Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Chair for
INFORMATICS I
Efficient Algorithms and
Knowledge-Based Systems

Institute for Informatics

Computational Geometry

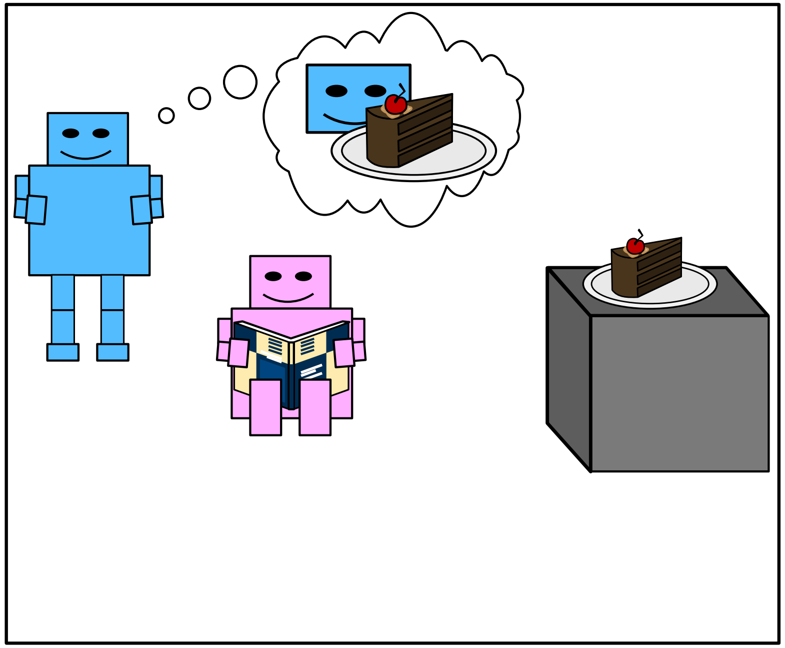
Visibility Graphs
or
Finding Shortest Paths

Lecture #12

Dr. Philipp Kindermann

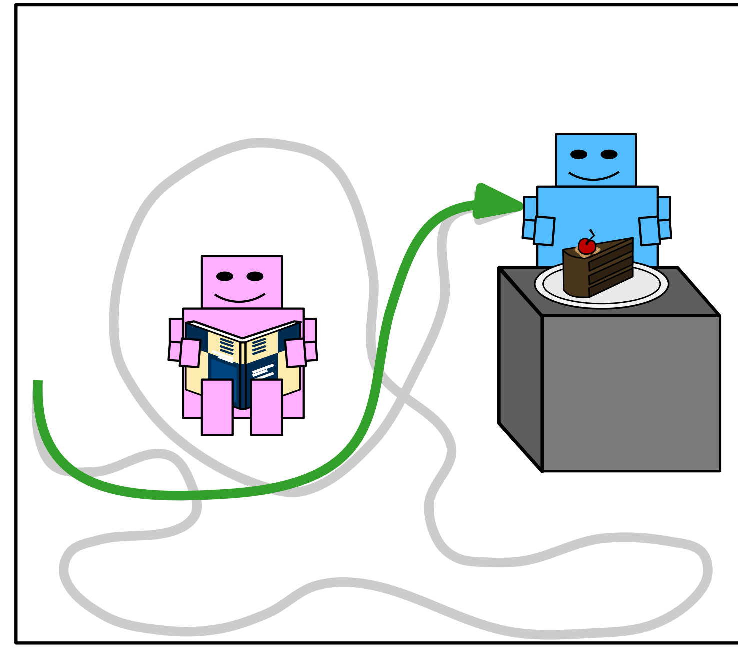
Winter Semester 2018/19

Path Planning



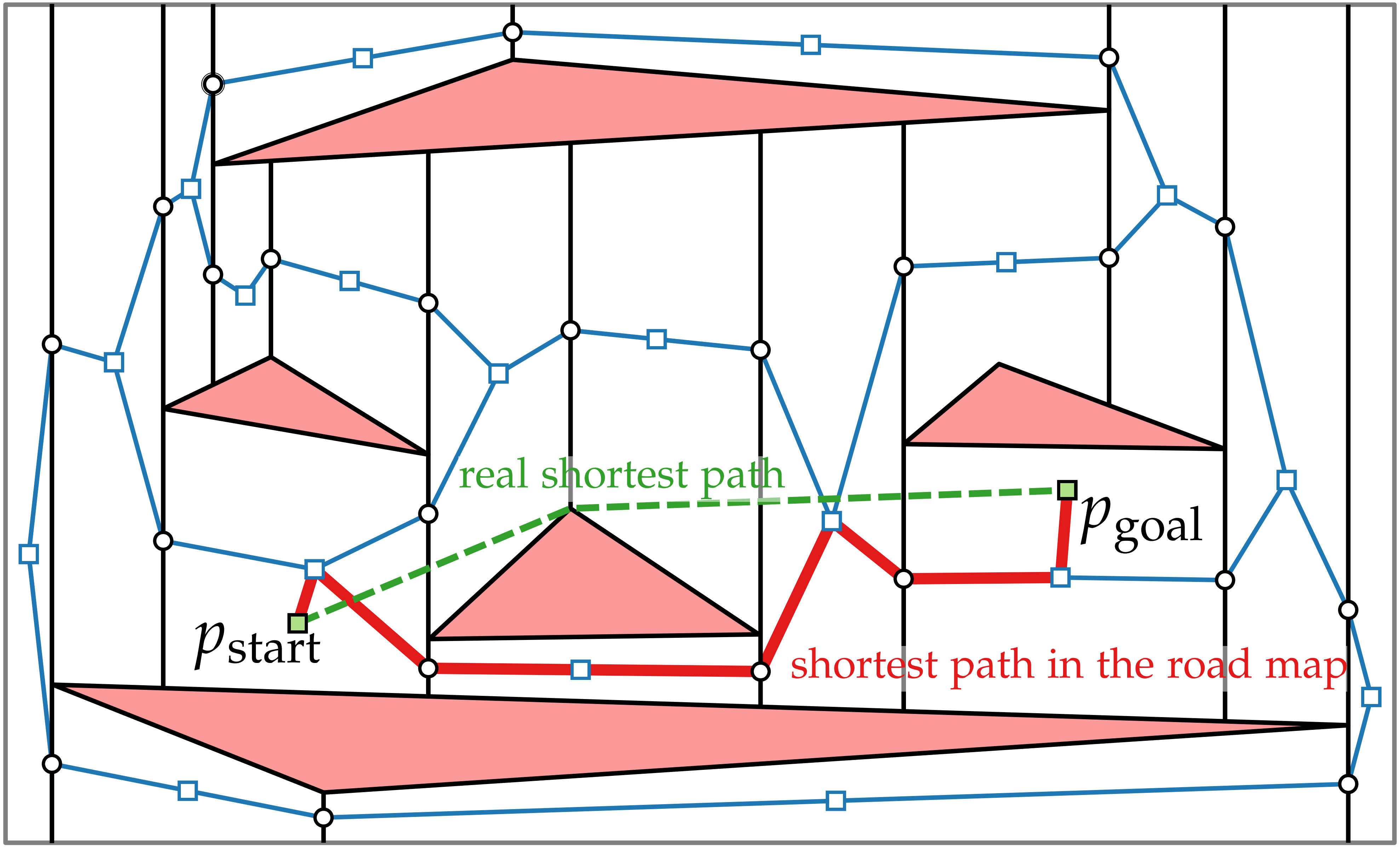
current location,
desired location

$\Rightarrow$



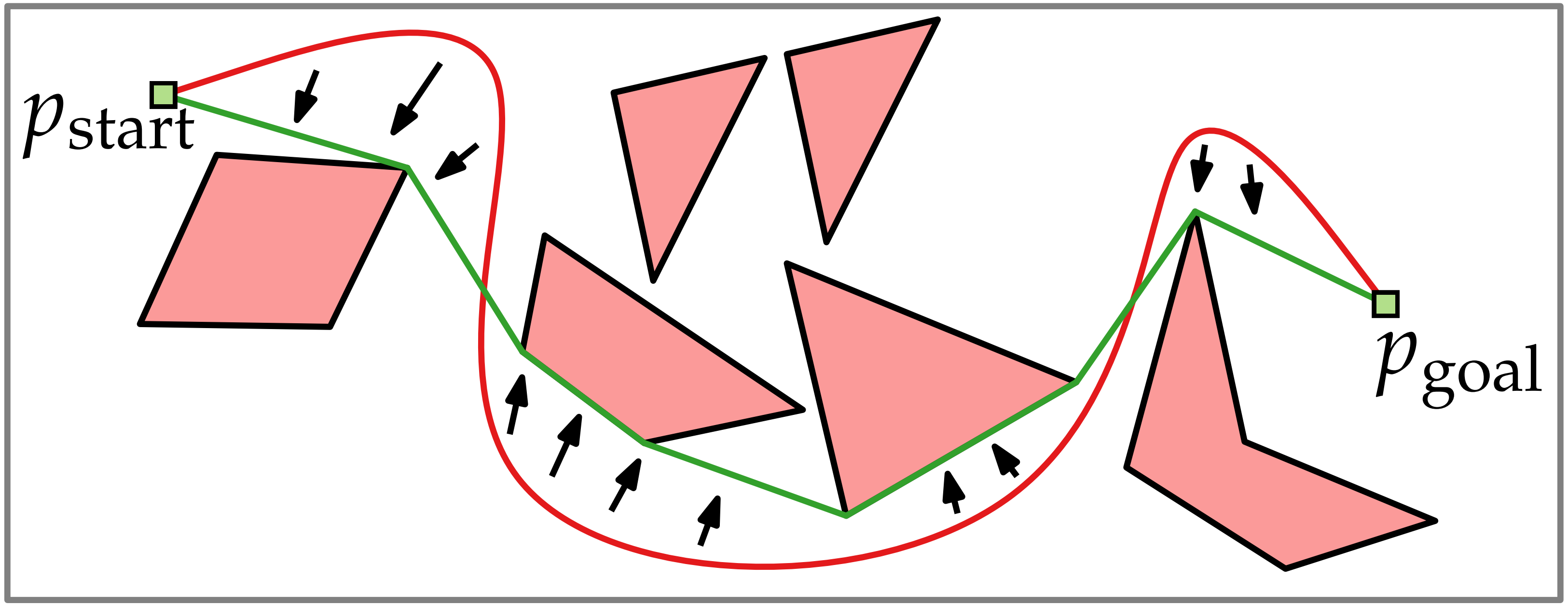
shortest path to reach the one from the other

Let's recap...


real shortest path
p_{goal}
p_{start}
shortest path in the road map

Characterization

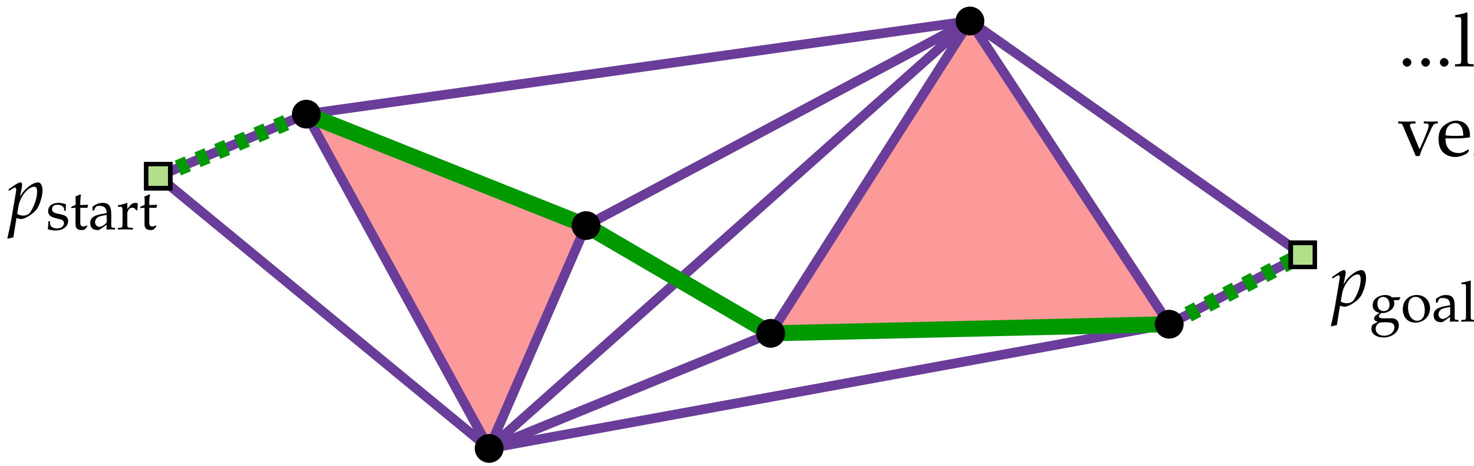
Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space, any shortest path between p_{start} and p_{goal} is a polygonal path whose inner vertices are vertices of S.



Proof.

Visibility Graph

Given a set S of disjoint (open) polygons...



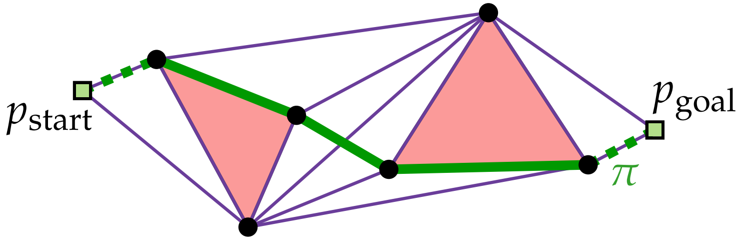
...let $V(S)$ be the vertex set of S.

Let $G_{\text{vis}}(S) = (V(S), E_{\text{vis}}(S))$ be the *visibility graph* of S, where $E_{\text{vis}}(S) = \{uv \mid u, v \in V(S), u \text{ sees } v\}$ and $w(uv) = |uv|$.

We define: u sees v $:\Leftrightarrow \overline{uv} \subset \mathcal{C}_{\text{free}} \quad (= \mathbb{R}^2 \setminus \bigcup S)$

Corollary. A shortest path between p_{start} and p_{goal} corresponds to a *shortest* path in $G_{\text{vis}}(S^\star)$, where $S^\star = S \cup \{p_{\text{start}}, p_{\text{goal}}\}$.

Algorithm



$\textsc{ShortestPath}(S, p_{\text{start}}, p_{\text{goal}})$ $n = |V(S)|$, $m = |E_{\text{vis}}(S)|$

- $G_{\text{vis}} \leftarrow \textsc{VisibilityGraph}(S \cup \{p_{\text{start}}, p_{\text{goal}}\})$?
- **foreach** $uv \in E_{\text{vis}}$ **do** $O(m)$
  - $w(uv) = d_{\text{Eucl.}}(u, v)$
- $\pi \leftarrow \textsc{Dijkstra}(G_{\text{vis}}, w, p_{\text{start}}, p_{\text{goal}})$ $O(m + n \log n)$
- **return** π

Running time?

Computing the Visibility Graph

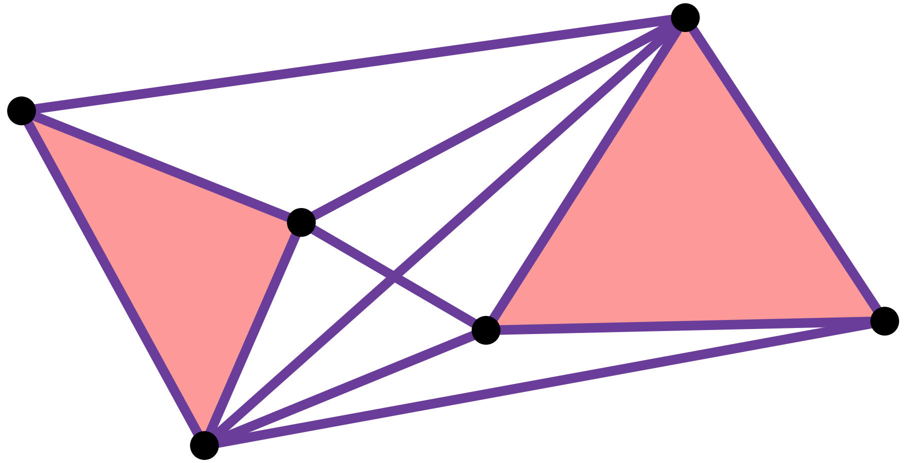
$\textsc{VisibilityGraph}(S)$

Input: a set S of disjoint polygons
Output: $G_{\text{vis}}(S)$

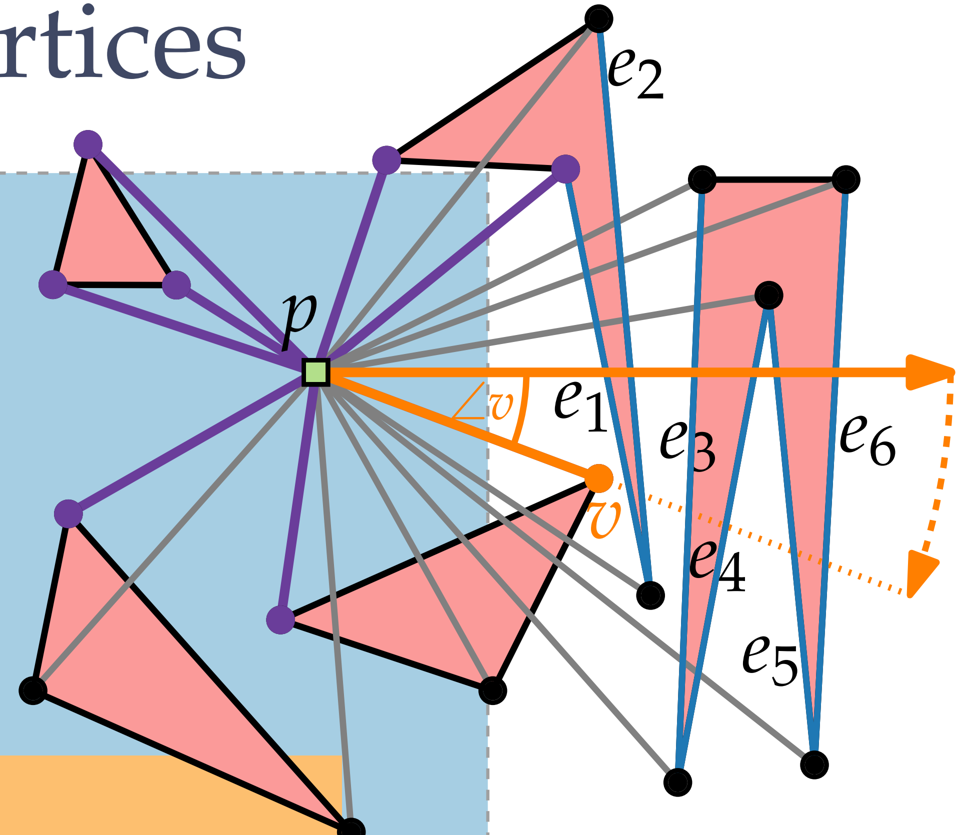
$E \leftarrow \emptyset$
foreach $v \in V(S)$ **do**
  $W = \textsc{VisibleVertices}(v, S)$
  $E \leftarrow E \cup \{vw \mid w \in W\}$
return $(V(S), E)$

$O(n) \cdot$

?



Computing Visible Vertices



VISIBLEVERTICES(p, S)

$r \leftarrow p + \mathbb{R}_{\geq 0}\binom{1}{0}$

$I \leftarrow \{e \in E(S) \mid e \cap r \neq \emptyset\}$

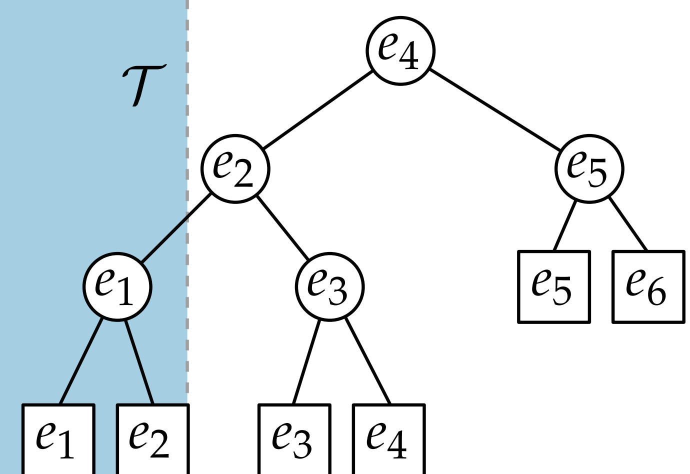
$\mathcal{T} \leftarrow$ balancedBinaryTree(I)

sort $V(S)$ $\qquad v \prec v' :\Leftrightarrow$

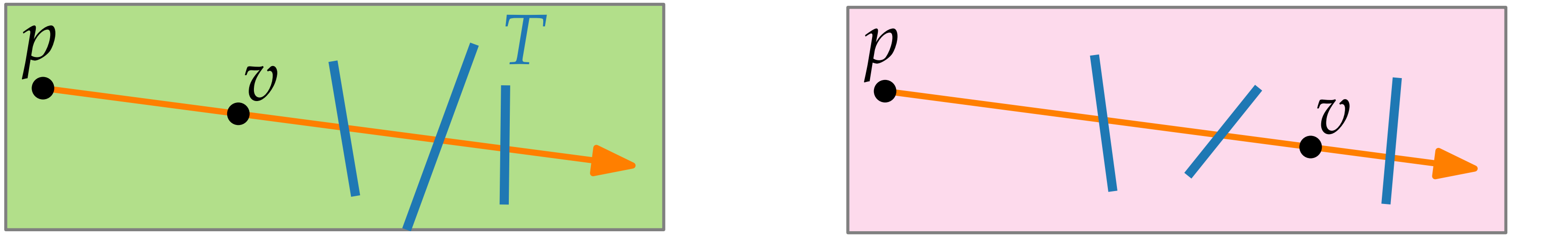
$\angle v < \angle v'$ or
($\angle v = \angle v'$ and $|pv| < |pv'|$)

rotational plane sweep

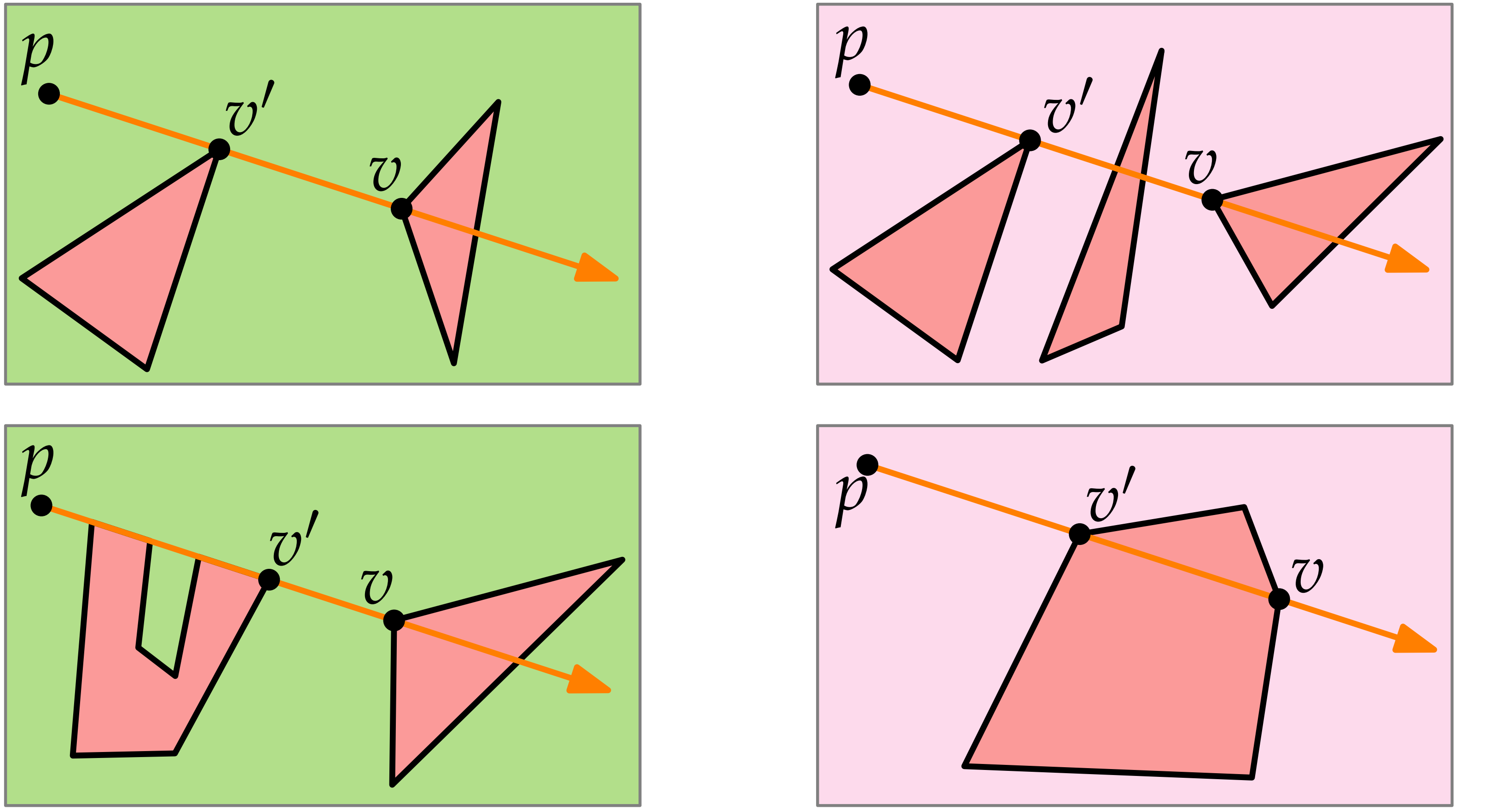
$\mathcal{T}$



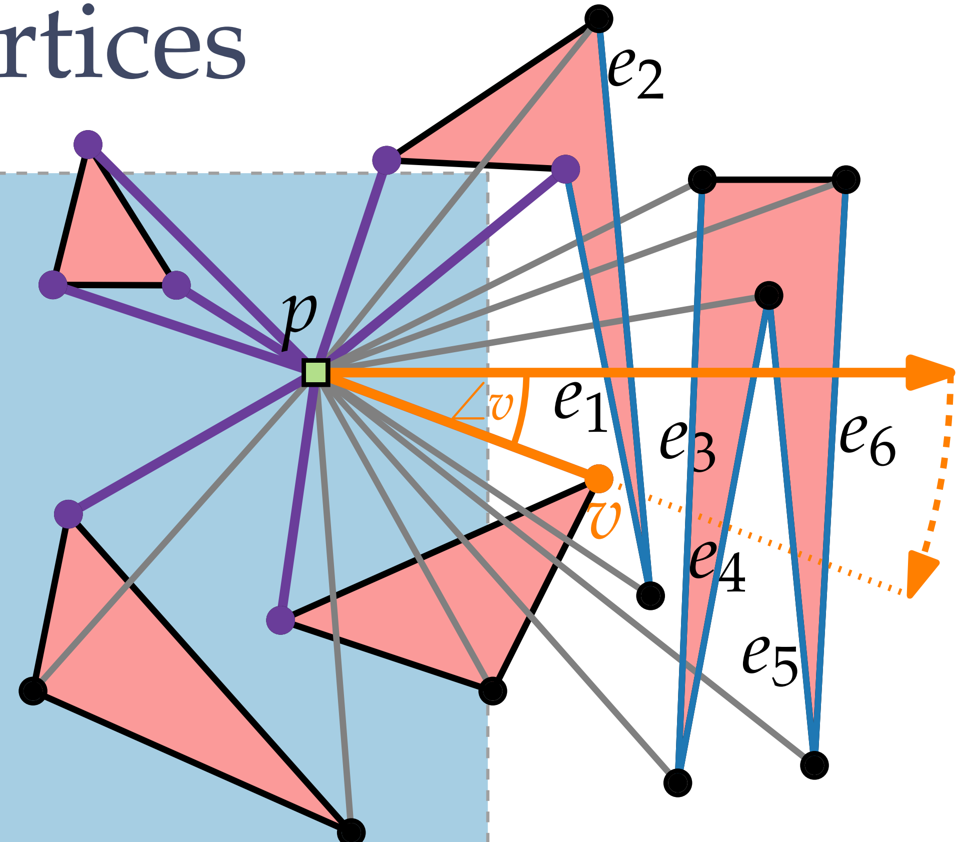
Cases



Let v' be the immediate predecessor of v according to $\prec$.

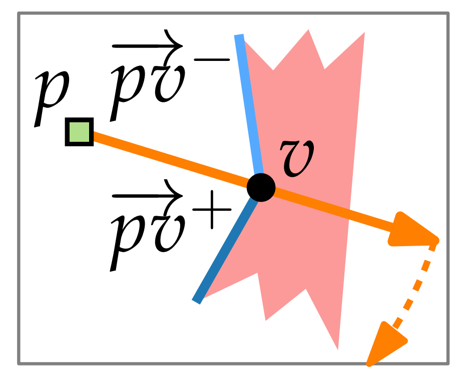


Computing Visible Vertices



VisibleVertices(p, S)

- $r \leftarrow p + \mathbb{R}_{\geq 0}\binom{1}{0}$
- $I \leftarrow \{e \in E(S) \mid e \cap r \neq \emptyset\}$
- $\mathcal{T} \leftarrow \text{balancedBinaryTree}(I)$
- sort $V(S)$ — $v \prec v' :\Leftrightarrow \angle v < \angle v'$ or $(\angle v = \angle v'$ and $|pv| < |pv'|)$
- $W \leftarrow \emptyset$
- **foreach** $v \in V(S)$ **do**
  - **if** Visible(v) **then** $O(1)$
    - $W \leftarrow W \cup \{v\}$
  - insert into $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^{+}$
  - delete from $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^{-}$
- **return** W



$O(n \log n)$

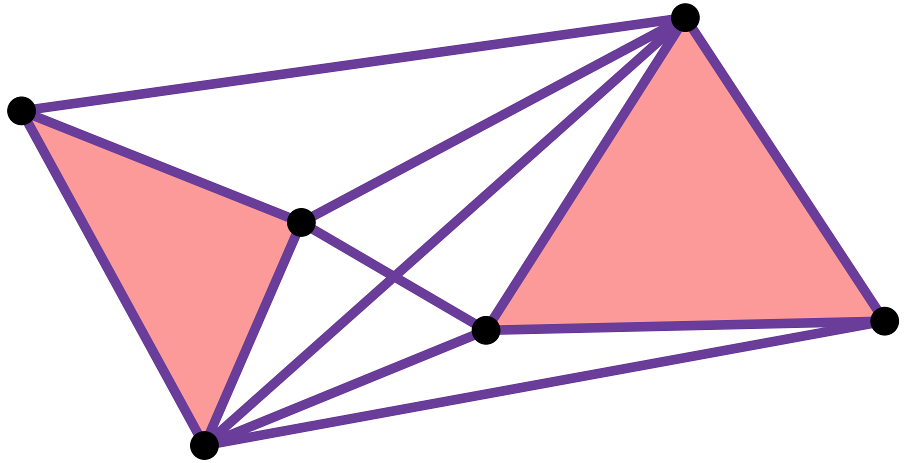
Computing the Visibility Graph

VisibilityGraph(S)

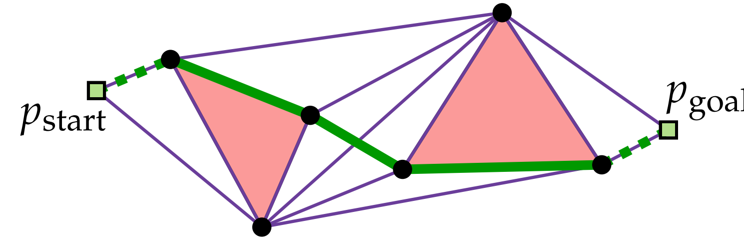
Input: a set S of disjoint polygons
Output: $G_{\text{vis}}(S)$

$E \leftarrow \emptyset$
foreach $v \in V(S)$ **do**
  $W =$ VisibleVertices(v, S)
  $E \leftarrow E \cup \{vw \mid w \in W\}$
return $(V(S), E)$

$O(n) \cdot$
$O(n \log n)$



Algorithm



ShortestPath($S, p_{\text{start}}, p_{\text{goal}}$) $n = |V(S)|$, $m = |E_{\text{vis}}(S)|$

$G_{\text{vis}} \leftarrow$ VisibilityGraph($S \cup \{p_{\text{start}}, p_{\text{goal}}\}$) $O(n^2 \log n)$

foreach $uv \in E_{\text{vis}}$ **do** $O(m)$

$w(uv) = d_{\text{Eucl.}}(u, v)$

$\pi \leftarrow$ Dijkstra($G_{\text{vis}}, w, p_{\text{start}}, p_{\text{goal}}$) $O(m + n \log n)$

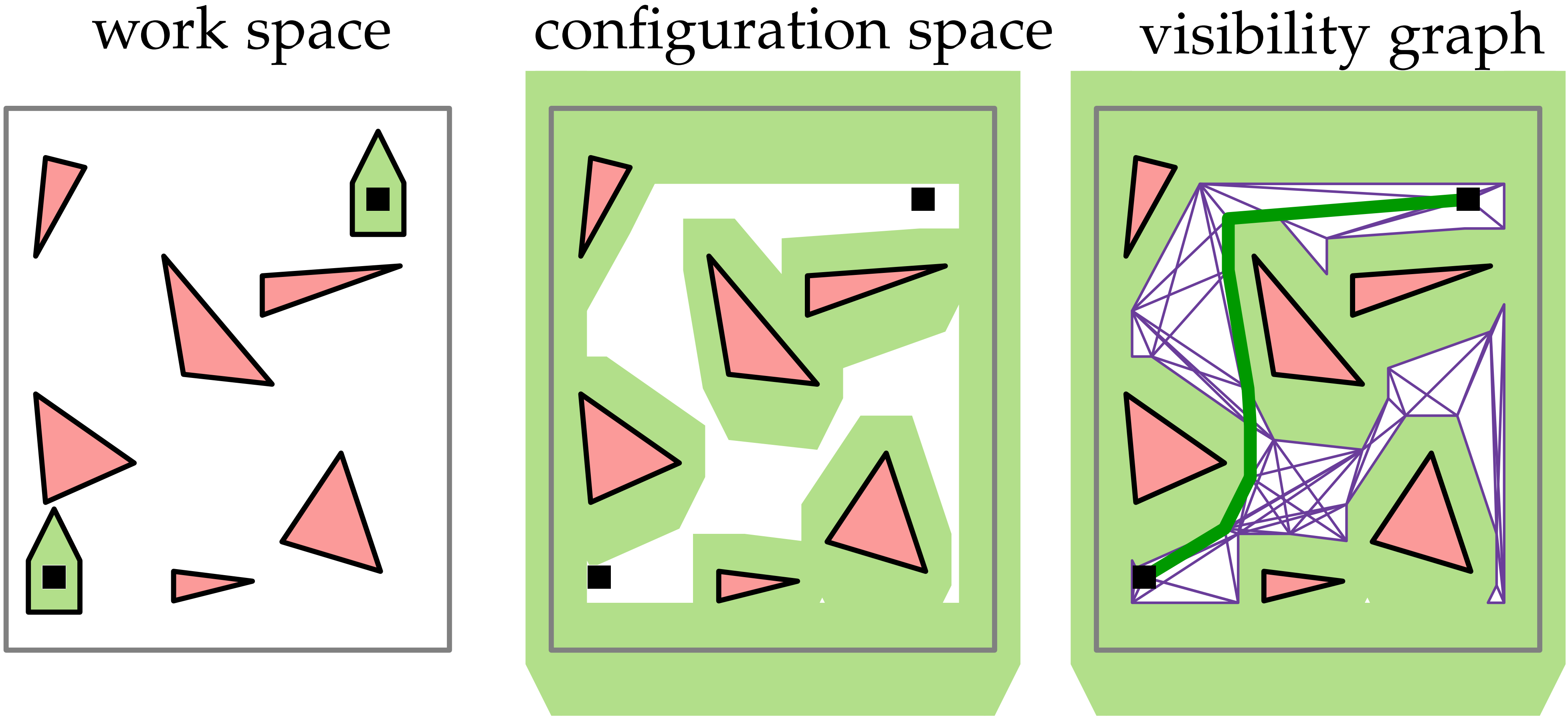
return π

Running time? $O(n^2 \log n)$

Theorem. The visibility graph of a set of disjoint polygonal obstacles with n edges in total can be computed in $O(n^2 \log n)$ time.

Theorem. A shortest path between two points among a set of [...] can be computed in $O(n \log n + m)$ time with $O(n^2 \log n)$ preproc.

Translating Polygonal Robots



Theorem: For a convex constant-complexity translating robot, a shortest collision-free path among a set of polygonal obstacles with n edges in total can be computed in $O(n^2 \log n)$ time.