Linear time $8/3$-approx. of r-star guards in simple orthogonal art galleries

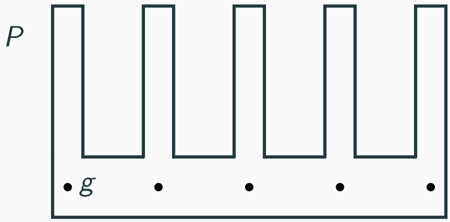
Ervin Győri & Tamás Róbert Mezei[1]

ICGT 2018, July 9, Lyon

[1]Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences

The Art Gallery problem - for Orthogonal Polygons

- Art gallery: $P \subset \mathbb{R}^2$, a simple orthogonal polygon
- Point guard: fixed point $g \in P$, has 360° line of sight vision
- Objective: place guards in the gallery so that any point in P is seen by at least one of the guards



The art gallery theorem for orthogonal polygons

Theorem (Kahn, Klawe and Kleitman, 1980)

$\left\lfloor \frac{n}{4} \right\rfloor$ guards are sometimes necessary and always sufficient to cover the interior of a simple orthogonal polygon of n vertices.

The art gallery theorem for orthogonal polygons

Theorem (Kahn, Klawe and Kleitman, 1980)

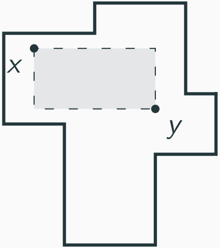
$\left\lfloor \frac{n}{4} \right\rfloor$ guards are sometimes necessary and always sufficient to cover the interior of a simple orthogonal polygon of n vertices.

Theorem (Schuchardt and Hecker, 1995)

Finding a minimum size point guard system is *NP*-hard in simple orthogonal polygons

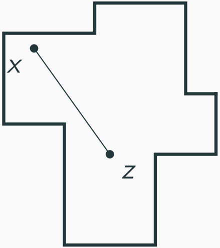
Rectangular or r-vision

Rectangular vision: two points $x, y \in P$ have r-vision of each other if there is an axis-parallel rectangle inside P, containing x and y.



Rectangular or r-vision

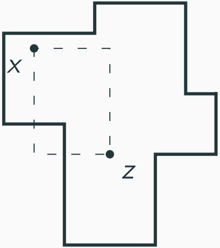
Rectangular vision: two points $x, y \in P$ have r-vision of each other if there is an axis-parallel rectangle inside P, containing x and y.



x and z have unrestricted vision of each other

Rectangular or r-vision

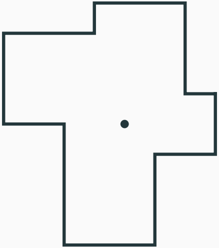
Rectangular vision: two points $x, y \in P$ have r-vision of each other if there is an axis-parallel rectangle inside P, containing x and y.



x and z do not have rectangular vision of each other

Rectangular or r-vision

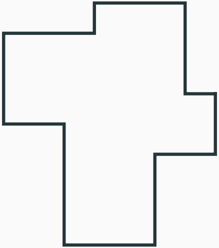
Rectangular vision: two points $x, y \in P$ have r-vision of each other if there is an axis-parallel rectangle inside P, containing x and y.



r-star: an orthogonal polygon that can be covered by one guard equipped with r-vision

Rectangular or r-vision

Rectangular vision: two points $x, y \in P$ have r-vision of each other if there is an axis-parallel rectangle inside P, containing x and y.



During the rest of the talk, vision means r-vision

Complexity results for r-vision

Theorem (Worman and Keil, 2007)

There is an $\tilde{\mathcal{O}}(n^{17})$ time algorithm that computes the minimum size set of point guards equipped with r-vision covering an n-vertex simple orthogonal polygon.

Complexity results for r-vision

Theorem (Worman and Keil, 2007)

There is an $\tilde{\mathcal{O}}(n^{17})$ time algorithm that computes the minimum size set of point guards equipped with r-vision covering an n-vertex simple orthogonal polygon.

Theorem (Lingas, Wasylewicz, Żyliński, 2012)

There is a linear time 3-approximation algorithm for minimum size point guard system with rectangular vision.

Complexity results for r-vision

Theorem (Worman and Keil, 2007)

There is an $\tilde{\mathcal{O}}(n^{17})$ time algorithm that computes the minimum size set of point guards equipped with r-vision covering an n-vertex simple orthogonal polygon.

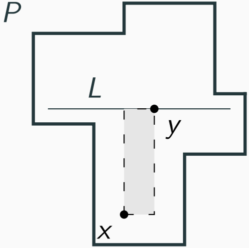
Theorem (Lingas, Wasylewicz, Żyliński, 2012)

There is a linear time 3-approximation algorithm for minimum size point guard system with rectangular vision.

The novelty of our algorithm is not so much the lower approximation ratio, but the extremal style of the result (we will see this)

Mobile guards in orthogonal polygons

A mobile guard is an axis-parallel line segment L inside the gallery. The guard sees a point $x \in P$ iff there is a point $y \in L$ such that x is visible from y.



Sliding camera (introduced by Katz and Morgenstern, 2011): a mobile guard whose line segment is maximal, equipped with r-vision

Results on the complexity of sliding camera problems

Theorem (Győri and M, 2016)

There is a linear time algorithm that finds a covering set of mobile guards of cardinality at most $\left\lfloor \frac{3n+4}{16} \right\rfloor$, even if the patrols are required to be pairwise disjoint.

The complexity of the optimization problem is open.

Our result

p: minimum number of point guards required to cover P

m_V: min. number of vertical sliding cameras required to cover P

m_H: min. number of horizontal sliding cameras required to cover P

Our result

p: minimum number of point guards required to cover P

m_V: min. number of vertical sliding cameras required to cover P

m_H: min. number of horizontal sliding cameras required to cover P

Theorem

For any simple orthogonal polygon there is a linear time algorithm which finds a point guard of size at most

$$\frac{4}{3}\left(m_V + m_H - 1\right).$$

Our result

p: minimum number of point guards required to cover P

m_V: min. number of vertical sliding cameras required to cover P

m_H: min. number of horizontal sliding cameras required to cover P

Theorem

For any simple orthogonal polygon there is a linear time algorithm which finds a point guard of size at most

$$\frac{4}{3}\left(m_V + m_H - 1\right).$$

Since $m_V, m_H \leq p$, we have

$$\frac{4}{3}(m_V + m_H - 1) \leq \frac{8}{3}p,$$

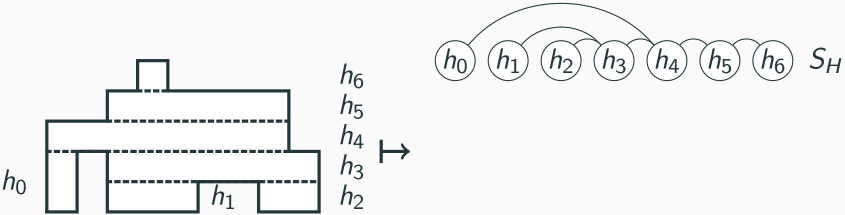
so the algorithm provides an $\frac{8}{3}$-approximation solution.

High level description of the algorithm

Horizontal and vertical R-trees

ortho. poly. P

tree T_H

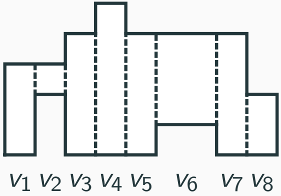


Cut horizontally at each reflex vertex, join touching slices by an edge

Győri et. al. (1995) drafts that T_H can be computed in linear time

Horizontal and vertical R-trees

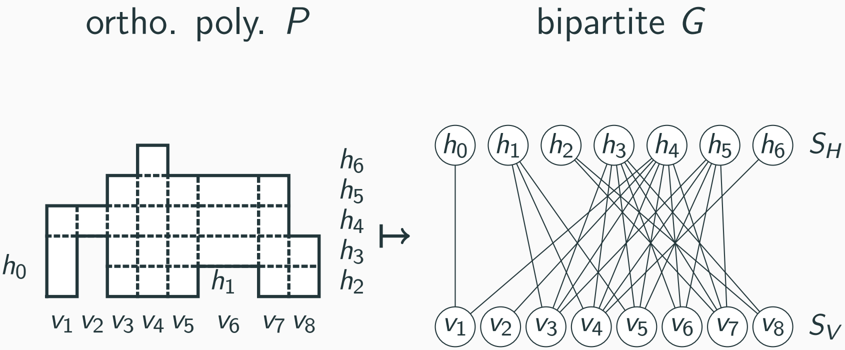
ortho. poly. P tree T_V



$\mapsto$

Do the same for vertical slices

Horizontal and vertical R-trees

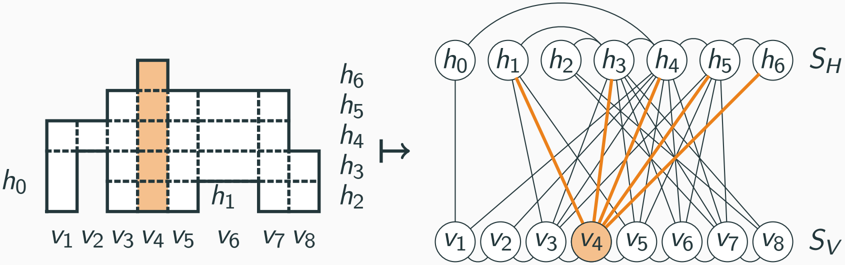


Join two slices iff their interiors intersect

G may have $\Omega(n^2)$ edges

Horizontal and vertical R-trees

ortho. poly. P



Each neighborhood in G forms a path in the appropriate R-tree

Observation 1: by storing only the ends of the path formed by the neighborhood of each vertex of G, the graph can be described in $\mathcal{O}(n)$ space

Working with the sparse representation of G

- Choose arbitrary roots in the R-trees
- The lowest common ancestors algorithm (for eg. the one due to Gabow and Tarjan, 1985) requires linear time

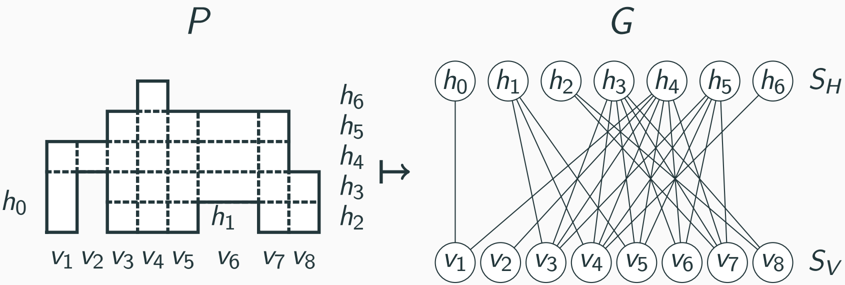
Working with the sparse representation of G

- Choose arbitrary roots in the R-trees
- The lowest common ancestors algorithm (for eg. the one due to Gabow and Tarjan, 1985) requires linear time
- Observation 2: for any $v_1, v_2 \in S_V$, the ends of the path formed by $N_G(v_1) \cap N_G(v_2)$ in T_H can be computed using 6 LCA queries

Working with the sparse representation of G

- Choose arbitrary roots in the R-trees
- The lowest common ancestors algorithm (for eg. the one due to Gabow and Tarjan, 1985) requires linear time
- Observation 2: for any $v_1, v_2 \in S_V$, the ends of the path formed by $N_G(v_1) \cap N_G(v_2)$ in T_H can be computed using 6 LCA queries
- Observation 3: if G is 2-connected then for any $v_1 v_2 \in E(T_V)$ we have $|N_G(v_1) \cap N_G(v_2)| \geq 2$

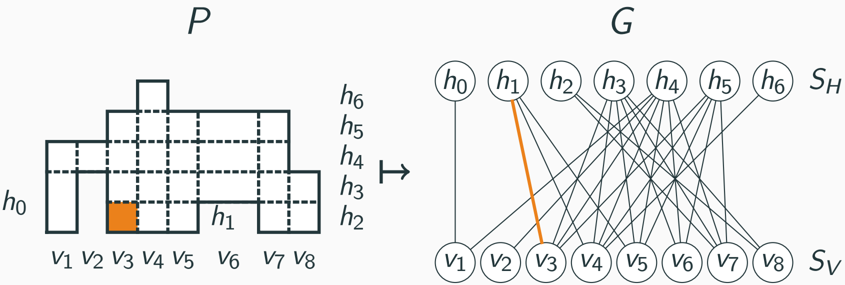
Pixelation graph



The intersection graph structure in connection with mobile guards has been studied by Kosowski, Małafiejski, and Żyliński (2007)

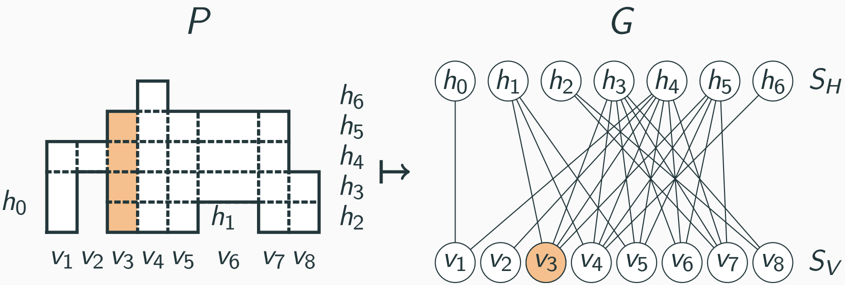
With respect to rectangular vision, it is enough to know the pixels containing the points.

Pixelation graph



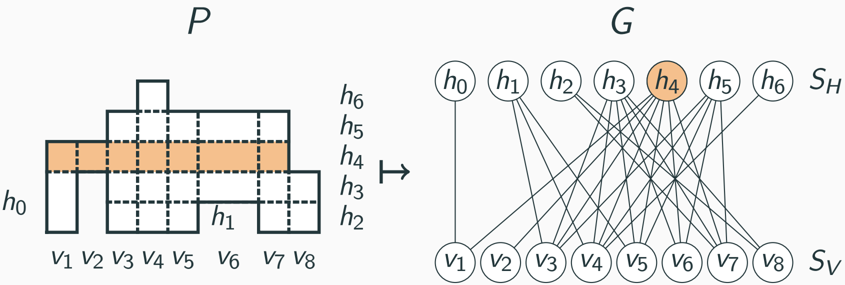
Point guard $\leftrightarrow$ Edge

Pixelation graph



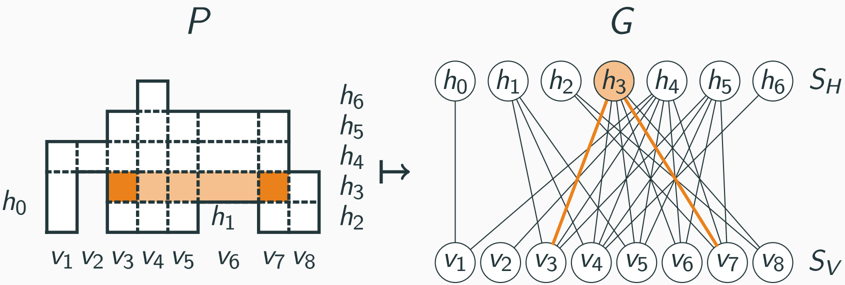
Sliding camera $\leftrightarrow$ Vertex

Pixelation graph



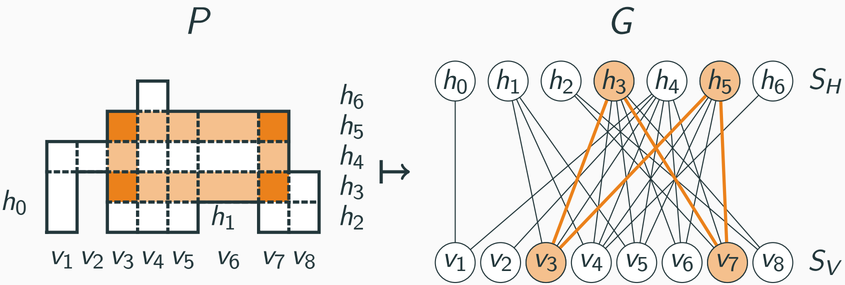
Sliding camera $\leftrightarrow$ Vertex

Pixelation graph



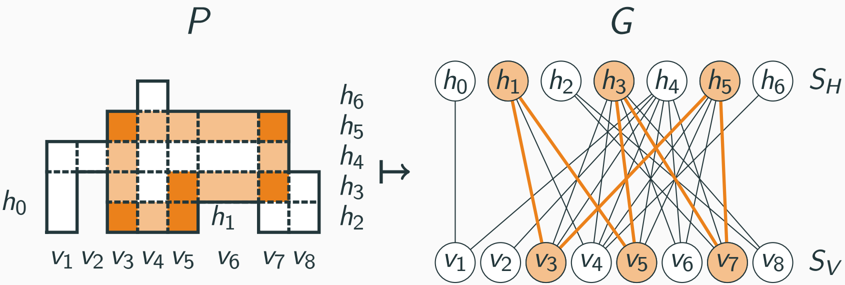
Rectangular vision ($e_1 \cap e_2 \neq \emptyset$)

Pixelation graph



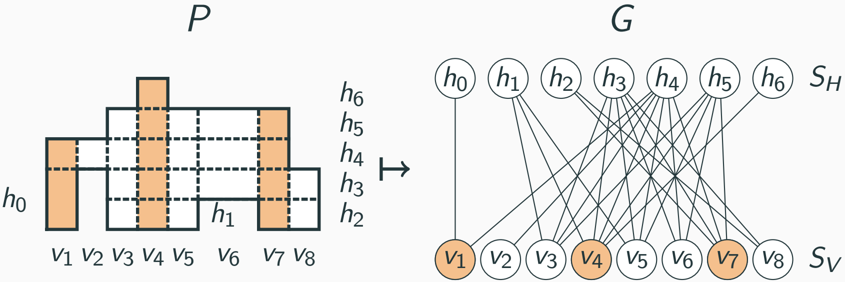
Rectangular vision ($G[e_1 \cup e_2] \cong C_4$)

Pixelation graph



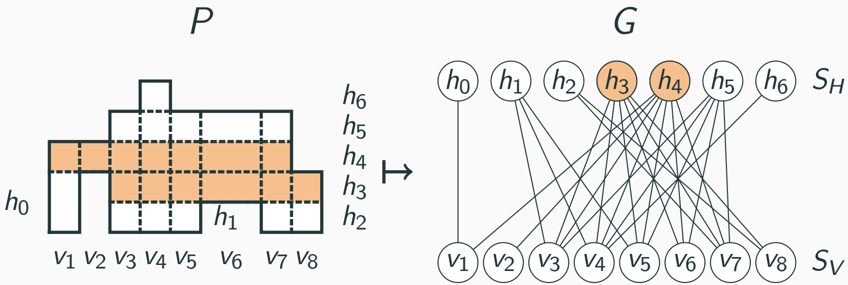
G is chordal bipartite: any cycle of length at least 6 has a chord (eg.: $h_5 v_5$)

Pixelation graph



Covering set of vert. sliding cameras $\leftrightarrow$ $M_V \subseteq S_V$ dominating S_H

Pixelation graph



Covering set of horiz. sliding cameras $\leftrightarrow$ $M_H \subseteq S_H$ dominating S_V

More about the structure

- Dirac: $\nu = \tau$ for a family subtrees of a tree $\Rightarrow M_V$ and M_H can be computed in linear time (Győri and M, 2018)

More about the structure

- Dirac: $\nu = \tau$ for a family subtrees of a tree $\Rightarrow M_V$ and M_H can be computed in linear time (Győri and M, 2018)

- $M := G[M_H \cup M_V]$ is also chordal bipartite; for simplicity assume it is connected

More about the structure

- Dirac: $\nu = \tau$ for a family subtrees of a tree $\Rightarrow M_V$ and M_H can be computed in linear time (Győri and M, 2018)
- $M := G[M_H \cup M_V]$ is also chordal bipartite; for simplicity assume it is connected
- Given an edge $h_0 v_0 \in E(G)$, there exists $h_1 \in M_H$ and $v_1 \in M_V$ s.t. $h_0 v_1 \in E(G)$ and $h_1 v_0 \in E(G)$

More about the structure

- Dirac: $\nu = \tau$ for a family subtrees of a tree $\Rightarrow M_V$ and M_H can be computed in linear time (Győri and M, 2018)
- $M := G[M_H \cup M_V]$ is also chordal bipartite; for simplicity assume it is connected
- Given an edge $h_0 v_0 \in E(G)$, there exists $h_1 \in M_H$ and $v_1 \in M_V$ s.t. $h_0 v_1 \in E(G)$ and $h_1 v_0 \in E(G)$
- There is a path in M joining h_1 to v_1 which together with $h_0 v_0$ forms a cycle

More about the structure

- Dirac: $\nu = \tau$ for a family subtrees of a tree $\Rightarrow M_V$ and M_H can be computed in linear time (Győri and M, 2018)

- $M := G[M_H \cup M_V]$ is also chordal bipartite; for simplicity assume it is connected

- Given an edge $h_0 v_0 \in E(G)$, there exists $h_1 \in M_H$ and $v_1 \in M_V$ s.t. $h_0 v_1 \in E(G)$ and $h_1 v_0 \in E(G)$

- There is a path in M joining h_1 to v_1 which together with $h_0 v_0$ forms a cycle

- G chordal $\Rightarrow \exists v_2 h_2 \in E(M)$ s.t. $h_0 v_0 h_2 v_2$ is a 4-cycle in G (or $v_2 = v_0$ or $h_2 = h_0$)

Computing the structures induced by M_H and M_V

- $E(M)$ is a point guard (given that M is connected), but may contain $\Omega(n^2)$ edges

Computing the structures induced by M_H and M_V

- $E(M)$ is a point guard (given that M is connected), but may contain $\Omega(n^2)$ edges
- M connected $\Rightarrow \exists$ induced R-trees on M_H and M_V

Computing the structures induced by M_H and M_V

- $E(M)$ is a point guard (given that M is connected), but may contain $\Omega(n^2)$ edges
- M connected $\Rightarrow \exists$ induced R-trees on M_H and M_V
- Observations 1-3 apply to M as well!

Computing the structures induced by M_H and M_V

- $E(M)$ is a point guard (given that M is connected), but may contain $\Omega(n^2)$ edges
- M connected $\Rightarrow \exists$ induced R-trees on M_H and M_V
- Observations 1-3 apply to M as well!
- The observations allow us to compute the 2-connected components $M_1, \ldots, M_q$ of M efficiently

Computing the structures induced by M_H and M_V

- $E(M)$ is a point guard (given that M is connected), but may contain $\Omega(n^2)$ edges
- M connected $\Rightarrow \exists$ induced R-trees on M_H and M_V
- Observations 1-3 apply to M as well!
- The observations allow us to compute the 2-connected components $M_1, \ldots, M_q$ of M efficiently
- For each i, we have a subproblem given by M_i and $N_G(V(M_i))$; so from now on assume that M is 2-connected

Finding point guards

- Case 1: M is an edge: the only edge of M guards G
- Case 2: M is a non-trivial 2-connected graph: any edge is contained in a 4-cycle, so we define

$$P[M] = \bigcup_{\{e_1,e_2,e_3,e_4\} \text{ is a } C_4 \text{ in } M} \mathrm{Conv}\left(\bigcup_{i=1}^{4} \cap e_i\right)$$

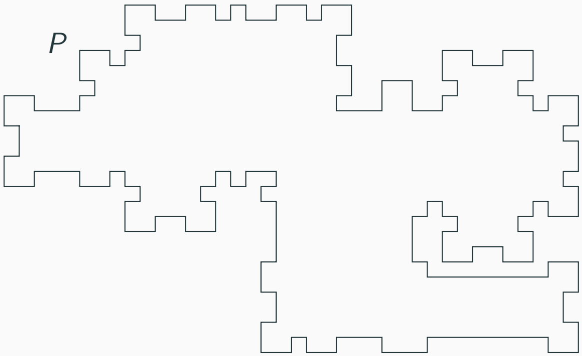
Finding point guards

- Case 1: M is an edge: the only edge of M guards G
- Case 2: M is a non-trivial 2-connected graph: any edge is contained in a 4-cycle, so we define

$$P[M] = \bigcup_{\{e_1,e_2,e_3,e_4\} \text{ is a } C_4 \text{ in } M} \mathrm{Conv}\left(\bigcup_{i=1}^{4} \cap e_i\right)$$

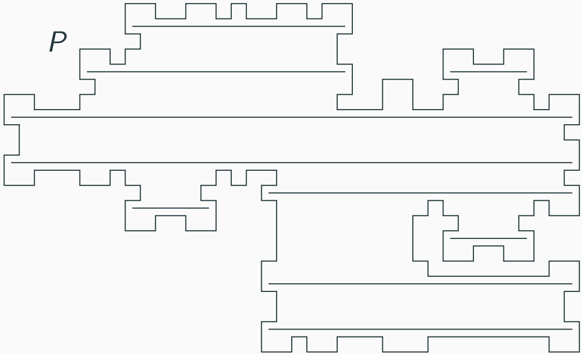
Lemma: $P[M]$ is simply connected

M is a non-trivial 2-connected graph



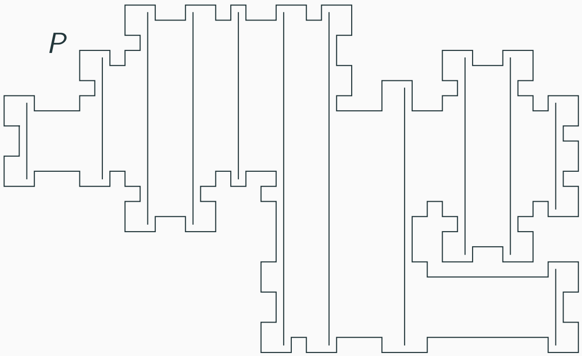
A large simple orthogonal polygon with $n = 160$

M is a non-trivial 2-connected graph



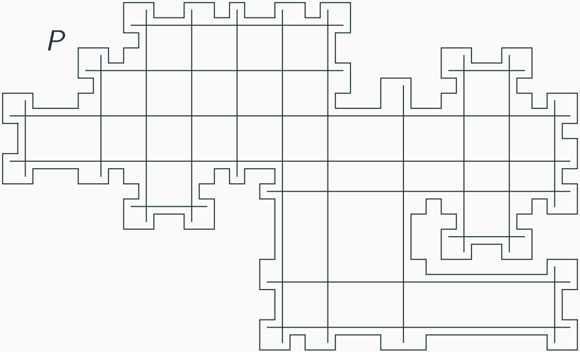
$|M_H| = 10$

M is a non-trivial 2-connected graph



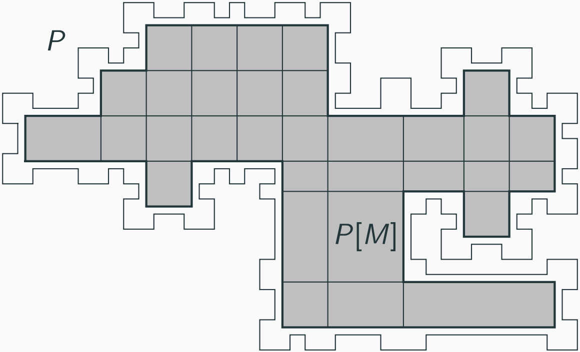
$$|M_V| = 12$$

M is a non-trivial 2-connected graph



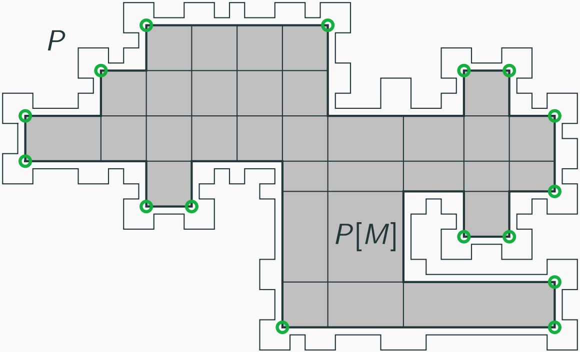
$$|M_H| + |M_V| = 22$$

M is a non-trivial 2-connected graph



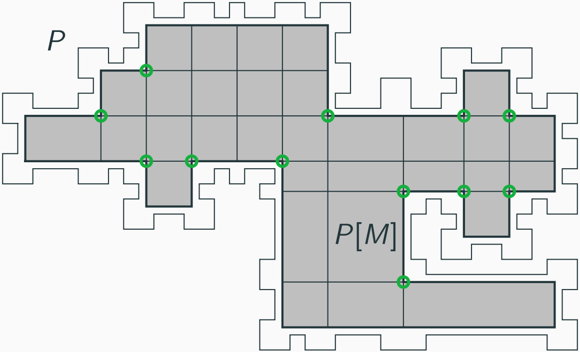
The pixels on the boundary of $P[M]$ form a point guard of P!

M is a non-trivial 2-connected graph



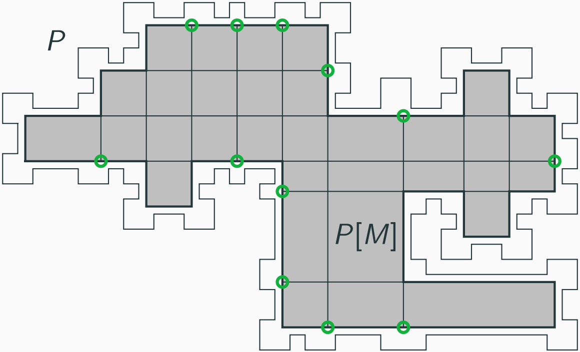
c pixels at convex vertices of $P[M]$

M is a non-trivial 2-connected graph



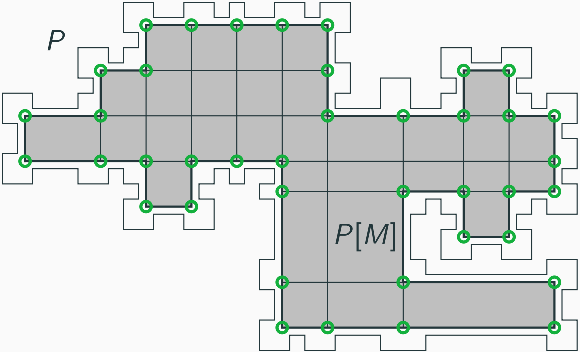
r pixels at reflex vertices of $P[M]$

M is a non-trivial 2-connected graph



s pixels on the boundary but not at a vertex of $P[M]$

M is a non-trivial 2-connected graph

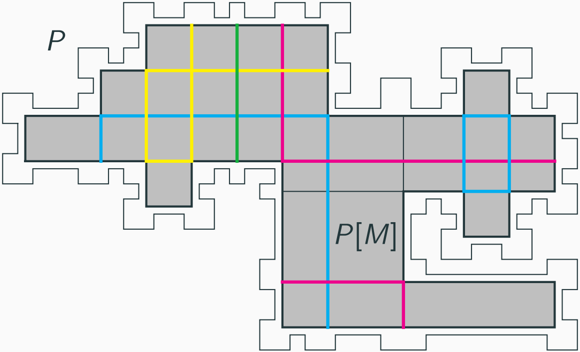


$$|M_H| + |M_V| = c + \frac{1}{2}s \text{ and } r = c - 4, \text{ so}$$

$$c + r + s = 2c + s - 4 = 2(|M_H| + |M_V|) - 4$$

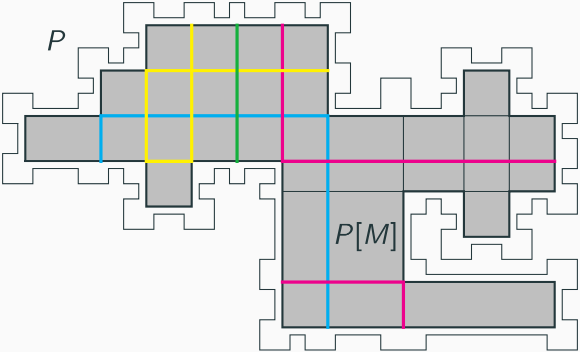
The pixels on the boundary of $P[M]$ is a 4-approximation solution

M is a non-trivial 2-connected graph



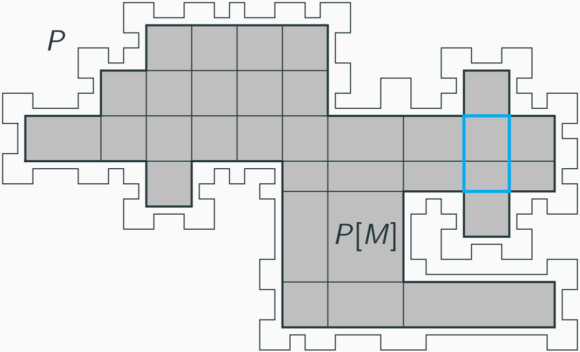
We can do better: some reflex and side edges can be omitted

M is a non-trivial 2-connected graph



Path of ℓ reflex and side pixels: only $\lceil \frac{\ell}{3} \rceil$ guards needed

M is a non-trivial 2-connected graph



Cycle of ℓ reflex and side pixels: only $\lfloor \frac{\ell}{3} \rfloor$ guards needed

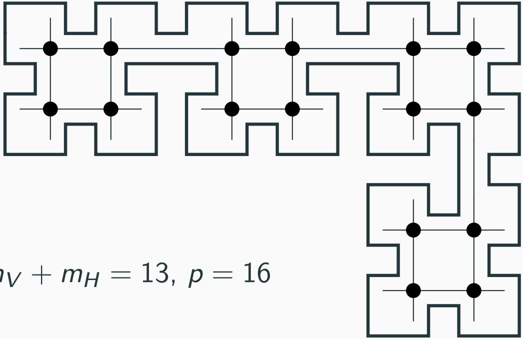
Summing it up

$$
\begin{aligned}
c + \sum_{\text{paths}} \lceil \frac{\ell}{3} \rceil + \sum_{\text{cycles}} \lfloor \frac{\ell}{3} \rfloor &\leq c + \sum_{\text{paths}} \frac{\ell + 2}{3} + \sum_{\text{cycles}} \frac{\ell}{3} \leq \\
&\leq c + \frac{1}{3}(r + s) + \sum_{\text{paths}} \frac{2}{3} \leq \\
&\leq c + \frac{1}{3}(c - 4 + s) + \frac{1}{3} s \leq \\
&\leq \frac{4}{3}(c + \frac{1}{2} s - 1) \leq \\
&\leq \frac{4}{3}(|M_H| + |M_V| - 1)
\end{aligned}
$$

Finishing the proof

- If M is connected, but not 2-connected, we recursively construct the guard sets for each 2-connected component
- Technical: if M has t connected components, at most $t - 1$ extra guards are necessary beyond what the recursive construction gives

Sharpness



$m_V + m_H = 13,\ p = 16$

A new block requires 4 more point guards, but only 3 more vertical + horizontal mobile guards.

Translating the problem to the pixelation graph

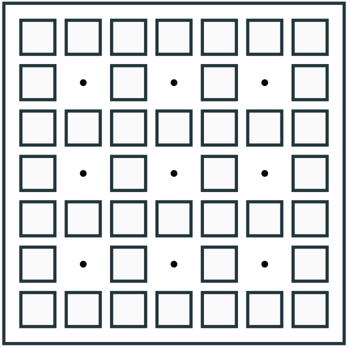
Orthogonal polygon	**Pixelation graph**
Mobile guard	Vertex
Point guard	Edge
Simply connected	Chordal bipartite ($\Rightarrow$, but $\not\Leftarrow$)
r-vision of two points	$e_1 \cap e_2 \neq \emptyset$ or $G[e_1 \cup e_2] \cong C_4$
Horiz. mobile guard cover	$M_H \subseteq S_H$ dominating S_V
Covering set of mobile guards	Dominating set

Making the definitions precise

- Degenerate-vision is prohibited
- The vertical and horizontal lines containing a point/mobile guard may not pass through a vertex of the polygon.
- These may be assumed without loss of generality, by using applying the following transformation to the gallery:

Simply connectedness is essential

For an orthogonal polygon with orthogonal holes, the ratio of $m_V + m_H$ and p is not bounded: no two of the black dots can be covered by a single point guard.



$m_V + m_H = 4k + 4$, but $p \geq k^2$

Approximation algorithms for line of sight vision

Theorem (Krohn and Nilsson, 2012)

There is a polynomial time algorithm that computes a guard cover of size $\mathcal{O}(OPT^2)$ in a simple orthogonal polygon P, where OPT is the size of the smallest guard cover for P.

Approximation algorithms for line of sight vision

Theorem (Krohn and Nilsson, 2012)

There is a polynomial time algorithm that computes a guard cover of size $\mathcal{O}(OPT^2)$ in a simple orthogonal polygon P, where OPT is the size of the smallest guard cover for P.

Theorem (Ghosh, 2010)

There is an $\mathcal{O}(n^5)$ time algorithm that computes a solution the point guard problem in a polygon (with or without holes) with an $\mathcal{O}(\log n)$ approximation ratio.

Approximation algorithms for line of sight vision

Theorem (Krohn and Nilsson, 2012)

There is a polynomial time algorithm that computes a guard cover of size $\mathcal{O}(OPT^2)$ in a simple orthogonal polygon P, where OPT is the size of the smallest guard cover for P.

Theorem (Ghosh, 2010)

There is an $\mathcal{O}(n^5)$ time algorithm that computes a solution the point guard problem in a polygon (with or without holes) with an $\mathcal{O}(\log n)$ approximation ratio.

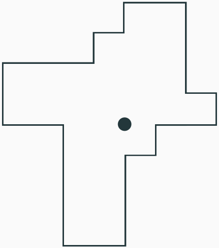
Theorem (Eidenbenz, Stamm, and Widmayer, 2001)

The point guard problem in simple polygons is APX-hard.

Covering by r-stars

Theorem (Hoffmann and Kaufmann, 1991)

Any n-vertex orthogonal polygon with holes can be partitioned into at most $\left\lfloor \frac{n}{4} \right\rfloor$ at most 16-vertex r-stars in $\tilde{\mathcal{O}}(n^{\frac{3}{2}})$ time.



A 16-vertex r-star.