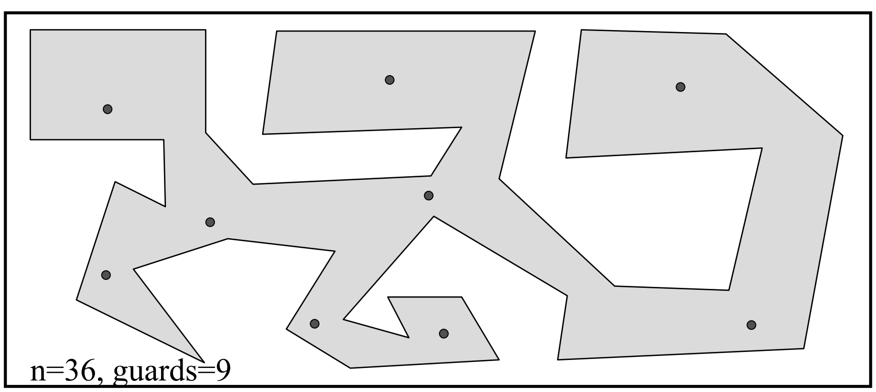
Computational Geometry

Exercise session #4: Polygon decompositions

- Art Gallery Theorem
- Trapezoidal decomposition
- Convex decomposition
- Solution to homework 1

The art gallery problem

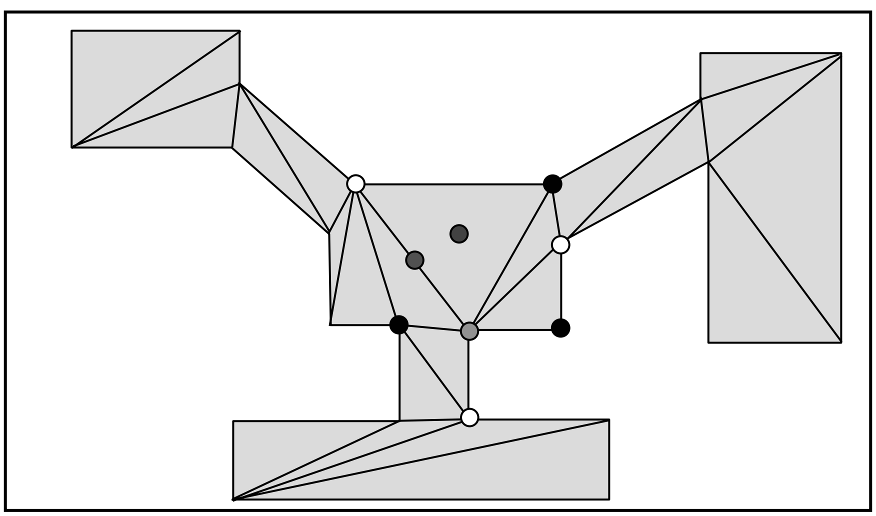
- Given a simple polygon, place point guards whose visibility regions cover the interior.



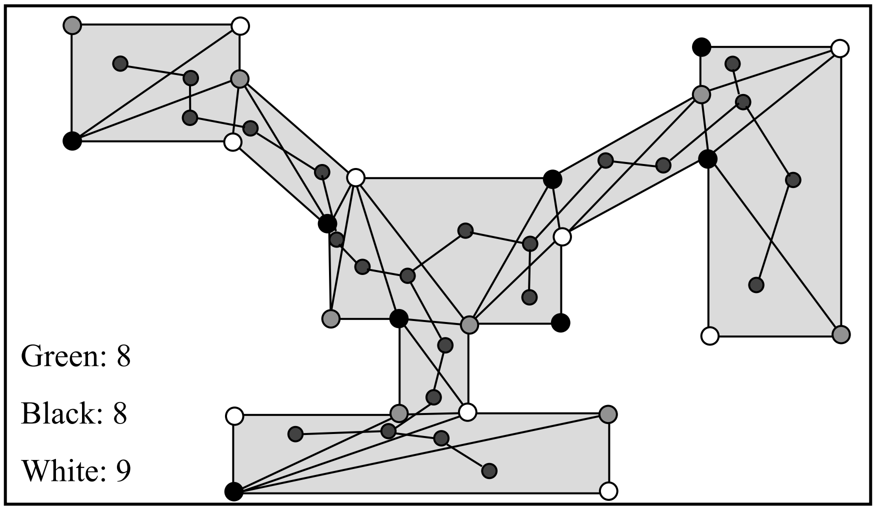
Art Gallery Theorem

- **Theorem**: For any simple polygon, $\lfloor n/3 \rfloor$ point guards are sufficient to guard the entire polygon. A placement of $\lfloor n/3 \rfloor$ guards can be computed in $O(n\log n)$ time.
- Proof idea: triangulate polygon, then position guards to cover each triangle.

Placing the guards



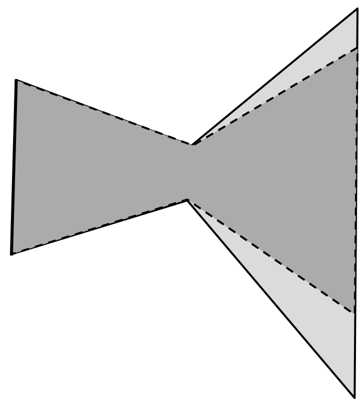
3-Coloring of Polygon Vertices



Correctness & Complexity

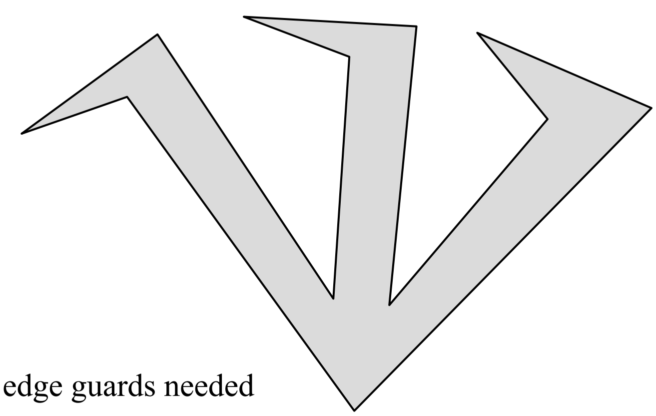
- Dual graph of triangulation is a tree.
- BFS traversal of tree visits vertices once.
- Neighboring triangle vertices get different colors.
- One vertex guard sees the vertices of different colors.
- Running time: triangulation + linear traversal of dual graph. Can be done in $O(n)$ time.

Guarding with Edge guards



How many edge guards are needed to guard a gallery?

Lower bound for edge guards



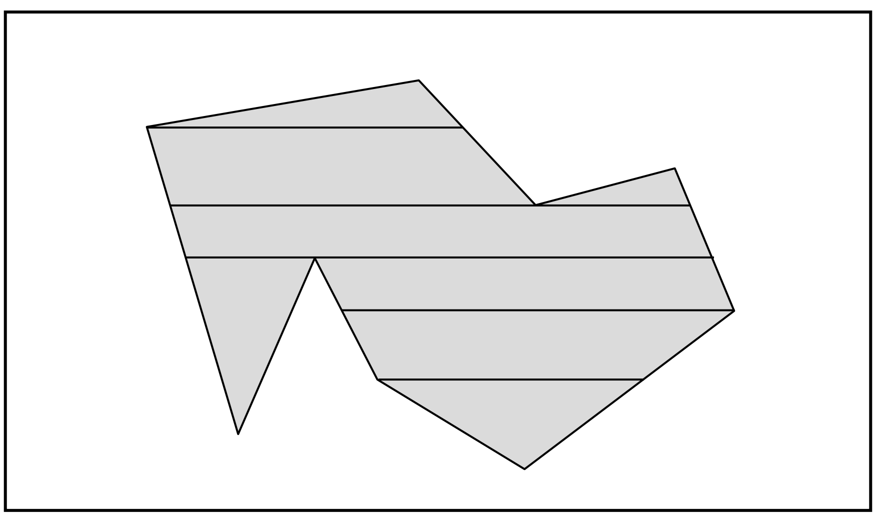
$\lfloor n/4 \rfloor$ edge guards needed

What about star-shaped polygons?

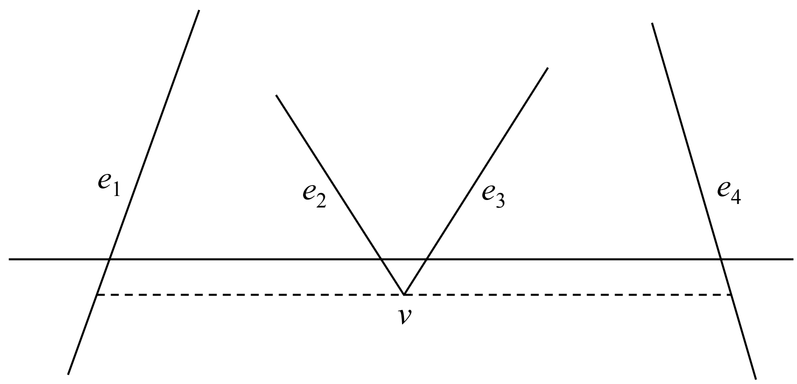
Trapezoidal decomposition

- Trapezoids are quadrilaterals with two parallel edges, or triangles (degenerate case).
- Trapezoids are easily triangulated.
- Decompose polygon into slabs which are easily searched.
- Computation using sweep line is easily achieved.
- Drawback: new vertices are added to the decomposition.

Example

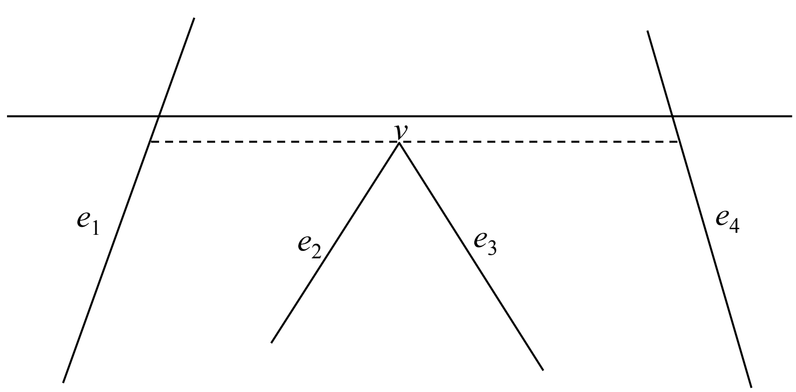


Sweep events I



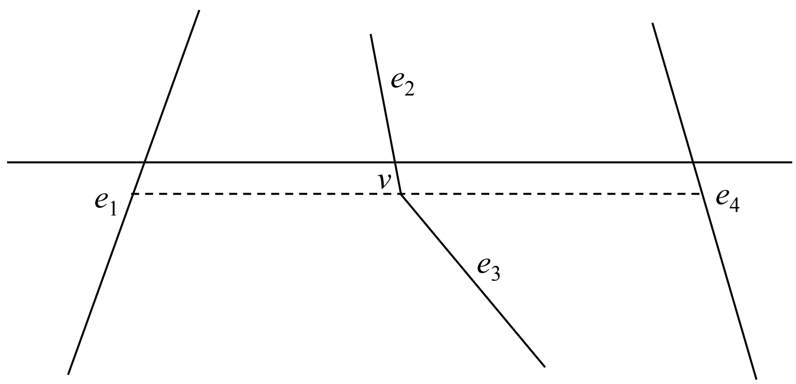
Sweepline status: (e_1,e_2,e_3,e_4) → (e_1,e_4)

Sweep events II



Sweepline status: (e_1,e_4) → (e_1,e_2,e_3,e_4)

Sweep events III

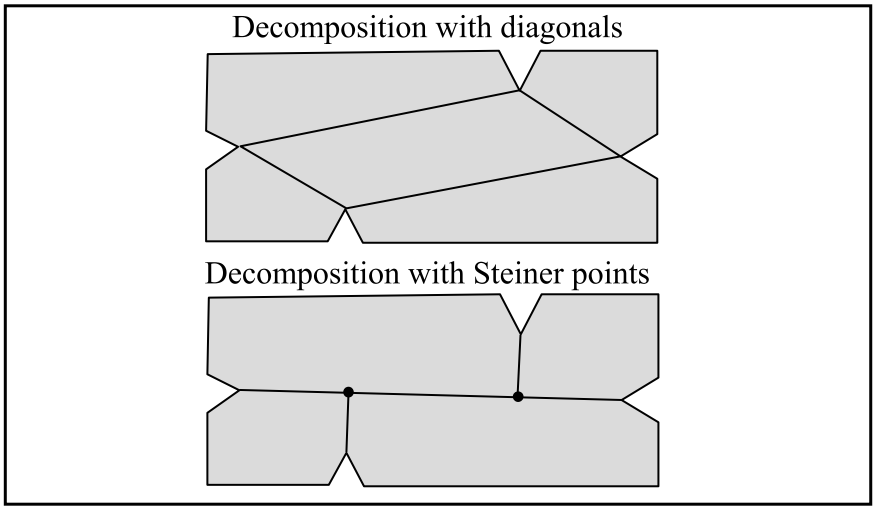


Sweepline status: $(e_1,e_2,e_4) \rightarrow (e_1,e_3,e_4)$

Convex decompositions

- Partition polygon into *minimal* number of *convex* pieces with *disjoint* interiors such they *cover* the polygon.
- Two versions:
  - Partition using only diagonals
  - Partition with additional vertices ('Steiner points')
- For a given polygon, which version results in less convex pieces?
- Which is harder to compute?

Decomposition example



Bounds on optimal decomposition

- Let Φ be the fewest number of convex pieces into which a polygon may be partitioned.
- **Theorem**: for a polygon of r reflex vertices:

$$\lceil r/2 \rceil + 1 \leq \Phi \leq r + 1$$

- *Proof*:
  - Drawing a segment that bisects a reflex vertex removes all reflex angles → upper bound (Steiner).
  - At most two reflex angles can be resolved by a single segment → lower bound (both versions).

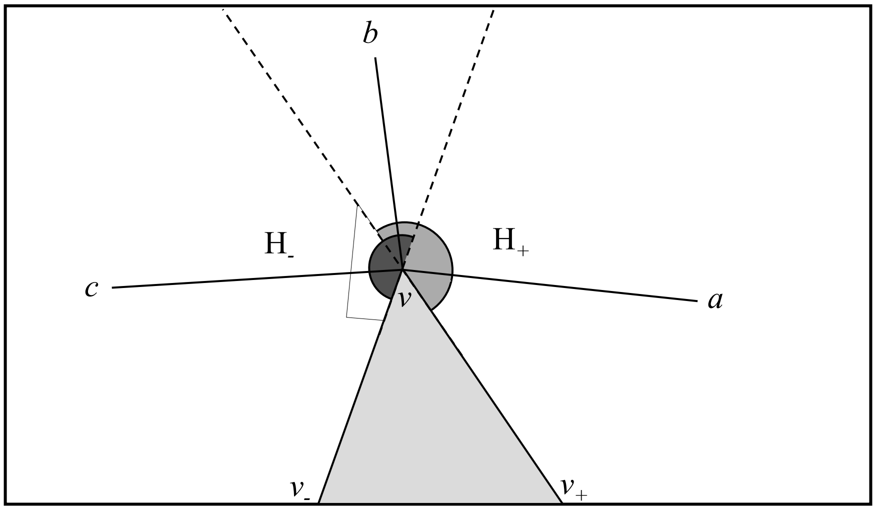
4-Approximation algorithm

- Hertel & Melhorn 1983.
- Quickly partition polygon such that number of pieces is at most four times the optimal.
- **Definition**: a diagonal *d* is *essential* for vertex v if removal of *d* creates a non-convex piece. Otherwise it is *inessential*.
- **Algorithm**: triangulate, then remove inessential diagonals one after one until none remain.

Quality of approximation

- Algorithm outputs convex partitions, since it starts with one and maintains this invariant throughout all steps.
- **Lemma 1**: There can be at most two diagonals essential for any reflex vertex.
- Lemma 2: The Hertel-Melhorn algorithm is never worse than four times the optimal number of pieces.

Proof of lemma 1



Proof of lemma 2

- When algorithm terminates, all diagonals are essential.
- According to lemma 1, every reflex vertex is responsible for at most 2 diagonals.
- Number of diagonals cannot be more than 2*r*.
- Number of convex pieces produced: $M \leq 2r+1$.
- Lower bound for Φ implies $M \leq 4\Phi$.

Optimal Convex partitioning

- Best algorithm for partitioning with diagonals: [Keil, 1985], runs in $O(r^2 n \log n)$ time.
- Best algorithm for partitioning with Steiner points: [Chazelle, 1980], runs in $O(n+r^3)$ time.