Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Chair for
INFORMATICS I
Efficient Algorithms and
Knowledge-Based Systems

Institute for Informatics

Computational Geometry

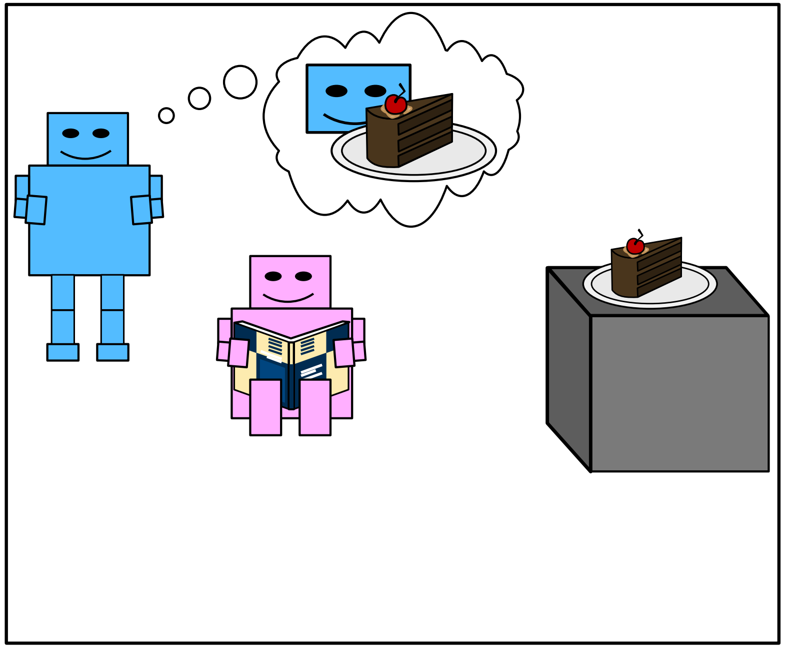
Visibility Graphs
or
Finding Shortest Paths

Lecture #12

Dr. Philipp Kindermann

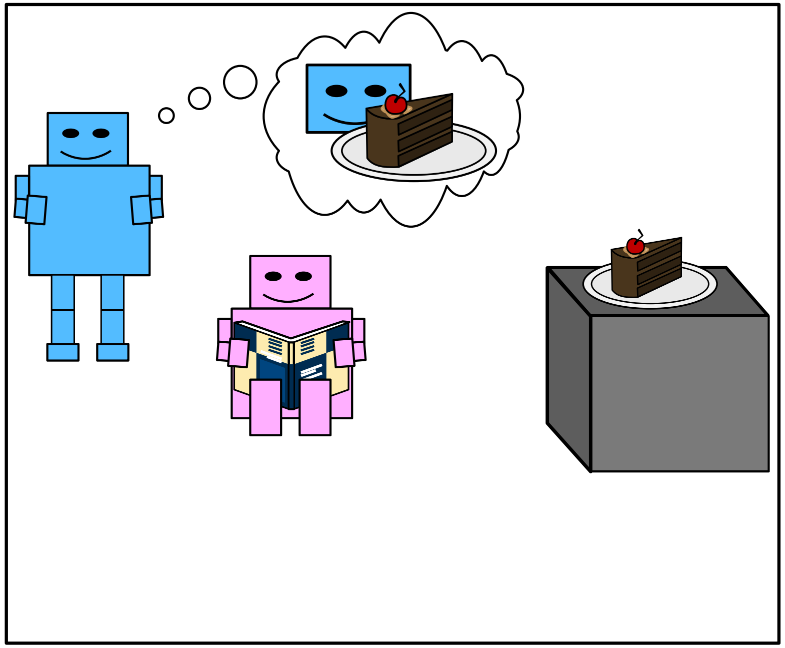
Winter Semester 2018/19

Path Planning



current location,
desired location

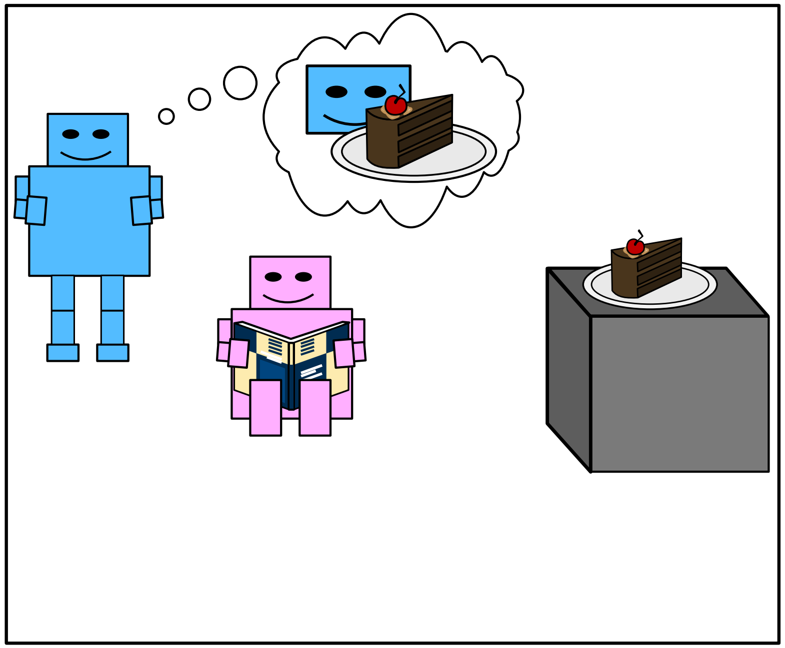
Path Planning



$\Rightarrow$

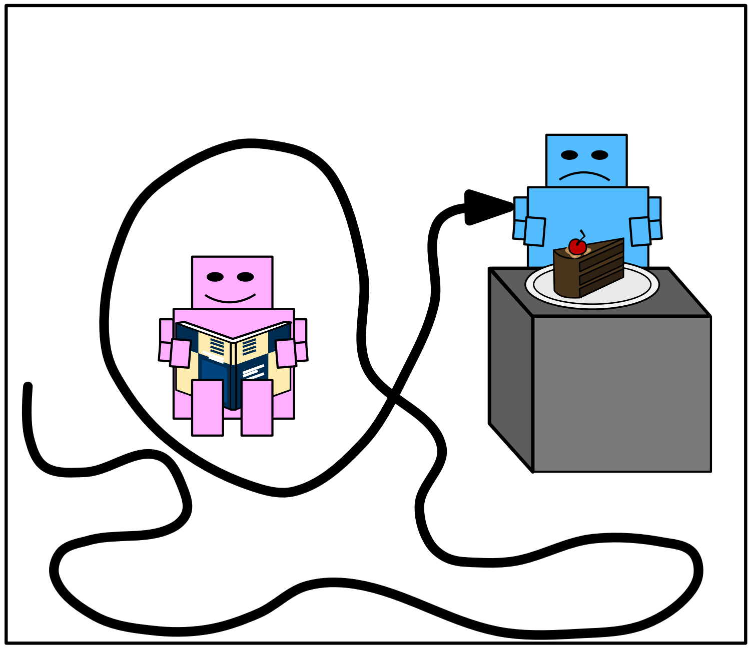
current location,
desired location

Path Planning



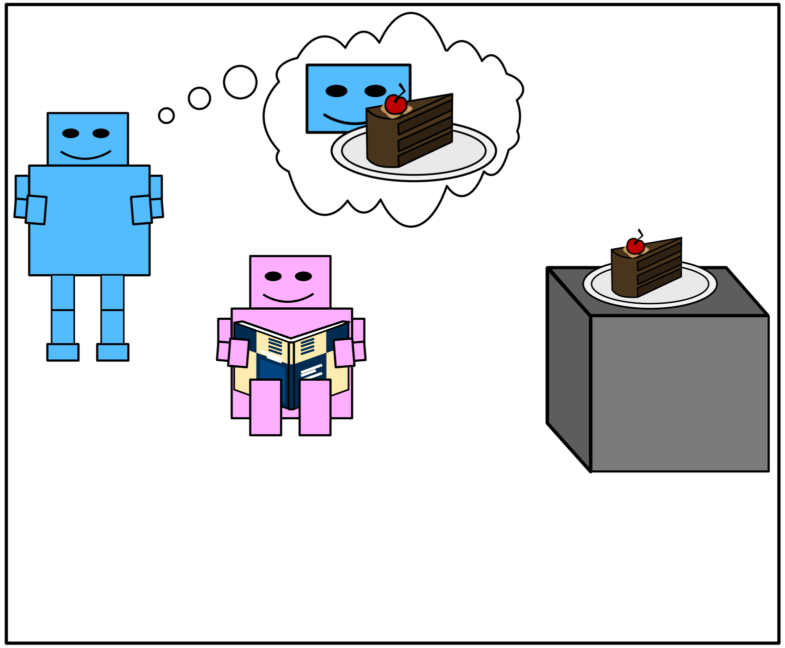
current location,
desired location

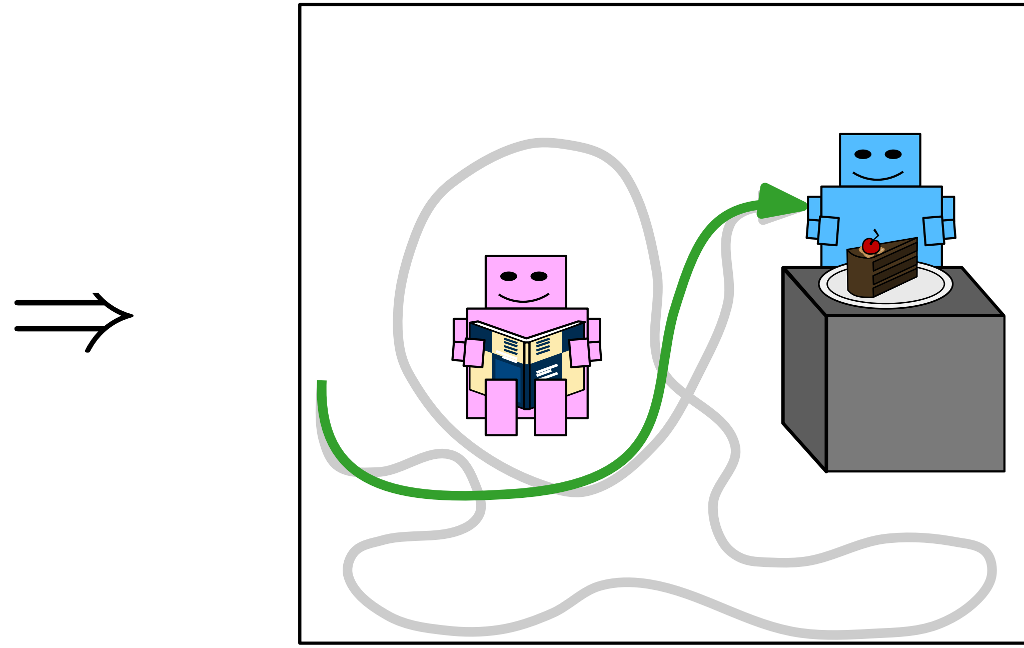
$\Rightarrow$



path to reach the
one from the other

Path Planning

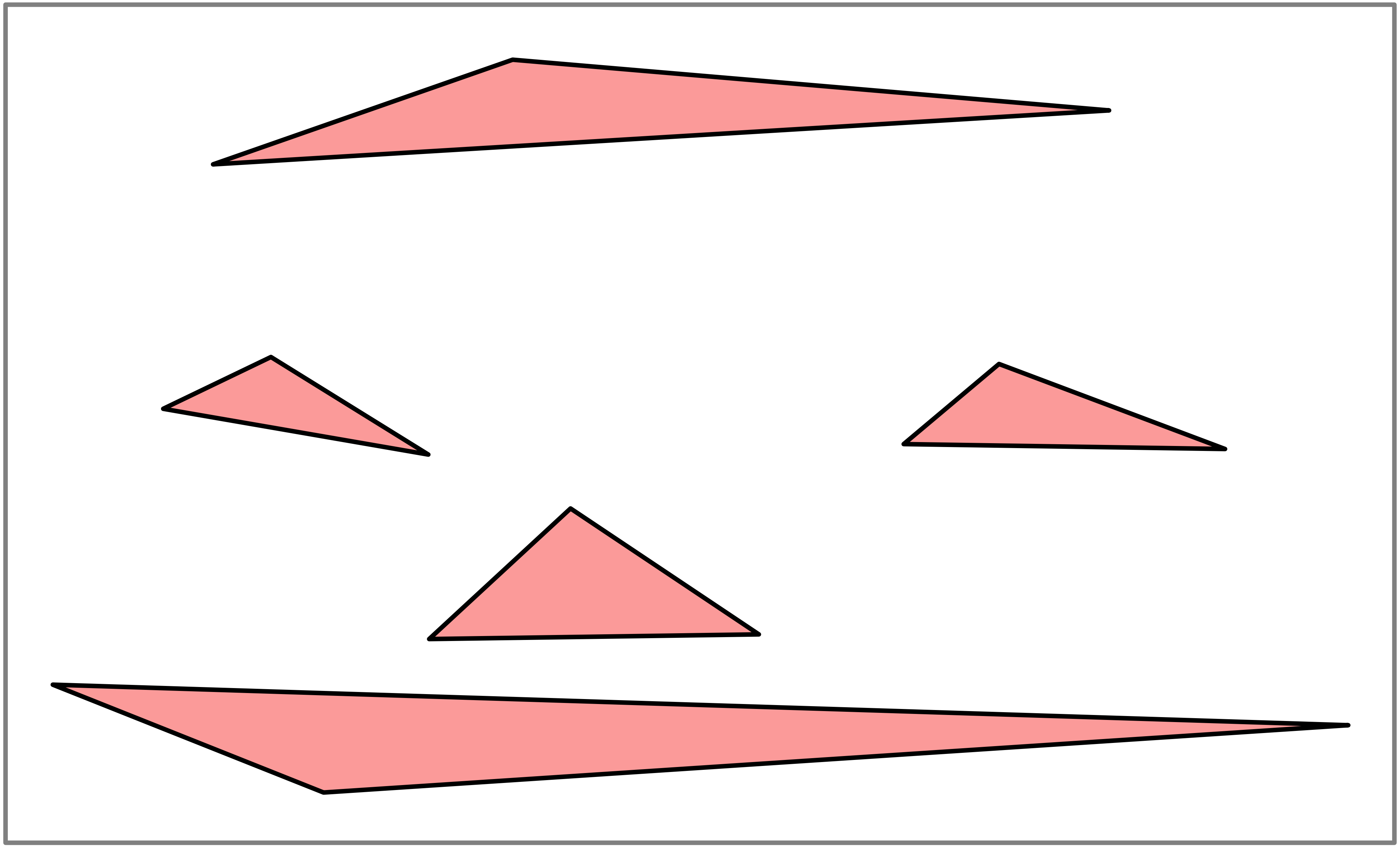




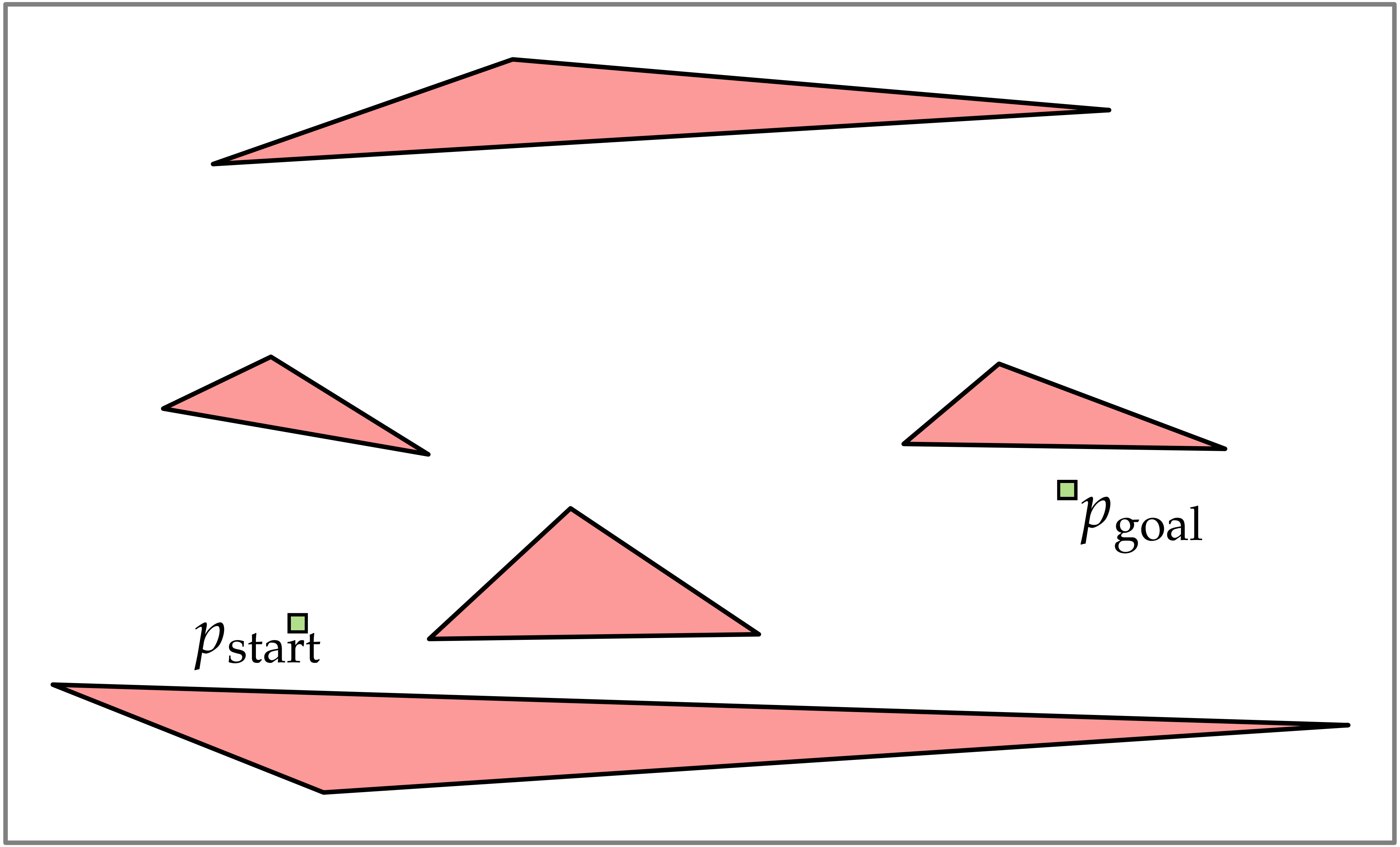
current location,
desired location

shortest path to reach the one from the other

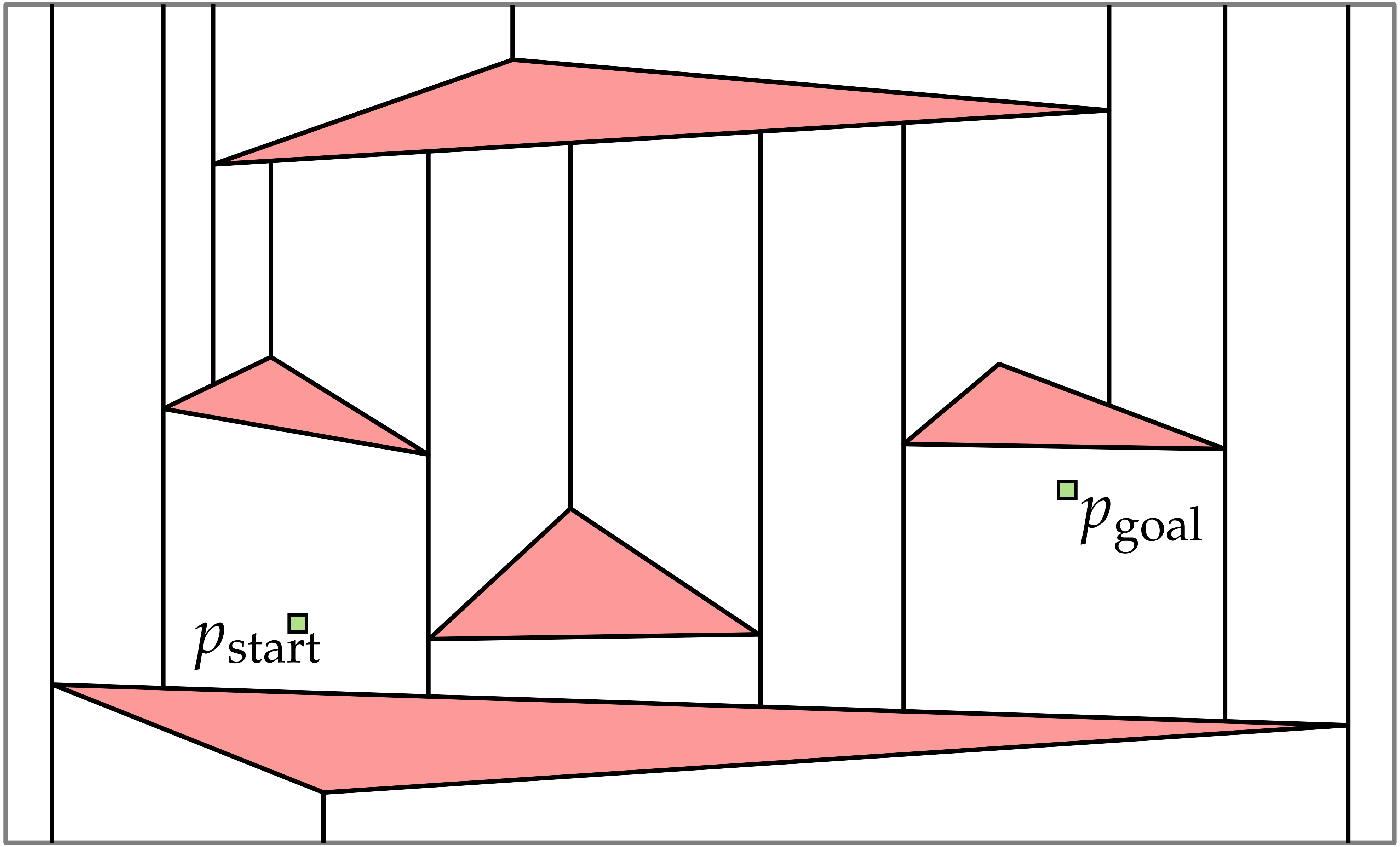
Let's recap...



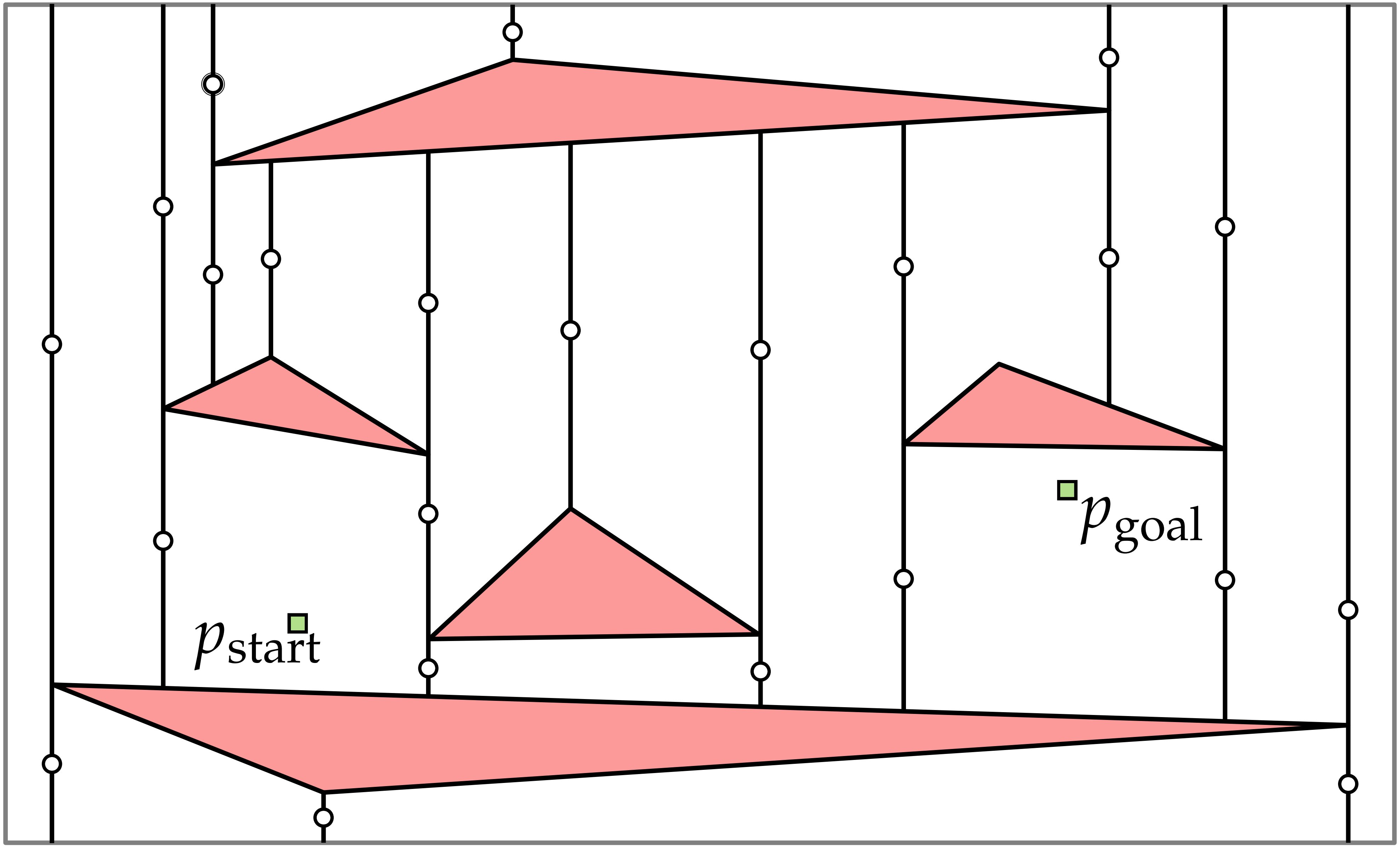
Let's recap...



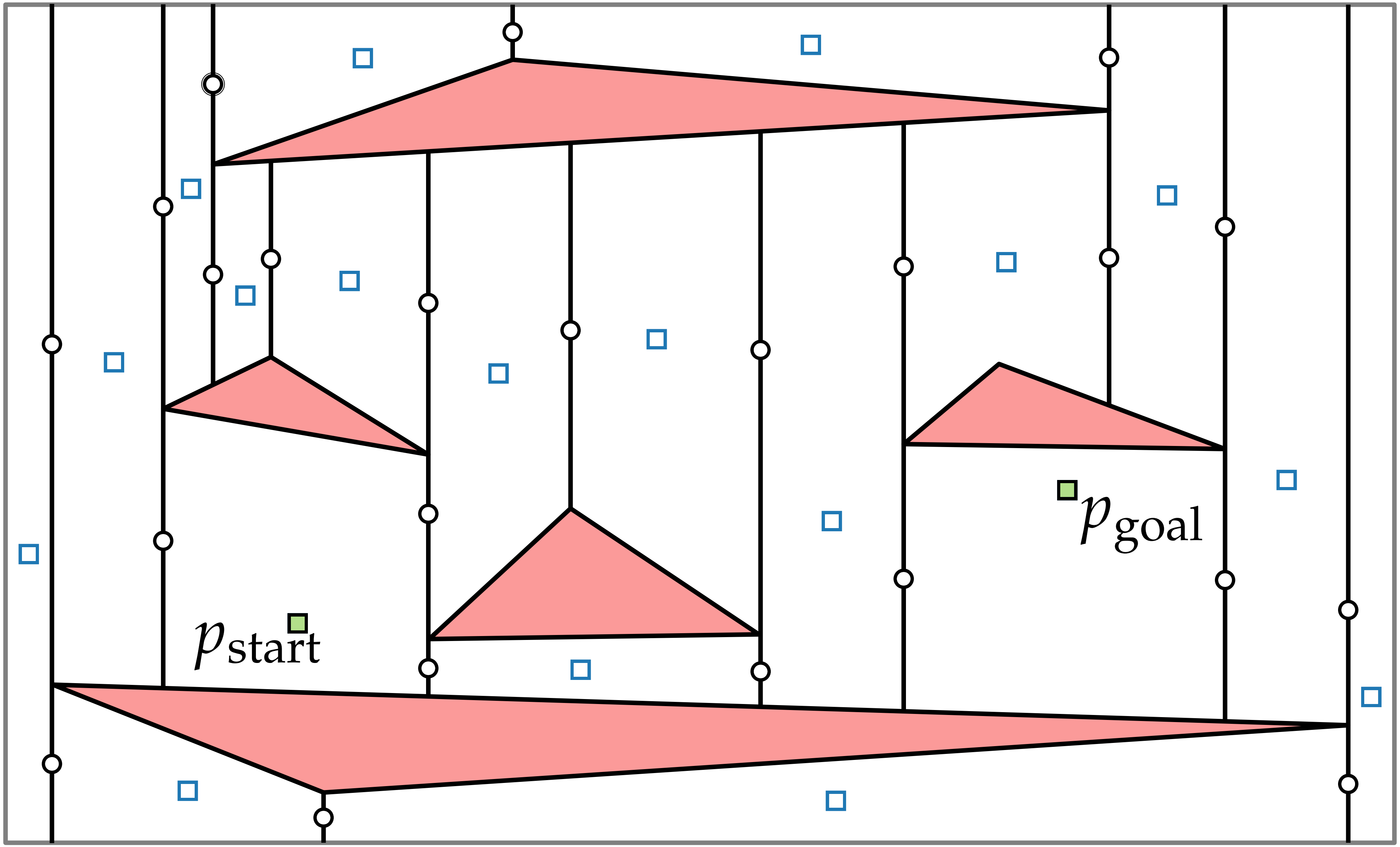
Let's recap...



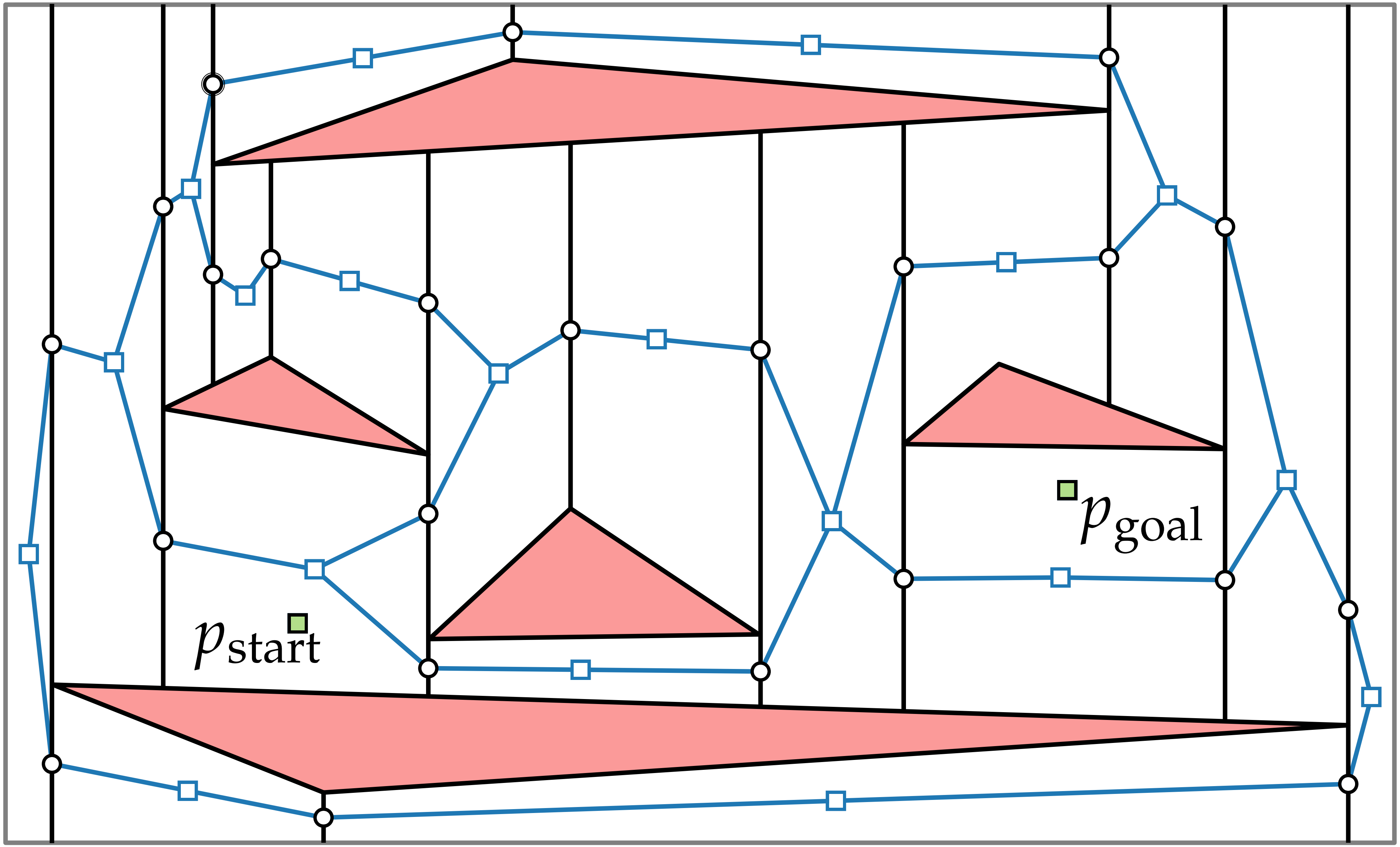
Let's recap...



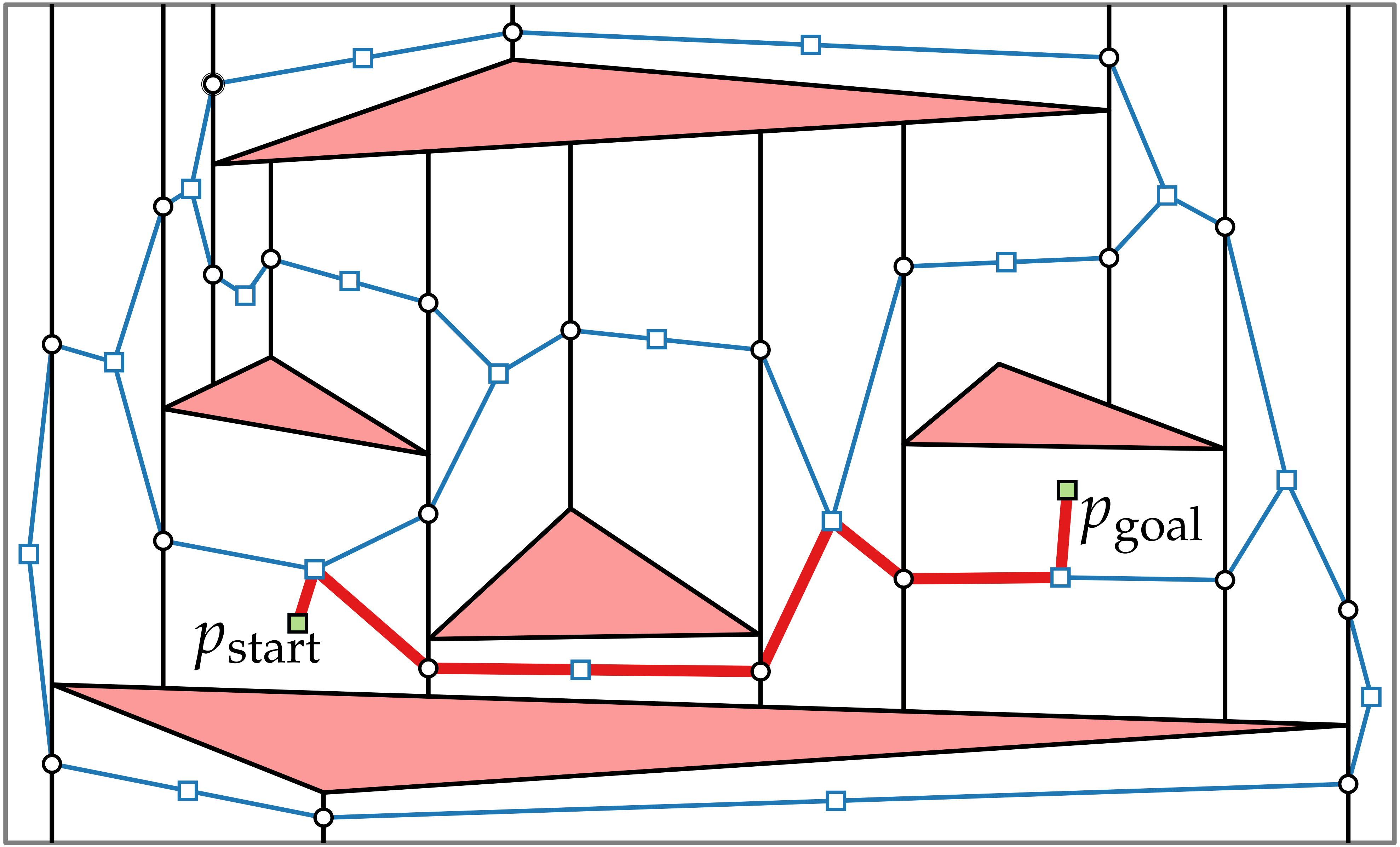
Let’s recap...



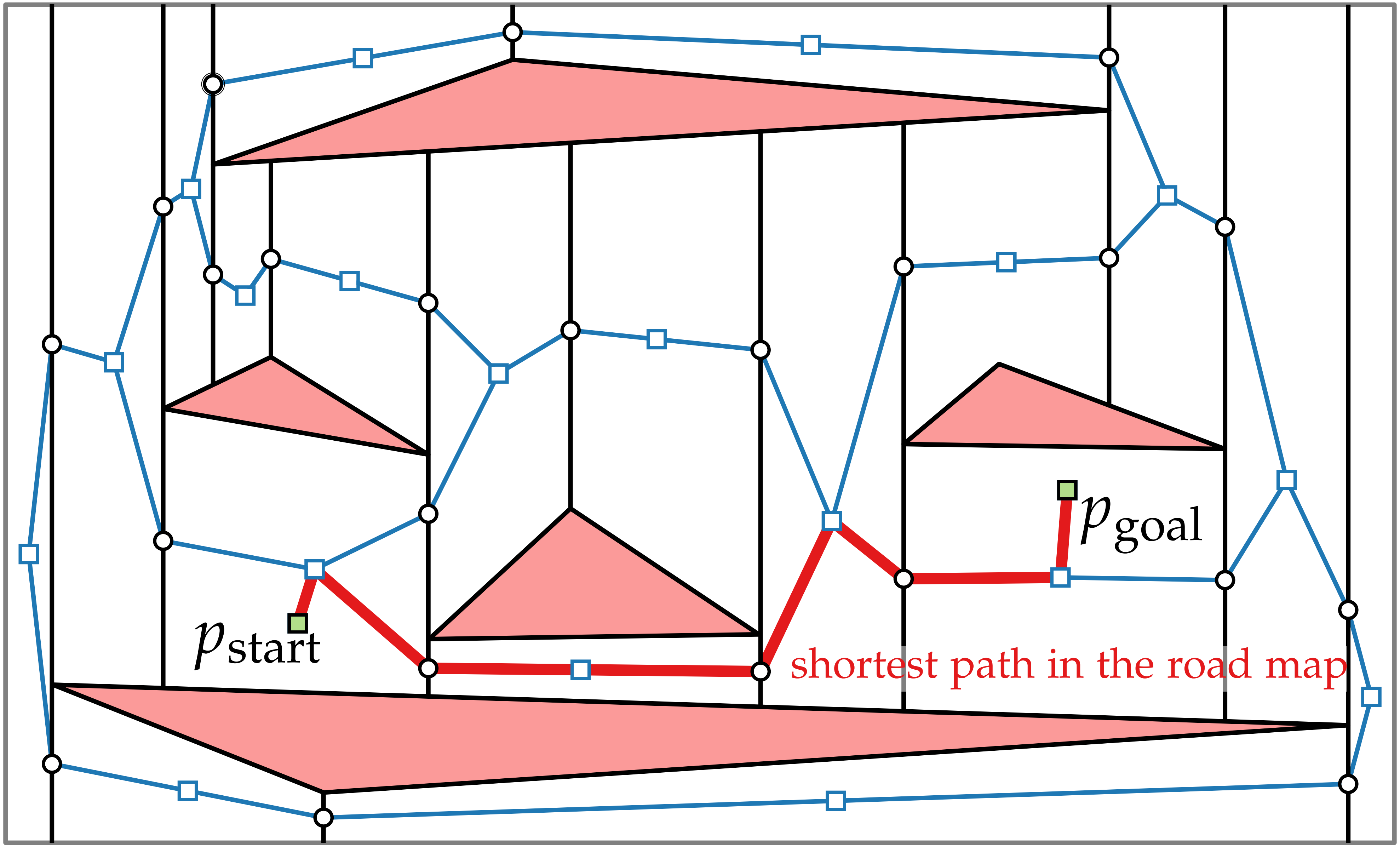
Let's recap...



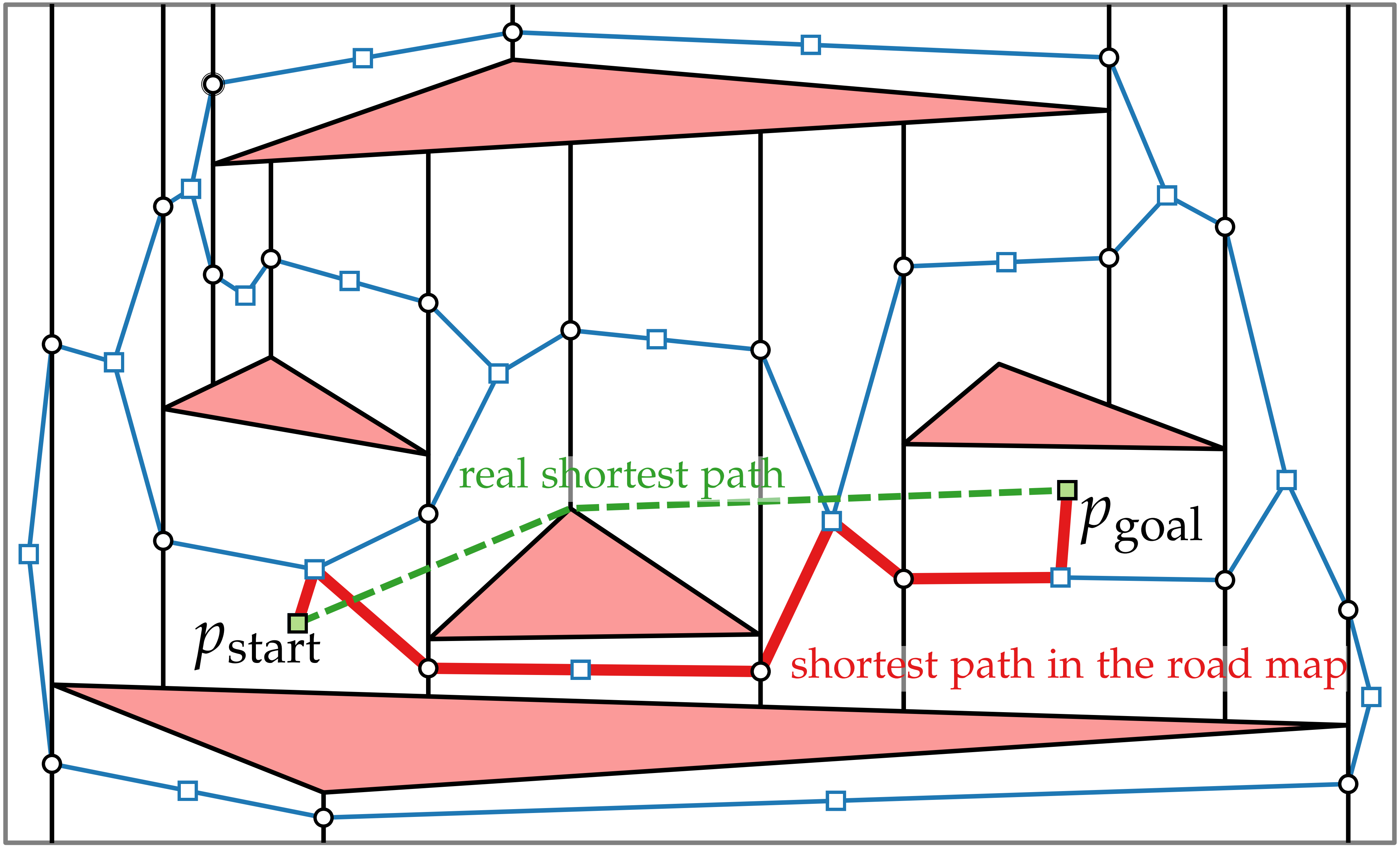
Let's recap...



Let's recap...

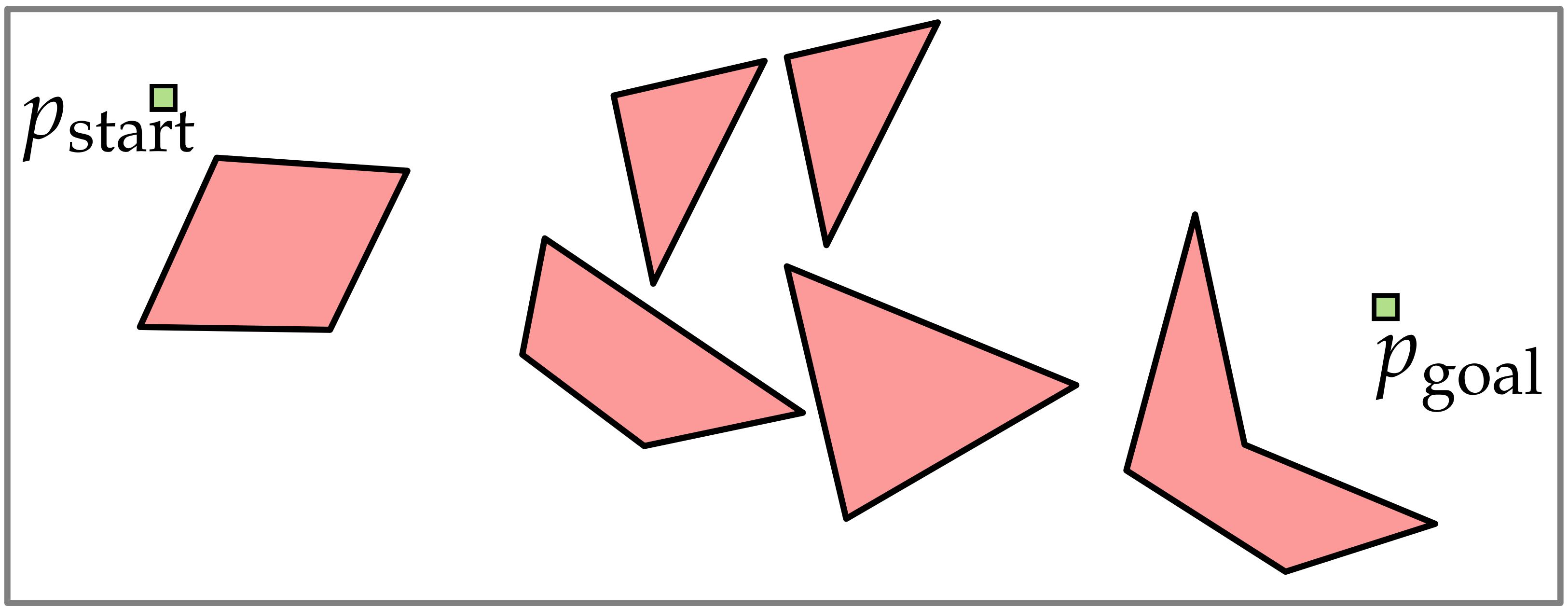


Let's recap...


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

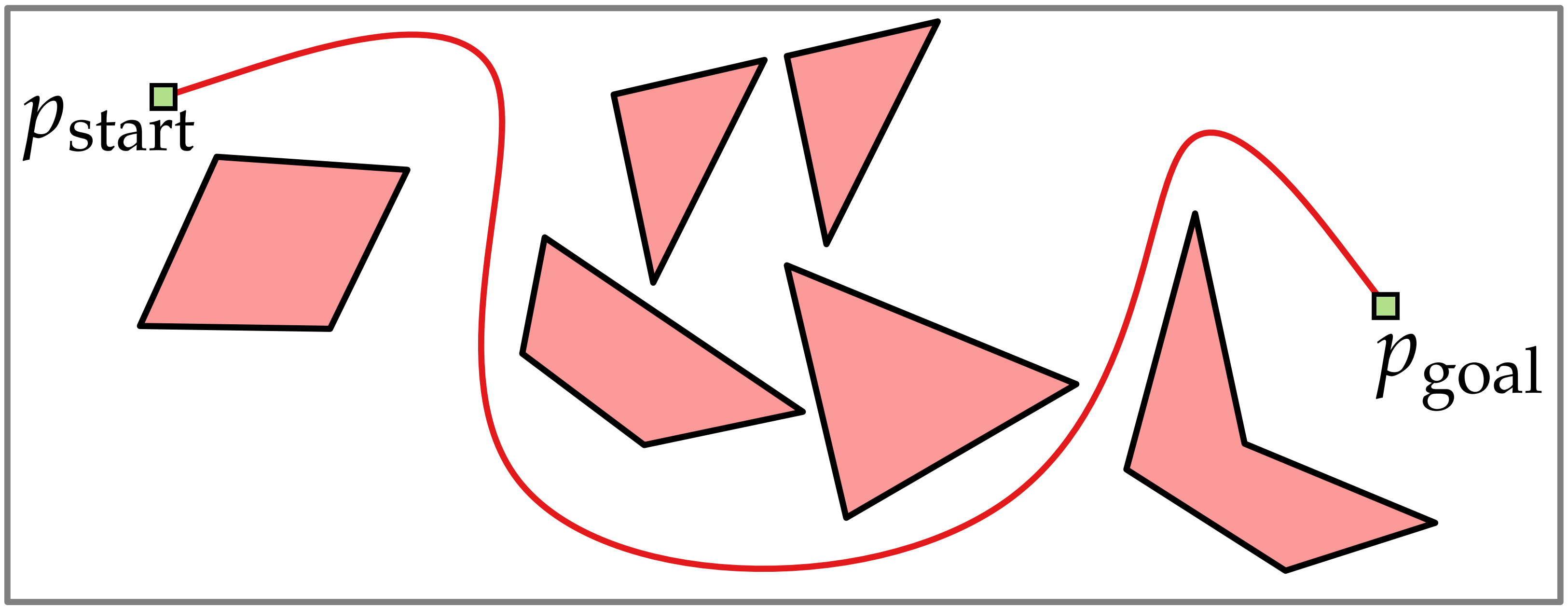
Characterization

Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space,



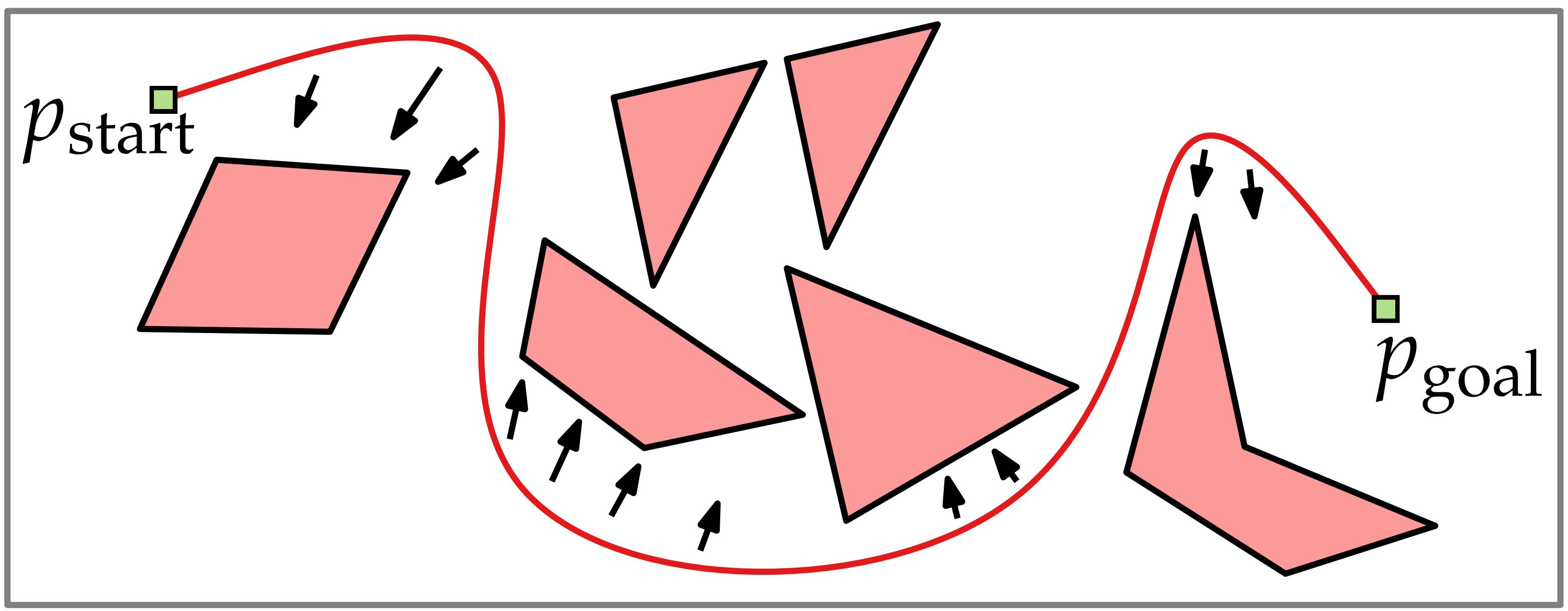
Characterization

Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space,



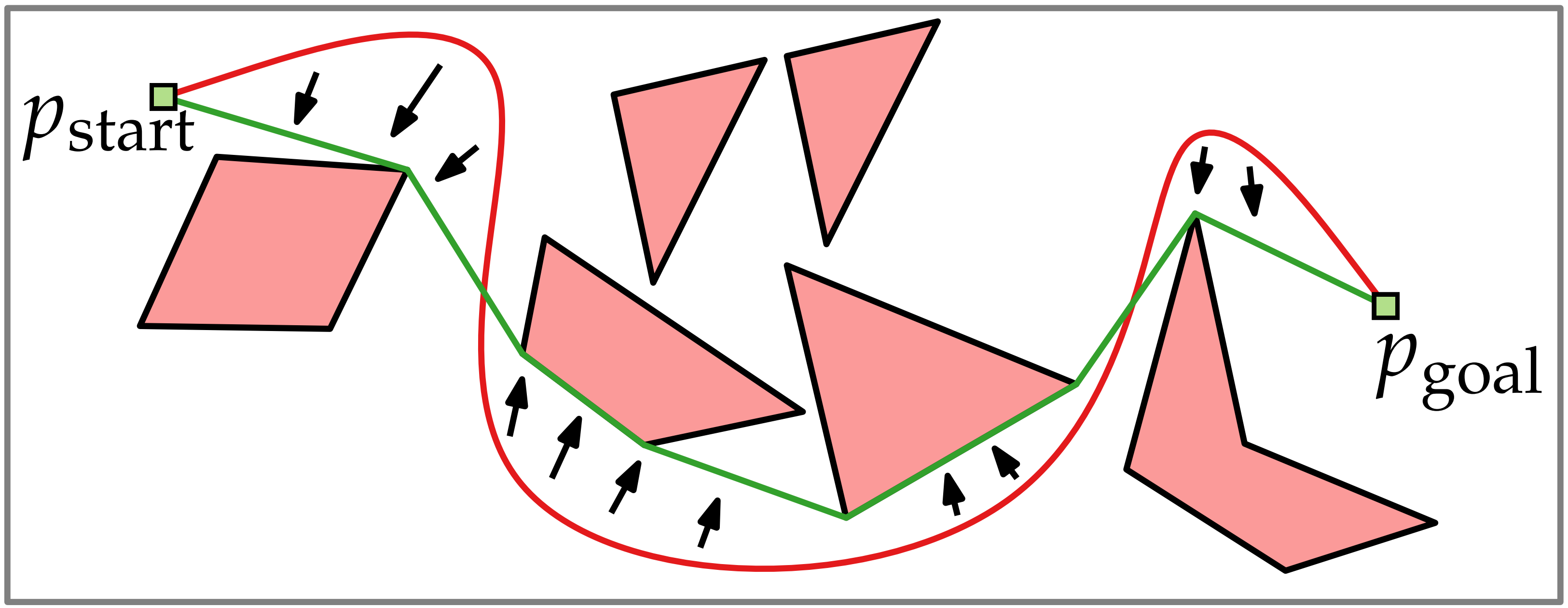
Characterization

Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space,



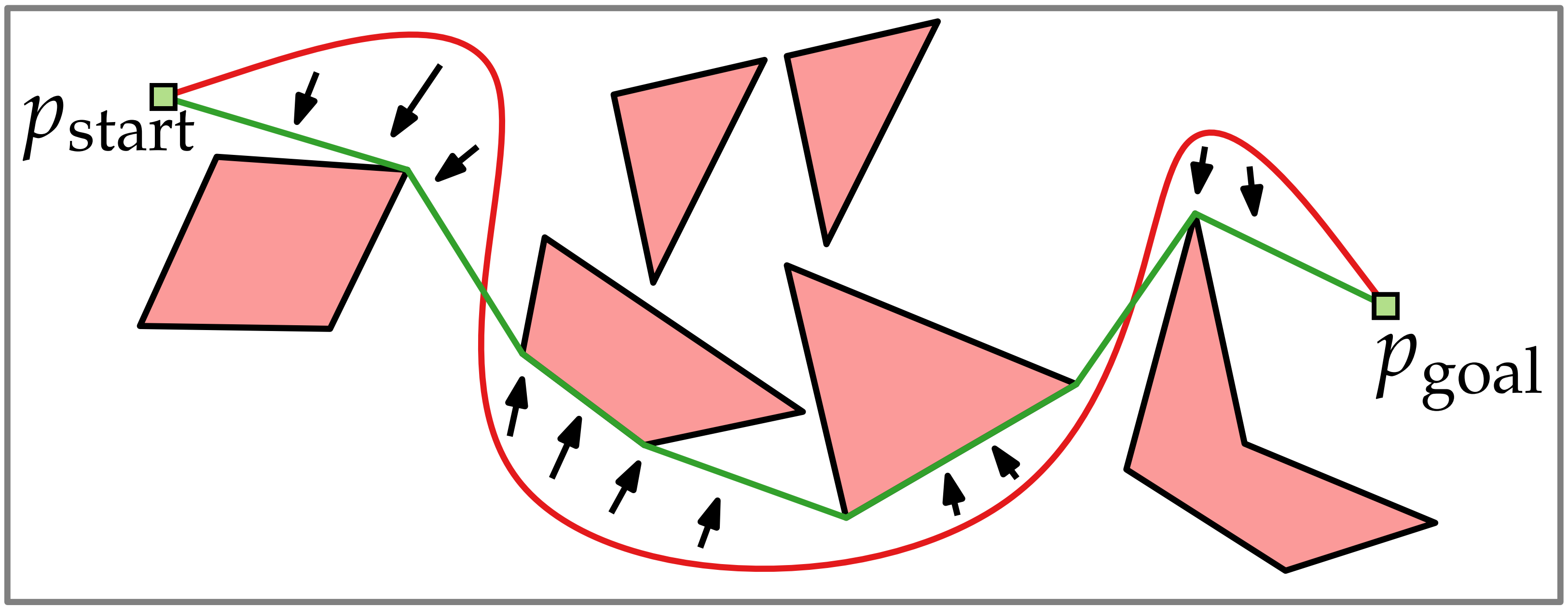
Characterization

Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space,



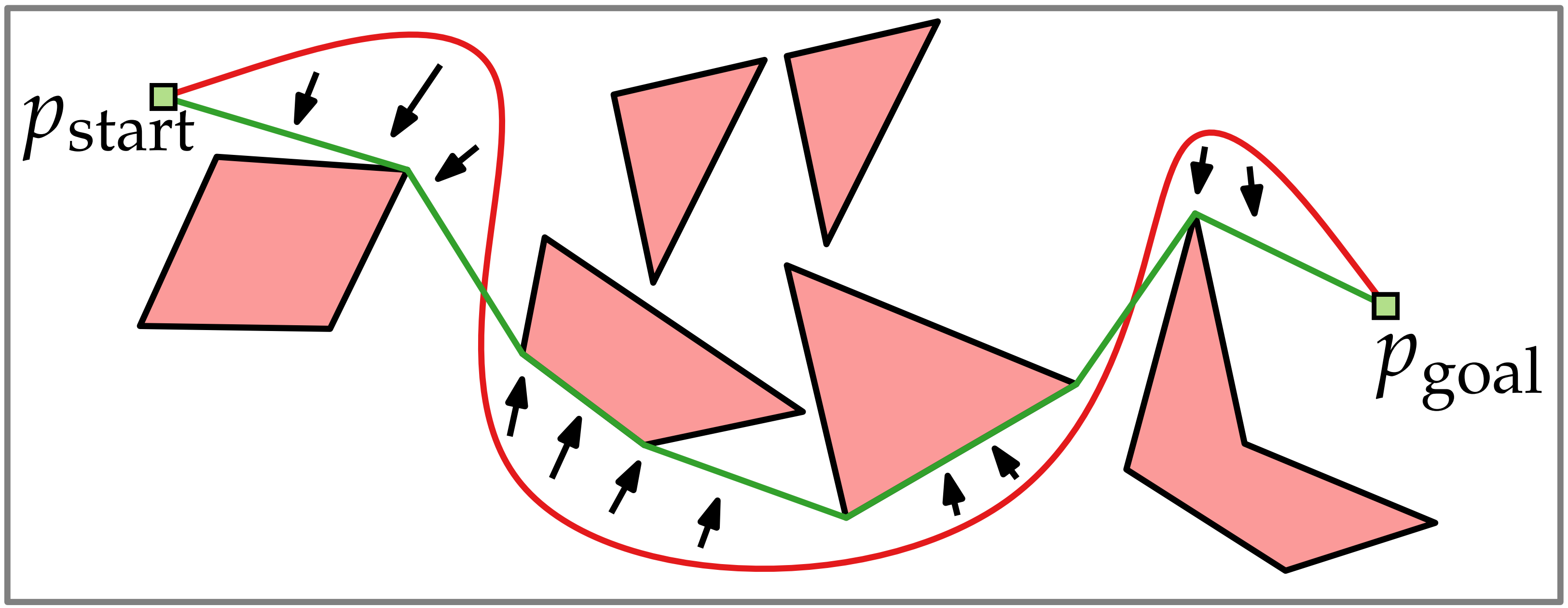
Characterization

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.



Characterization

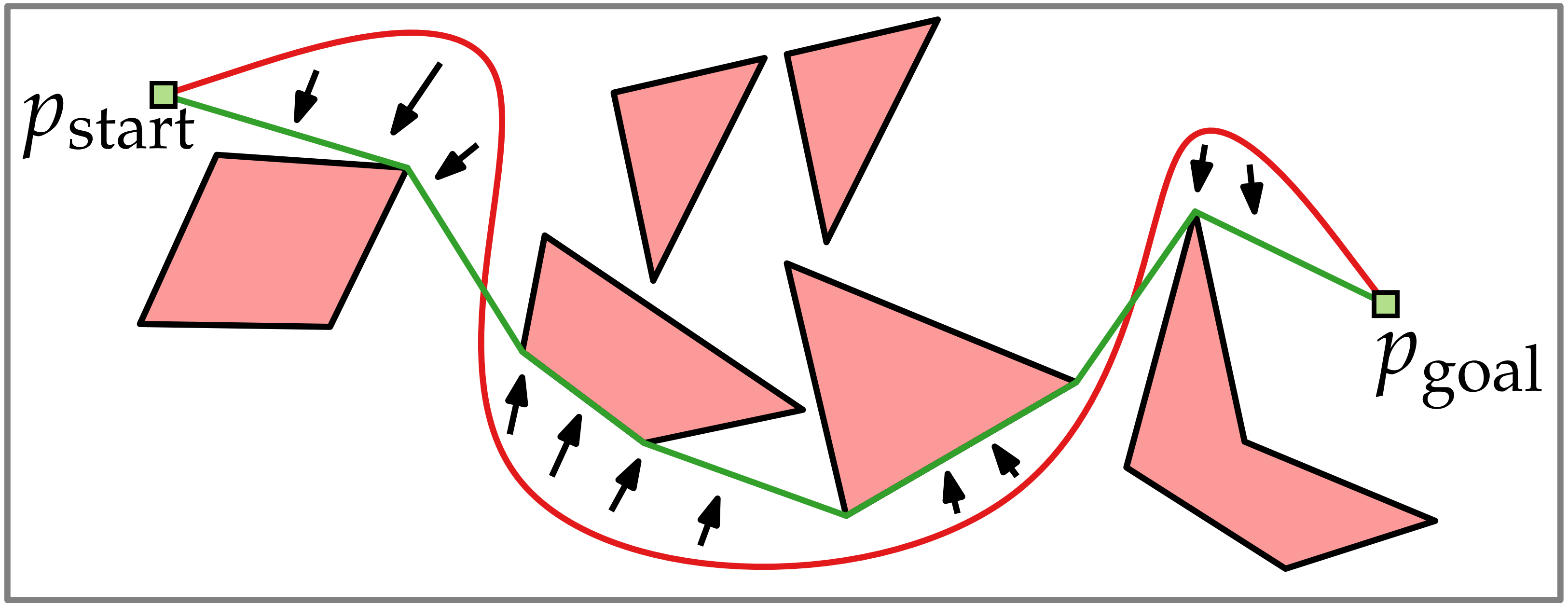
Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space, any shortest path between p_{start} and p_{goal} is a polygonal path whose inner vertices are vertices of S.



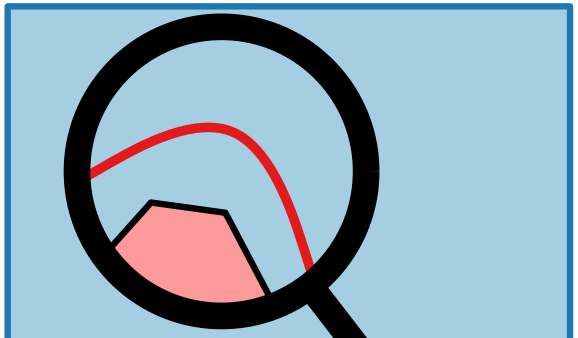
Proof.

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.

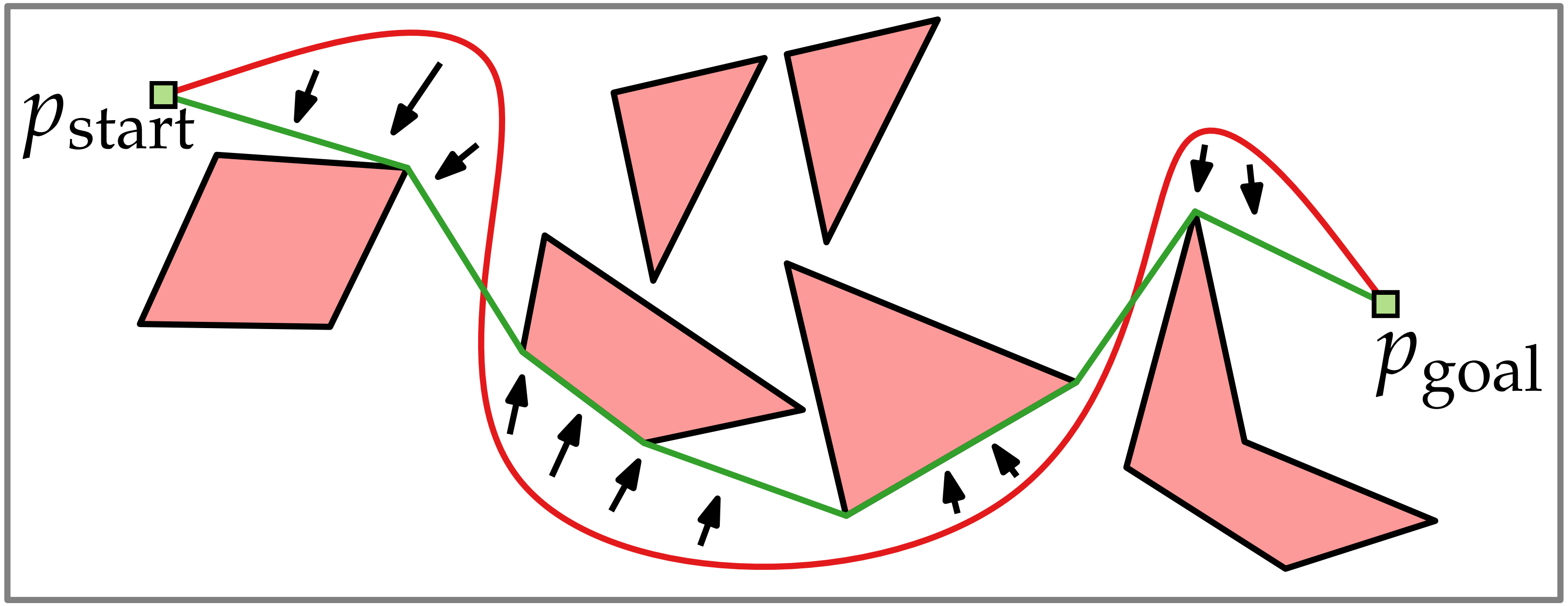


Proof.



Characterization

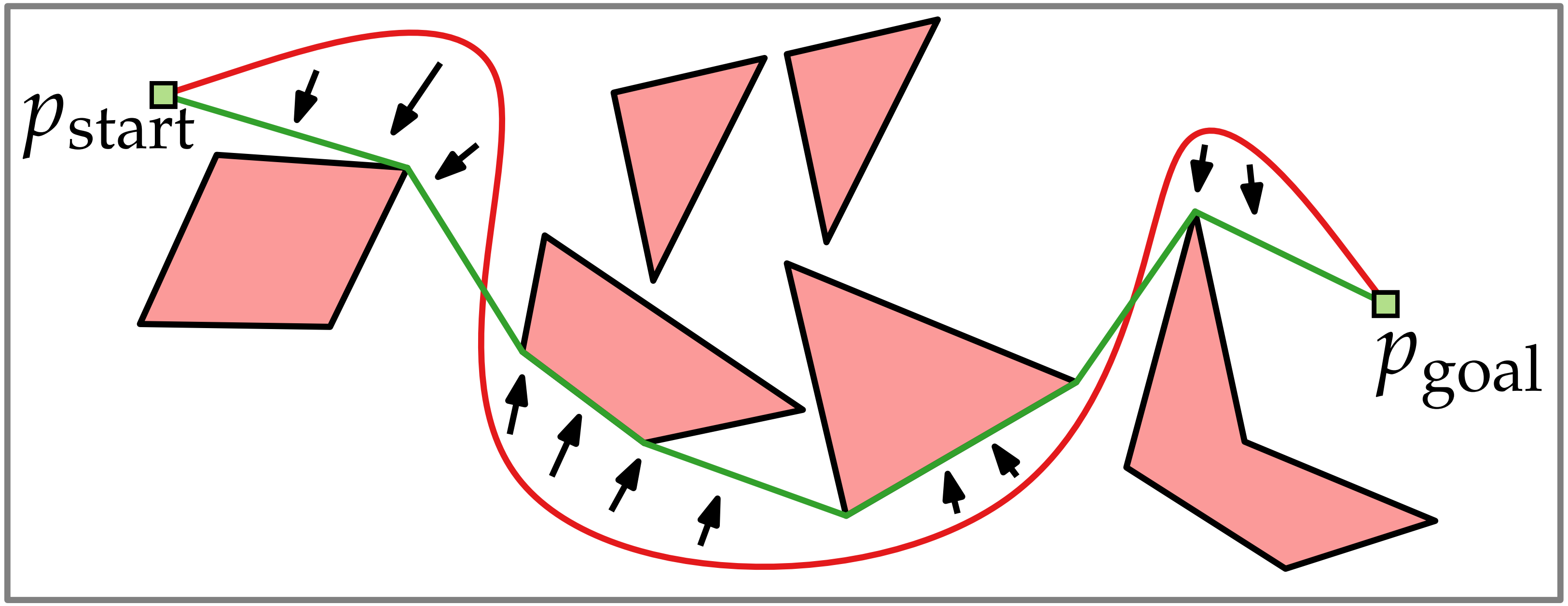
Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space, any shortest path between p_{start} and p_{goal} is a polygonal path whose inner vertices are vertices of S.



Proof.

Characterization

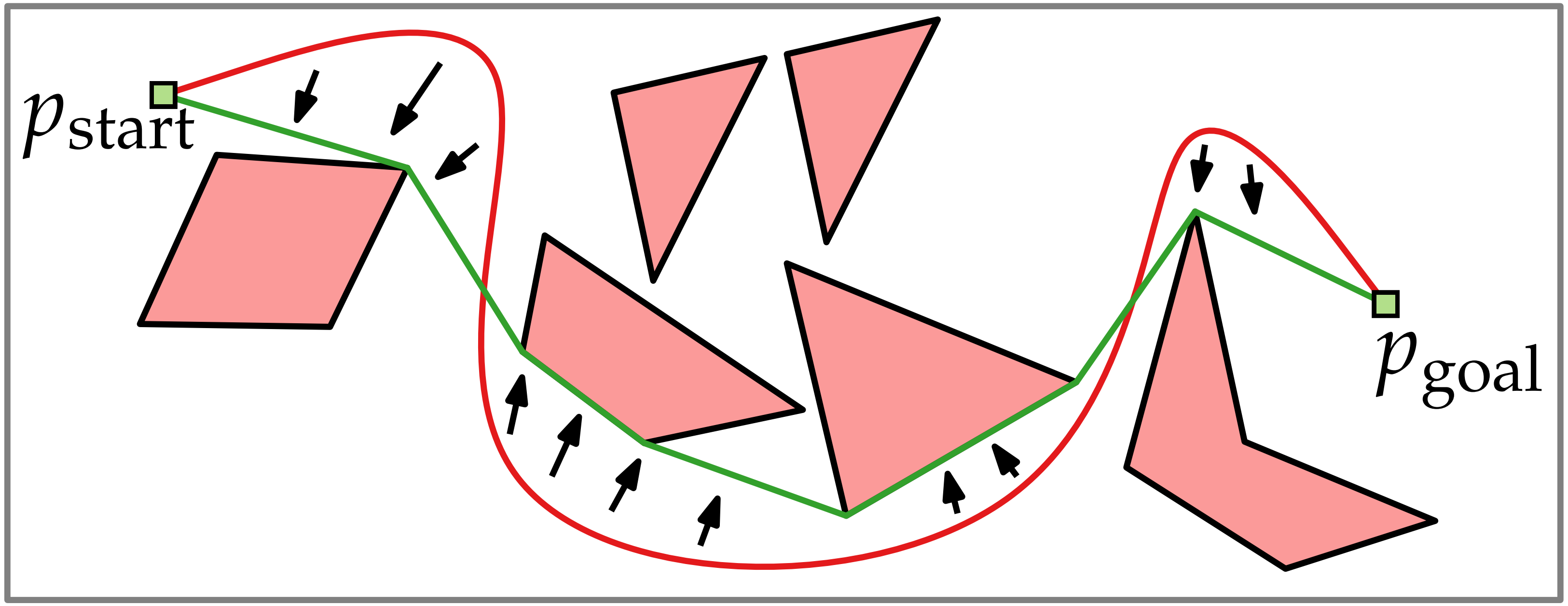
Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space, any shortest path between p_{start} and p_{goal} is a polygonal path whose inner vertices are vertices of S.



Proof.

Characterization

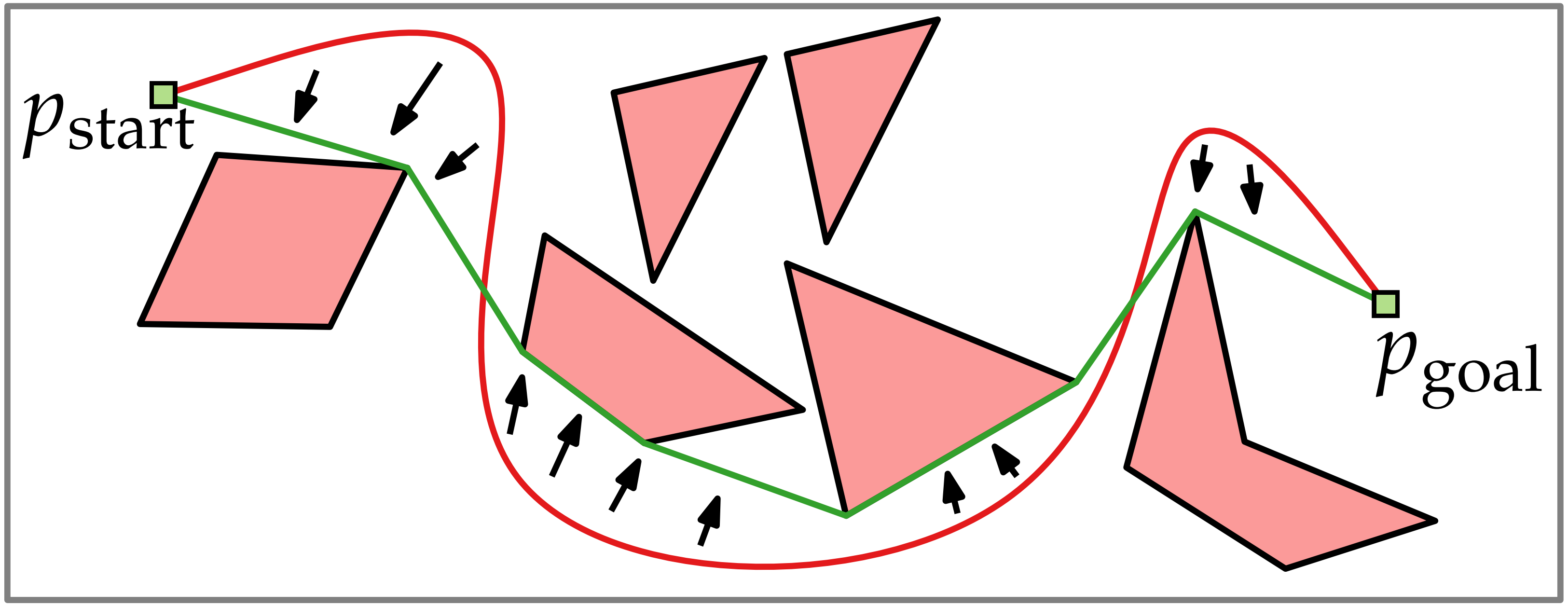
Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space, any shortest path between p_{start} and p_{goal} is a polygonal path whose inner vertices are vertices of S.



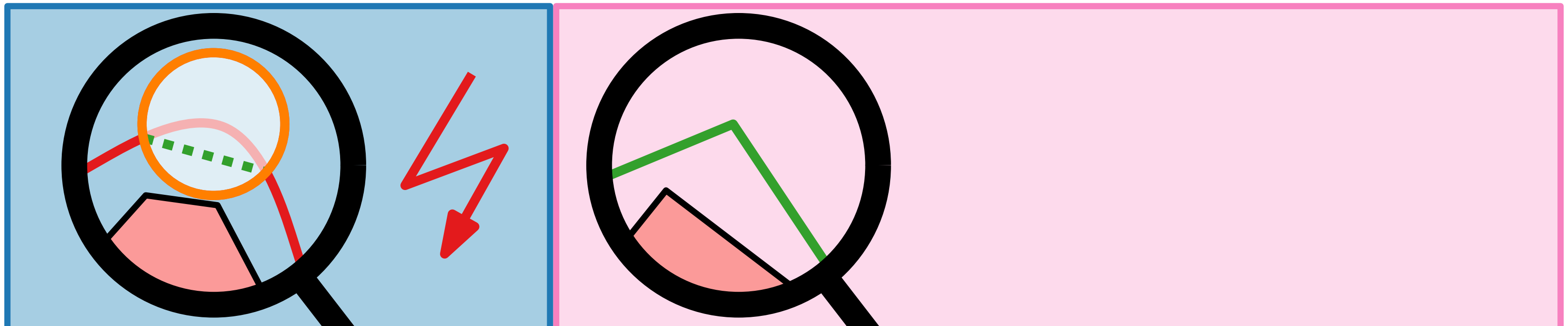
Proof.

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.

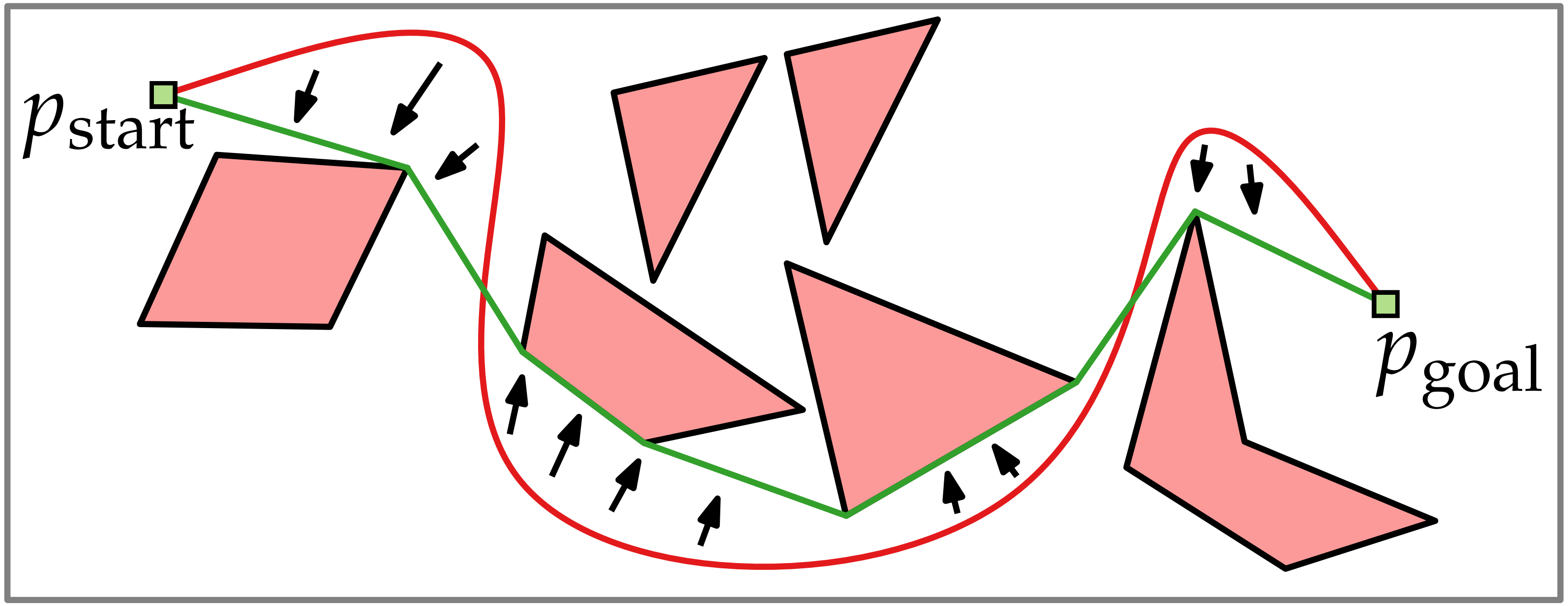


Proof.

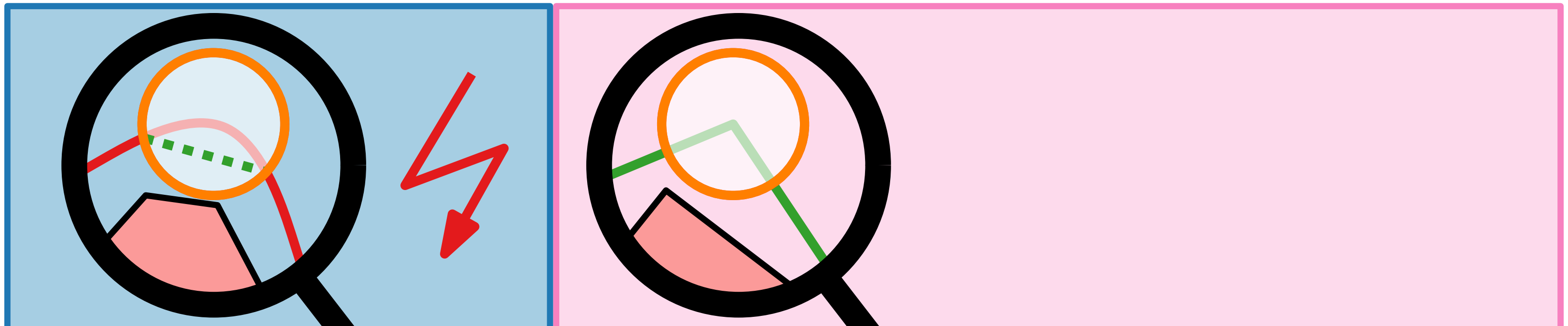


Characterization

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.

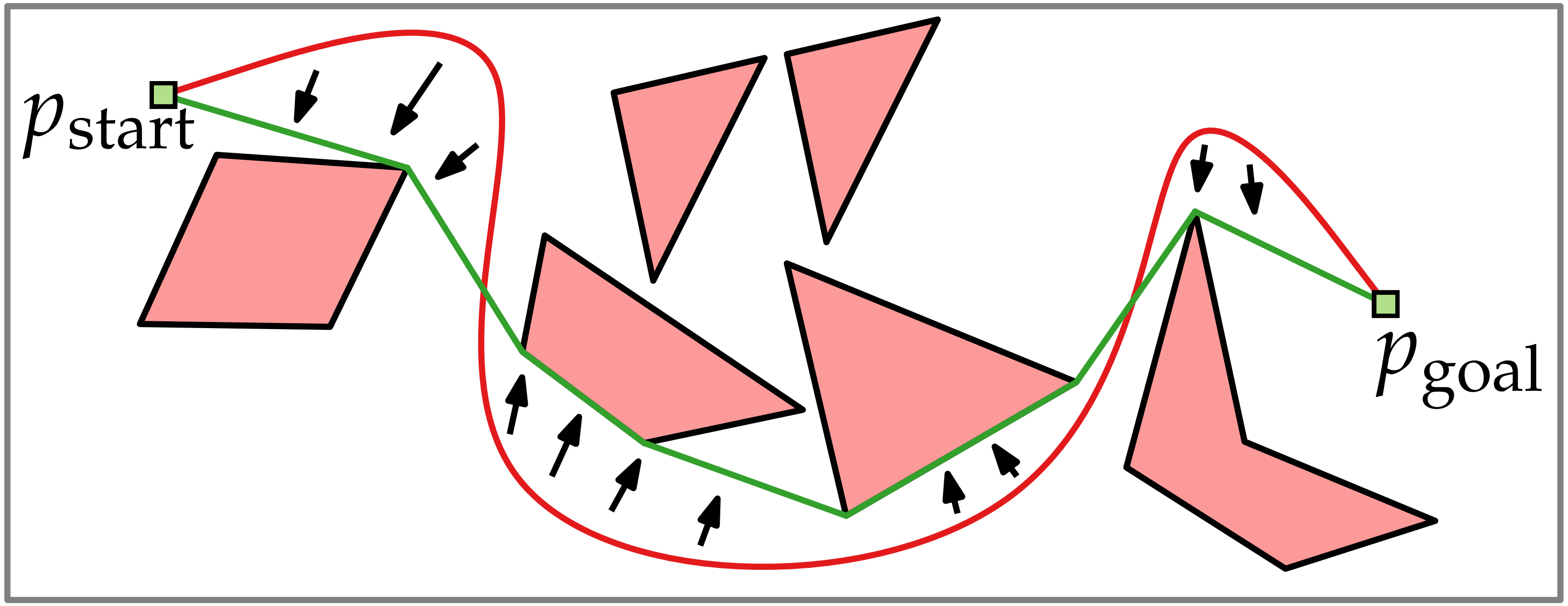


Proof.

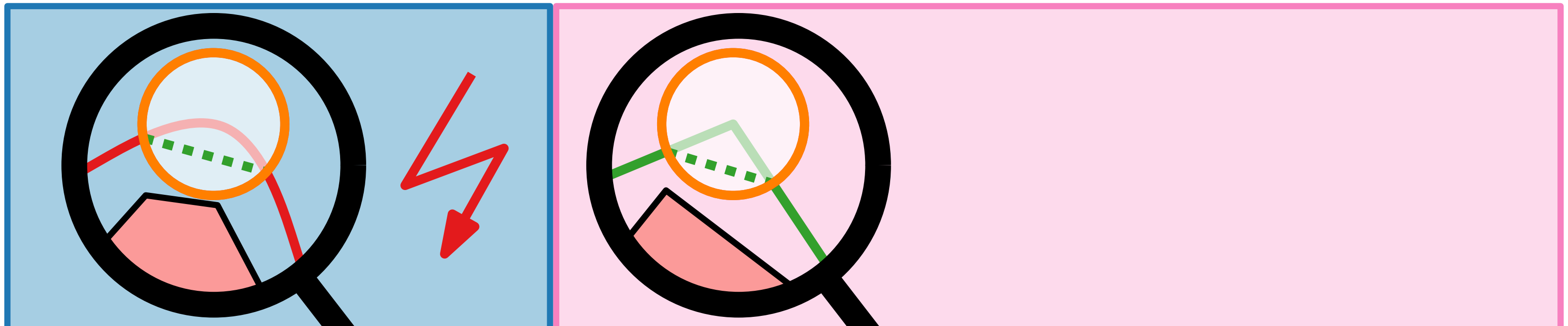


Characterization

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.

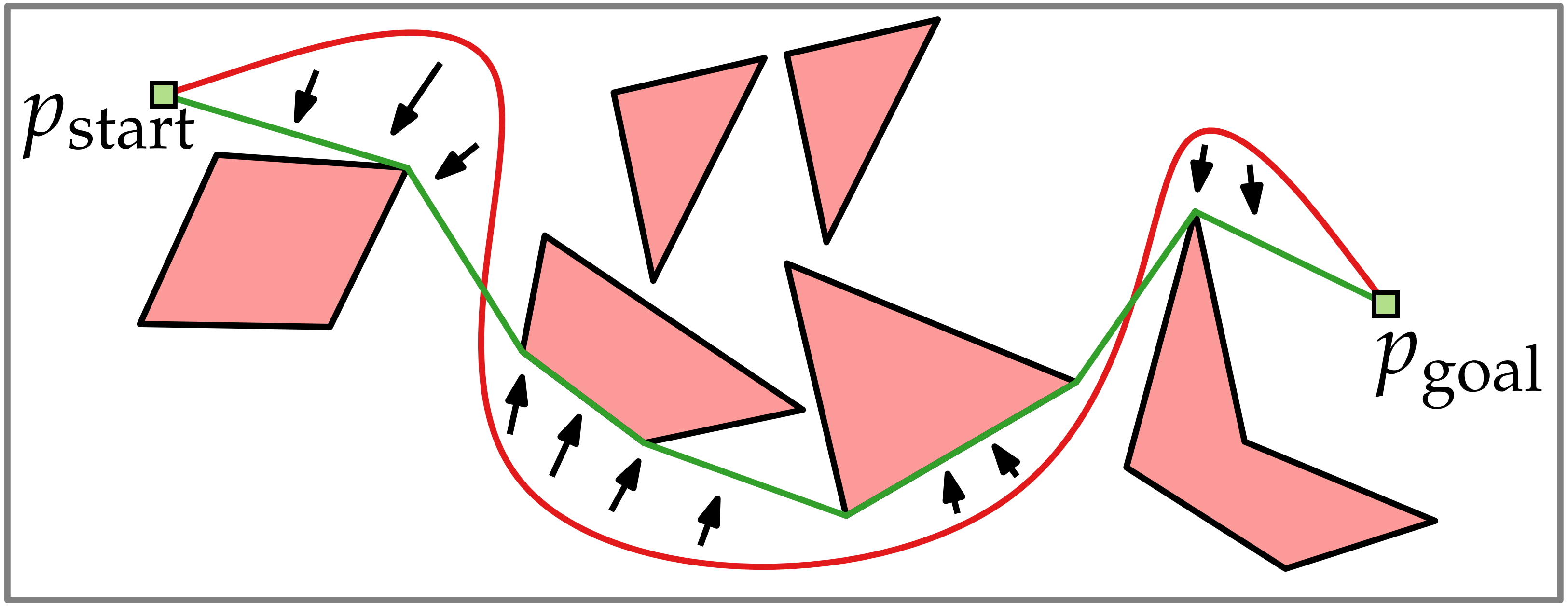


Proof.



Characterization

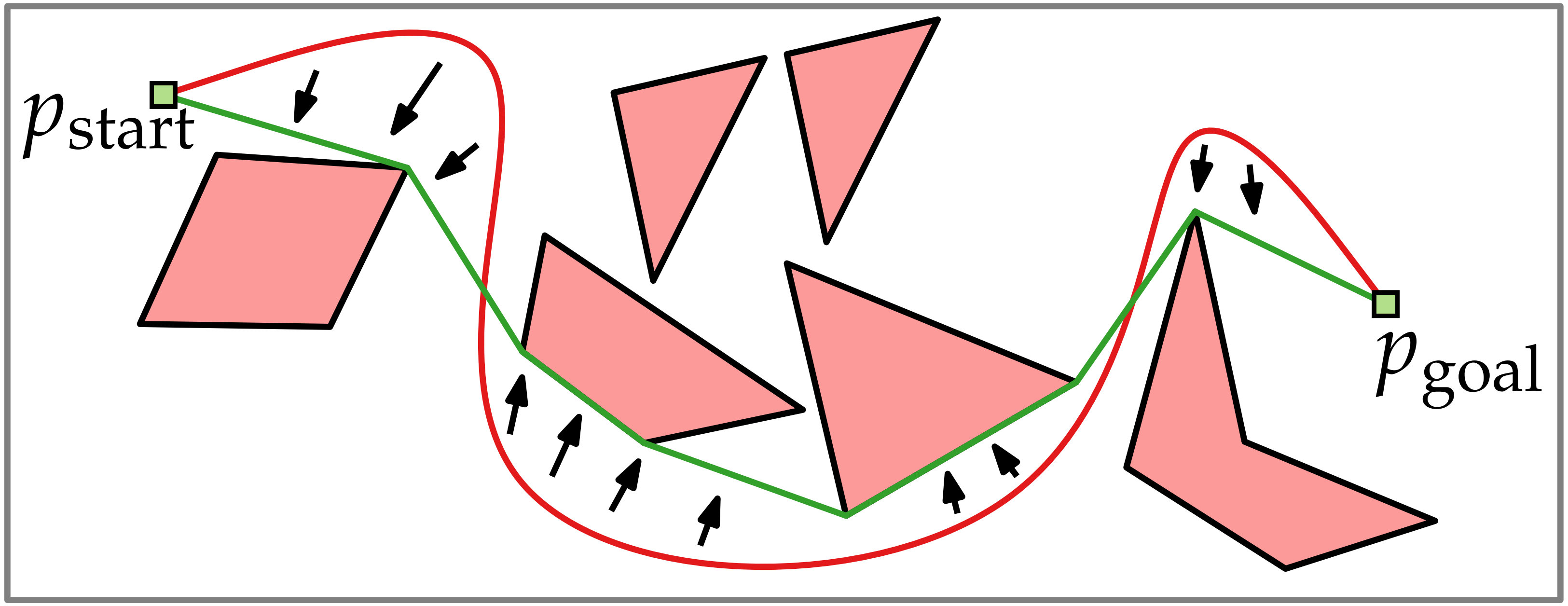
Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space, any shortest path between p_{start} and p_{goal} is a polygonal path whose inner vertices are vertices of S.



Proof.

Characterization

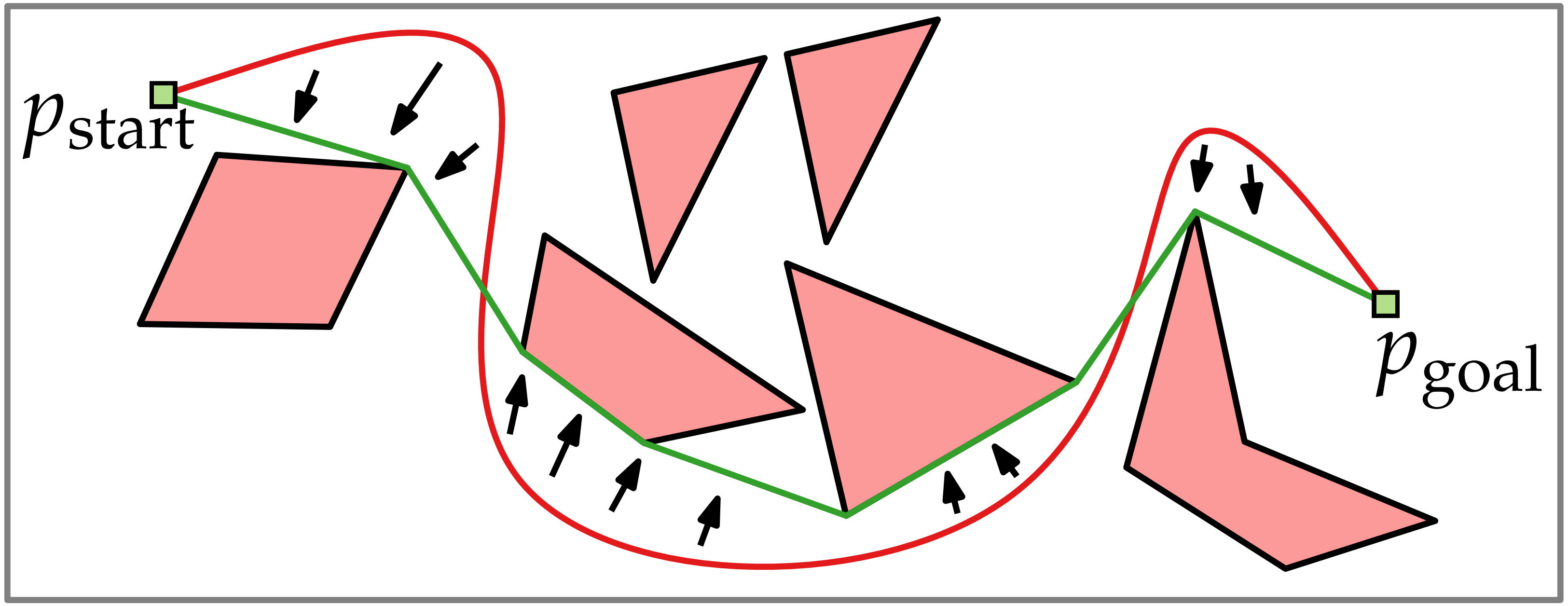
Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space, any shortest path between p_{start} and p_{goal} is a polygonal path whose inner vertices are vertices of S.



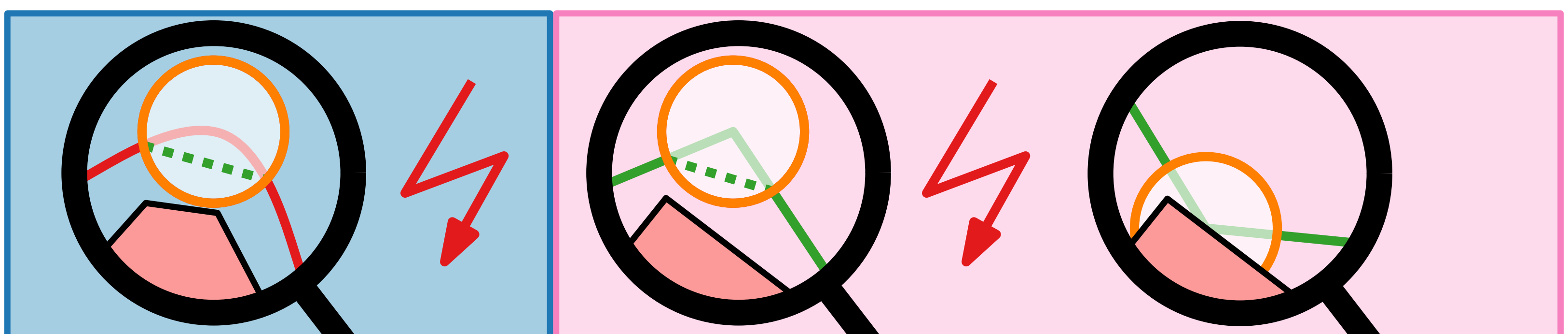
Proof.

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.

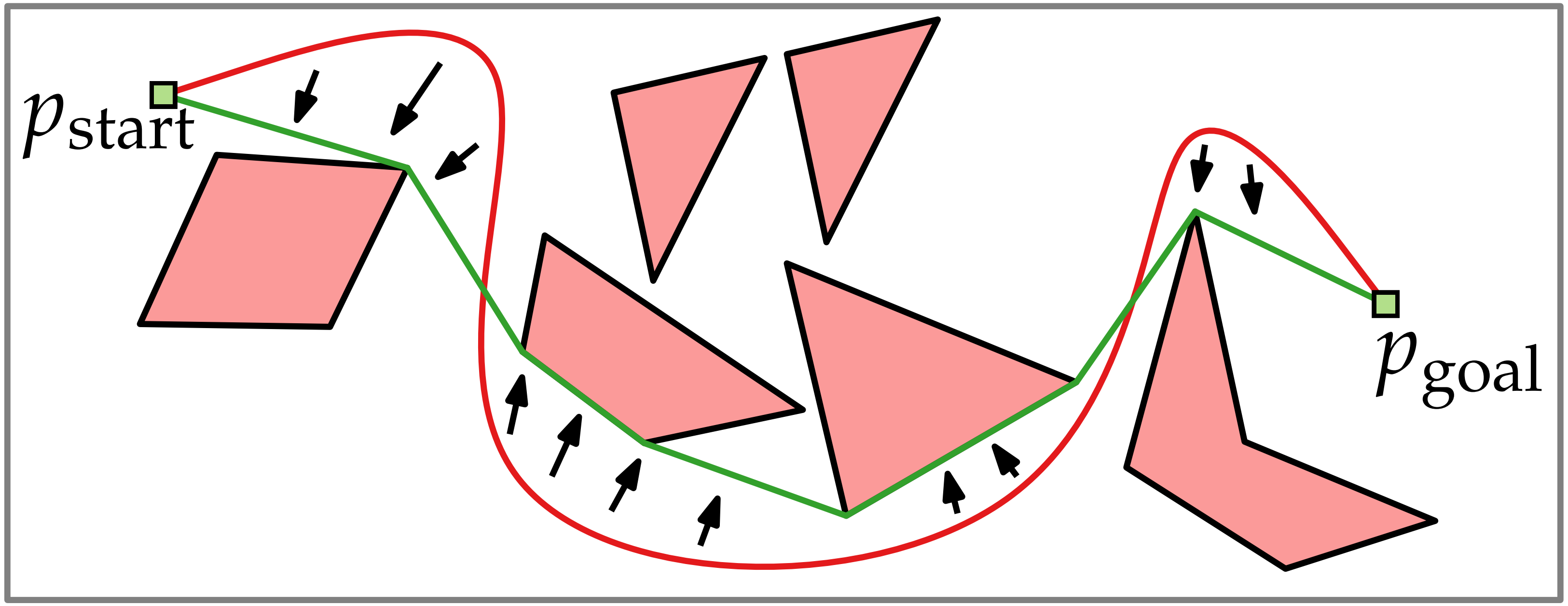


Proof.



Characterization

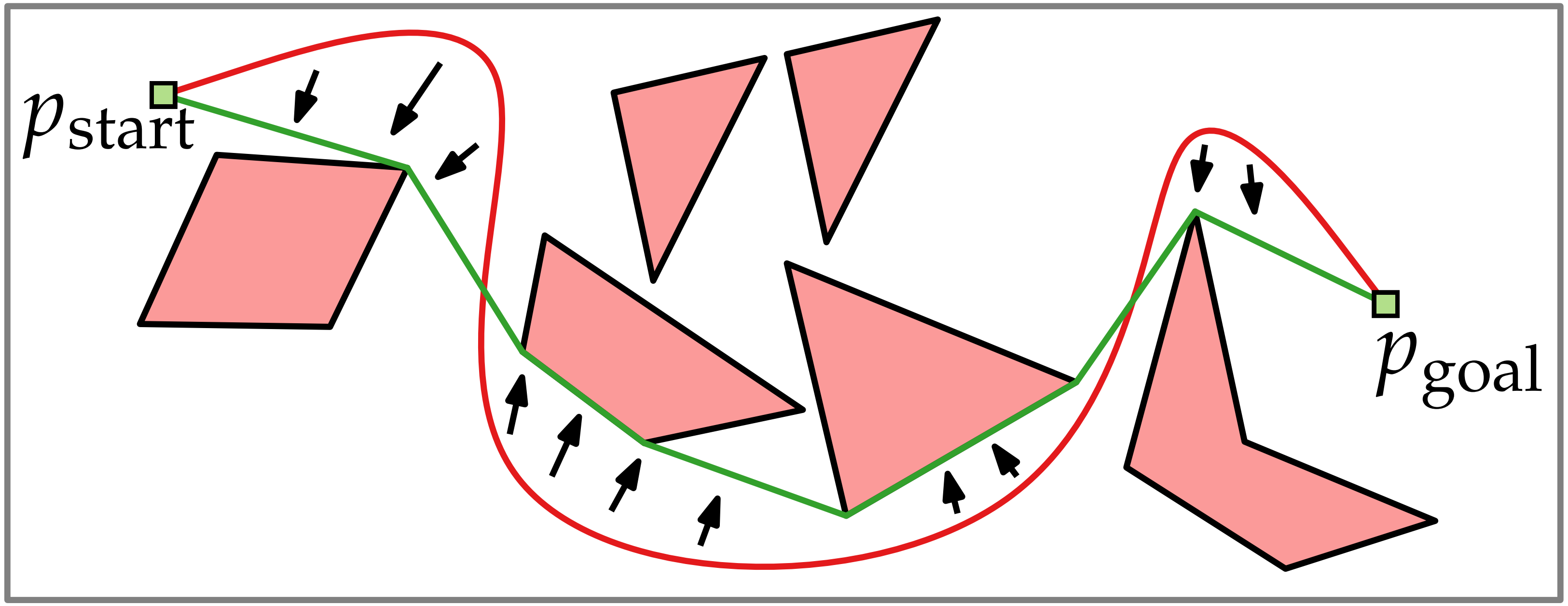
Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space, any shortest path between p_{start} and p_{goal} is a polygonal path whose inner vertices are vertices of S.



Proof.

Characterization

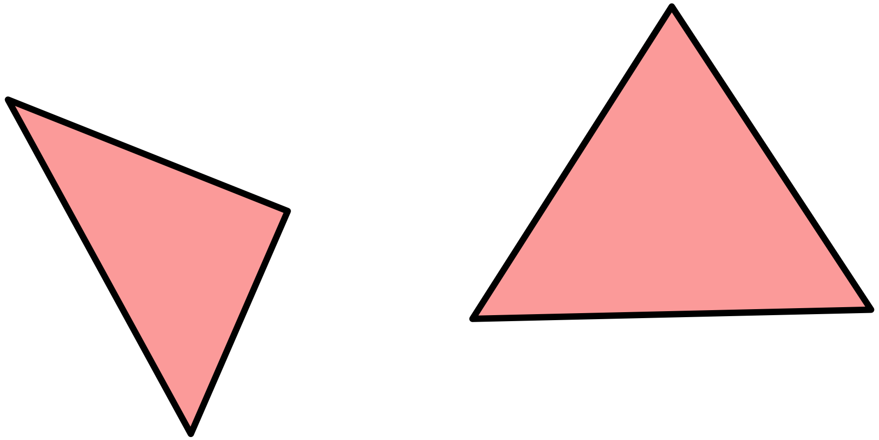
Lemma. Given a set S of disjoint polygonal obstacles in $\mathbb{R}^2$ and points p_{start} and p_{goal} in the free space, any shortest path between p_{start} and p_{goal} is a polygonal path whose inner vertices are vertices of S.



Proof.

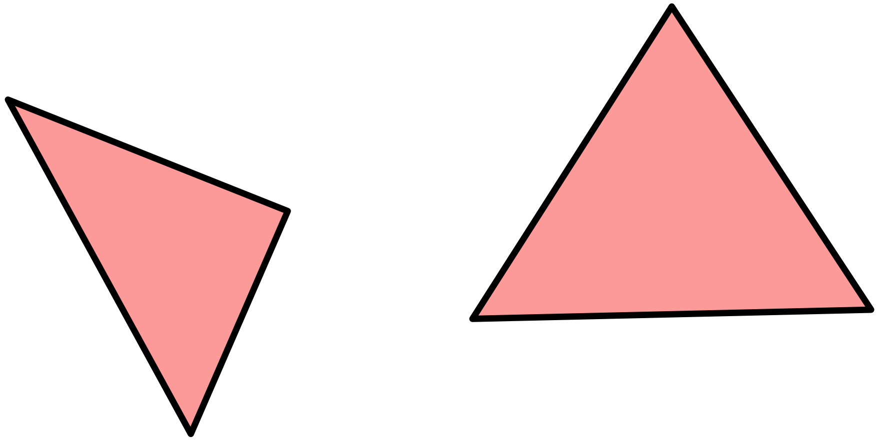
Visibility Graph

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



Visibility Graph

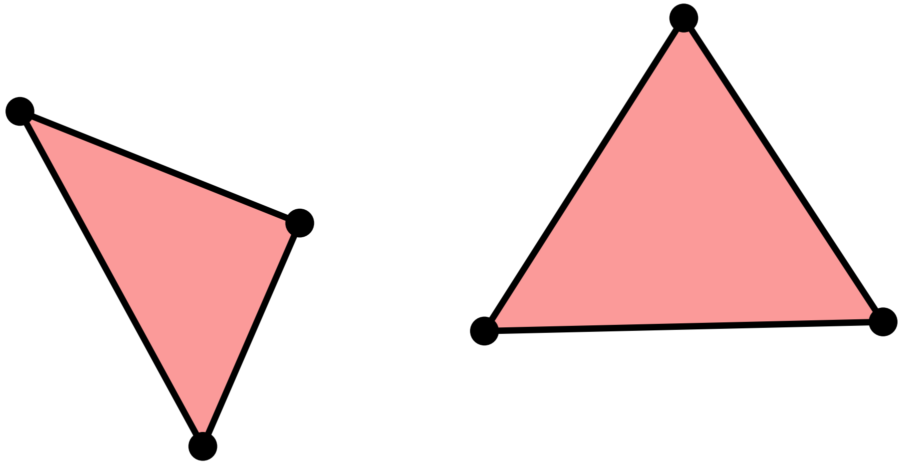
Given a set S of disjoint (open) polygons...



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

Visibility Graph

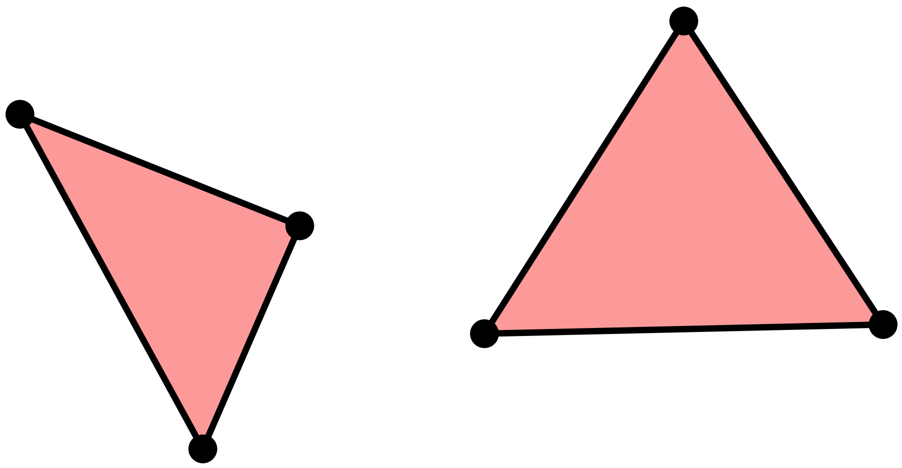
Given a set S of disjoint (open) polygons...



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

Visibility Graph

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

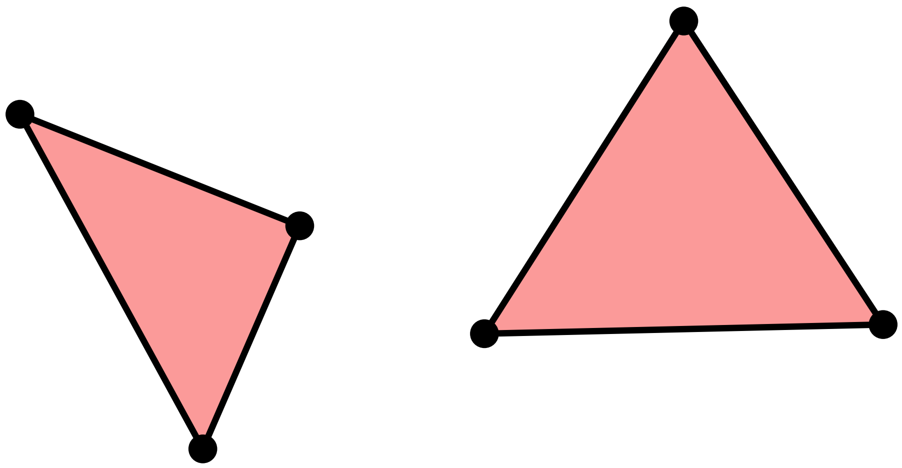


...let $V(S)$ be the vertex set of S.

Let $G_{\text{vis}}(S) = (V(S), E_{\text{vis}}(S))$ be the *visibility graph* of S,

Visibility Graph

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

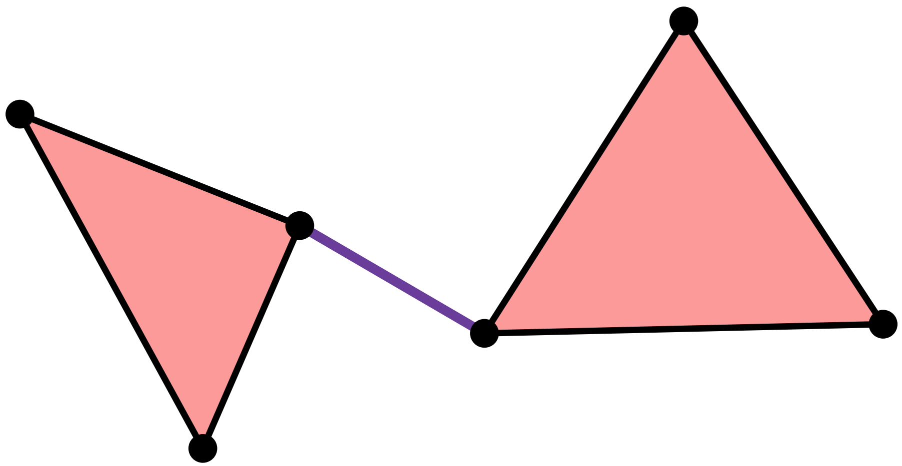


...let $V(S)$ be the vertex set of S.

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

Visibility Graph

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

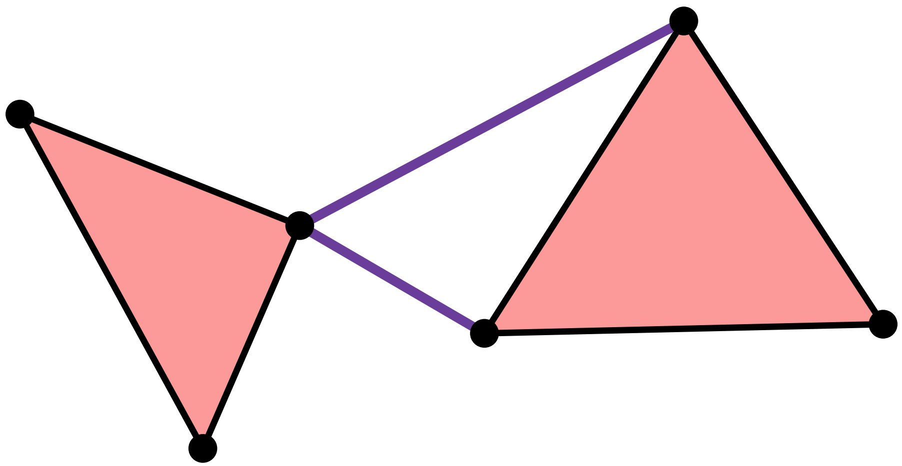


...let $V(S)$ be the vertex set of S.

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

Visibility Graph

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

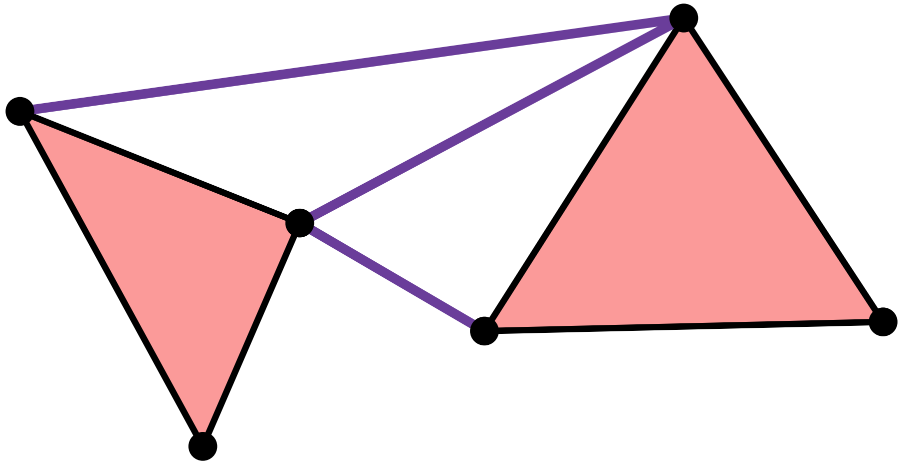


...let $V(S)$ be the vertex set of S.

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

Visibility Graph

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

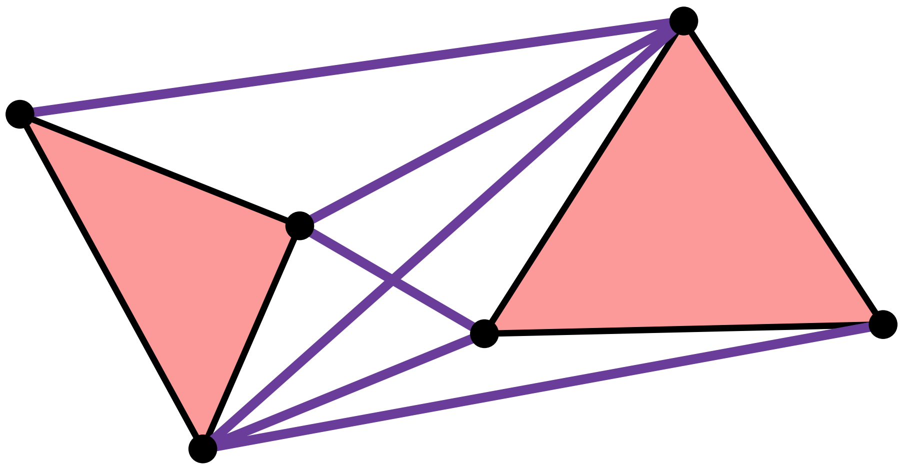


...let $V(S)$ be the vertex set of S.

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

Visibility Graph

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

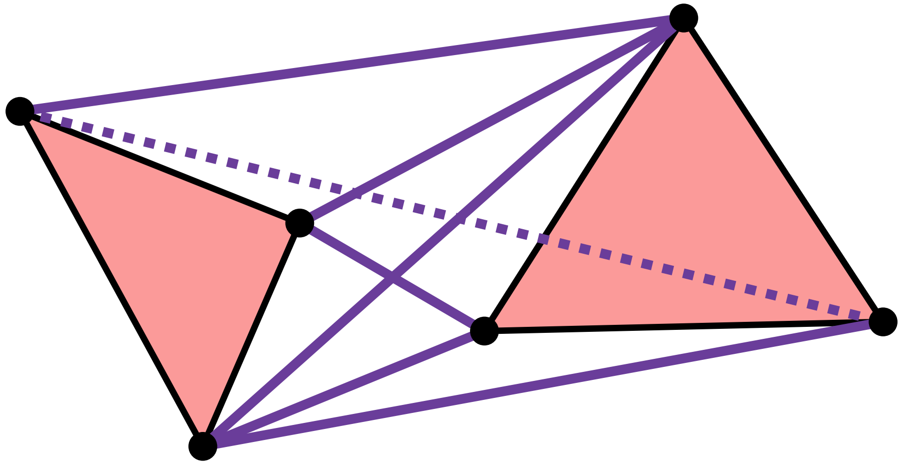


...let $V(S)$ be the vertex set of S.

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

Visibility Graph

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

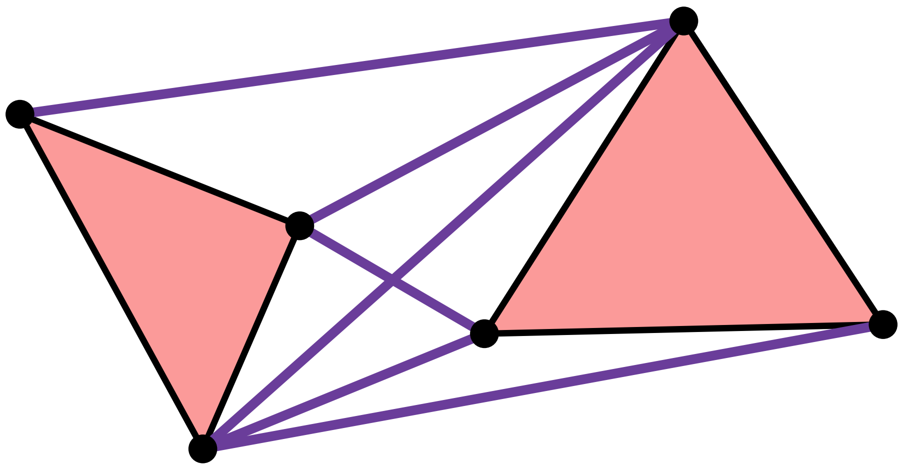


...let $V(S)$ be the vertex set of S.

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

Visibility Graph

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

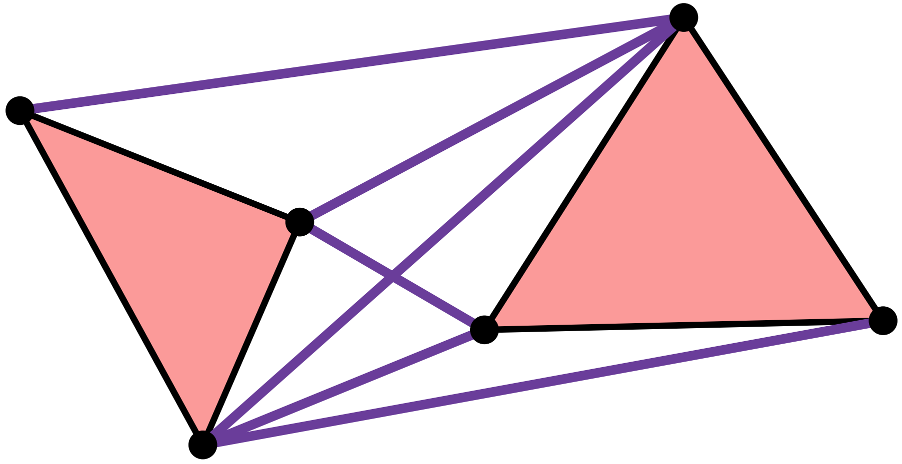


...let $V(S)$ be the vertex set of S.

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

Visibility Graph

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

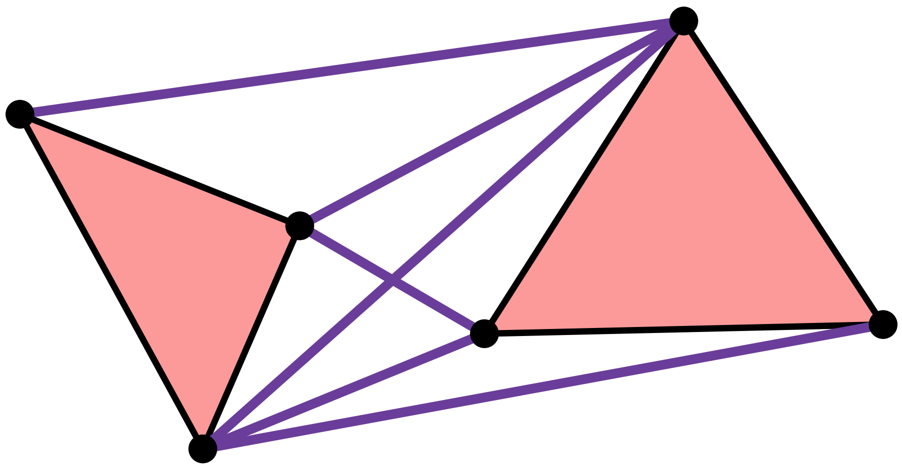


...let $V(S)$ be the vertex set of S.

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 $w(uv) = |uv|$.

Visibility Graph

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



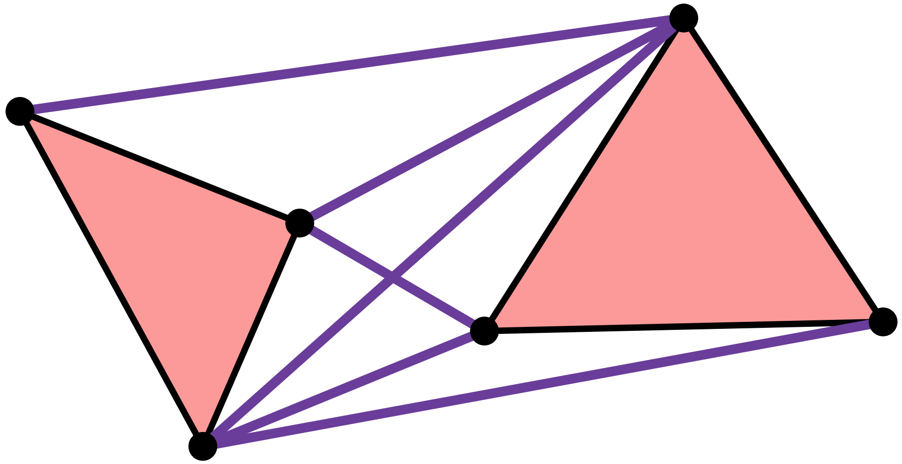
...let $V(S)$ be the vertex set of S.

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 $:\Leftrightarrow$

Visibility Graph

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



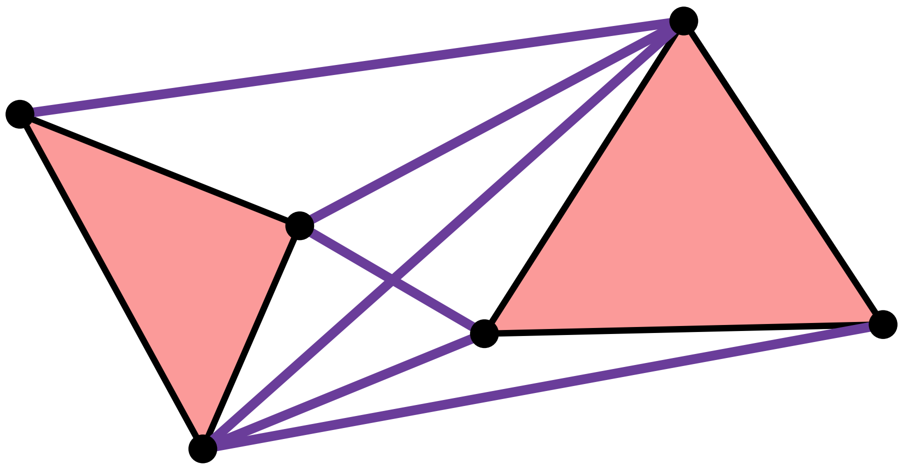
...let $V(S)$ be the vertex set of S.

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 $:\Leftrightarrow \overline{uv} \subset \mathcal{C}_{\text{free}}$

Visibility Graph

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



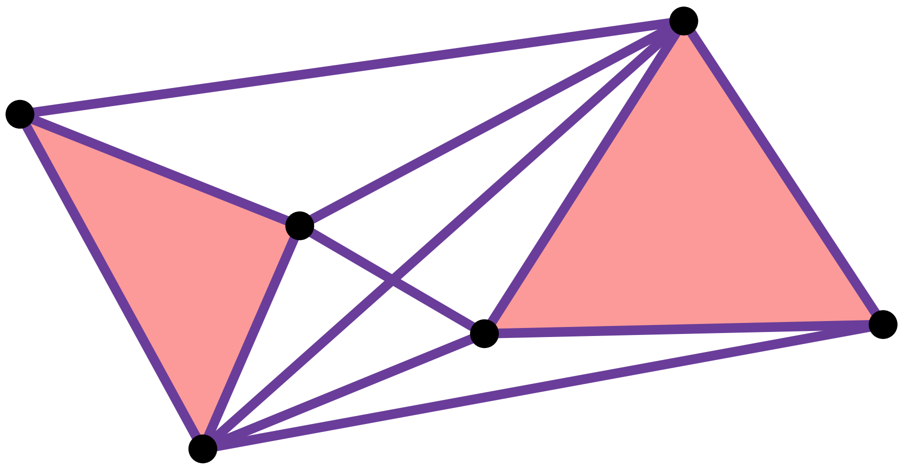
...let $V(S)$ be the vertex set of S.

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

Visibility Graph

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



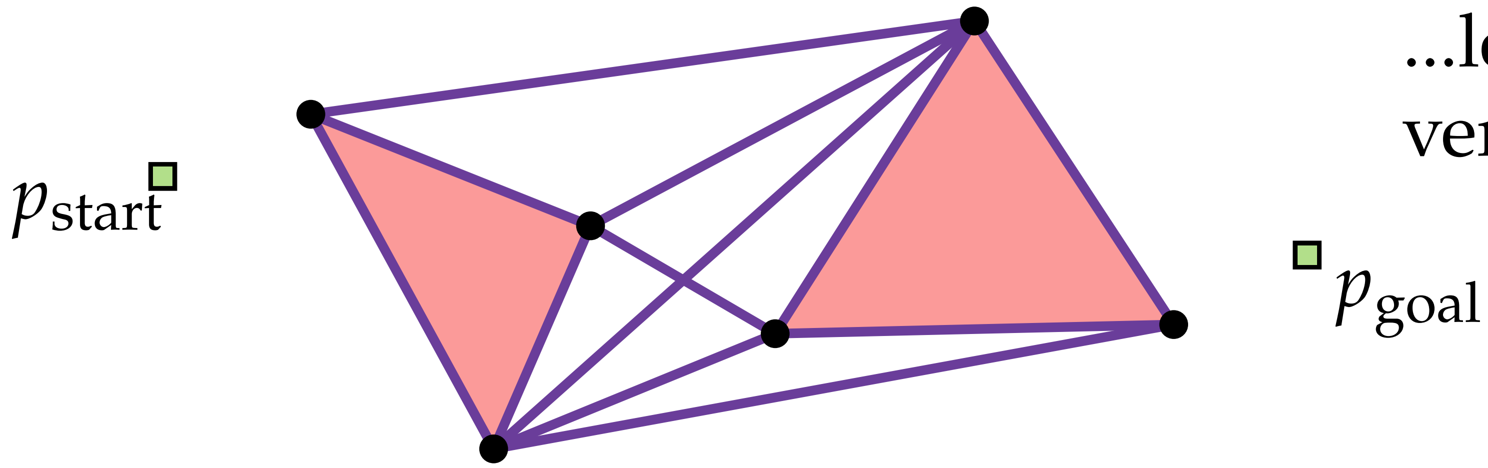
...let $V(S)$ be the vertex set of S.

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

Visibility Graph

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



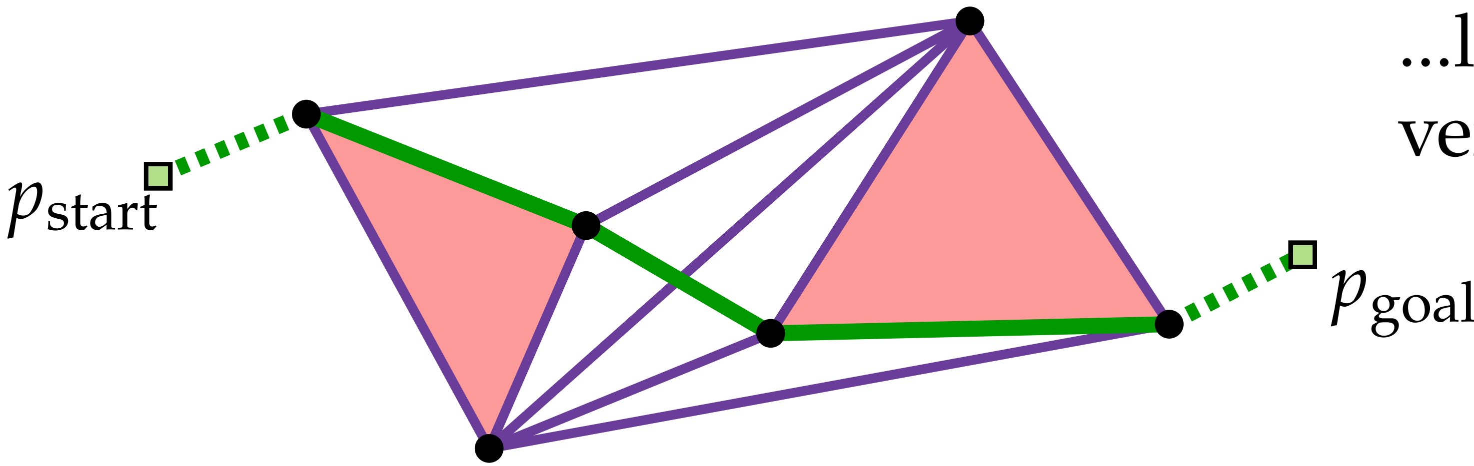
...let $V(S)$ be the vertex set of S.

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

Visibility Graph

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



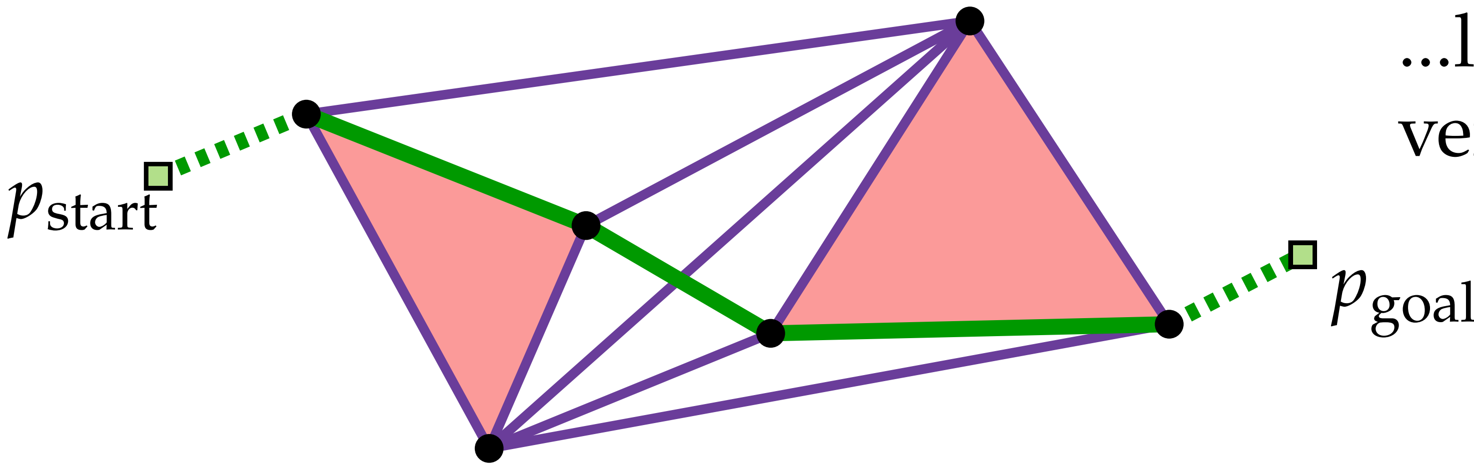
...let $V(S)$ be the vertex set of S.

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

Visibility Graph

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



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

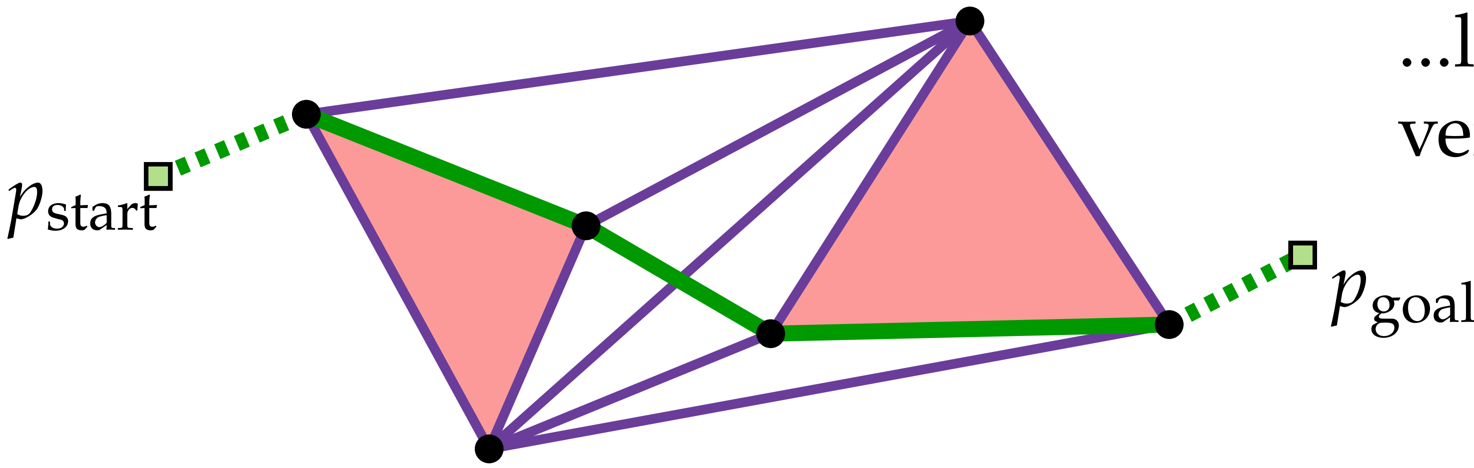
Let $G_{\text{vis}}(S) = (V(S), E_{\text{vis}}(S))$ be the *visibility graph* of S, where $E_{\text{vis}}(S) = \{uv \mid u, v \in V(S), u \text{ sees } v\}$ and $w(uv) = |uv|$.

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

$$S^\star = S \cup \{p_{\text{start}}, p_{\text{goal}}\}.$$

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 $w(uv) = |uv|$.

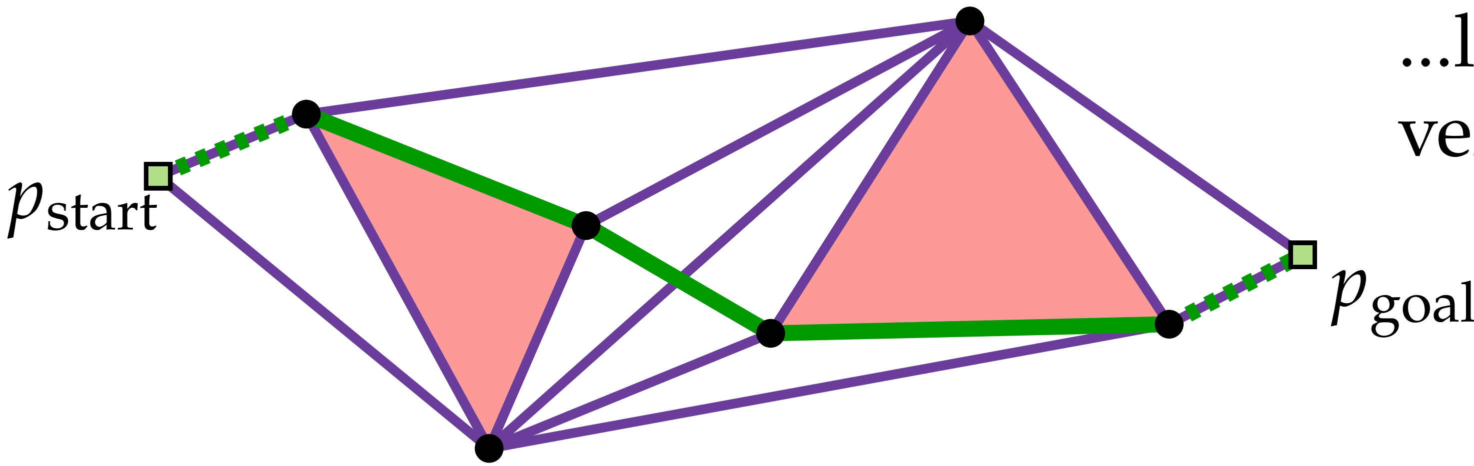
We define: u sees v $:\Leftrightarrow$ $\overline{uv} \subset \mathcal{C}_{\text{free}}$ $\quad (= \mathbb{R}^2 \setminus \bigcup S)$

$G_{\text{vis}}(S^\star)$

$$S^\star = S \cup \{p_{\text{start}}, p_{\text{goal}}\}.$$

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 $w(uv) = |uv|$.

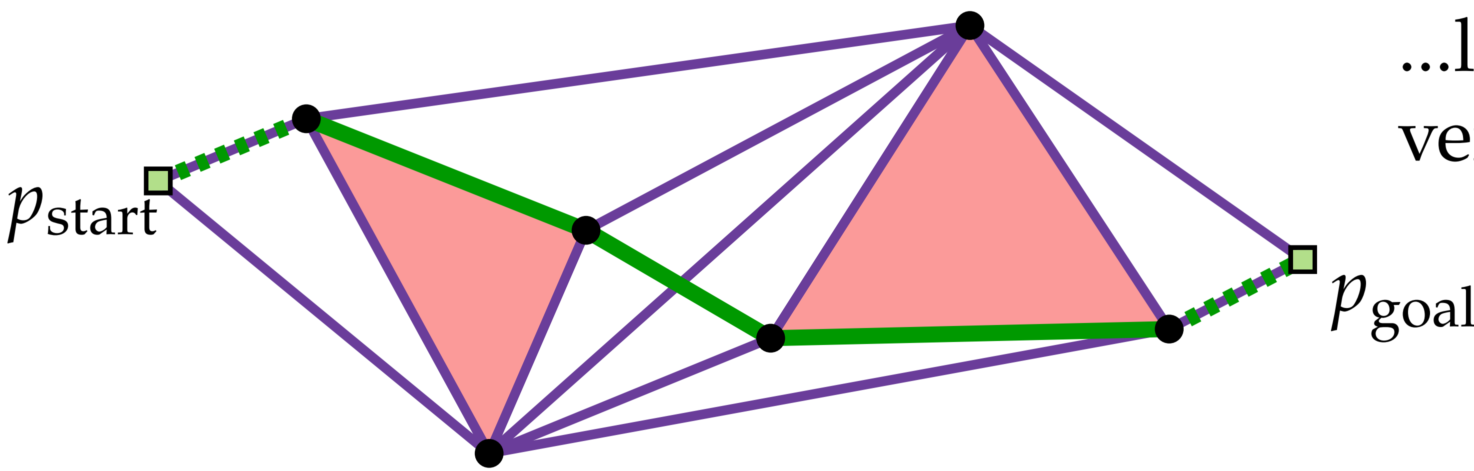
We define: u sees v $:\Leftrightarrow \overline{uv} \subset \mathcal{C}_{\text{free}} \quad (= \mathbb{R}^2 \setminus \bigcup S)$

$G_{\text{vis}}(S^\star)$

$$S^\star = S \cup \{p_{\text{start}}, p_{\text{goal}}\}.$$

Visibility Graph

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



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

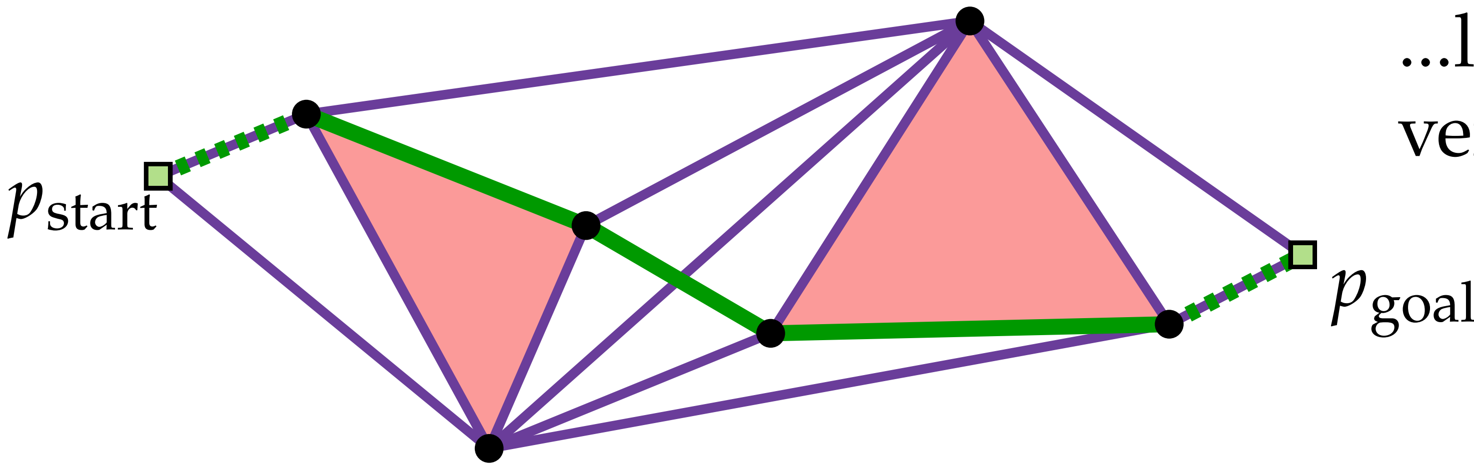
Let $G_{\text{vis}}(S) = (V(S), E_{\text{vis}}(S))$ be the *visibility graph* of S, where $E_{\text{vis}}(S) = \{uv \mid u, v \in V(S), u \text{ sees } v\}$ and $w(uv) = |uv|$.

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 path in $G_{\text{vis}}(S^\star)$, where $S^\star = S \cup \{p_{\text{start}}, p_{\text{goal}}\}$.

Visibility Graph

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



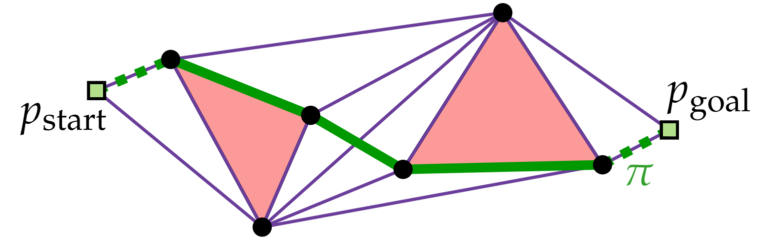
...let $V(S)$ be the vertex set of S.

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

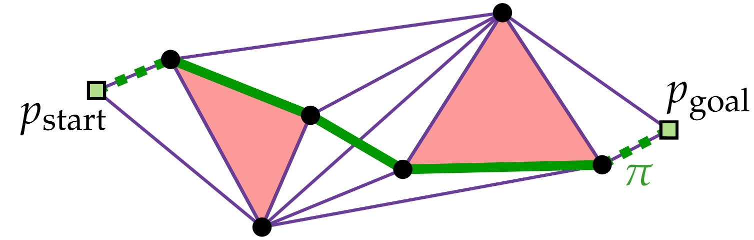
Corollary. A shortest path between p_{start} and p_{goal} corresponds to a *shortest* path in $G_{\text{vis}}(S^\star)$, where $S^\star = S \cup \{p_{\text{start}}, p_{\text{goal}}\}$.

Algorithm



SHORTESTPATH($S, p_{\text{start}}, p_{\text{goal}}$)

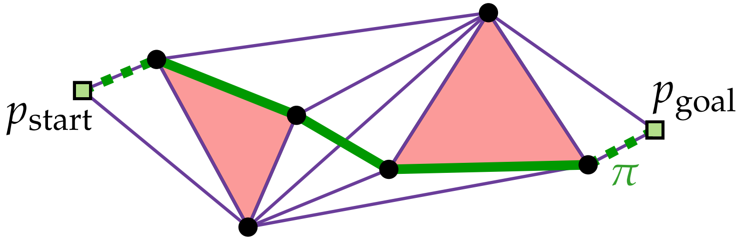
Algorithm



$\textsc{ShortestPath}(S, p_{\text{start}}, p_{\text{goal}})$

$G_{\text{vis}} \leftarrow \textsc{VisibilityGraph}(S \cup \{p_{\text{start}}, p_{\text{goal}}\})$

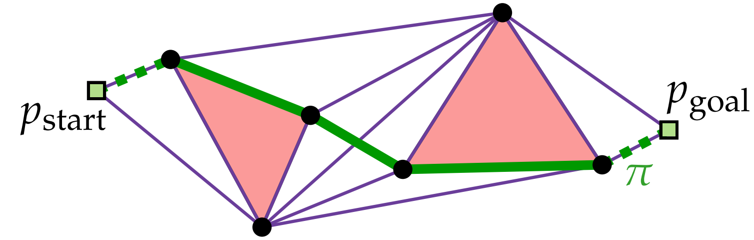
Algorithm



ShortestPath$(S, p_{\text{start}}, p_{\text{goal}})$

$G_{\text{vis}} \leftarrow$ **VisibilityGraph**$(S \cup \{p_{\text{start}}, p_{\text{goal}}\})$
foreach $uv \in E_{\text{vis}}$ **do**

Algorithm



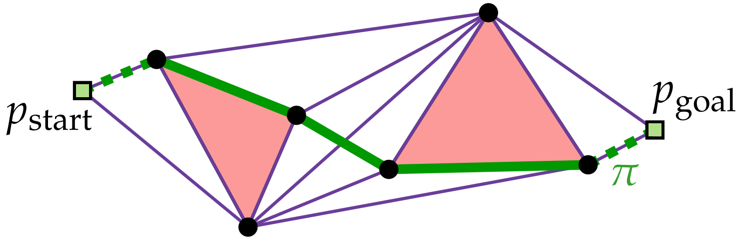
$\textsc{ShortestPath}(S, p_{\text{start}}, p_{\text{goal}})$

$G_{\text{vis}} \leftarrow \textsc{VisibilityGraph}(S \cup \{p_{\text{start}}, p_{\text{goal}}\})$

foreach $uv \in E_{\text{vis}}$ **do**

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

Algorithm



$\textsc{ShortestPath}(S, p_{\text{start}}, p_{\text{goal}})$

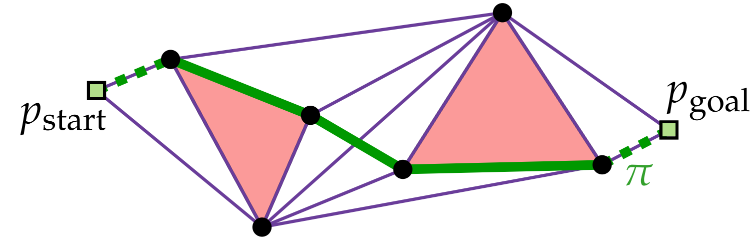
$G_{\text{vis}} \leftarrow \textsc{VisibilityGraph}(S \cup \{p_{\text{start}}, p_{\text{goal}}\})$

foreach $uv \in E_{\text{vis}}$ **do**

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

$\pi \leftarrow$

Algorithm



$\textsc{ShortestPath}(S, p_{\text{start}}, p_{\text{goal}})$

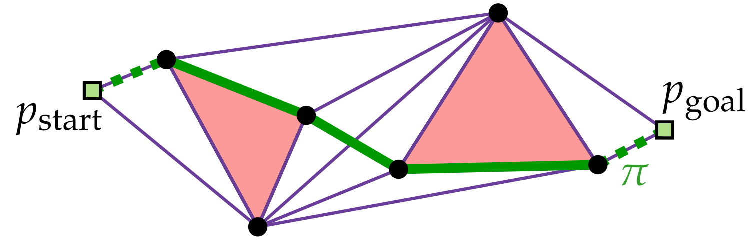
$G_{\text{vis}} \leftarrow \textsc{VisibilityGraph}(S \cup \{p_{\text{start}}, p_{\text{goal}}\})$

foreach $uv \in E_{\text{vis}}$ **do**

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

$\pi \leftarrow \textsc{Dijkstra}(G_{\text{vis}}, w, p_{\text{start}}, p_{\text{goal}})$

Algorithm



$\textsc{ShortestPath}(S, p_{\text{start}}, p_{\text{goal}})$

$G_{\text{vis}} \leftarrow \textsc{VisibilityGraph}(S \cup \{p_{\text{start}}, p_{\text{goal}}\})$

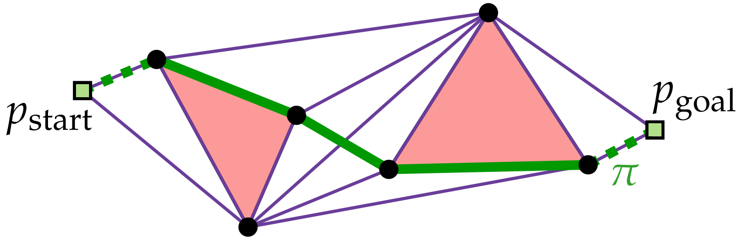
foreach $uv \in E_{\text{vis}}$ **do**

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

$\pi \leftarrow \textsc{Dijkstra}(G_{\text{vis}}, w, p_{\text{start}}, p_{\text{goal}})$

return π

Algorithm



$\textsc{ShortestPath}(S, p_{\text{start}}, p_{\text{goal}})$

$G_{\text{vis}} \leftarrow \textsc{VisibilityGraph}(S \cup \{p_{\text{start}}, p_{\text{goal}}\})$

foreach $uv \in E_{\text{vis}}$ **do**

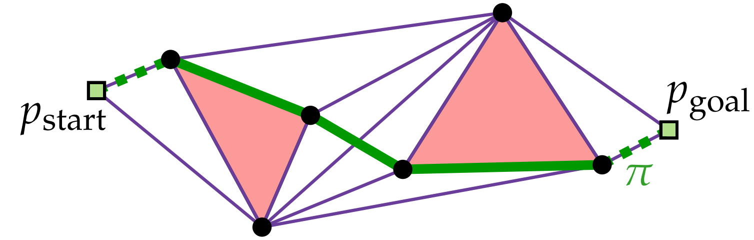
$\quad w(uv) = d_{\text{Eucl.}}(u, v)$

$\pi \leftarrow \textsc{Dijkstra}(G_{\text{vis}}, w, p_{\text{start}}, p_{\text{goal}})$

return π

Running time?

Algorithm

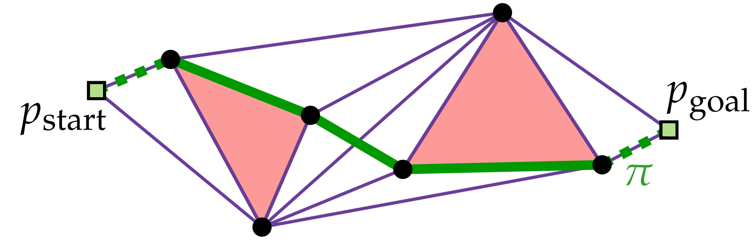


ShortestPath$(S, p_{\text{start}}, p_{\text{goal}})$ $\quad n = |V(S)|$

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

Running time?

Algorithm



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

$G_{vis} \leftarrow \textsc{VisibilityGraph}(S \cup \{p_{start}, p_{goal}\})$

foreach $uv \in E_{vis}$ **do**

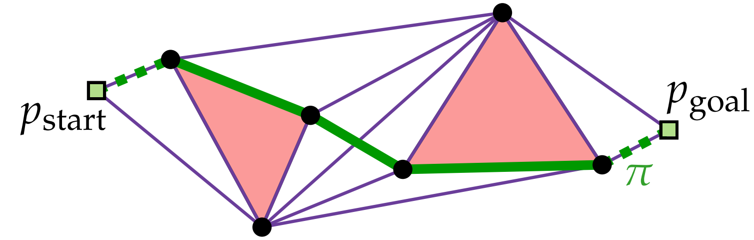
$\quad w(uv) = d_{Eucl.}(u, v)$

$\pi \leftarrow \textsc{Dijkstra}(G_{vis}, w, p_{start}, p_{goal})$

return π

Running time?

Algorithm



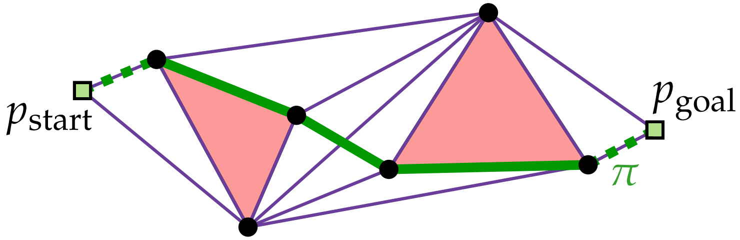
$n = |V(S)|$, $m = |E_{\text{vis}}(S)|$

ShortestPath$(S, p_{\text{start}}, p_{\text{goal}})$

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

Running time?

Algorithm



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

ShortestPath($S, p_{\text{start}}, p_{\text{goal}}$)

$G_{\text{vis}} \leftarrow$ VisibilityGraph($S \cup \{p_{\text{start}}, p_{\text{goal}}\}$)

foreach $uv \in E_{\text{vis}}$ **do**

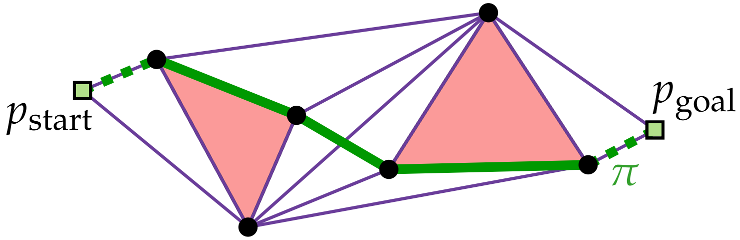
  $w(uv) = d_{\text{Eucl.}}(u, v)$ $O(m)$

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

return π

Running time?

Algorithm



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

$G_{\text{vis}} \leftarrow$ VISIBILITYGRAPH$(S \cup \{p_{\text{start}}, p_{\text{goal}}\})$?

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?

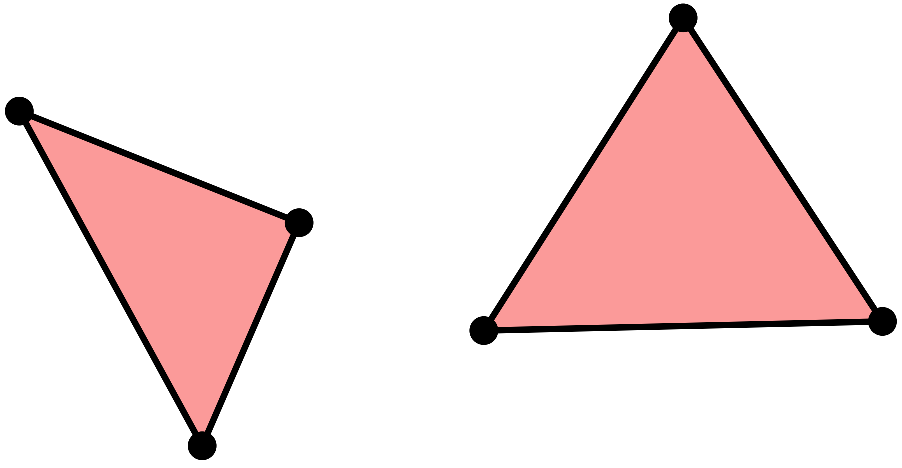
Computing the Visibility Graph

VisibilityGraph(S)

Computing the Visibility Graph

VisibilityGraph(S)

Input: a set S of disjoint polygons

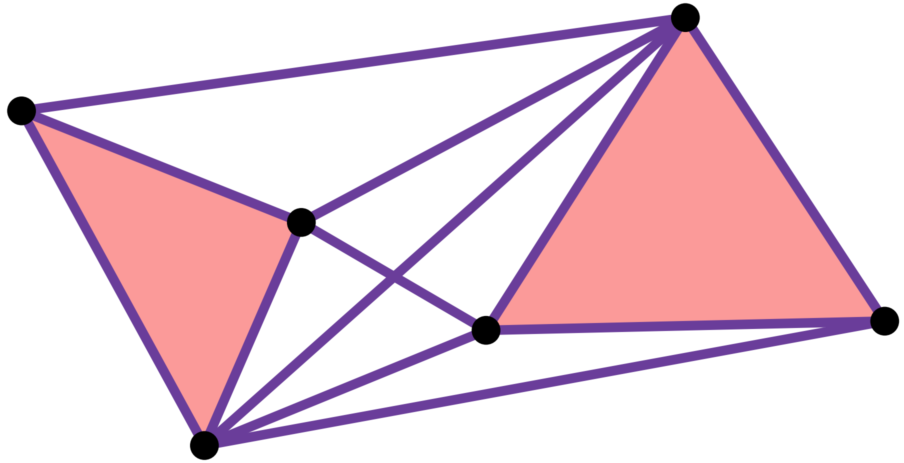


Computing the Visibility Graph

VisibilityGraph(S)

Input: a set S of disjoint polygons

Output: $G_{\text{vis}}(S)$

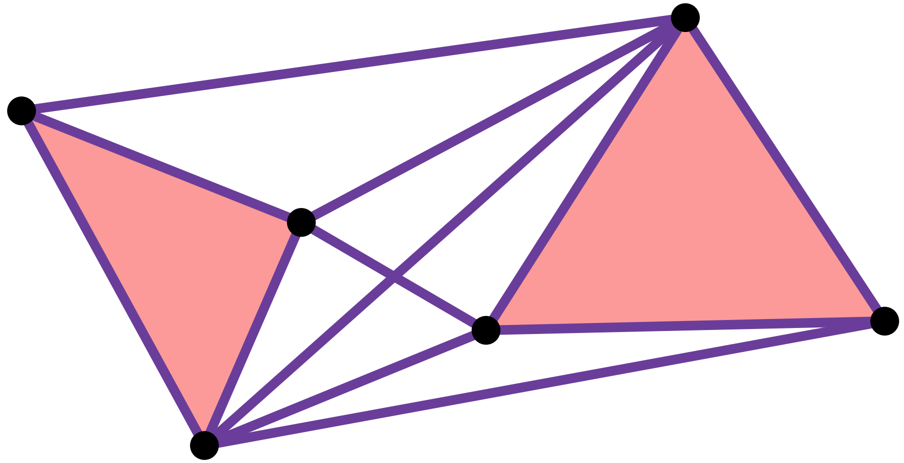


Computing the Visibility Graph

VisibilityGraph(S)

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

$E \leftarrow \emptyset$

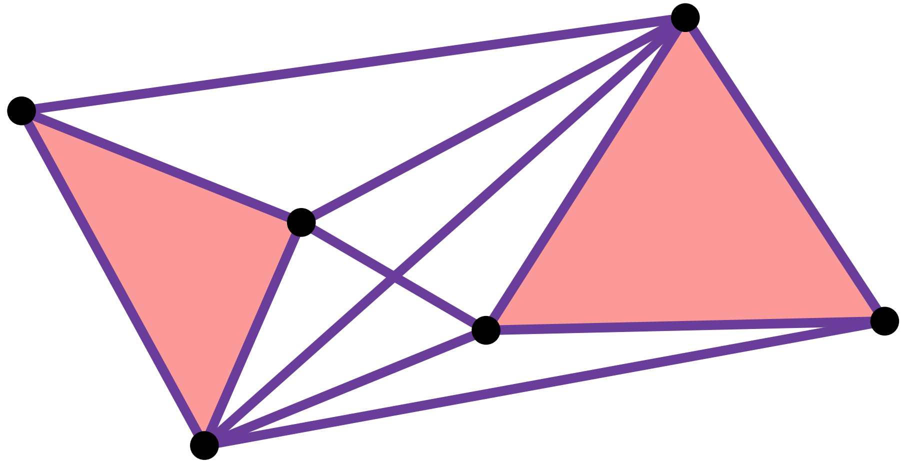


Computing the Visibility Graph

VisibilityGraph(S)

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

$E \leftarrow \emptyset$
foreach $v \in V(S)$ **do**

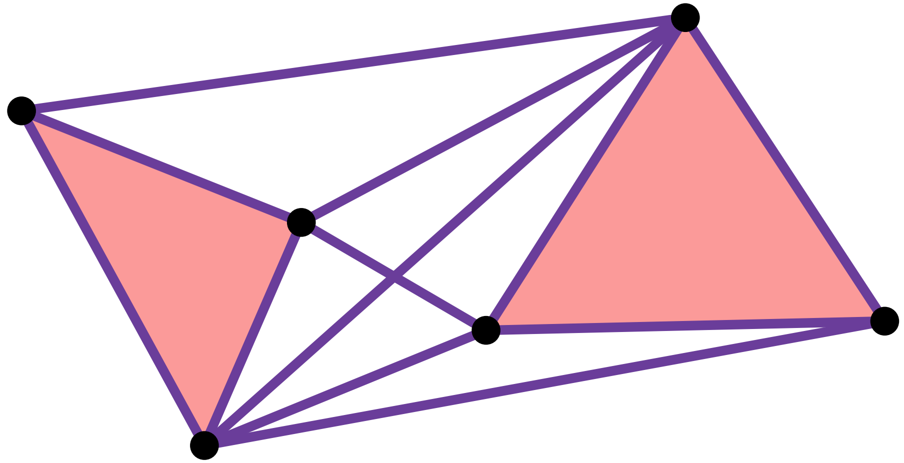


Computing the Visibility Graph

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)

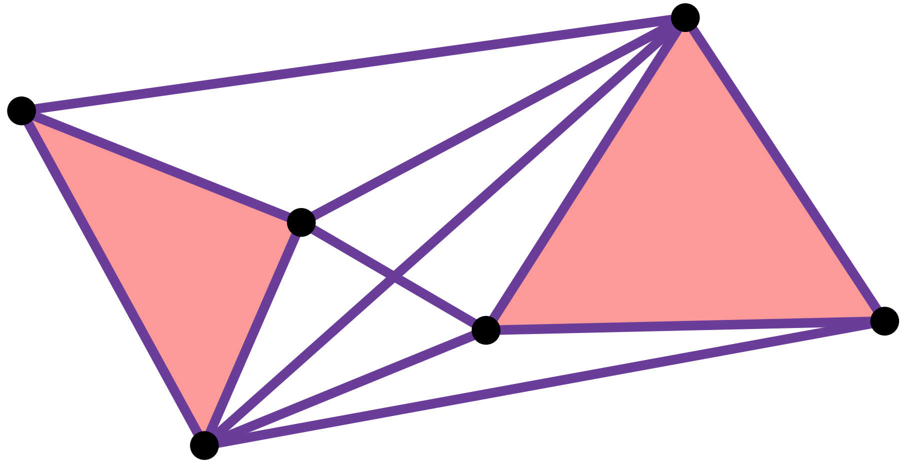


Computing the Visibility Graph

VisibilityGraph(S)

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

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

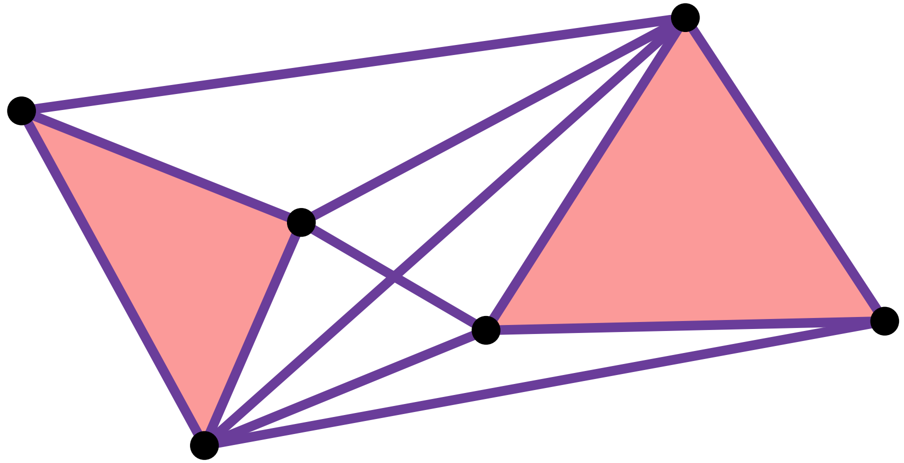


Computing the Visibility Graph

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 $(V(S), E)$



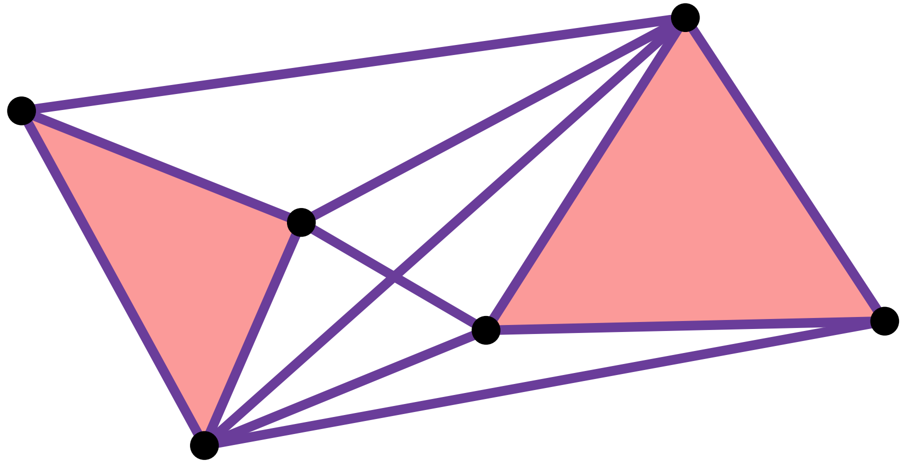
Computing the Visibility Graph

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 $(V(S), E)$

$O(n) \cdot$



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**

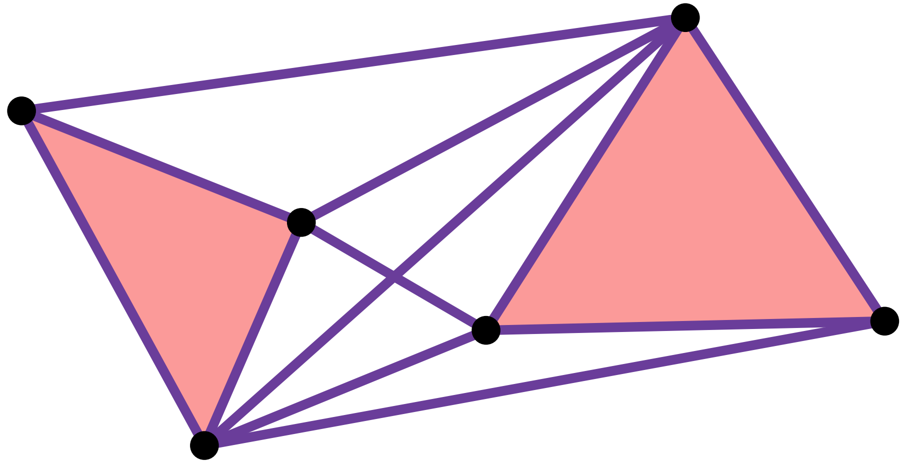
  $W =$ VisibleVertices(v, S)

  $E \leftarrow E \cup \{vw \mid w \in W\}$

return $(V(S), E)$

$O(n) \cdot$

?

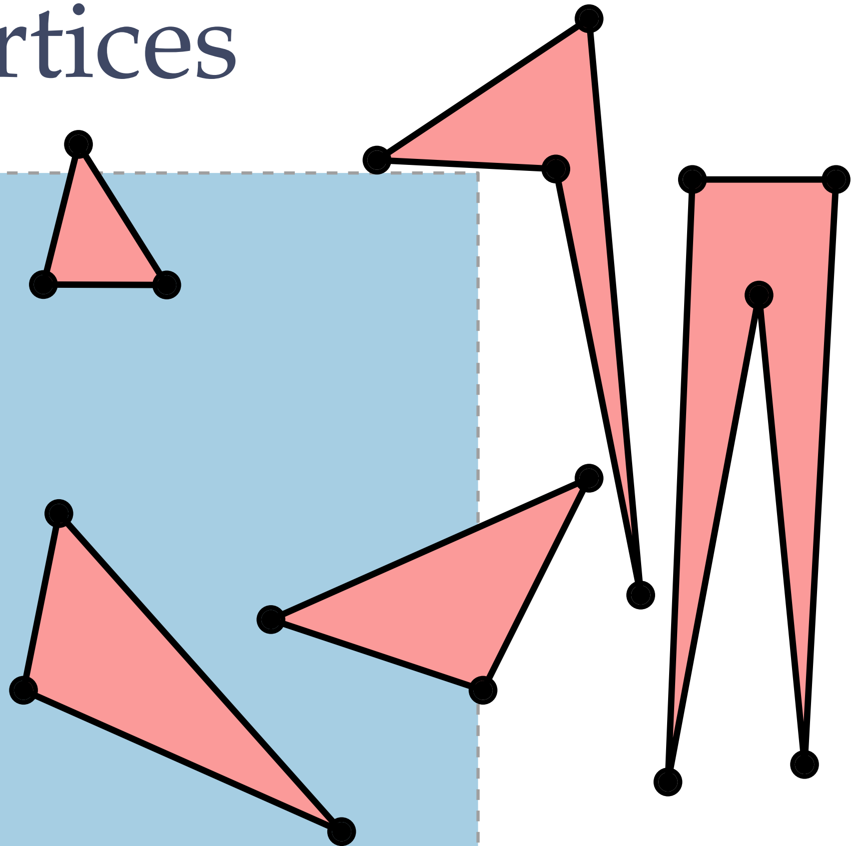


Computing Visible Vertices

VISIBLEVERTICES(p, S)

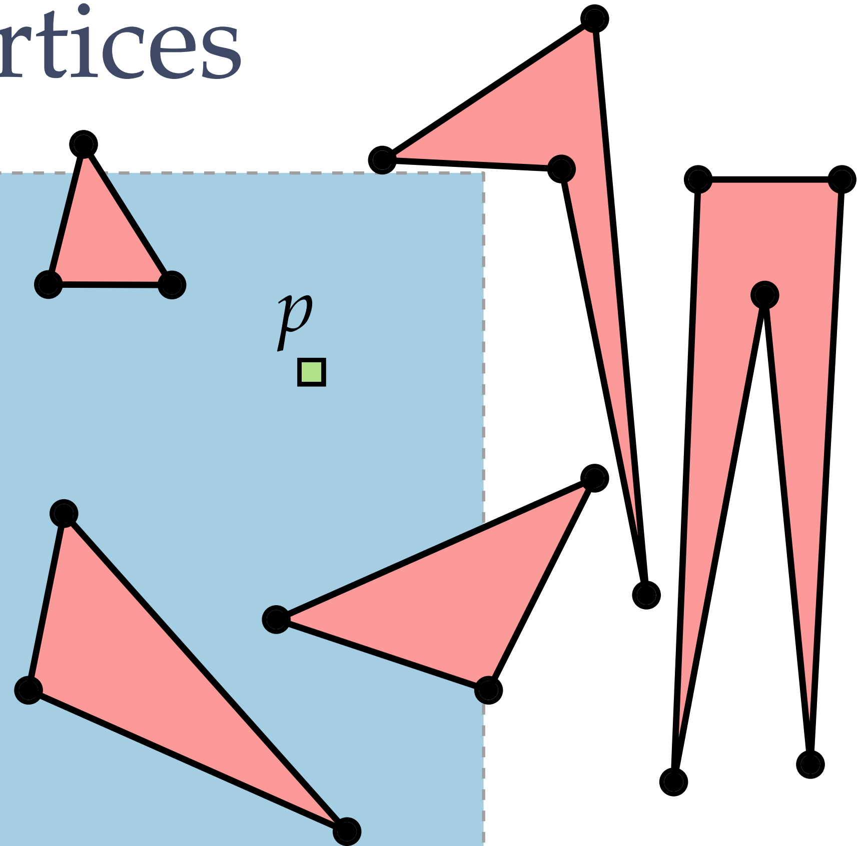
Computing Visible Vertices

VISIBLEVERTICES(p, S)



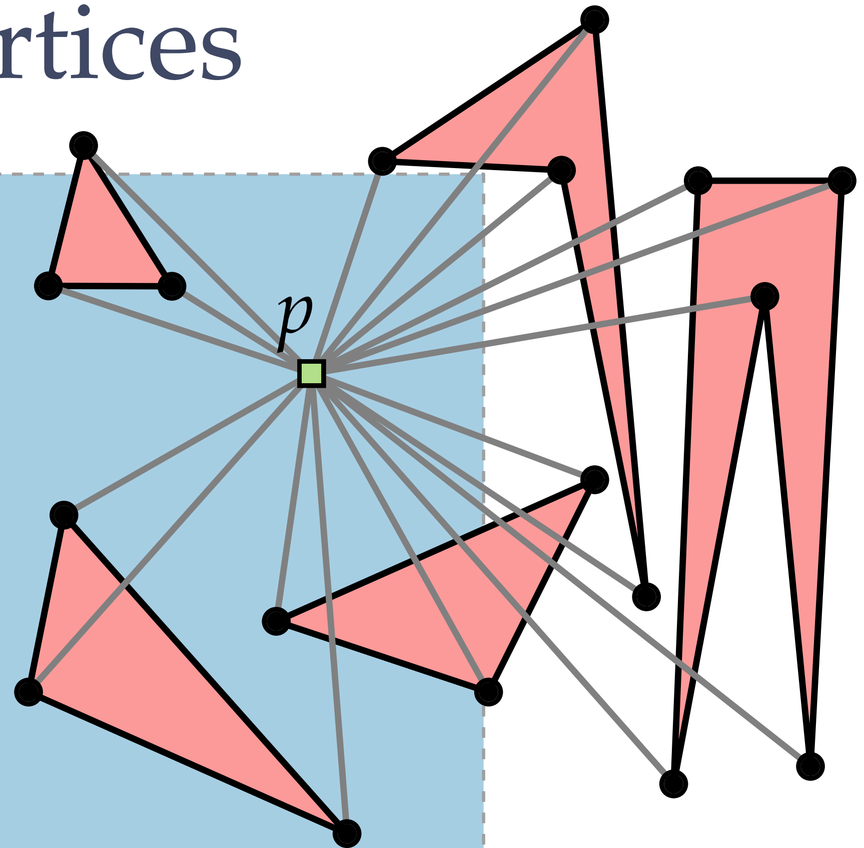
Computing Visible Vertices

VisibleVertices(p, S)



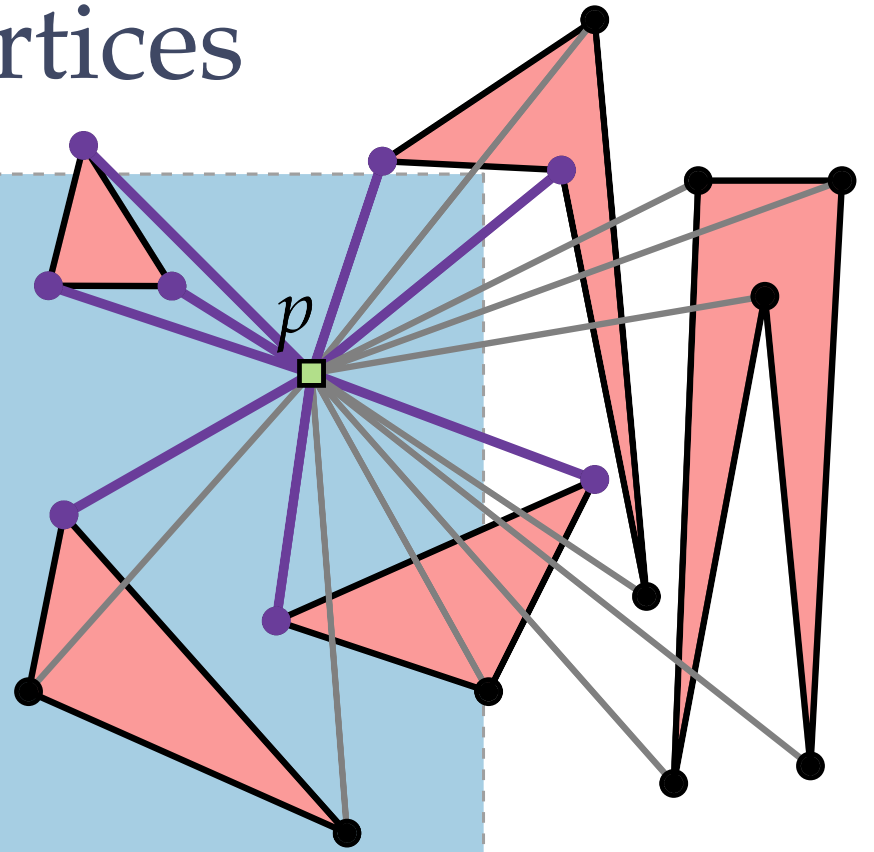
Computing Visible Vertices

VISIBLEVERTICES(p, S)



Computing Visible Vertices

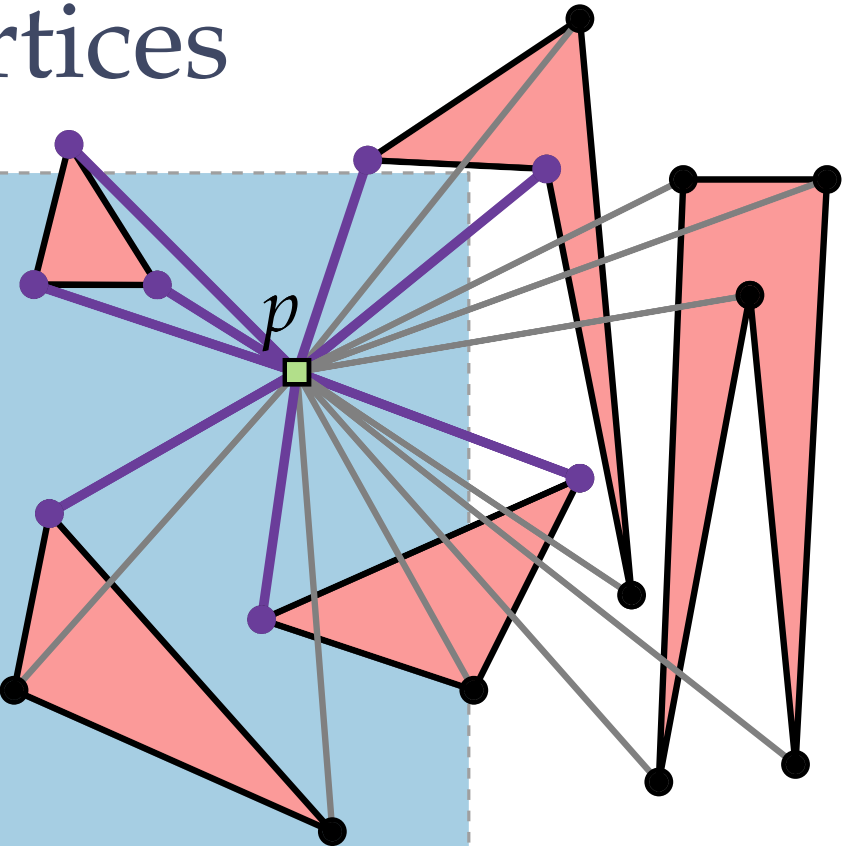
VisibleVertices(p, S)



Computing Visible Vertices

VisibleVertices(p, S)

Task: Separate the "**good**" from the "**evil**"



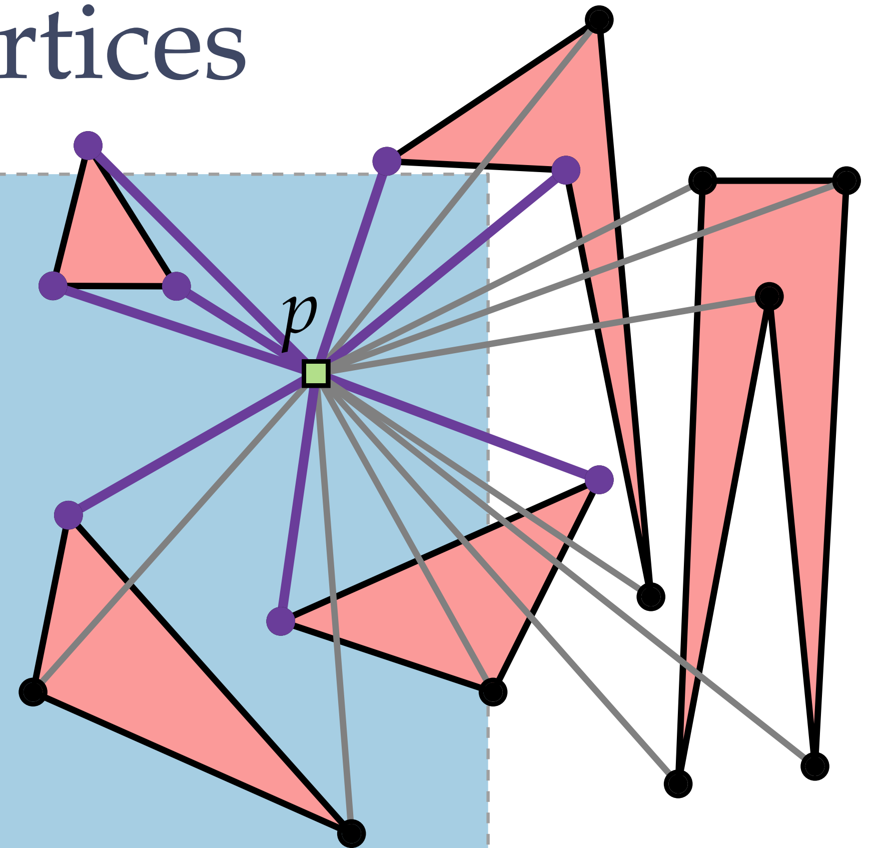
Computing Visible Vertices

VisibleVertices(p, S)

Task: Separate the "**good**" from the "**evil**":

Given p and S, find in $O(n \log n)$ time all vertices in $V(S)$ visible from p!

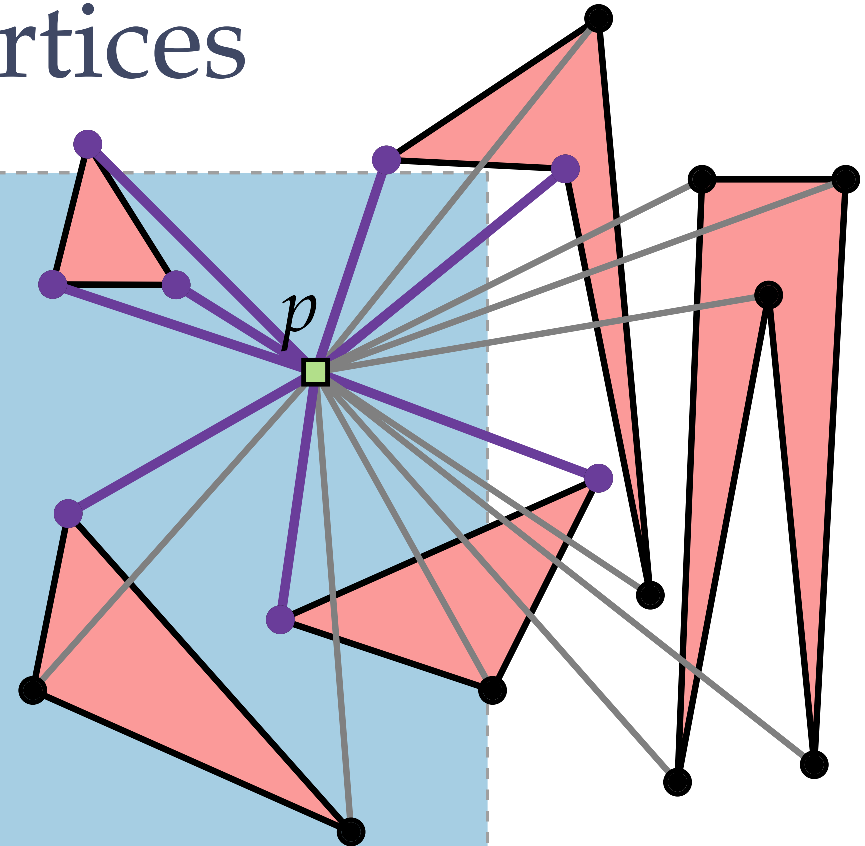
[3 min]



Computing Visible Vertices

VisibleVertices(p, S)

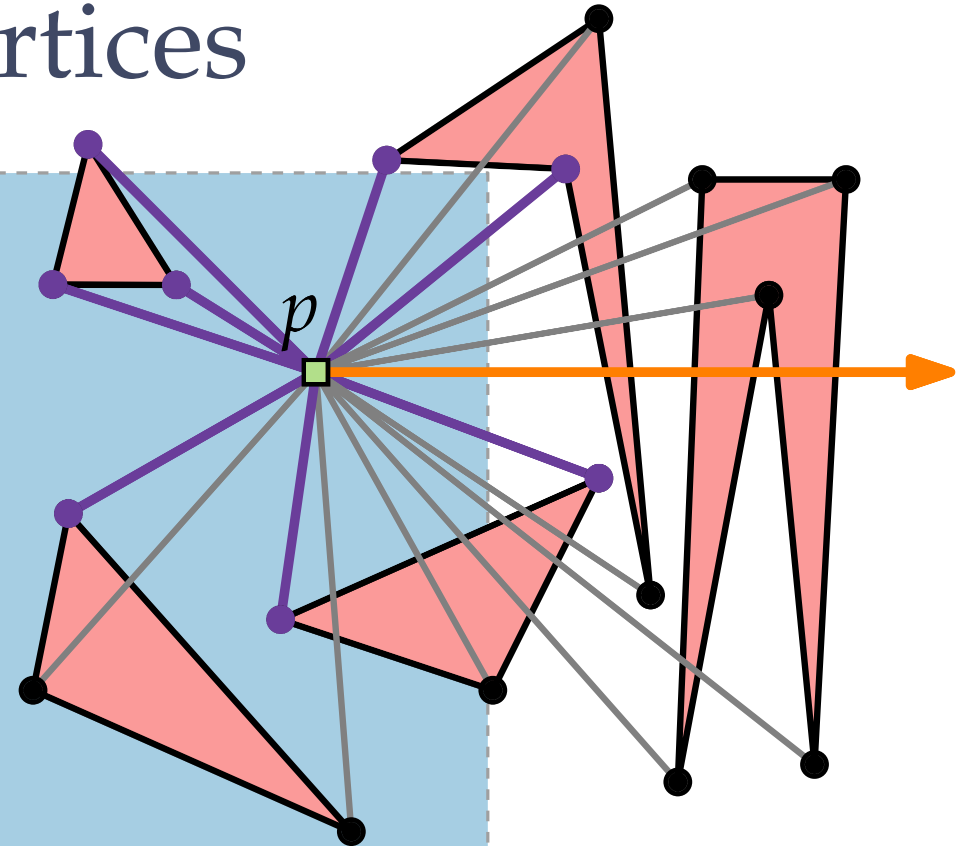
$r \leftarrow p + \mathbb{R}_{\geq 0}\binom{1}{0}$



Computing Visible Vertices

VISIBLEVERTICES(p, S)

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

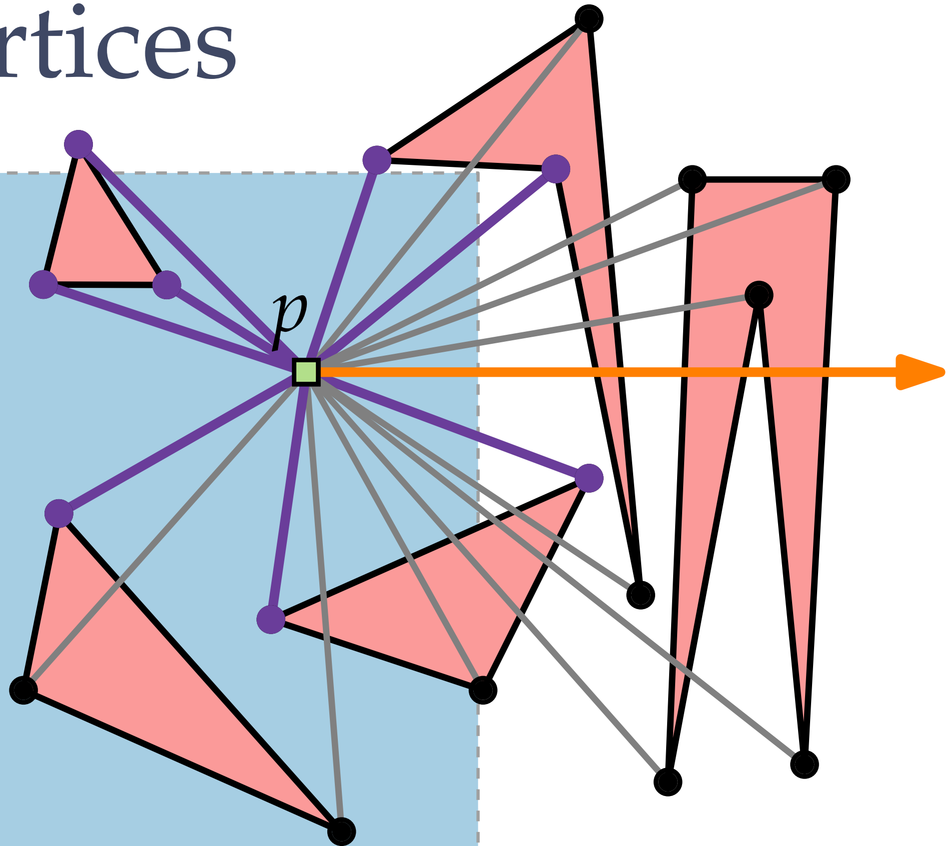


Computing Visible Vertices

VisibleVertices(p, S)

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

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

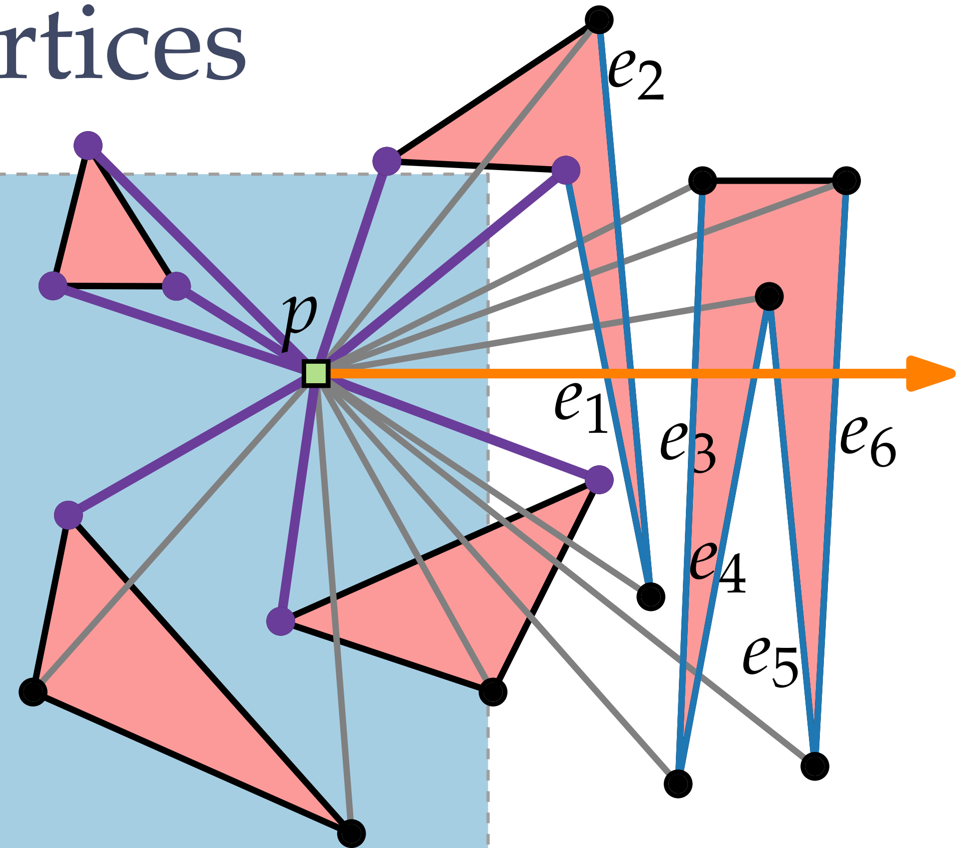


Computing Visible Vertices

$\textsc{VisibleVertices}(p, S)$

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

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



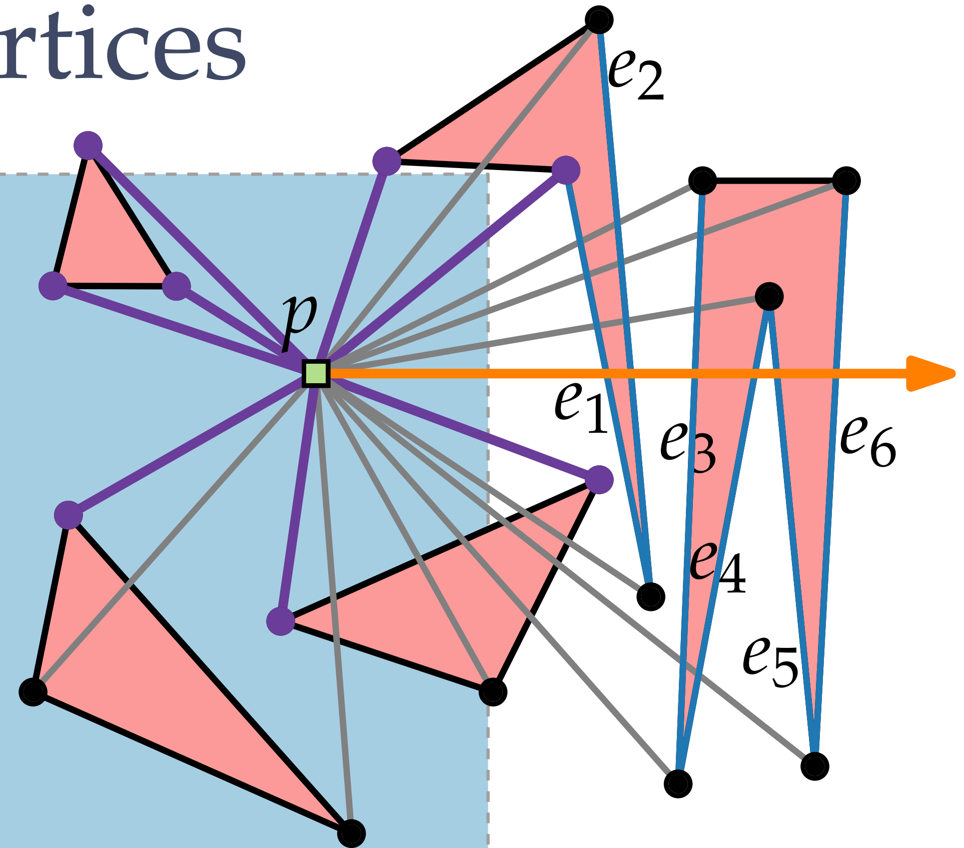
Computing Visible Vertices

VisibleVertices(p, S)

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

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

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



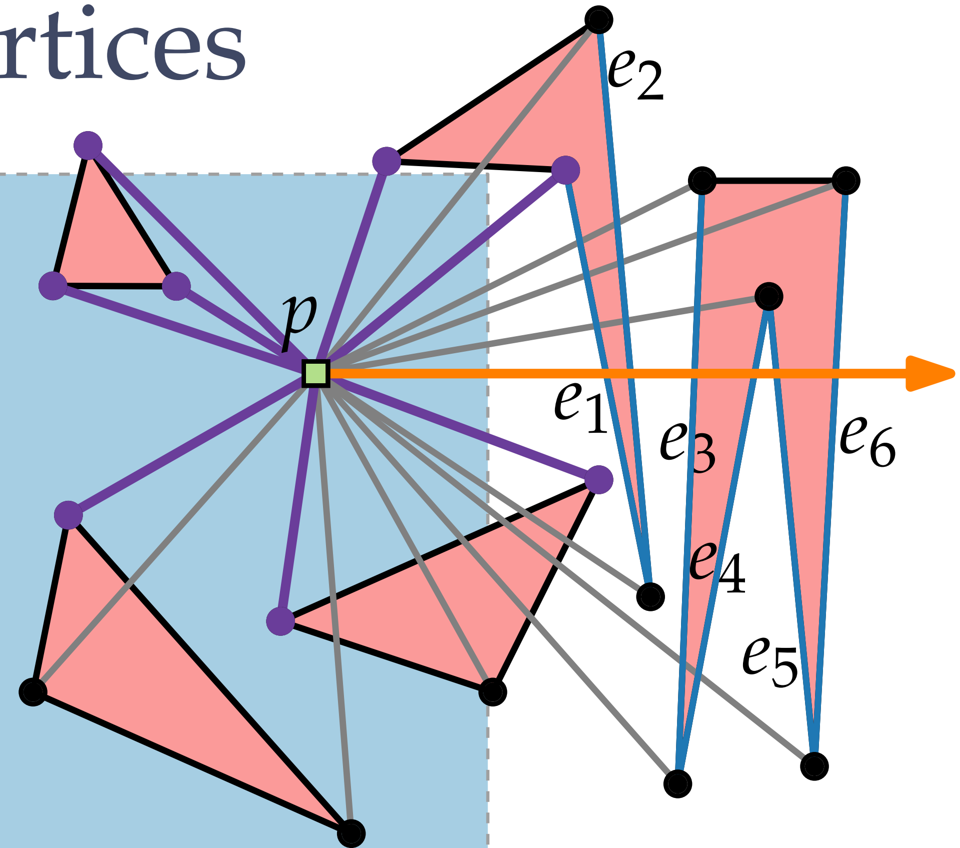
Computing Visible Vertices

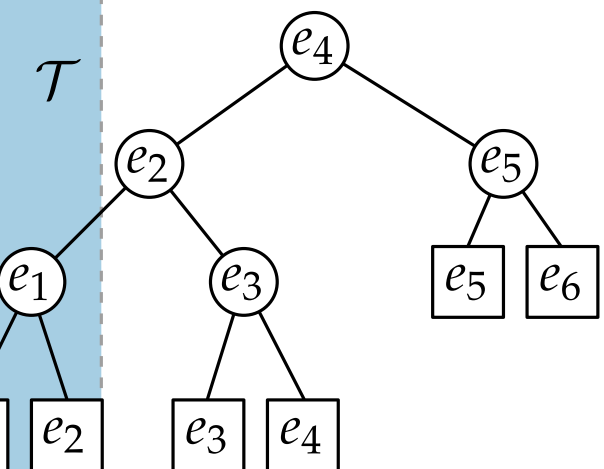
VISIBLEVERTICES(p, S)

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

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

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





Computing Visible Vertices

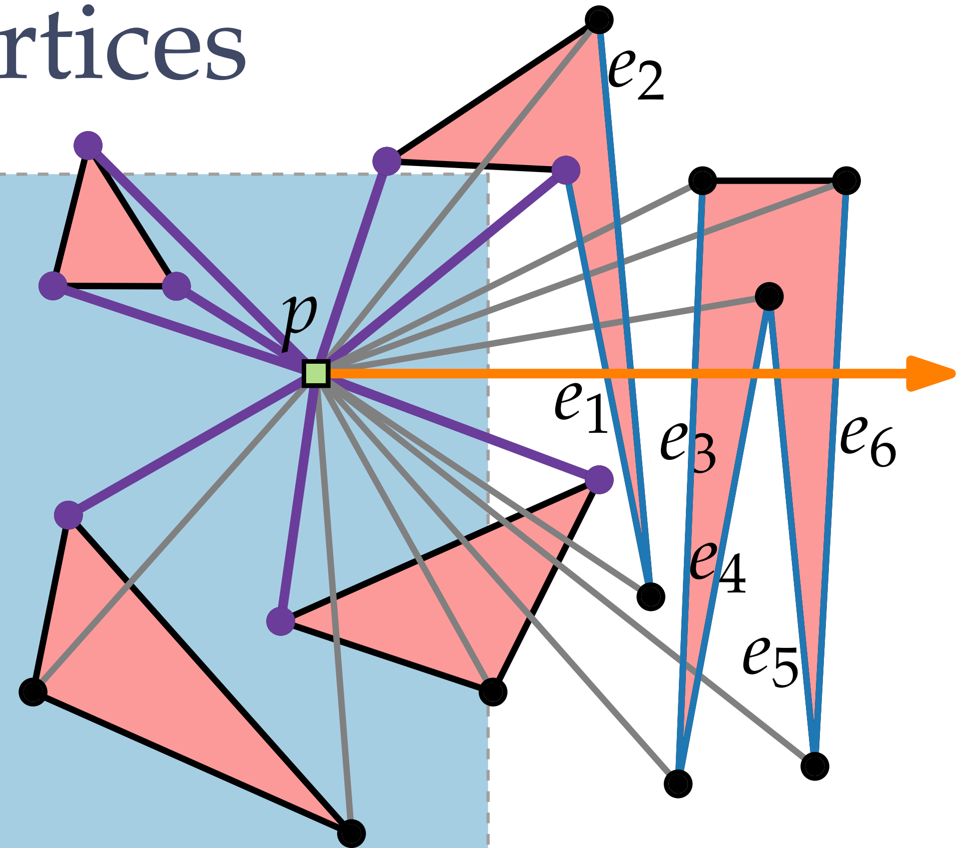
VisibleVertices(p, S)

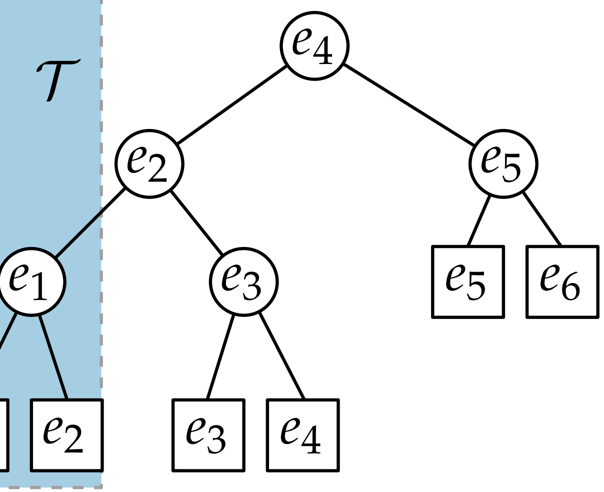
$r \leftarrow p + \mathbb{R}_{\geq 0}\binom{1}{0}$

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

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

sort $V(S)$





Computing Visible Vertices

VisibleVertices(p, S)

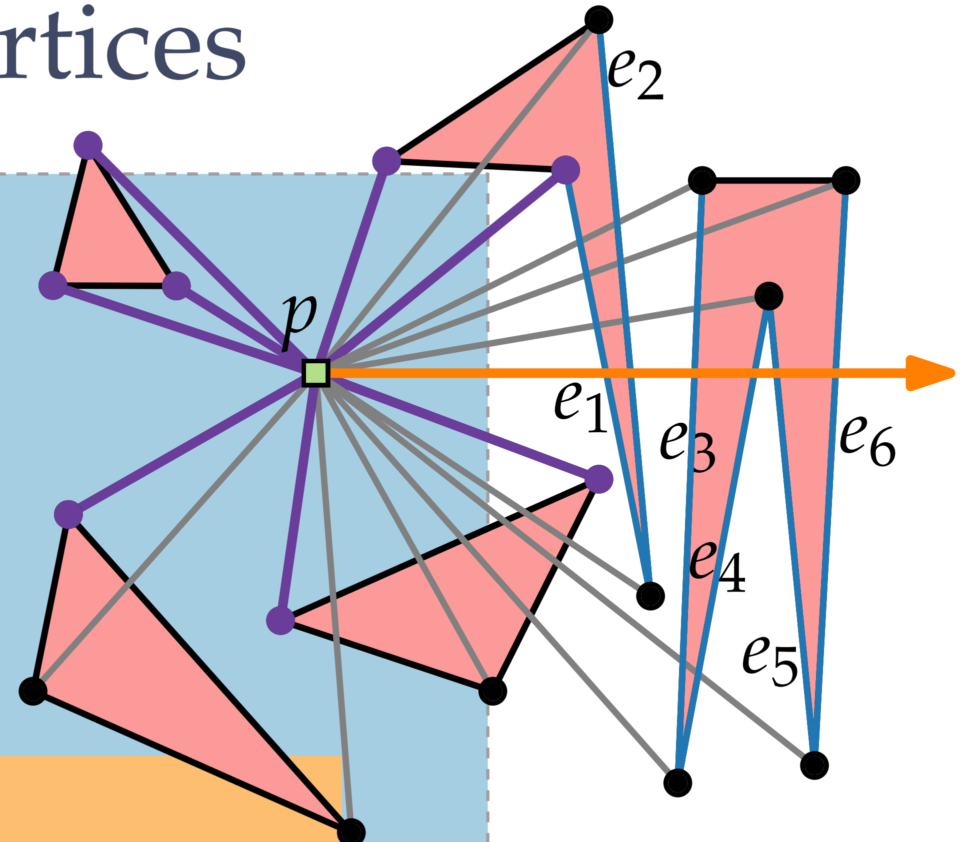
$r \leftarrow p + \mathbb{R}_{\geq 0}\binom{1}{0}$

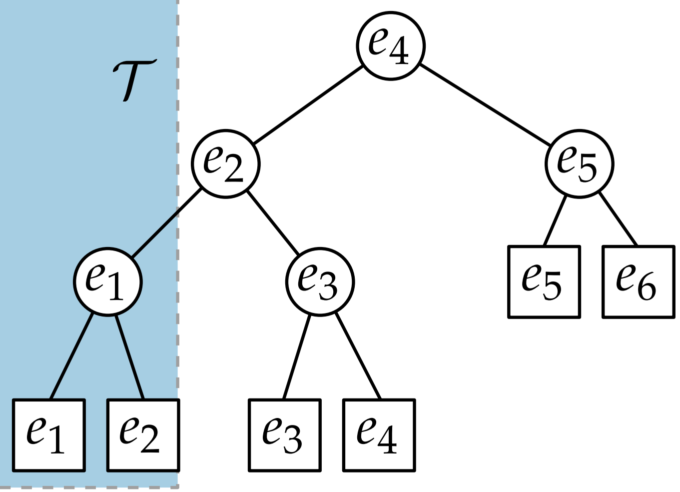
$I \leftarrow \{e \in E(S) \mid e \cap r \neq \emptyset\}$

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

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

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





Computing Visible Vertices

VisibleVertices(p, S)

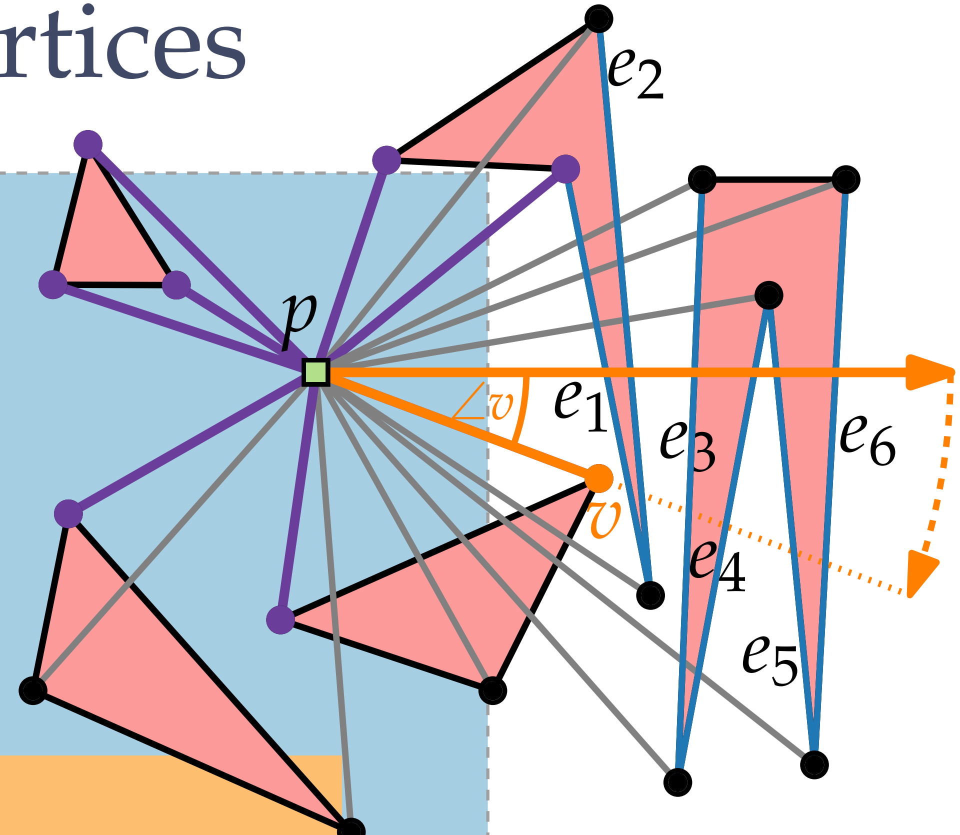
$r \leftarrow p + \mathbb{R}_{\geq 0}\binom{1}{0}$

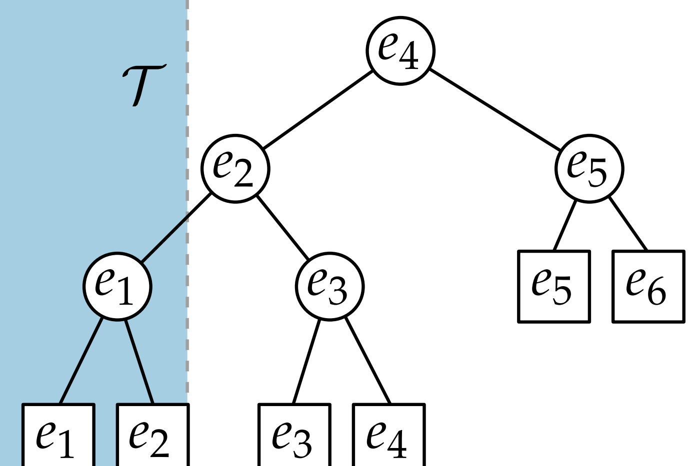
$I \leftarrow \{e \in E(S) \mid e \cap r \neq \emptyset\}$

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

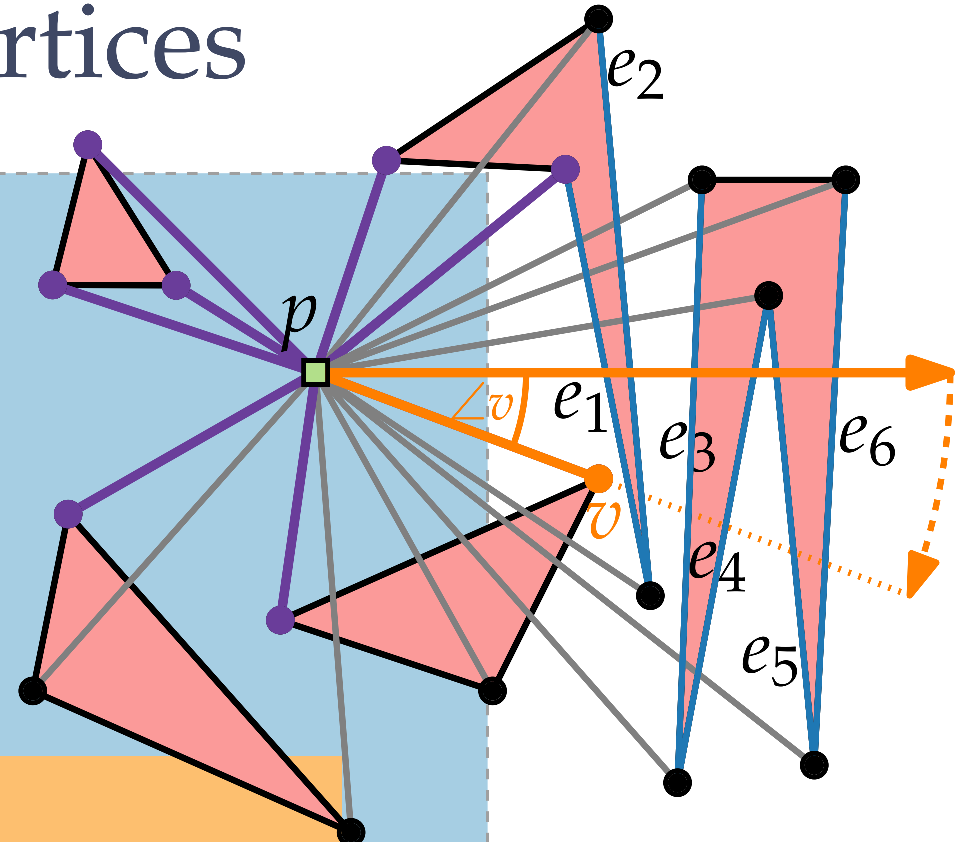
sort $V(S)$ $\qquad v \prec v' :\Leftrightarrow$

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





Computing Visible Vertices



VisibleVertices(p, S)

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

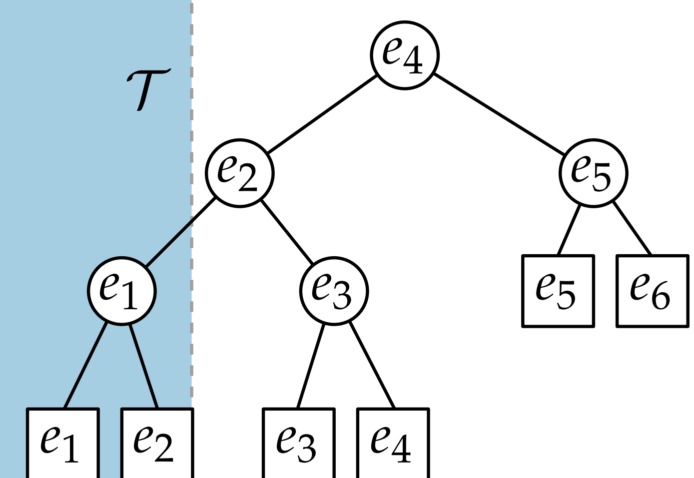
$I \leftarrow \{e \in E(S) \mid e \cap r \neq \emptyset\}$

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

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

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

rotational plane sweep



Computing Visible Vertices

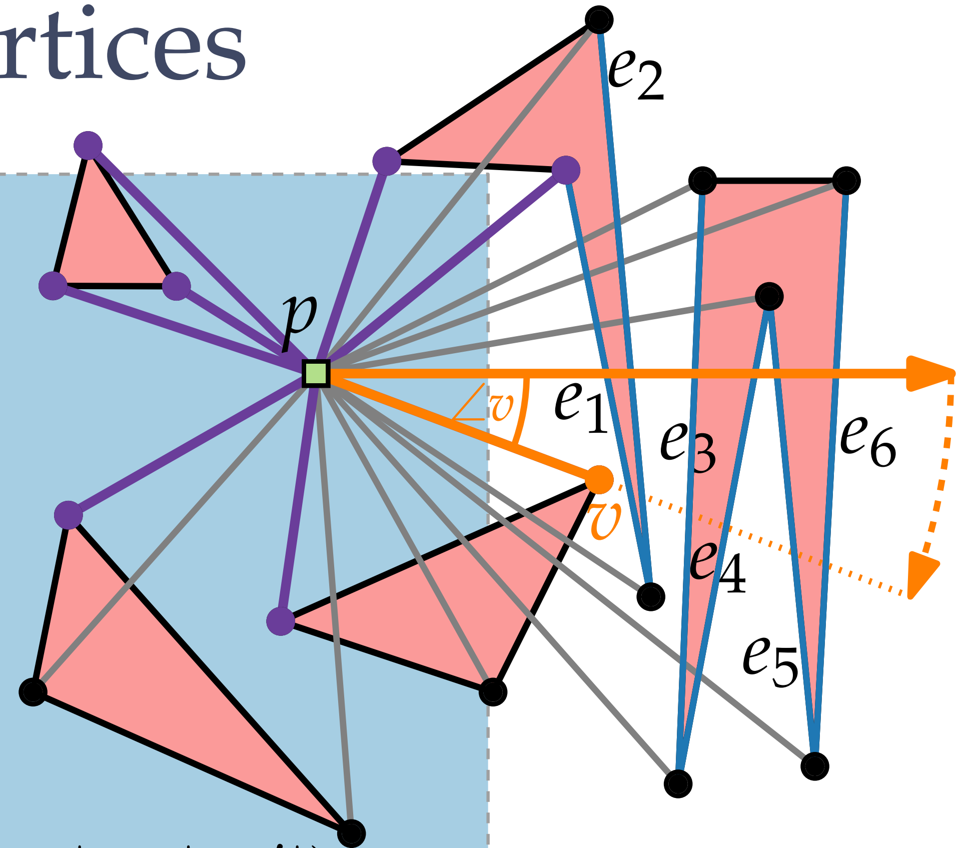
VISIBLEVERTICES(p, S)

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

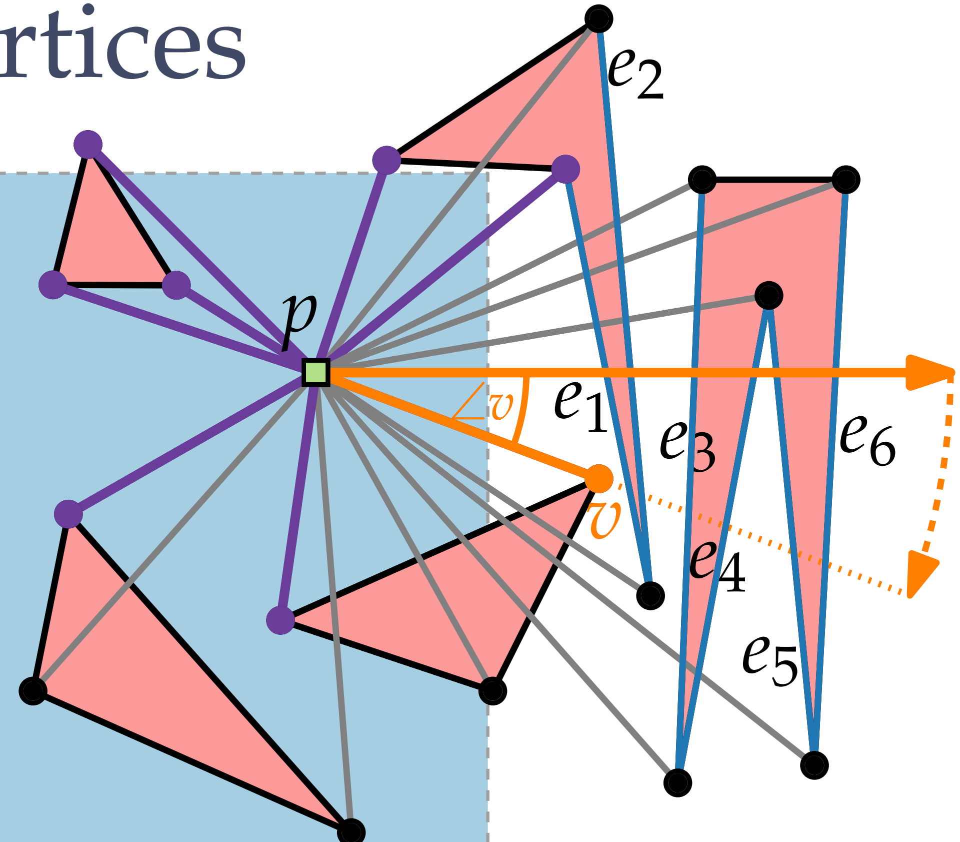
$I \leftarrow \{e \in E(S) \mid e \cap r \neq \emptyset\}$

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

sort $V(S)$ $\qquad v \prec v' :\Leftrightarrow$
$\angle v < \angle v'$ or
$(\angle v = \angle v'$ and $|pv| < |pv'|)$



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 $V(S)$ $\quad v \prec v' :\Leftrightarrow$
$\angle v < \angle v'$ or
$(\angle v = \angle v'$ and $|pv| < |pv'|)$

$W \leftarrow \emptyset$

Computing Visible Vertices

VisibleVertices(p, S)

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

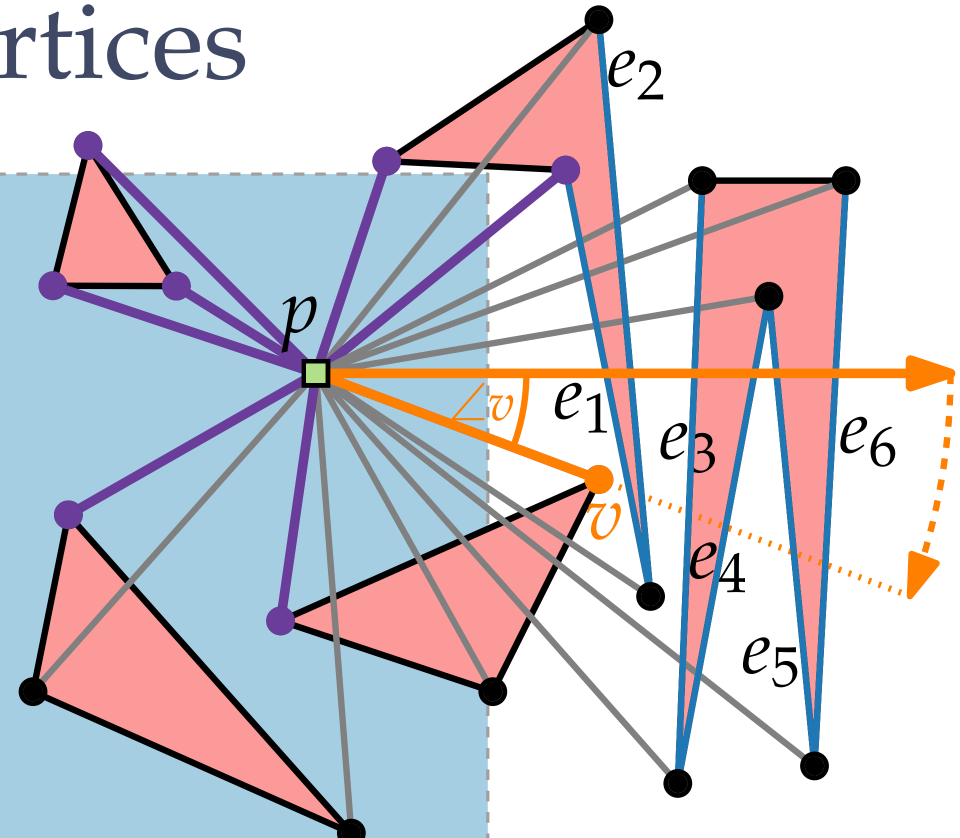
$I \leftarrow \{e \in E(S) \mid e \cap r \neq \emptyset\}$

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

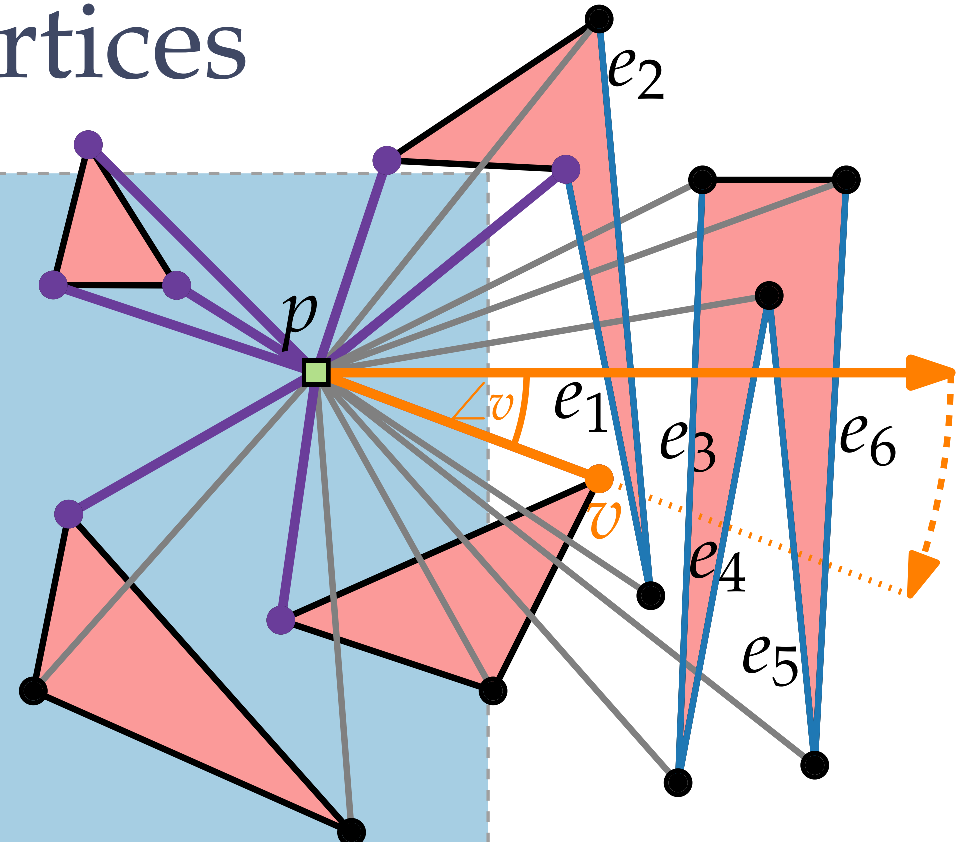
sort $V(S)$ $\quad v \prec v' :\Leftrightarrow$
$\angle v < \angle v'$ or
$(\angle v = \angle v'$ and $|pv| < |pv'|)$

$W \leftarrow \emptyset$

foreach $v \in V(S)$ **do**



Computing Visible Vertices



$\textsc{VisibleVertices}(p, S)$

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

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

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

sort $V(S)$ $\quad v \prec v' :\Leftrightarrow$ $\angle v < \angle v'$ or $(\angle v = \angle v'$ and $|pv| < |pv'|)$

$W \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

 if $\textsc{Visible}(v)$ **then**

Computing Visible Vertices

VISIBLEVERTICES(p, S)

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

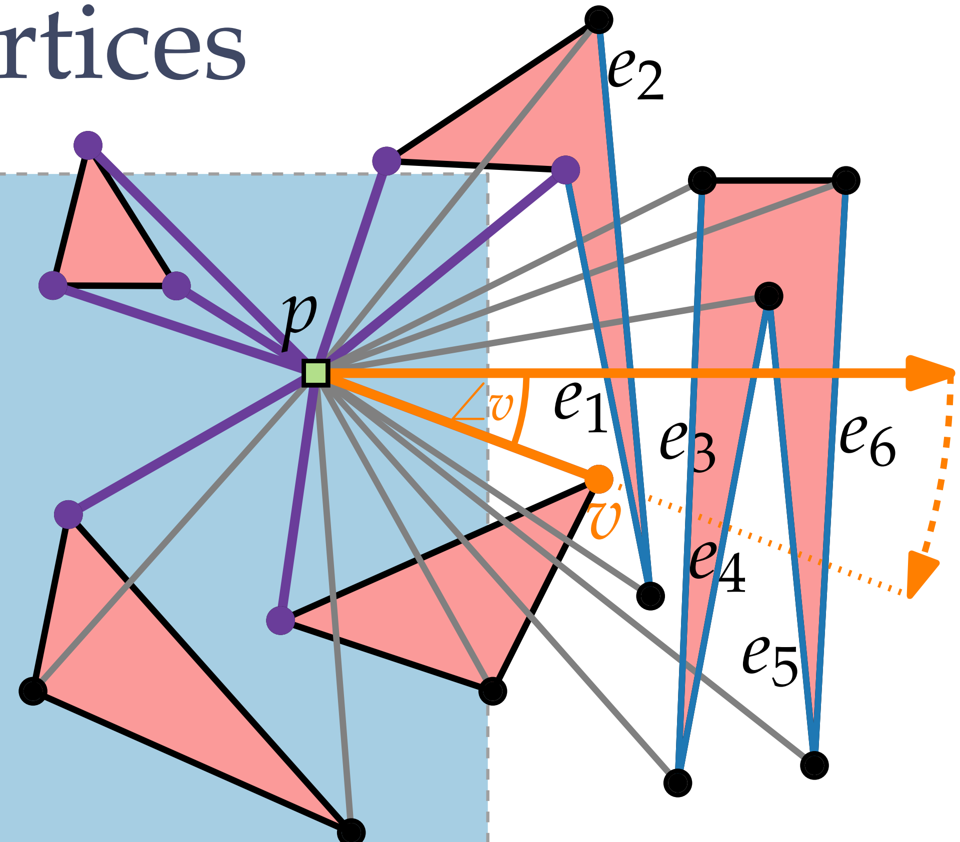
$I \leftarrow \{e \in E(S) \mid e \cap r \neq \emptyset\}$

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

sort $V(S)$ $\quad v \prec v' :\Leftrightarrow$
$\angle v < \angle v'$ or
$(\angle v = \angle v'$ and $|pv| < |pv'|)$

$W \leftarrow \emptyset$

foreach $v \in V(S)$ **do**
  if VISIBLE(v) **then**
    $W \leftarrow W \cup \{v\}$



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)

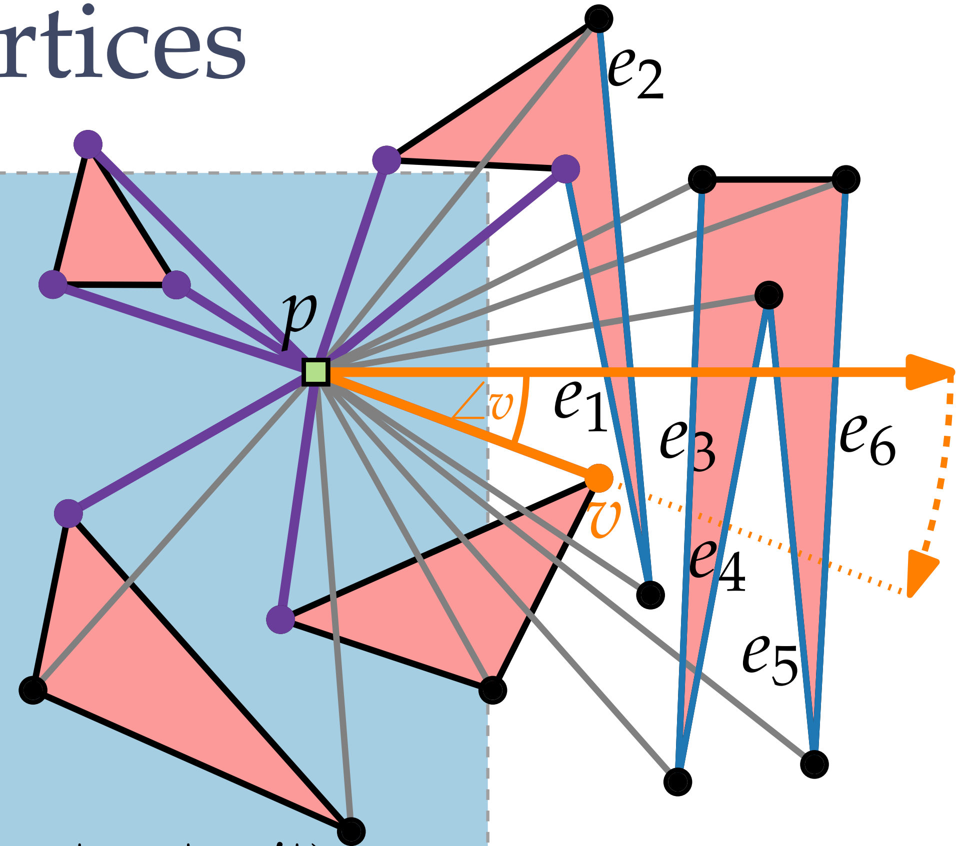
sort $V(S)$ $\quad v \prec v' :\Leftrightarrow$
$\angle v < \angle v'$ or
$(\angle v = \angle v'$ and $|pv| < |pv'|)$

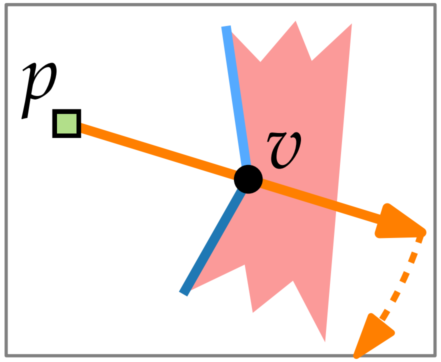
$W \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

  if Visible(v) **then**

    $W \leftarrow W \cup \{v\}$





Computing Visible Vertices

$\textsc{VisibleVertices}(p, S)$

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

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

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

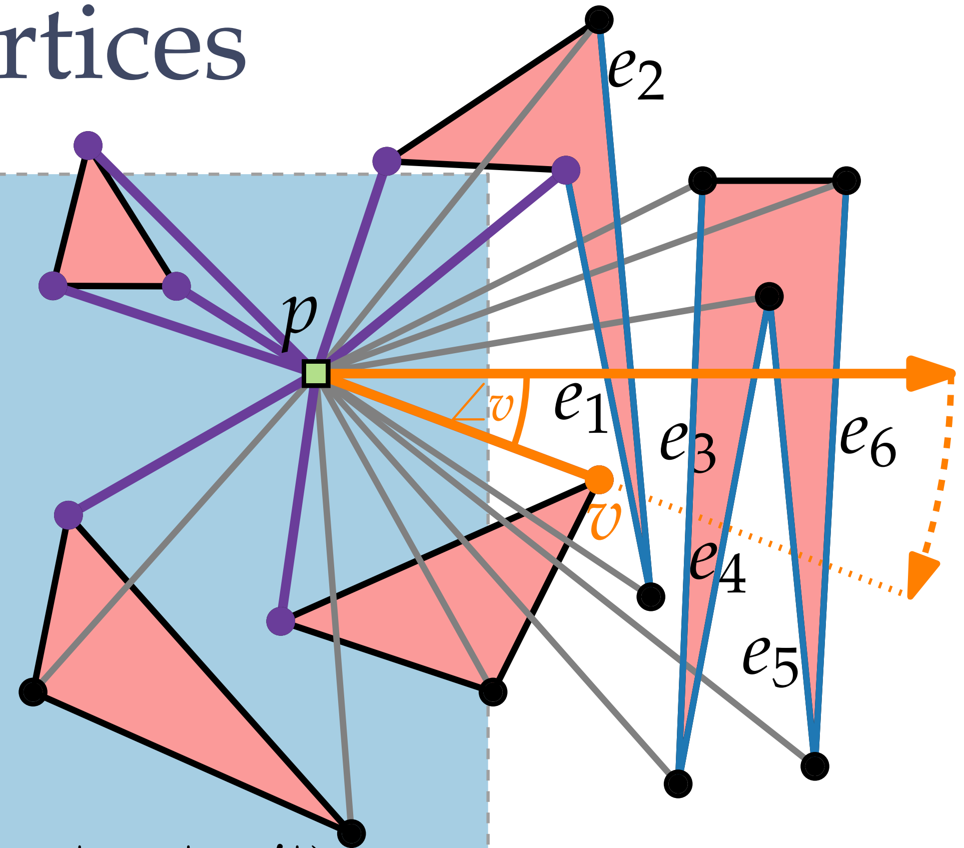
sort $V(S)$ $\quad v \prec v' :\Leftrightarrow$ $\angle v < \angle v'$ or $(\angle v = \angle v'$ and $|pv| < |pv'|)$

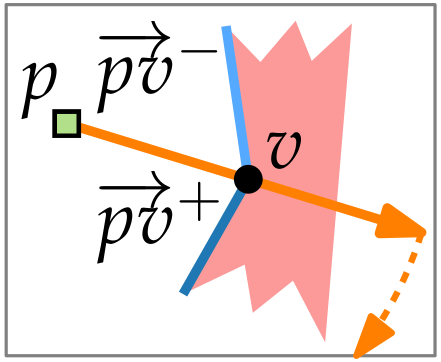
$W \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

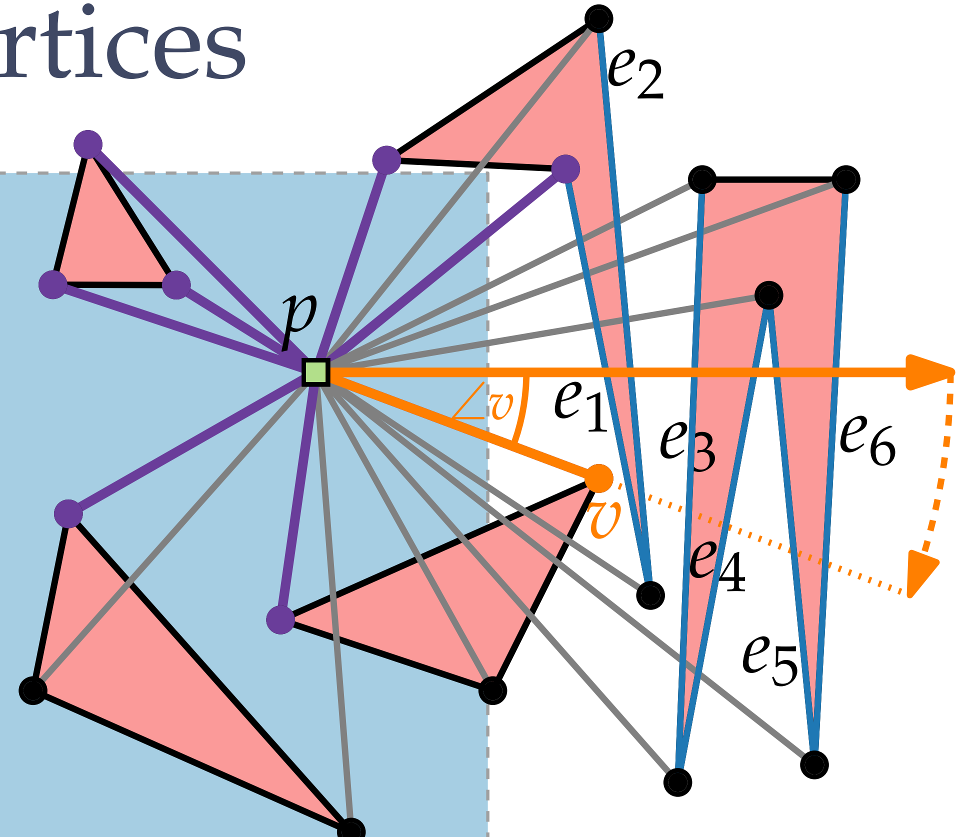
    if $\textsc{Visible}(v)$ **then**

        $W \leftarrow W \cup \{v\}$





Computing Visible Vertices



VisibleVertices(p, S)

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

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

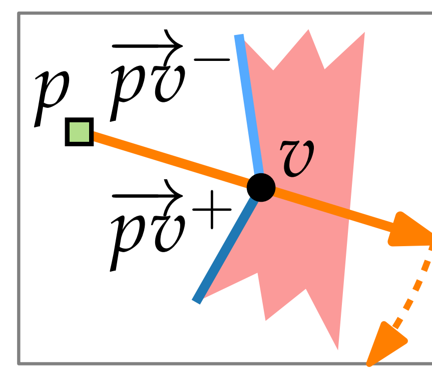
$\mathcal{T} \leftarrow \text{balancedBinaryTree}(I)$

sort $V(S)$ $\quad v \prec v' :\Leftrightarrow$
$\angle v < \angle v'$ or
$(\angle v = \angle v'$ and $|pv| < |pv'|)$

$W \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

- **if** Visible(v) **then**
  - $W \leftarrow W \cup \{v\}$
- insert into $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^{+}$



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 $V(S)$ $\quad v \prec v' :\Leftrightarrow$
$\angle v < \angle v'$ or
$(\angle v = \angle v'$ and $|pv| < |pv'|)$

$W \leftarrow \emptyset$

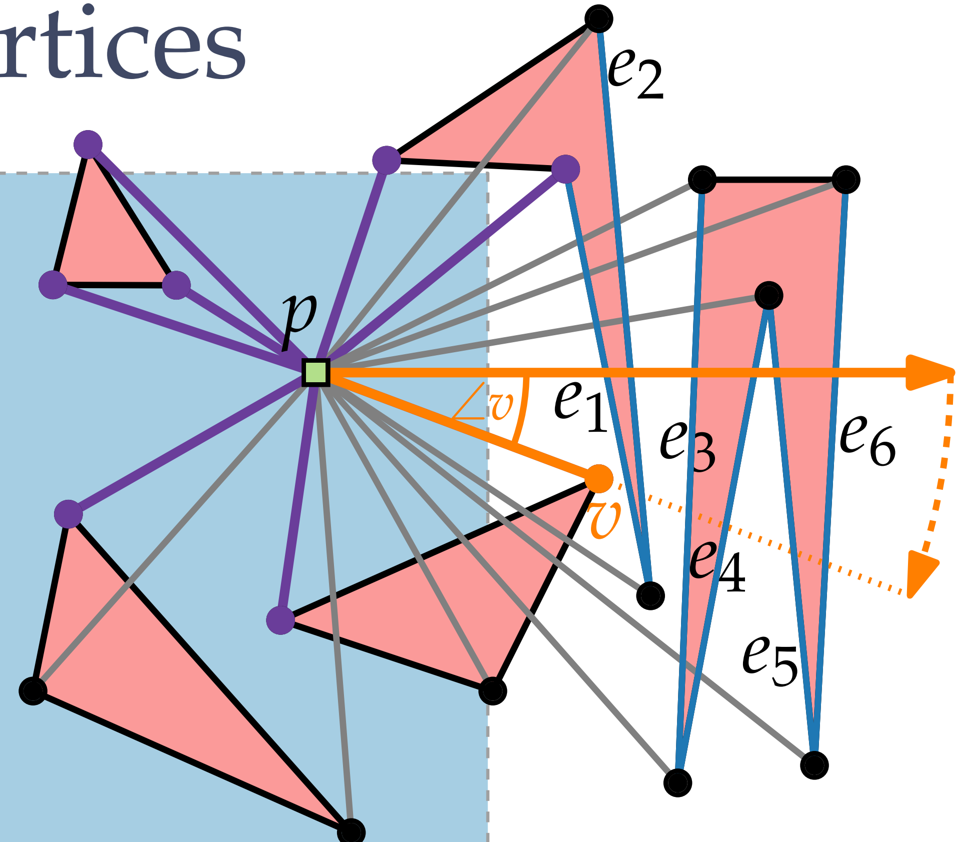
foreach $v \in V(S)$ **do**

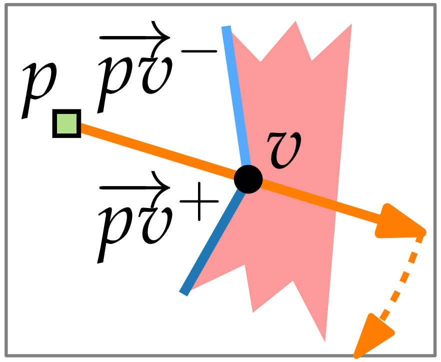
  if Visible(v) **then**

    $W \leftarrow W \cup \{v\}$

  insert into $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^{+}$

  delete from $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^{-}$





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 $V(S)$ $\quad v \prec v' :\Leftrightarrow$ $\angle v < \angle v'$ or $(\angle v = \angle v'$ and $|pv| < |pv'|)$

$W \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

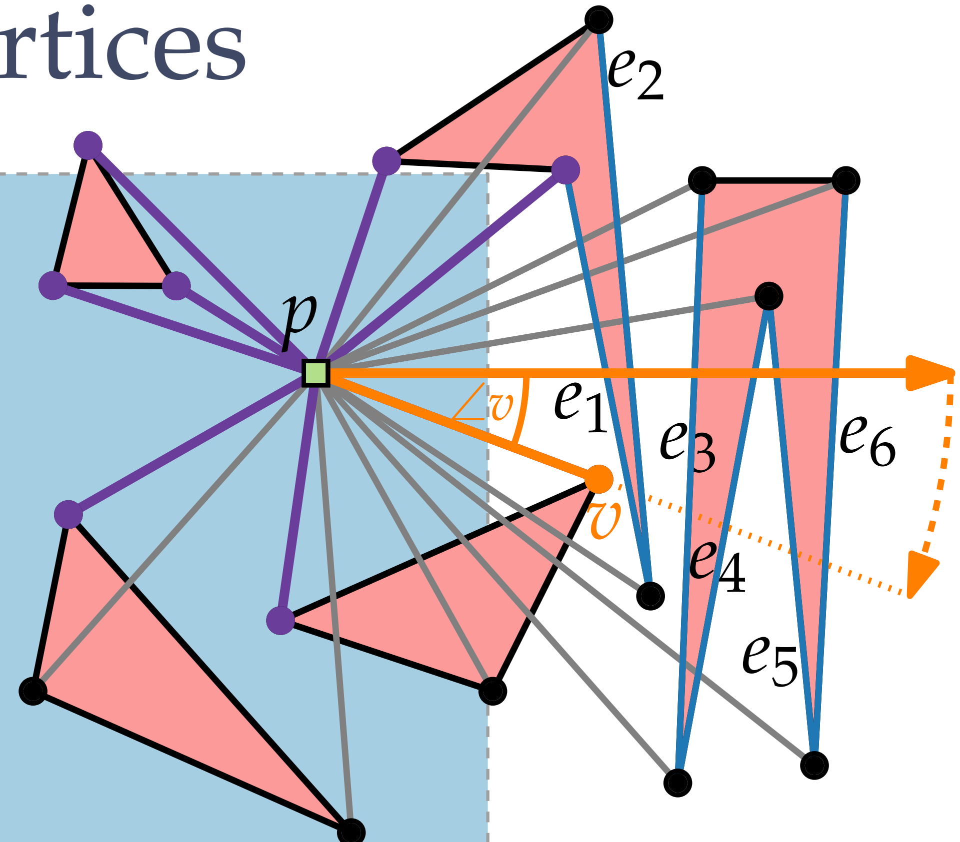
  if Visible(v) **then**

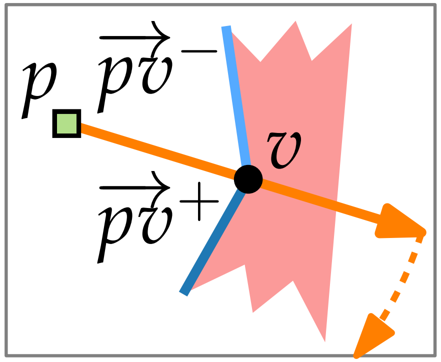
    $W \leftarrow W \cup \{v\}$

  insert into $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^{+}$

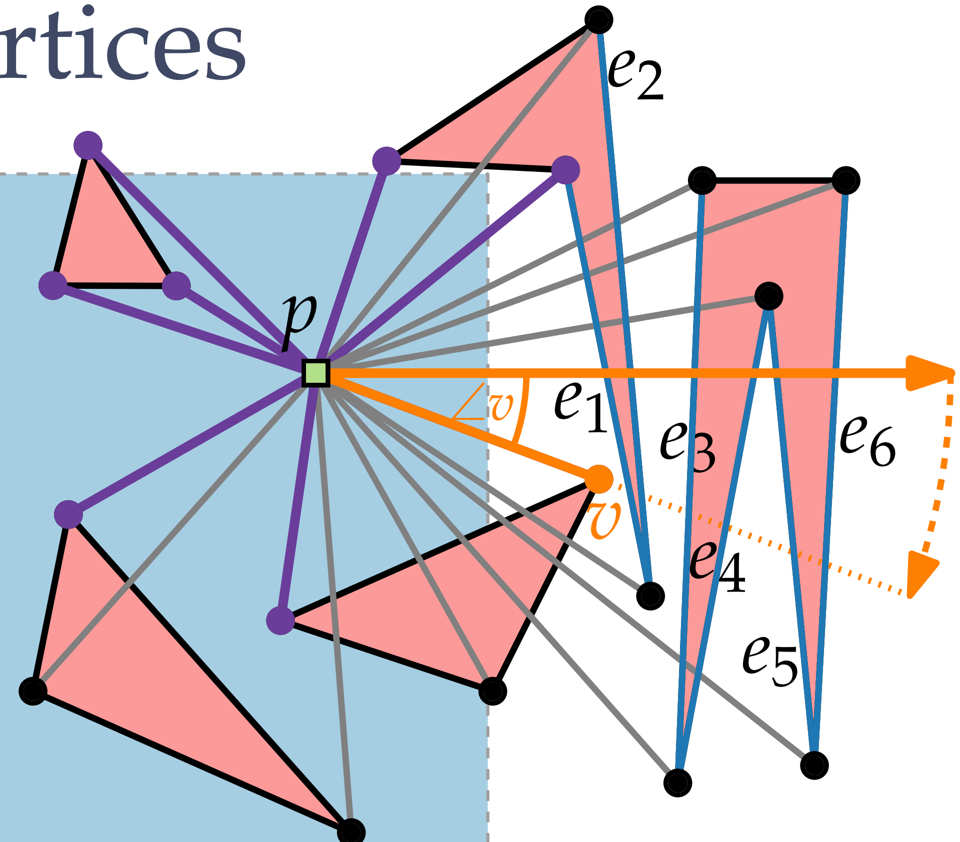
  delete from $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^{-}$

return W





Computing Visible Vertices



$\textsc{VisibleVertices}(p, S)$

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

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

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

sort $V(S)$ $\qquad v \prec v' :\Leftrightarrow \angle v < \angle v'$ or $(\angle v = \angle v'$ and $|pv| < |pv'|)$

$W \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

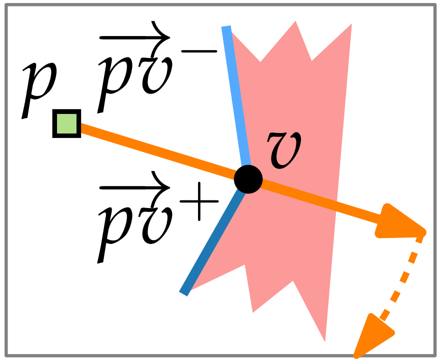
  if $\textsc{Visible}(v)$ **then**

    $W \leftarrow W \cup \{v\}$

  insert into $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^+$

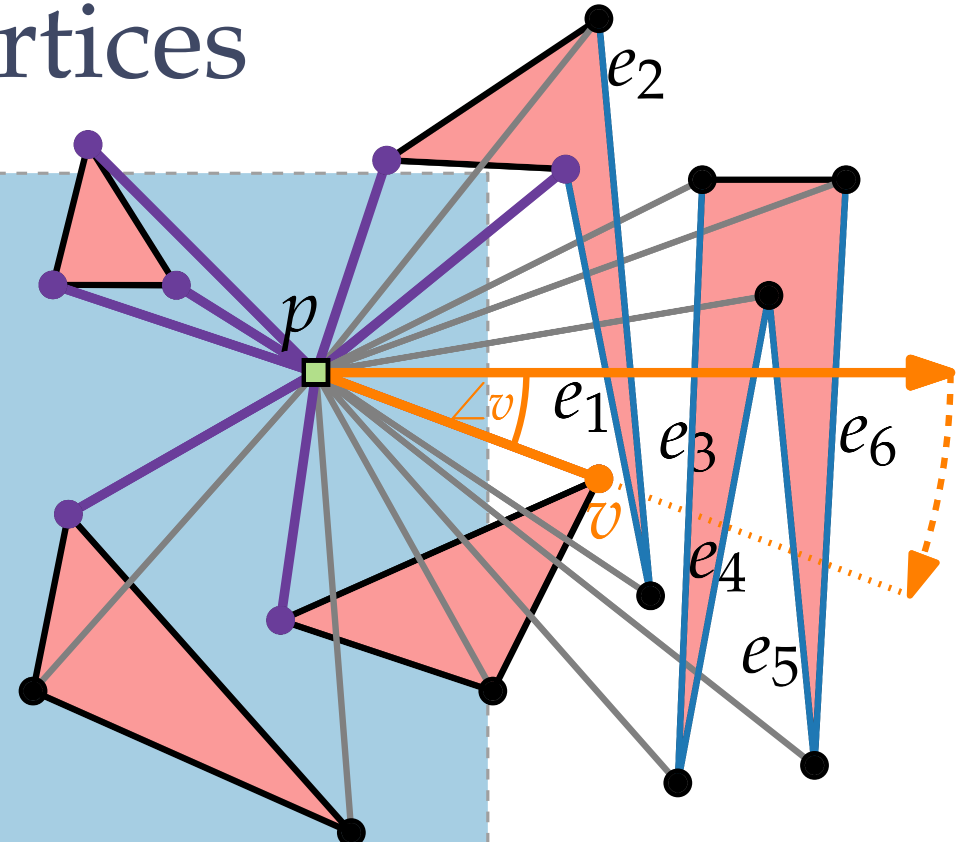
  delete from $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^-$

return W



$O(n \log n)$

Computing Visible Vertices



$\textsc{VisibleVertices}(p, S)$

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

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

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

sort $V(S)$ $\qquad v \prec v' :\Leftrightarrow$
$\angle v < \angle v'$ or
$(\angle v = \angle v'$ and $|pv| < |pv'|)$

$W \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

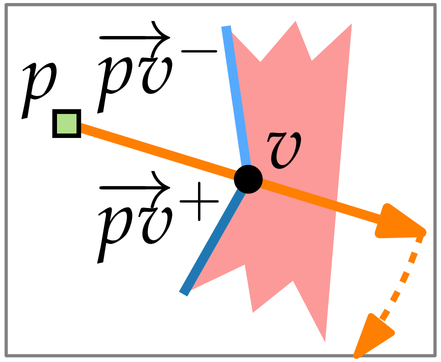
  if $\textsc{Visible}(v)$ **then** ?

    $W \leftarrow W \cup \{v\}$

  insert into $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^{+}$

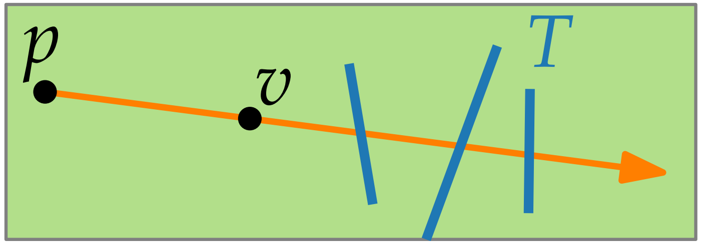
  delete from $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^{-}$

return W

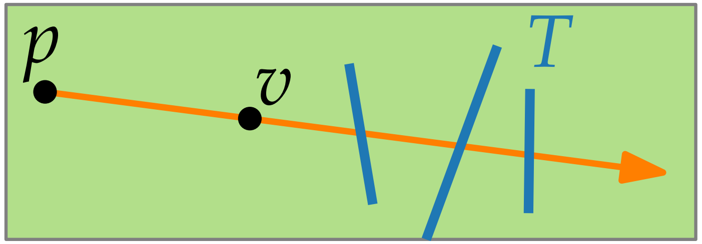


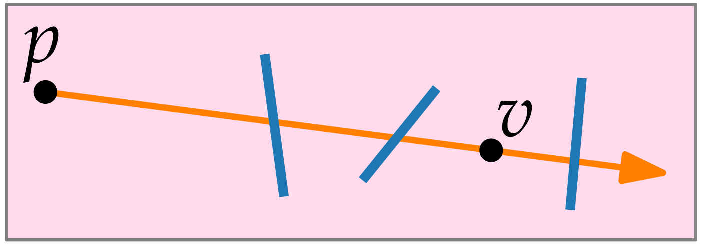
$O(n \log n)$

Cases

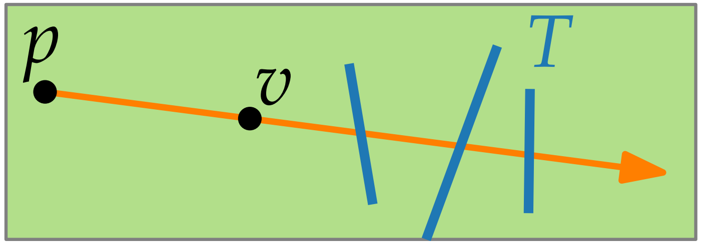


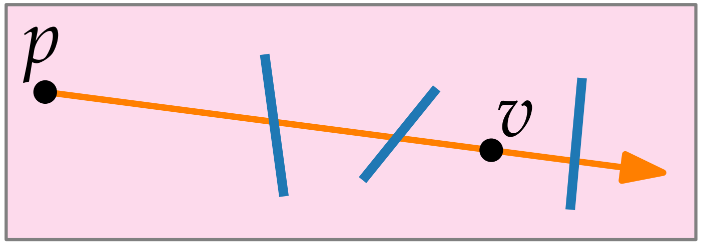
Cases

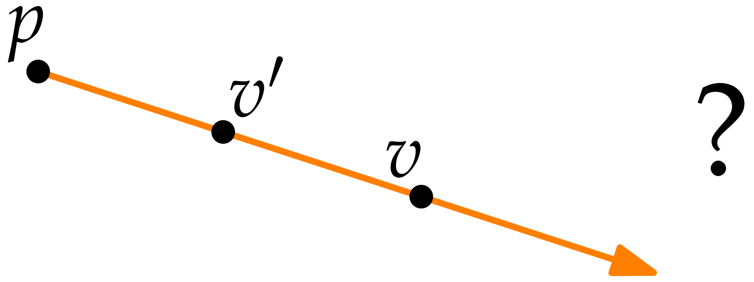




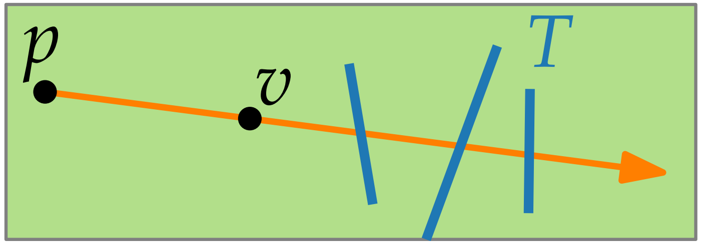
Cases

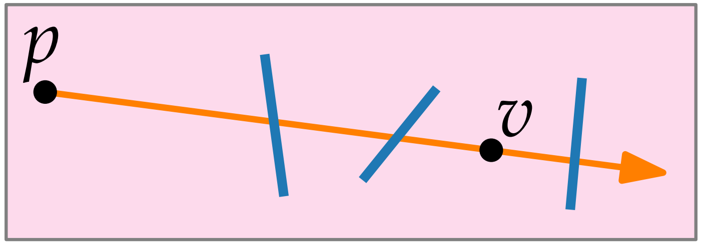






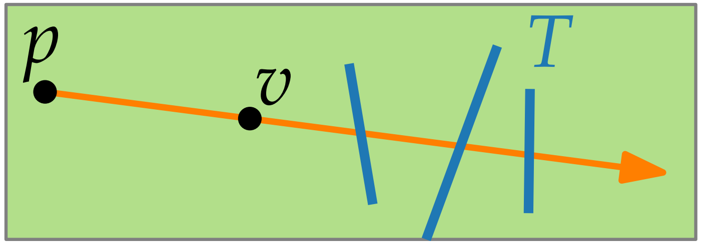
Cases

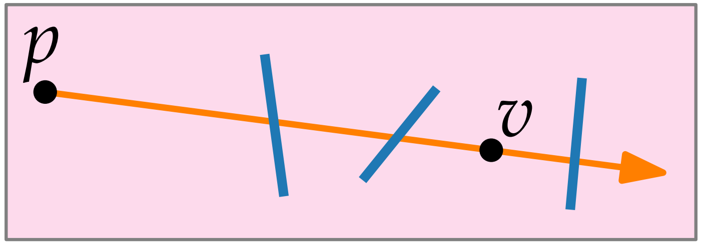




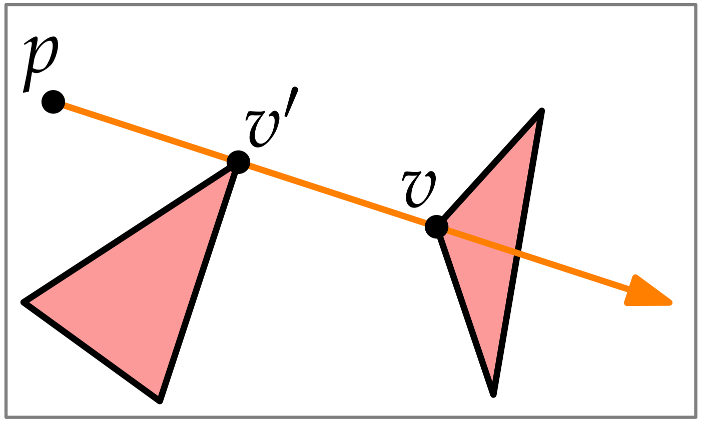
Let v' be the immediate predecessor of v according to $\prec$.

Cases

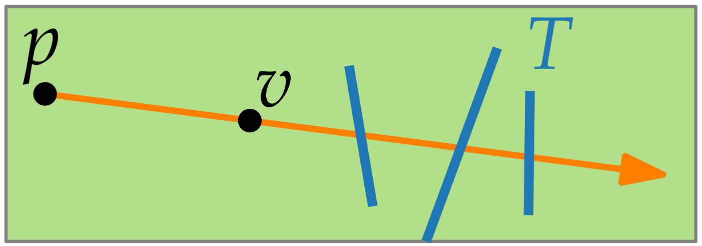


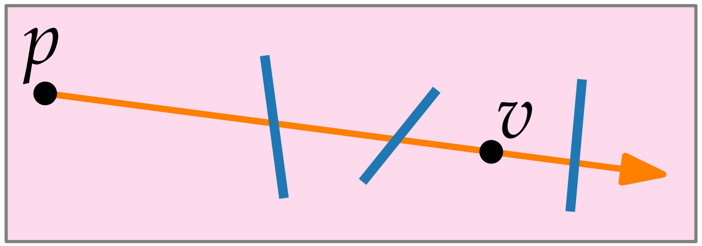


Let v' be the immediate predecessor of v according to $\prec$.

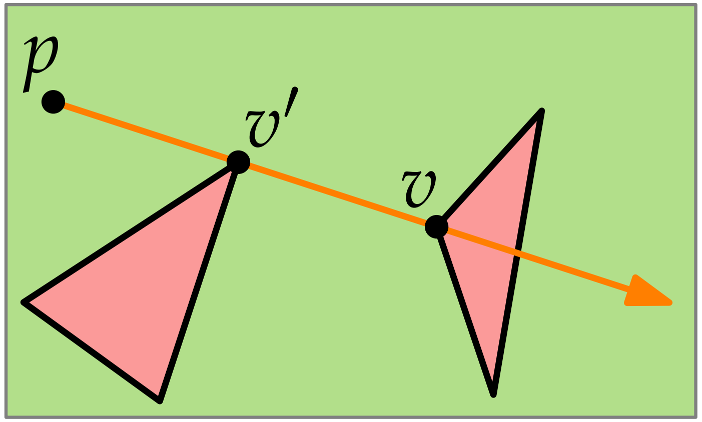


Cases

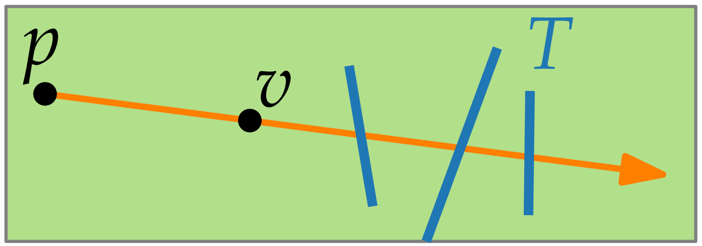


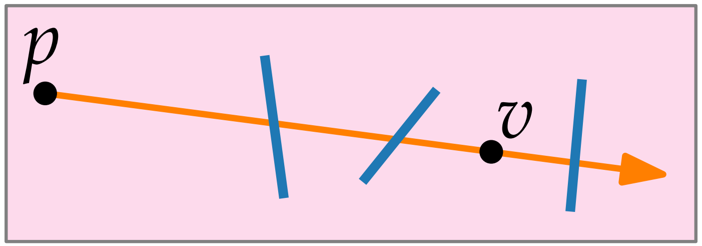


Let v' be the immediate predecessor of v according to $\prec$.

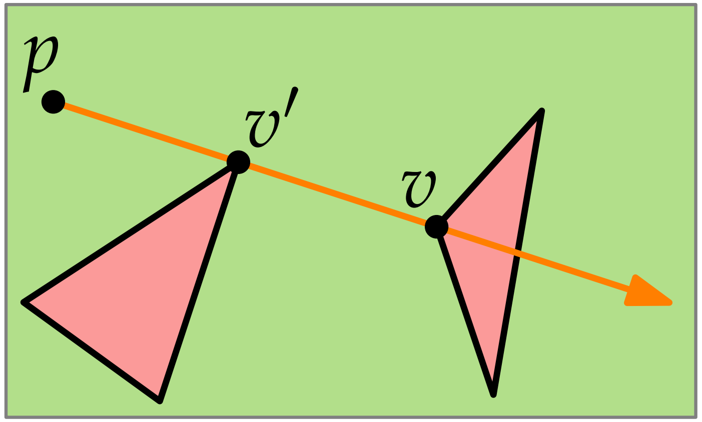


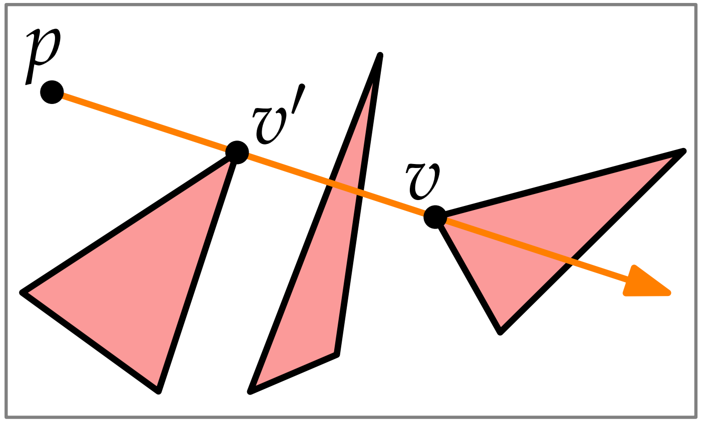
Cases



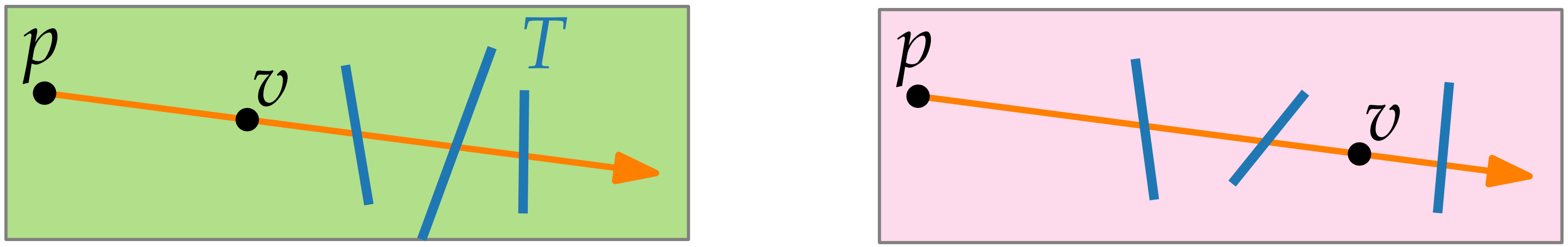


Let v' be the immediate predecessor of v according to $\prec$.

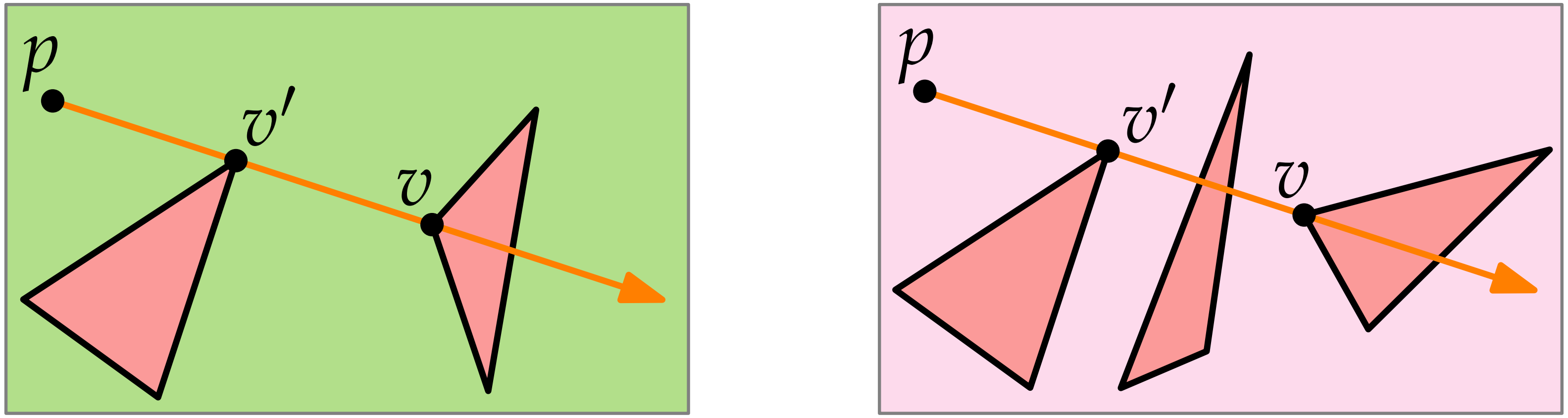




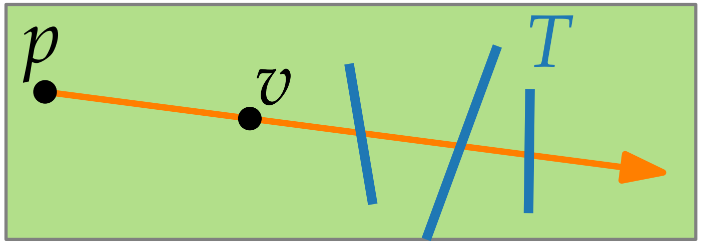
Cases

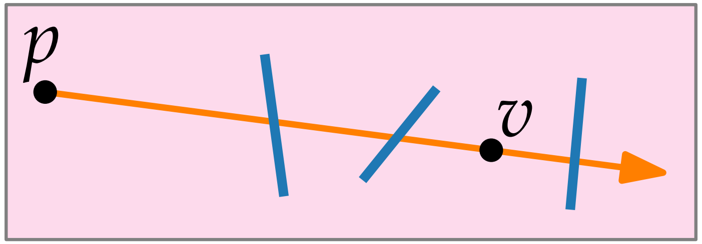


Let v' be the immediate predecessor of v according to $\prec$.

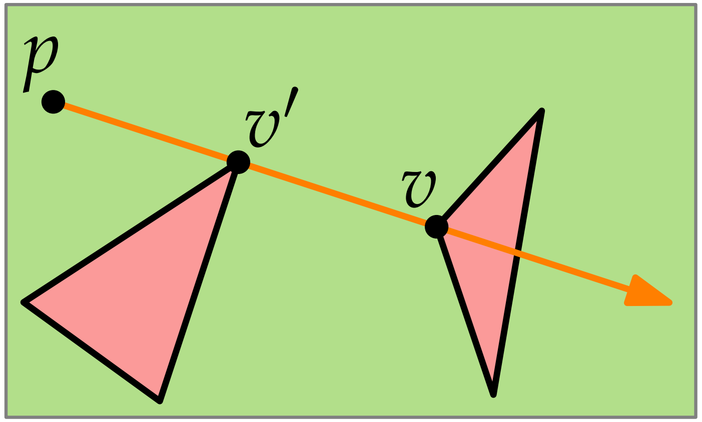


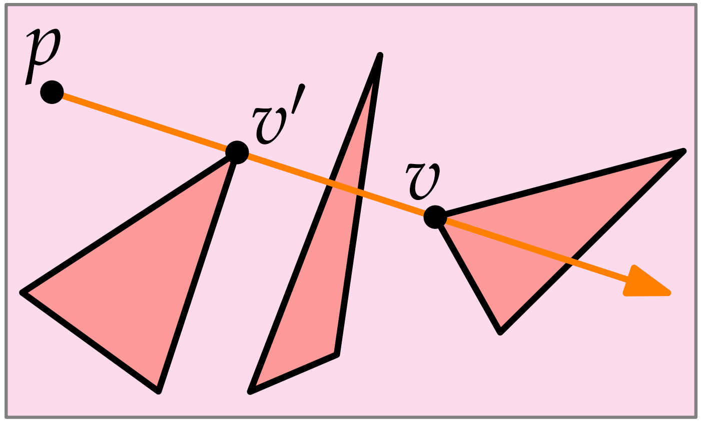
Cases

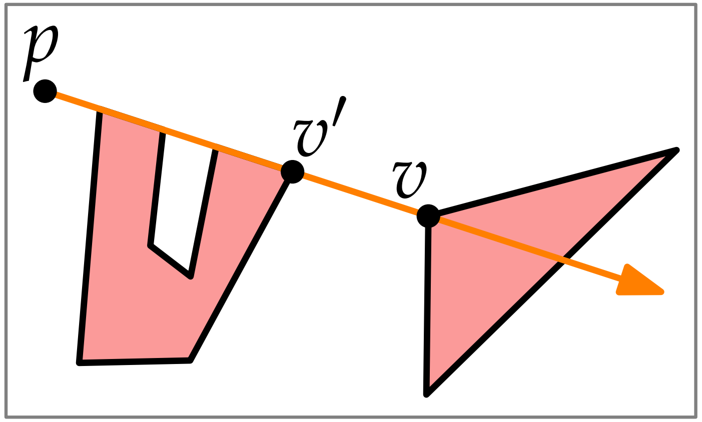




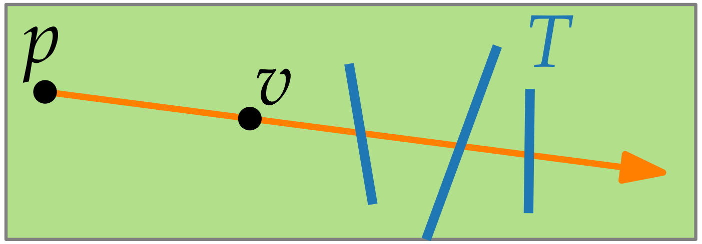
Let v' be the immediate predecessor of v according to $\prec$.

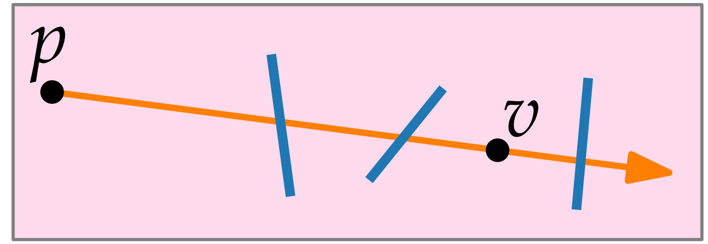




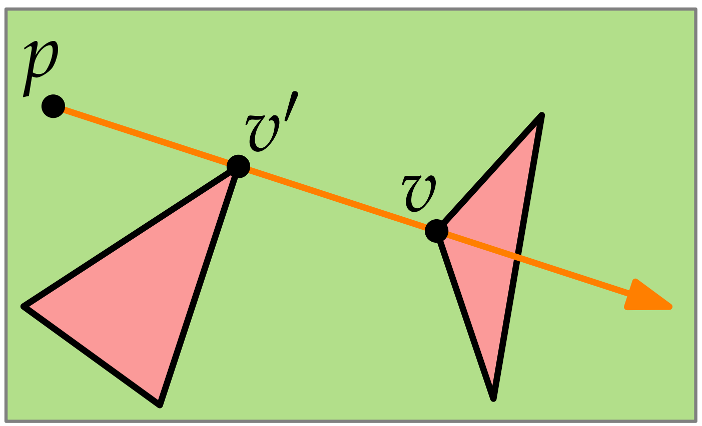


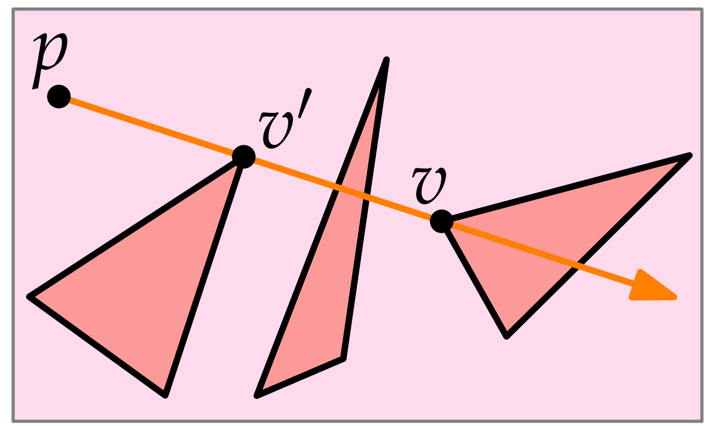
Cases

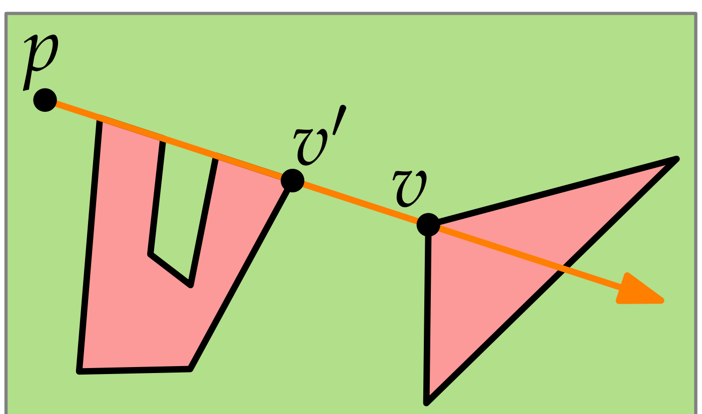




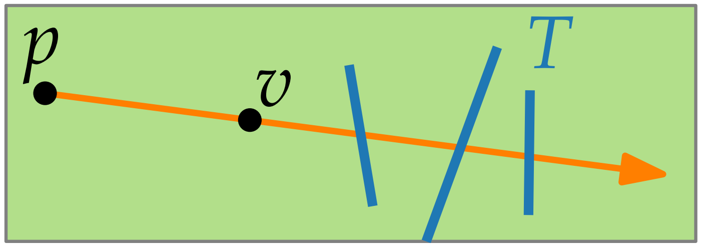
Let v' be the immediate predecessor of v according to $\prec$.

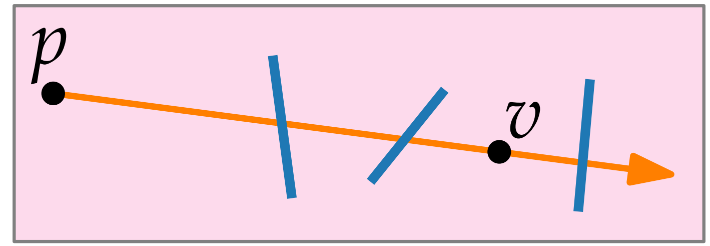




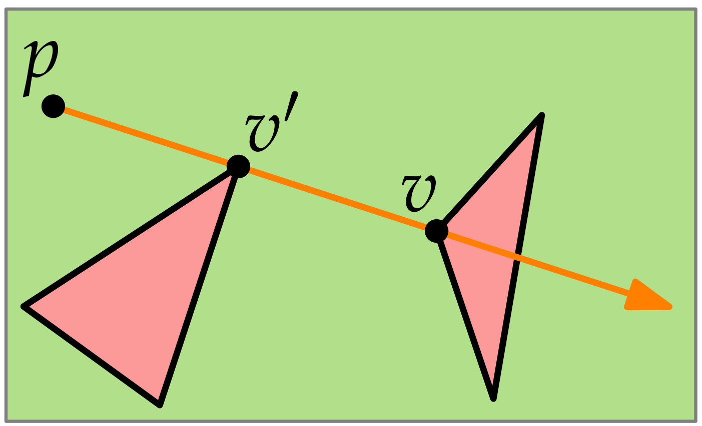


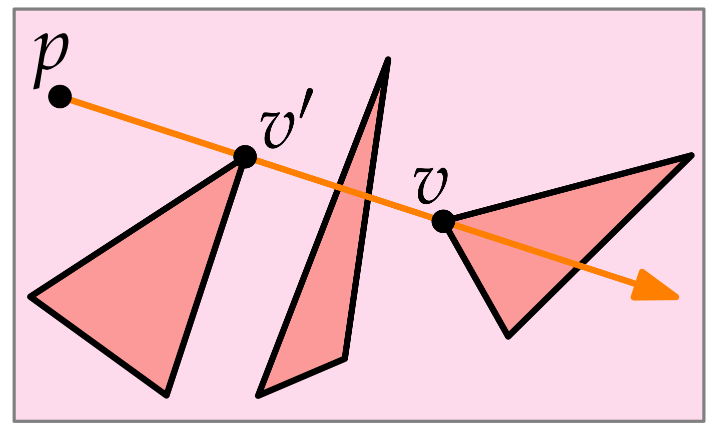
Cases

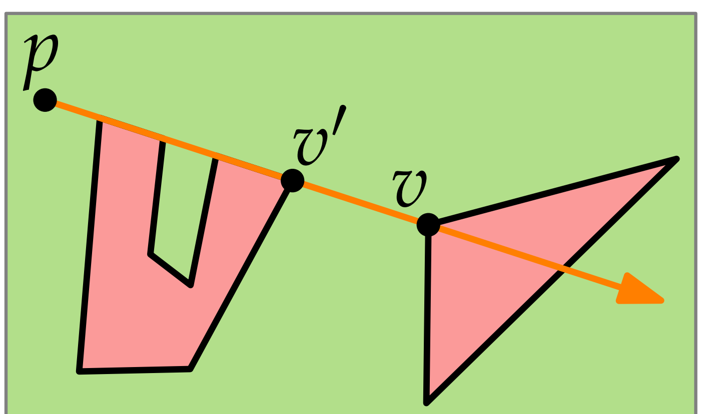


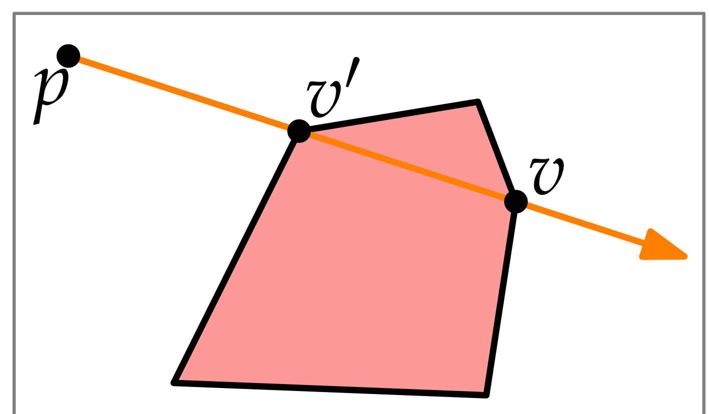


Let v' be the immediate predecessor of v according to $\prec$.

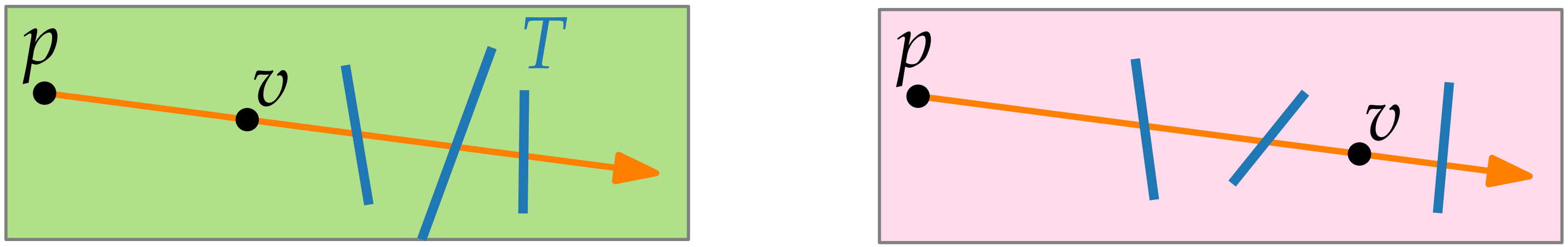




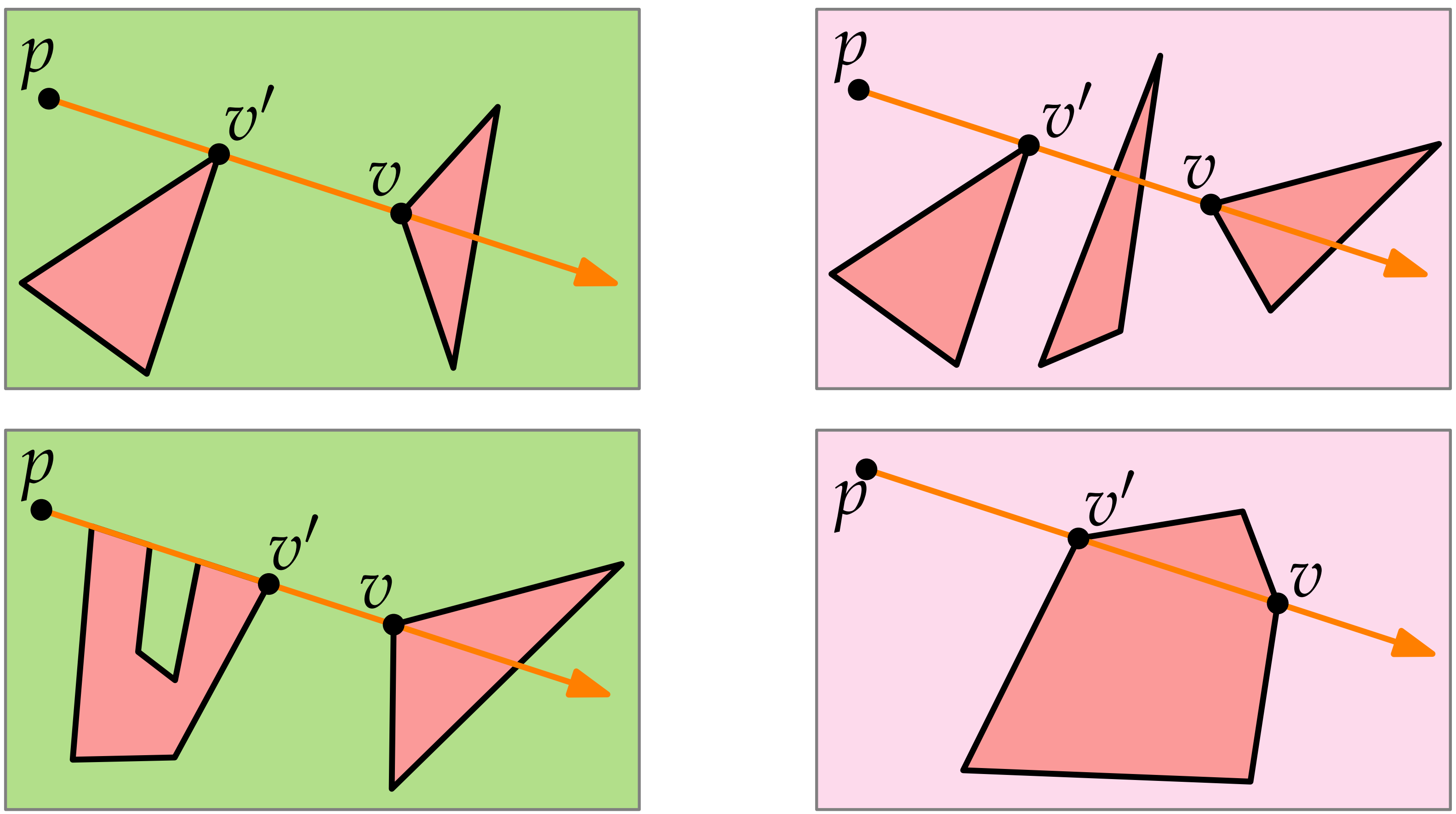




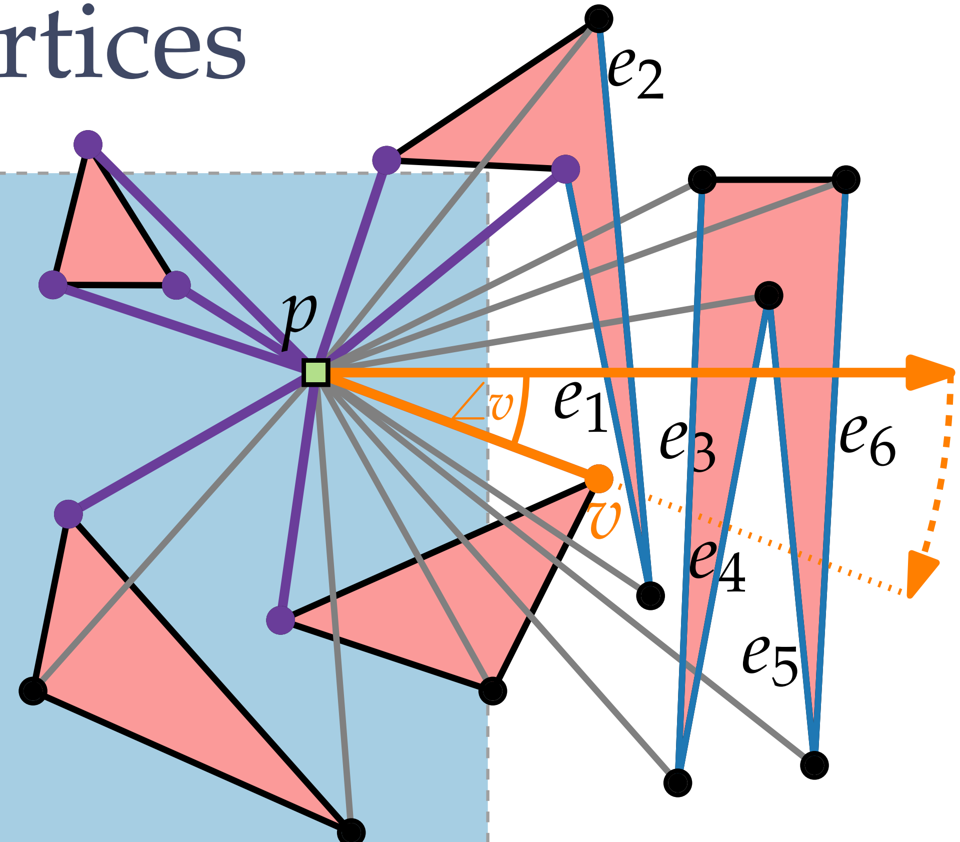
Cases



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



Computing Visible Vertices



VisibleVertices(p, S)

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

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

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

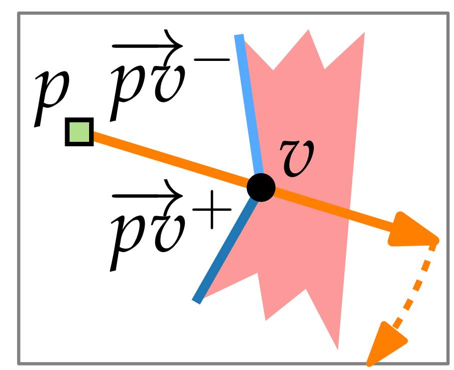
sort $V(S)$ $\quad$ $v \prec v' :\Leftrightarrow$ $\angle v < \angle v'$ or $(\angle v = \angle v'$ and $|pv| < |pv'|)$

$W \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

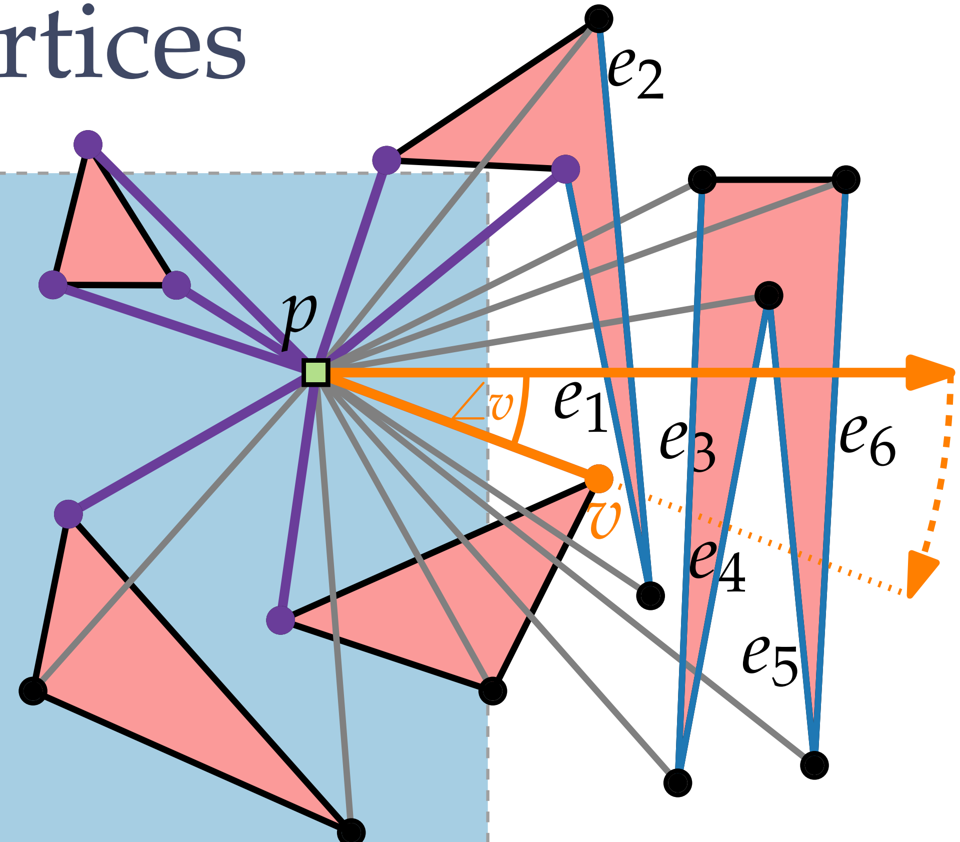
- **if** Visible(v) **then** ?
  - $W \leftarrow W \cup \{v\}$
- insert into $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^+$
- delete from $\mathcal{T}$ edges incident to v in $\overrightarrow{pv}^-$

return W



$O(n \log n)$

Computing 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 $V(S)$ $\qquad v \prec v' :\Leftrightarrow$ $\angle v < \angle v'$ or $(\angle v = \angle v'$ and $|pv| < |pv'|)$

$W \leftarrow \emptyset$

foreach $v \in V(S)$ **do**

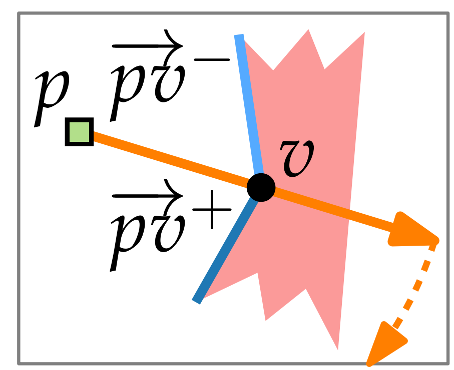
  if VISIBLE(v) **then** $\quad O(1)$

    $W \leftarrow W \cup \{v\}$

  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

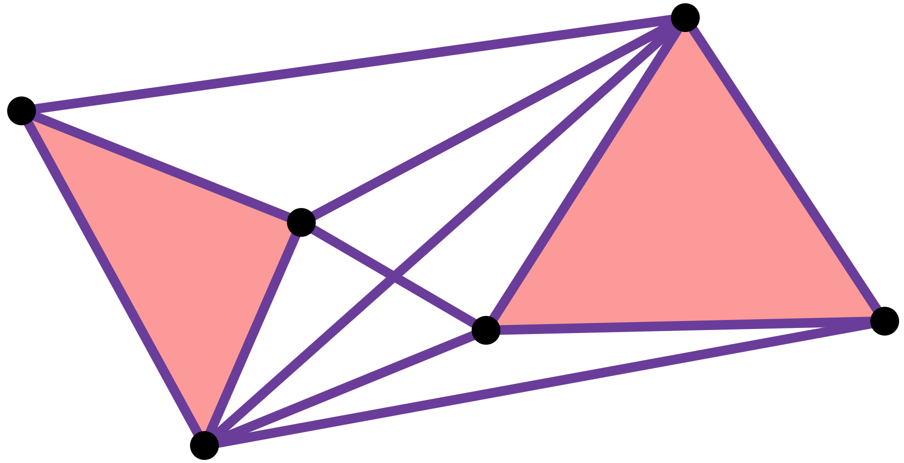
VisibilityGraph(S)

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 $(V(S), E)$

$O(n) \cdot$

?



Computing the Visibility Graph

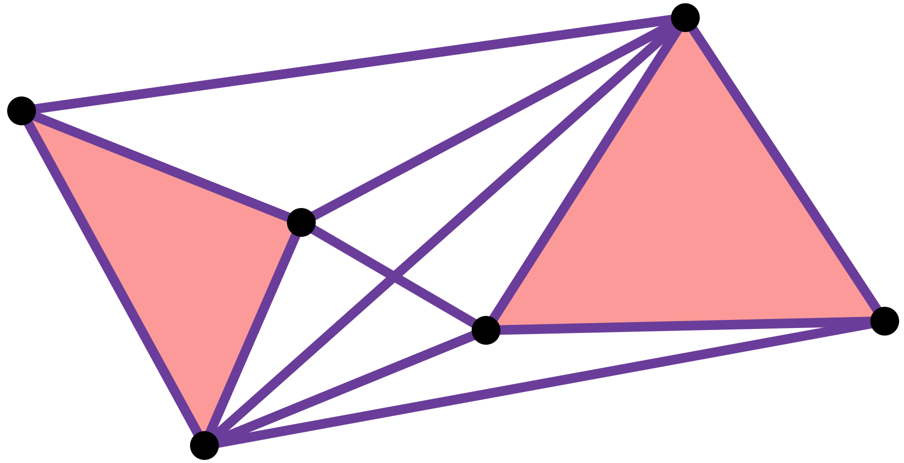
$\textsc{VisibilityGraph}(S)$

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

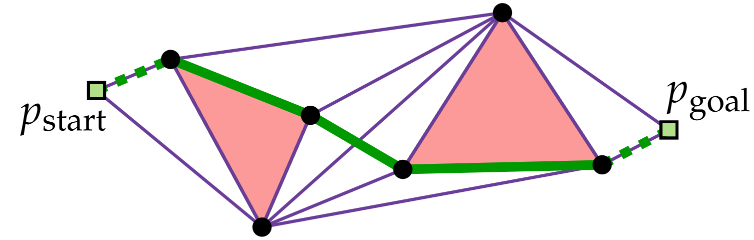
$E \leftarrow \emptyset$
foreach $v \in V(S)$ **do**
  $W = \textsc{VisibleVertices}(v, S)$
  $E \leftarrow E \cup \{vw \mid w \in W\}$
return $(V(S), E)$

$O(n) \cdot$

$O(n \log n)$



Algorithm



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

$G_{\text{vis}} \leftarrow \textsc{VisibilityGraph}(S \cup \{p_{\text{start}}, p_{\text{goal}}\})$?

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

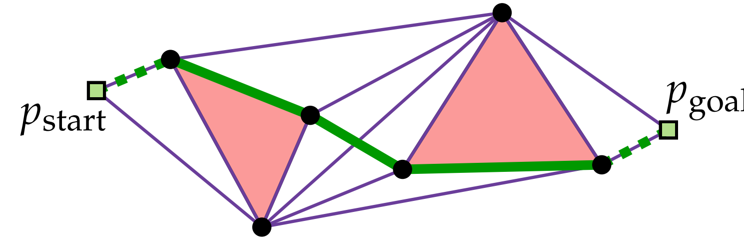
$w(uv) = d_{\text{Eucl.}}(u, v)$

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

return π

Running time?

Algorithm



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

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

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

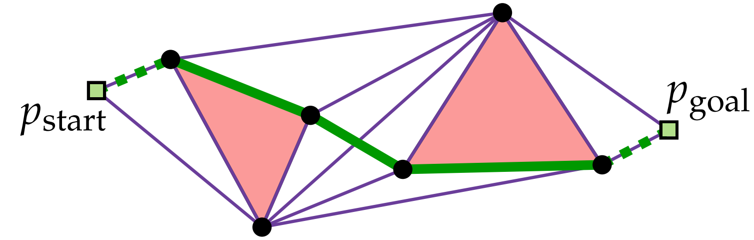
$\quad w(uv) = d_{\text{Eucl.}}(u, v)$

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

return π

Running time?

Algorithm



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

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

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

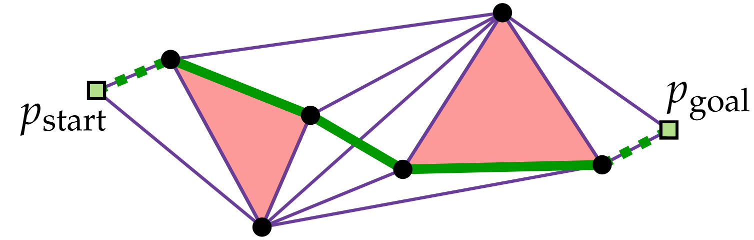
$w(uv) = d_{\text{Eucl.}}(u, v)$

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

return π

Running time?

Algorithm



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

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

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

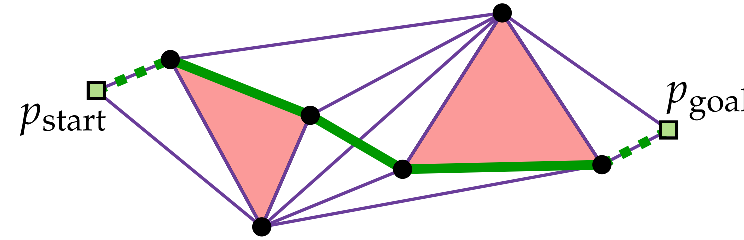
$\quad w(uv) = d_{\text{Eucl.}}(u, v)$

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

return π

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

Algorithm



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

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

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

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

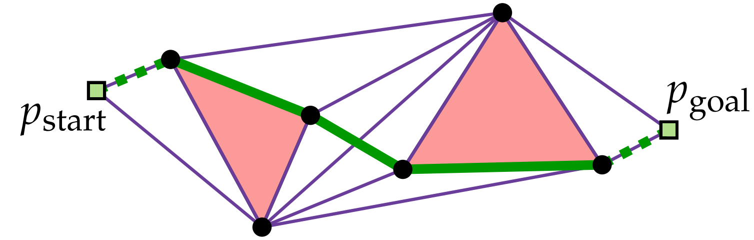
$\pi \leftarrow \textsc{Dijkstra}(G_{\text{vis}}, w, p_{\text{start}}, p_{\text{goal}})$ $O(m + n \log n)$

return π

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

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.

Algorithm



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

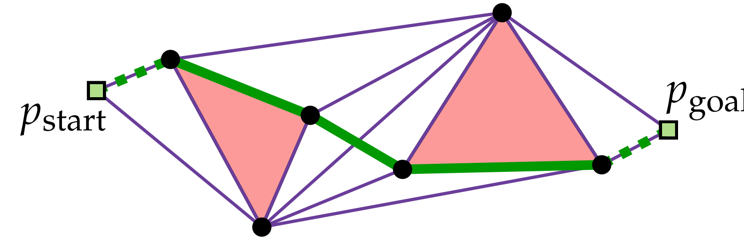
- $G_{\text{vis}} \leftarrow$ VisibilityGraph$(S \cup \{p_{\text{start}}, p_{\text{goal}}\})$ $\quad O(n^2 \log n)$
- **foreach** $uv \in E_{\text{vis}}$ **do** $\quad O(m)$
  - $w(uv) = d_{\text{Eucl.}}(u, v)$
- $\pi \leftarrow$ Dijkstra$(G_{\text{vis}}, w, p_{\text{start}}, p_{\text{goal}})$ $\quad O(m + n \log n)$
- **return** π

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

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.

Algorithm



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

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

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

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

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

return π

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

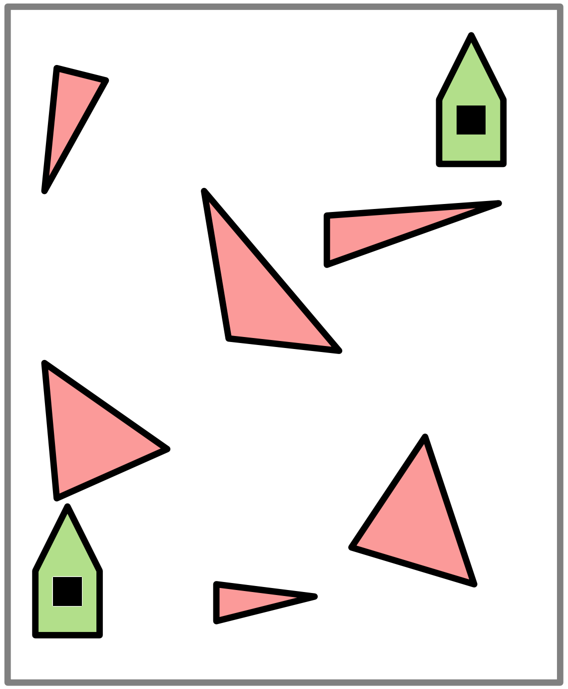
Theorem. The visibility graph of a set of disjoint polygonal obstacles with n edges in total can be computed in $O(n^2 \log n)$ time. [Ghosh & Mount]

($n \log n + m$)

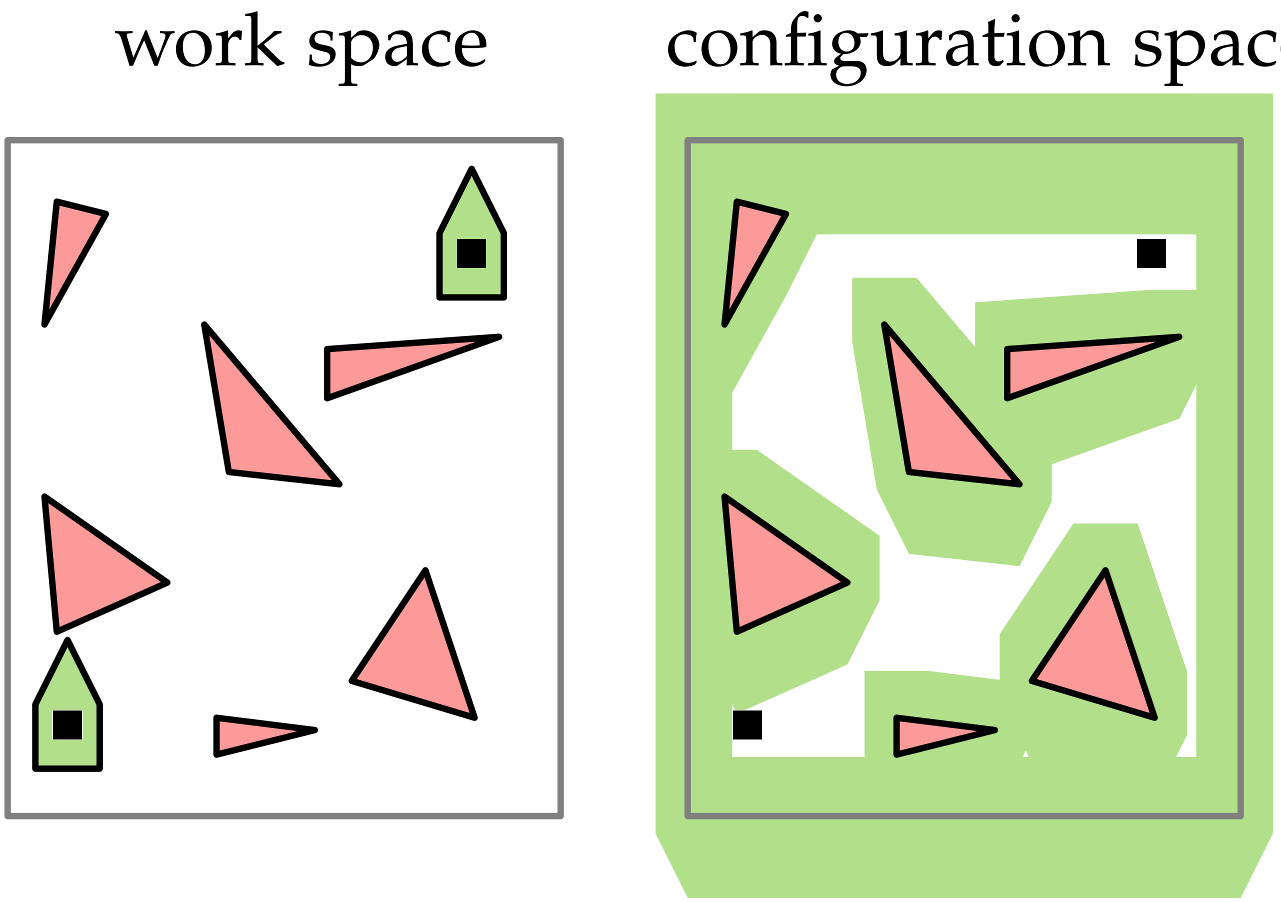
Theorem. A shortest path between two points among a set of [...] can be computed in $O(n \log n + m)$ time ~~with $O(n^2 \log n)$ preproc.~~

Translating Polygonal Robots

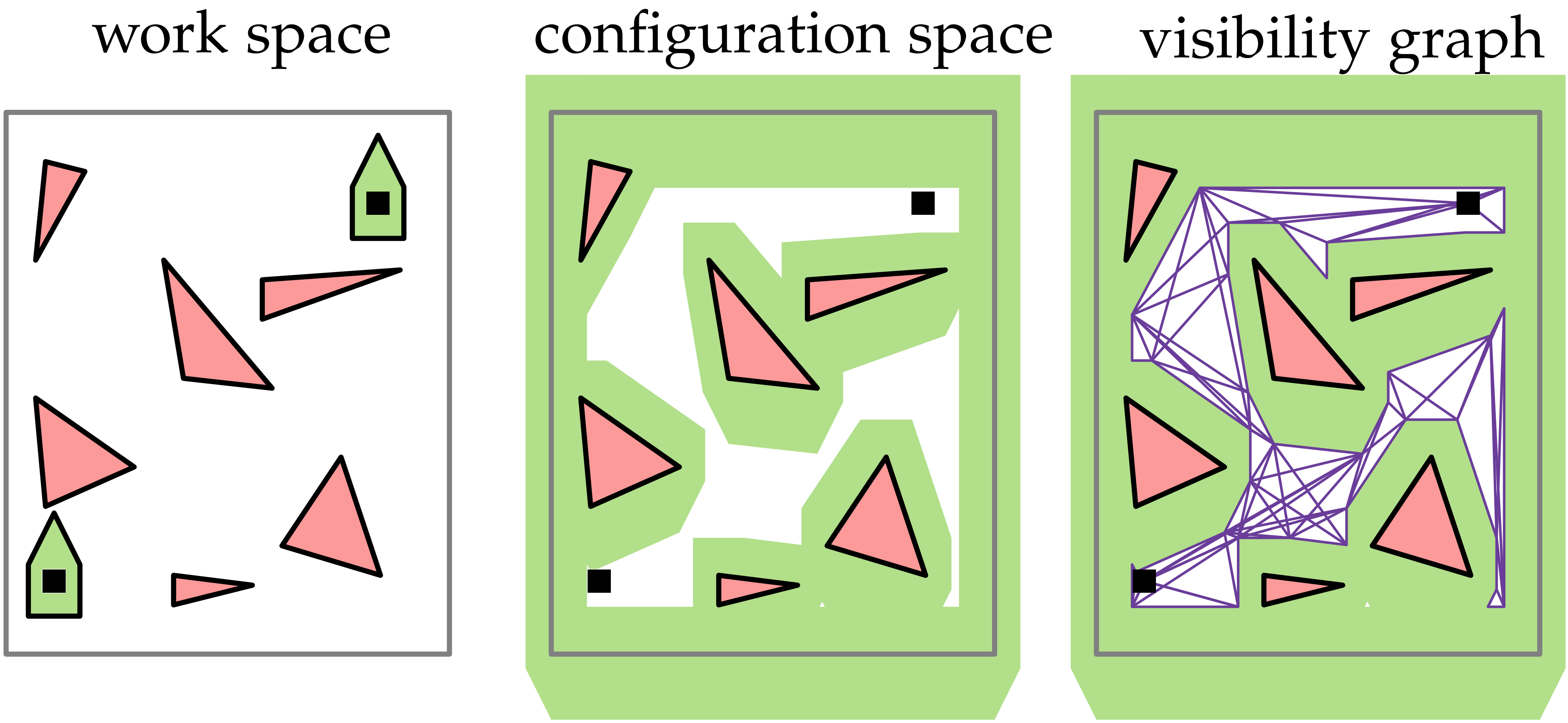
work space



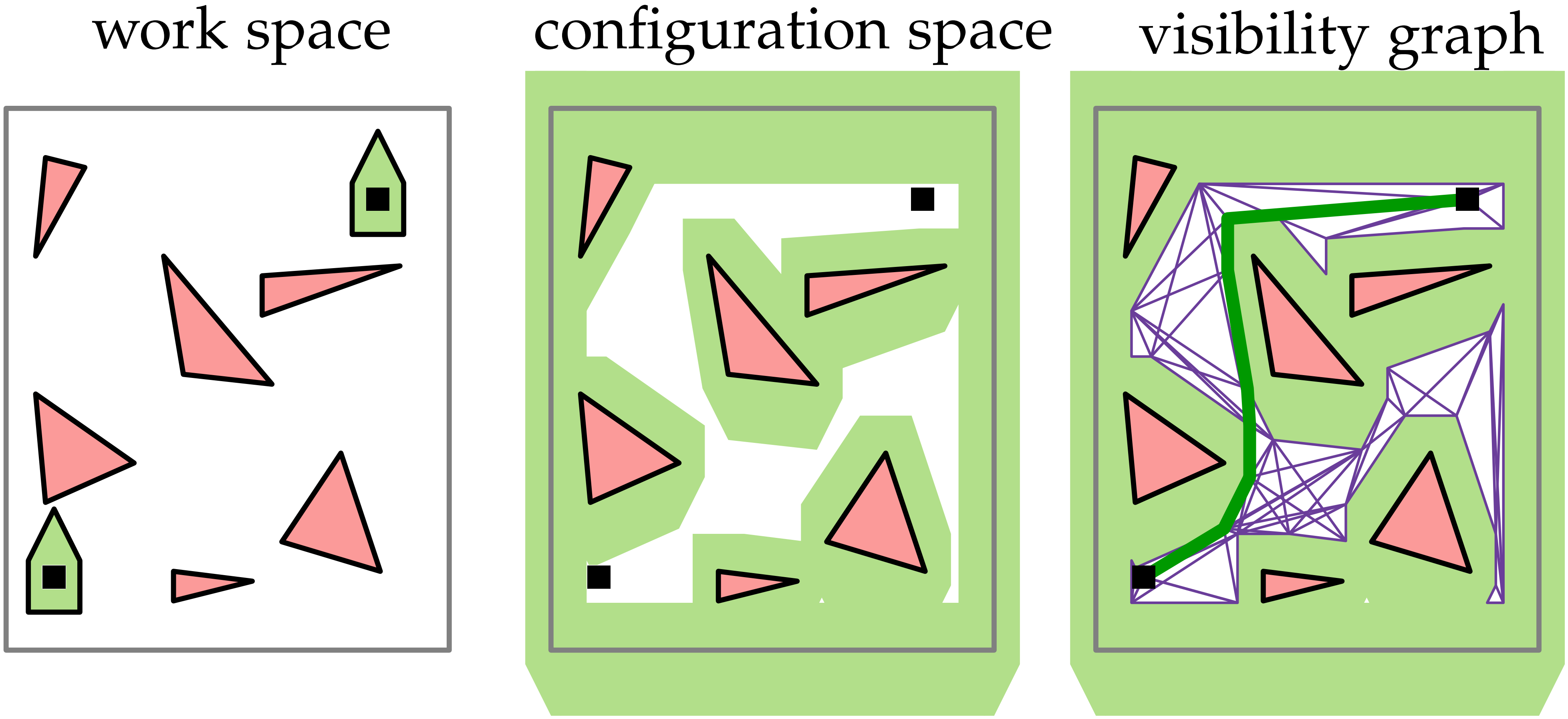
Translating Polygonal Robots



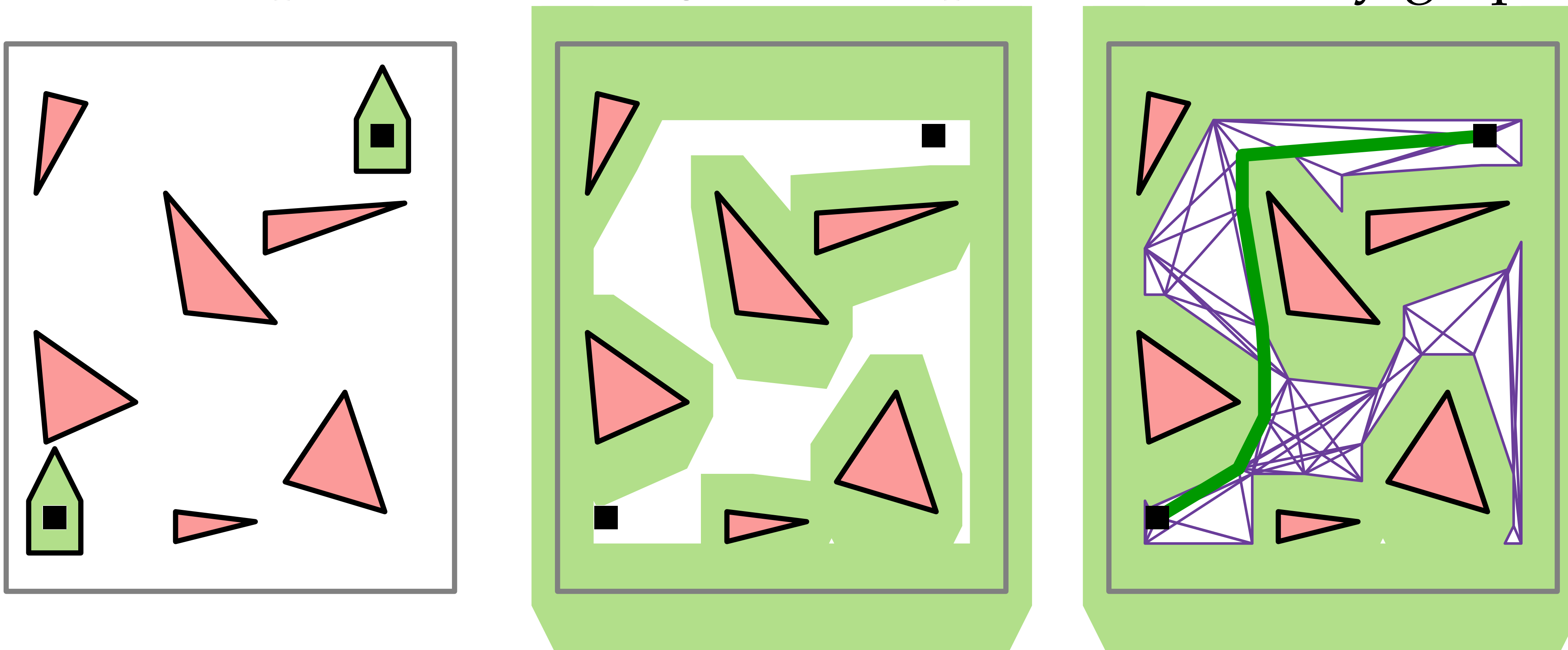
Translating Polygonal Robots



Translating Polygonal Robots

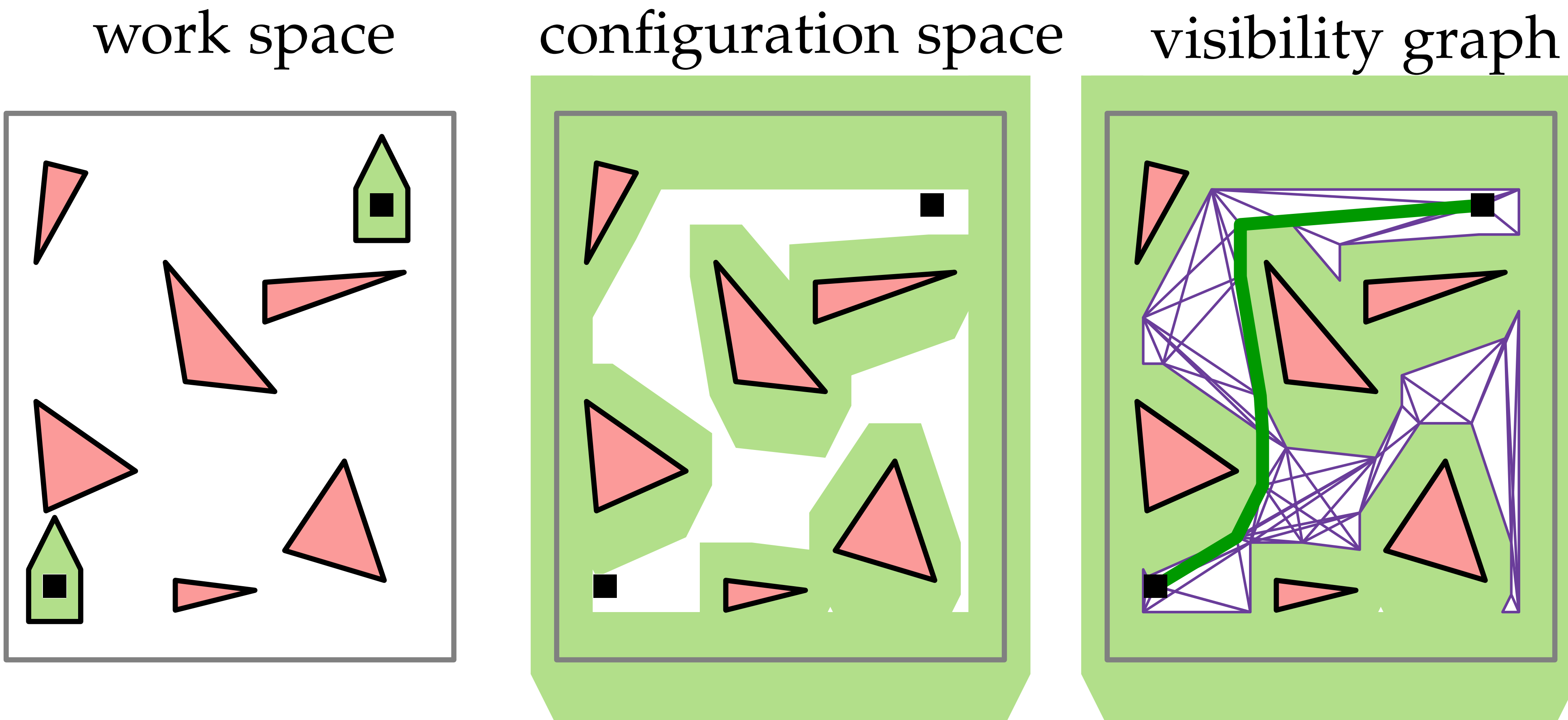


Translating Polygonal Robots



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.

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 $O(n^{\cancel{2}} \log n)$ time.

[Hershberger & Suri]