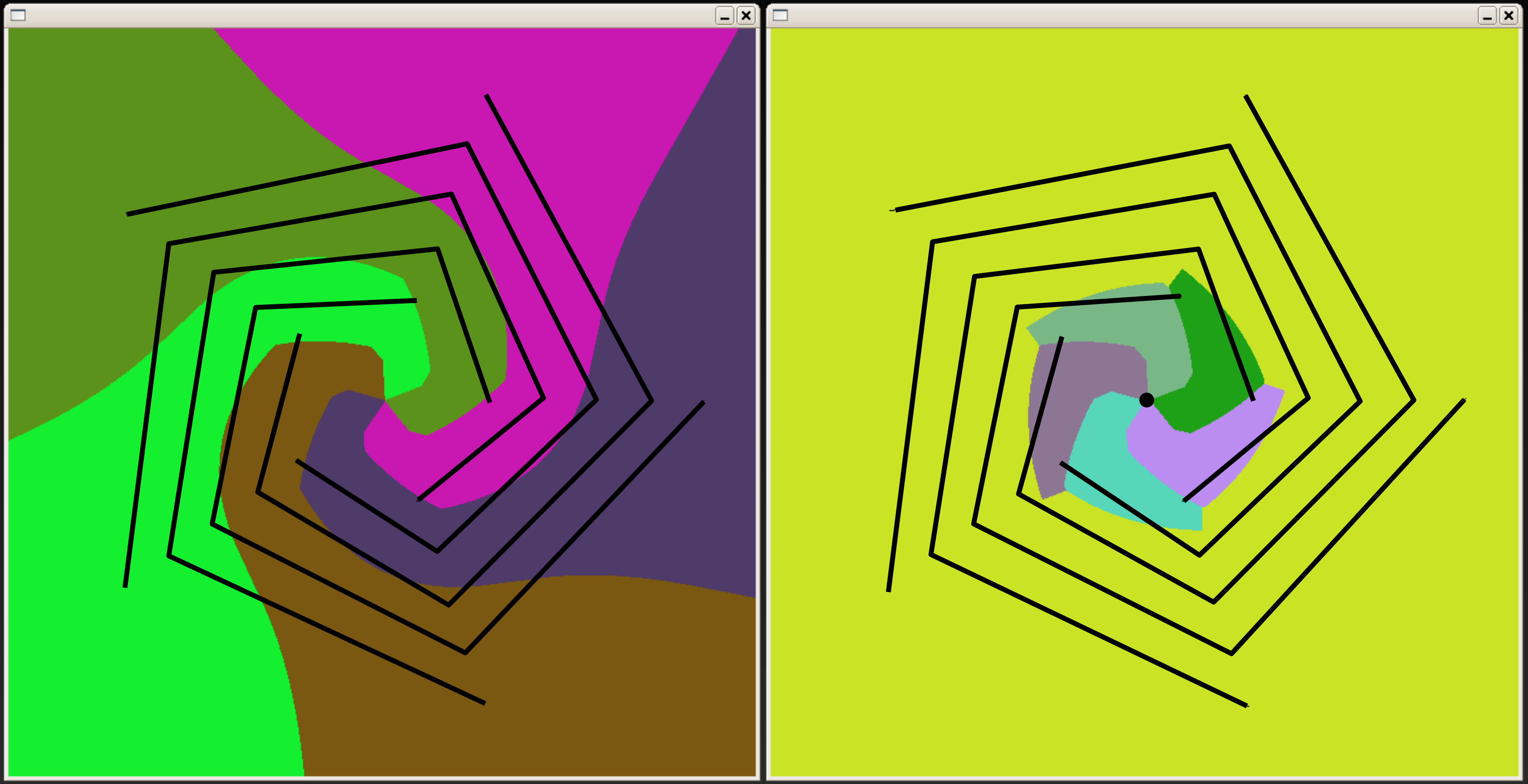
. . . their farthest-polygon Voronoi diagram

Illustrations





Illustrations

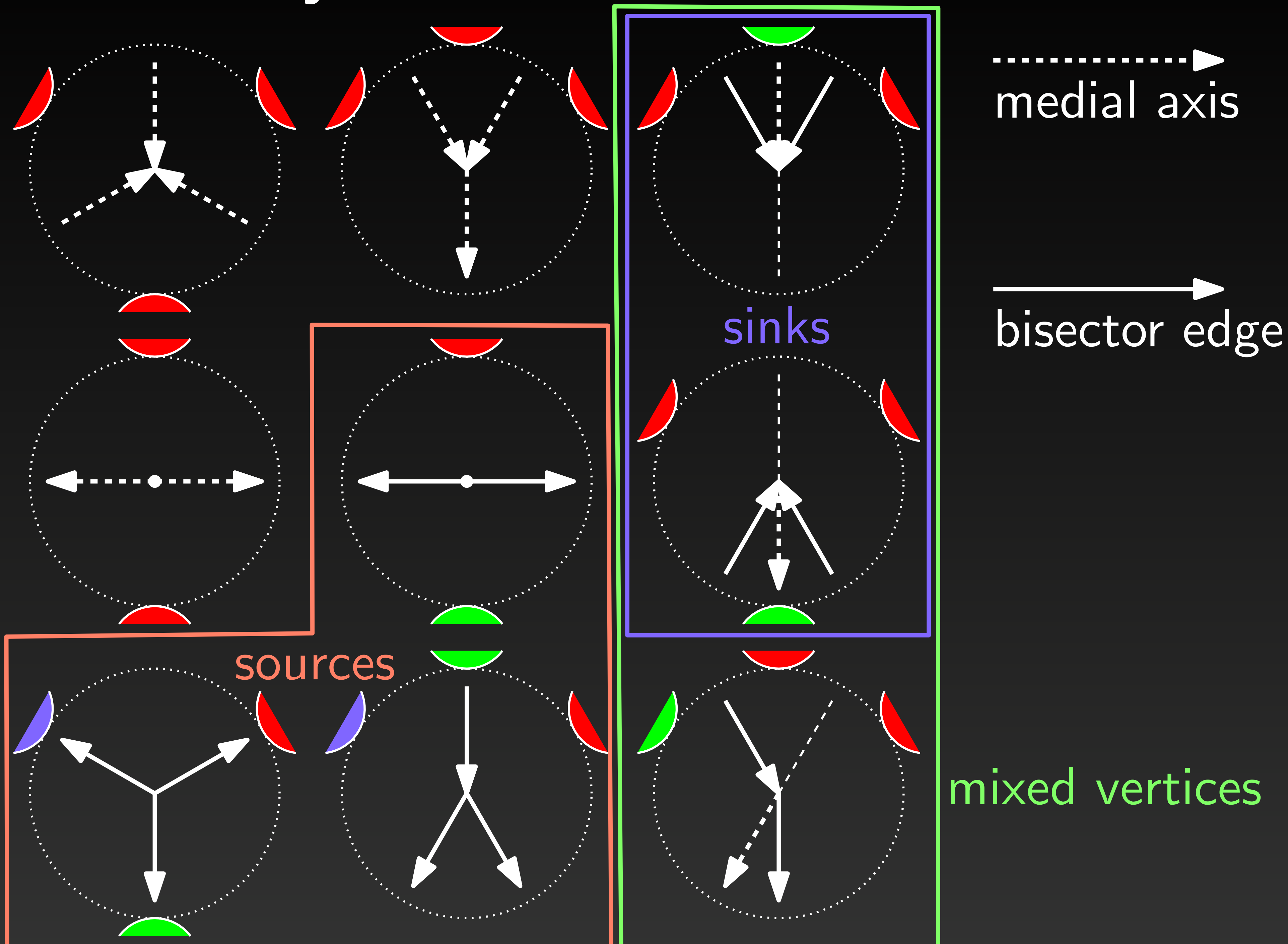


The FPolyVD has linear size

$\phi(x) =$ distance from x to its farthest polygon

1. Orient the edges of the FPolyVD along the gradient of ϕ.
2. Partition the edges into maximal oriented paths.
3. All vertices are source or sink.
4. (vertices at infinity are sinks.)
5. Bound the number of sinks.

The FPolyVD's vertices

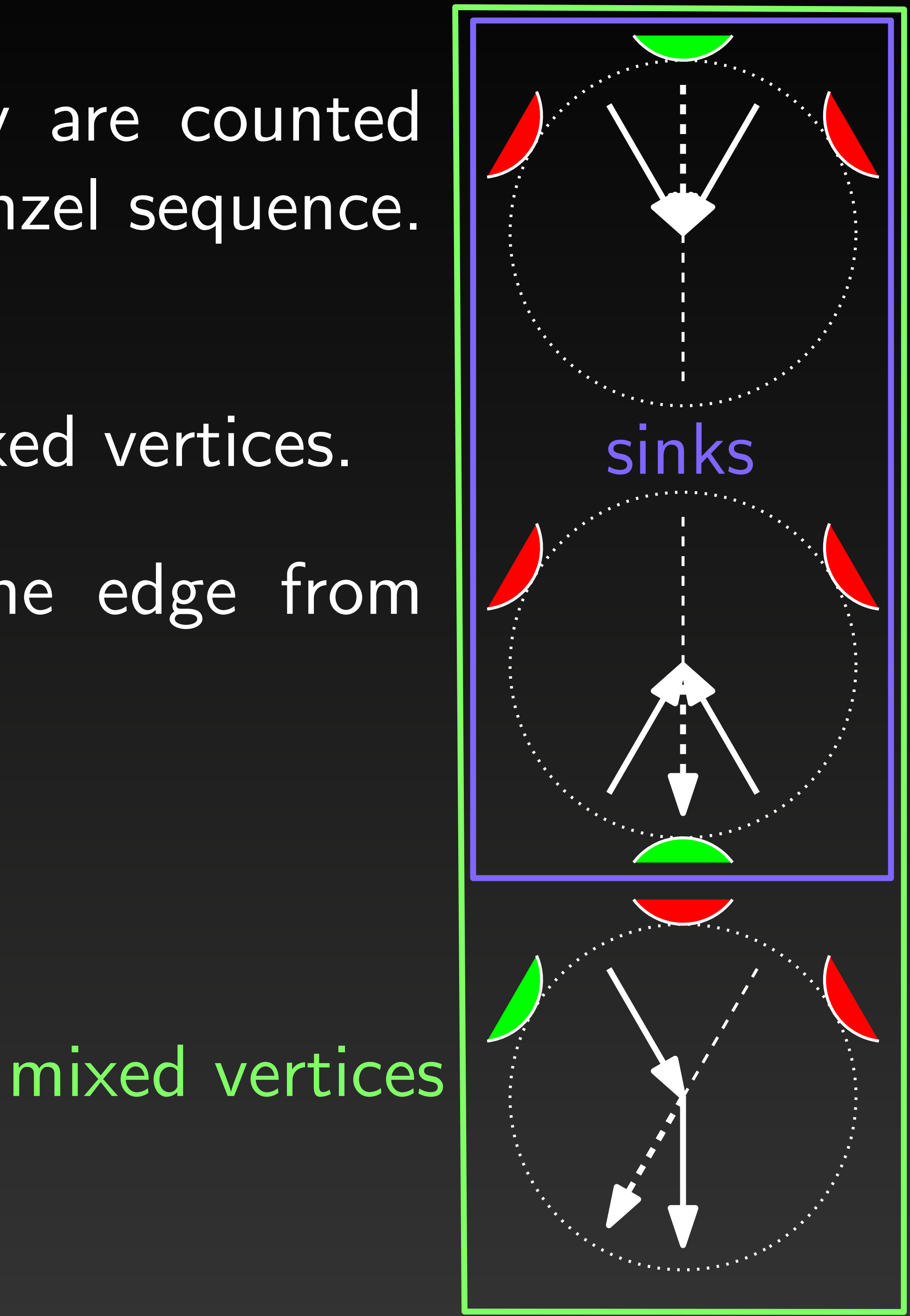


The FPolyVD's vertices

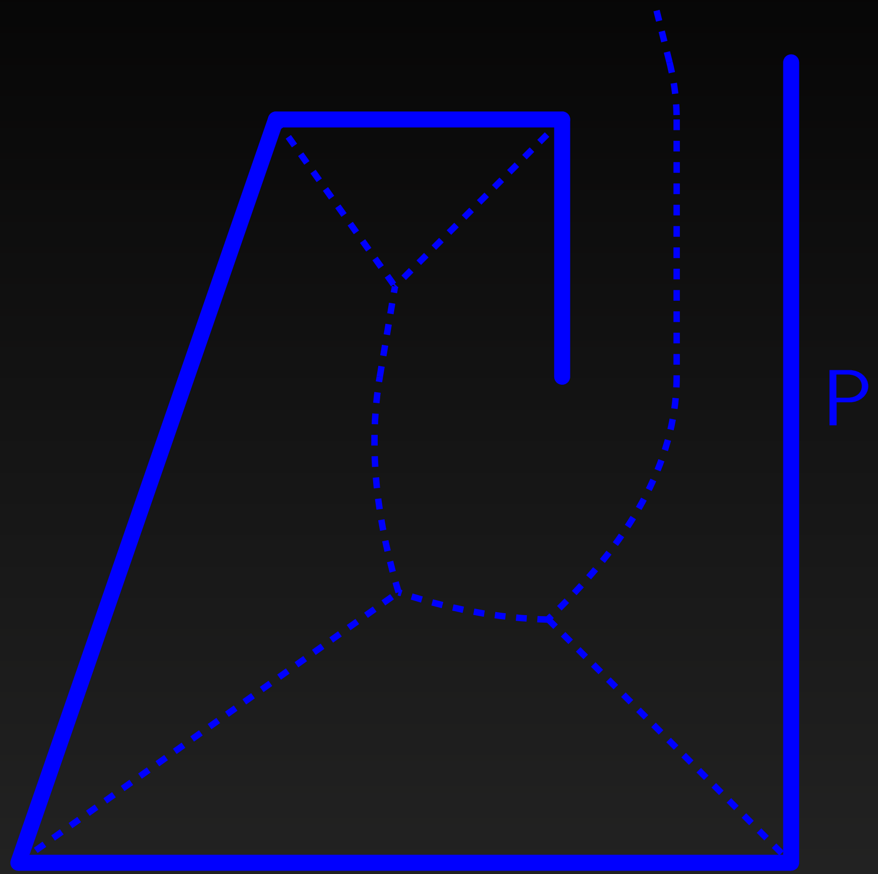
1. Sink vertices at infinity are counted using a Davenport-Schinzel sequence. Their number is linear.

2. It remains to bound mixed vertices.

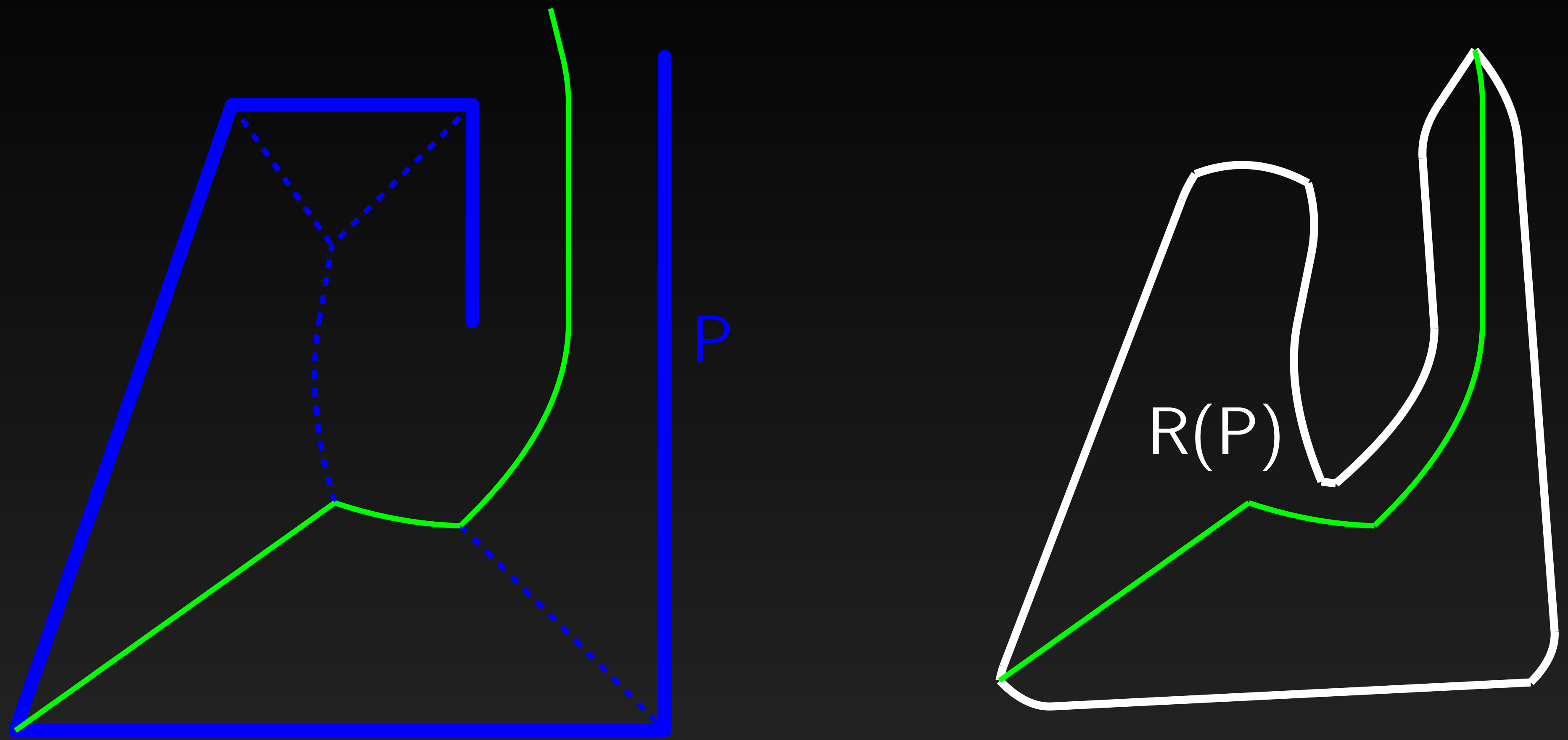
3. A mixed vertex has one edge from some medial axis. . .



The FPolyVD has linear size



The FPolyVD has linear size



Lemma. Any path in medial axis of P, intersects $R(P)$ in a connected path

The FPolyVD has linear size

Lemma

Any path in medial axis of P, intersects $R(P)$ in a connected path.

Corollary

The medial axis of P intersects $R(P)$ in a connected tree. which has a linear number of leaves (the mixed vertices).

Corollary

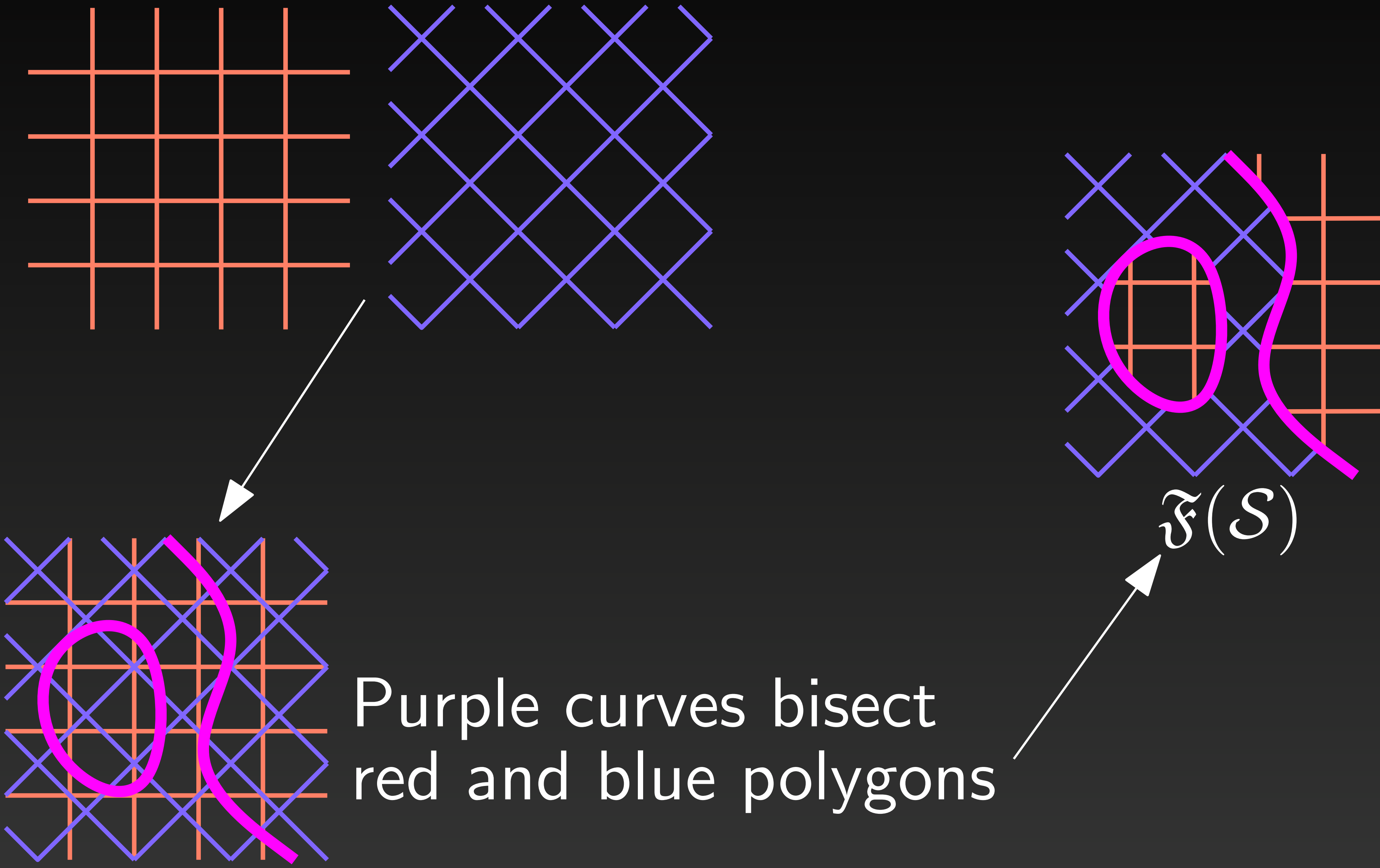
The number of mixed vertices is linear.

Conclusion

The FPolyVD has linear size.

Construction of the FPolyVD

$\mathfrak{F}(\mathcal{S}_1)$ and $\mathfrak{F}(\mathcal{S}_2)$ are constructed recursively



$\mathfrak{F}(\mathcal{S})$

Purple curves bisect red and blue polygons

Construction of the FPolyVD

Divide and conquer algorithm:

- Split $\mathcal{P}$ into $\mathcal{P}_1 \sqcup \mathcal{P}_2$ of roughly equal size
- Compute $\mathcal{F}(\mathcal{P}_i)$, $i = 1, 2$, recursively
- Merge $\mathcal{F}(\mathcal{P}_1)$ and $\mathcal{F}(\mathcal{P}_2)$

The merging step takes $O(|\mathcal{P}| \log^2 |\mathcal{P}|)$ time...

$\Rightarrow$ total time complexity is $O(n \log^3 n)$.

Thank you