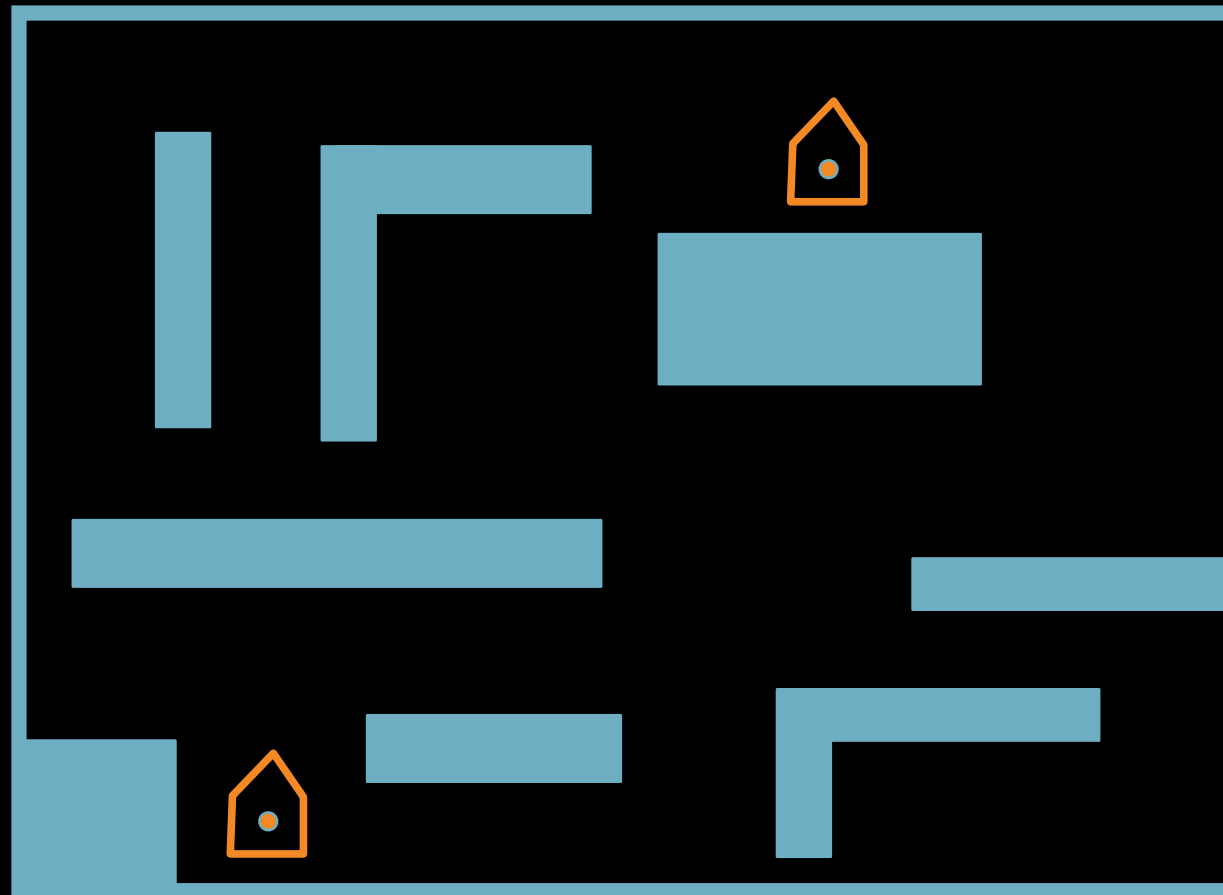


# **Movimento de Robos**

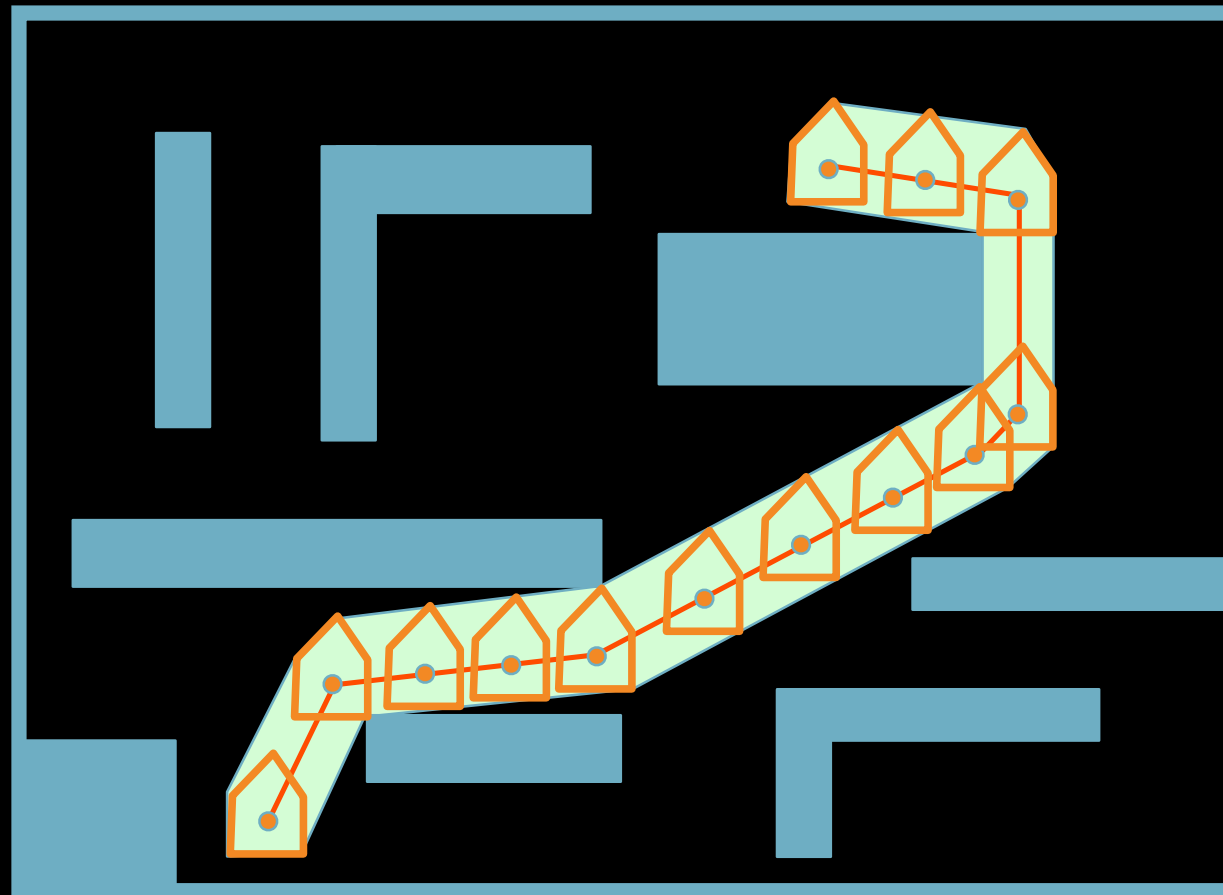
## **Grafos de Visibilidade**

**João Comba**

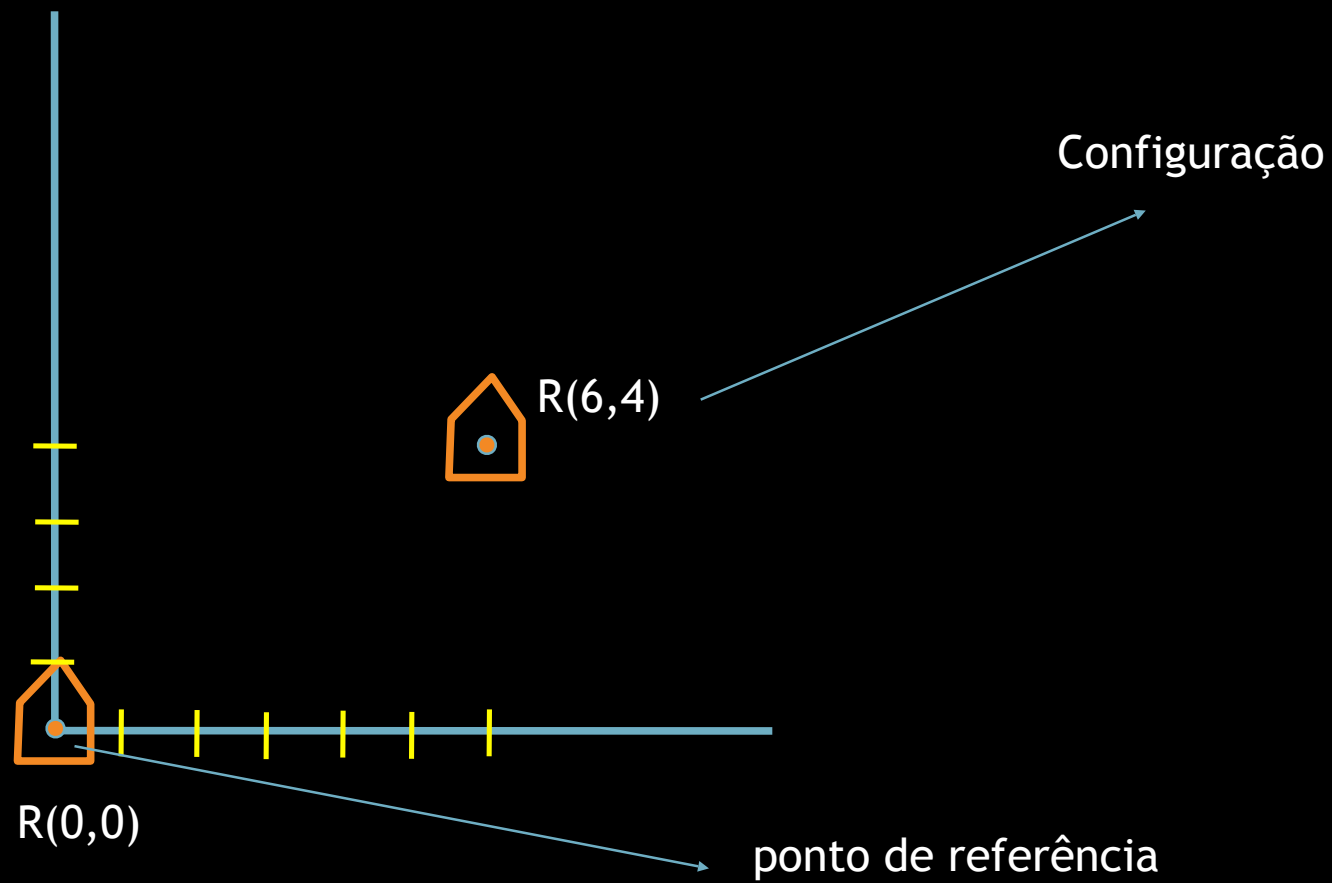
# Problema



# Problema

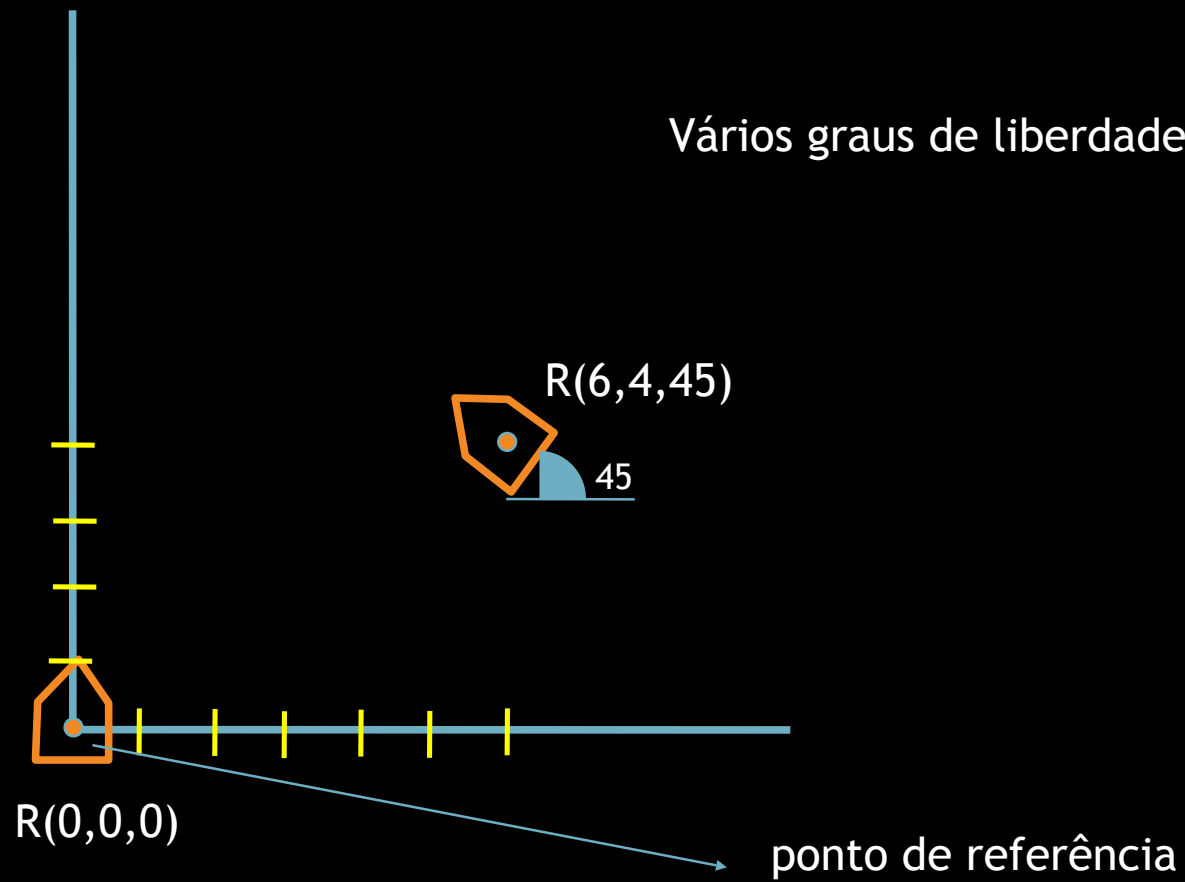


# Espaço de trabalho



# Espaço de trabalho

Vários graus de liberdade (DOF)



# Espaço de Configuração

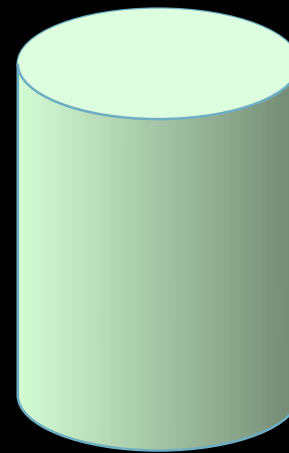
Espaço dos parâmetros de um robô R

$$(x, y, \Phi) \rightarrow R(x, y, \Phi)$$

# Espaço de Configuração

Espaço dos parâmetros de um robô R

$$(x, y, \Phi) \rightarrow R(x, y, \Phi)$$



$$\mathbb{R}^2 \times [0:360)$$

## **Espaço de Configuração Proibido**

Espaço dos parâmetros de um robô  $R$  onde o robô colide com o ambiente

## **Espaço de Configuração Livre**

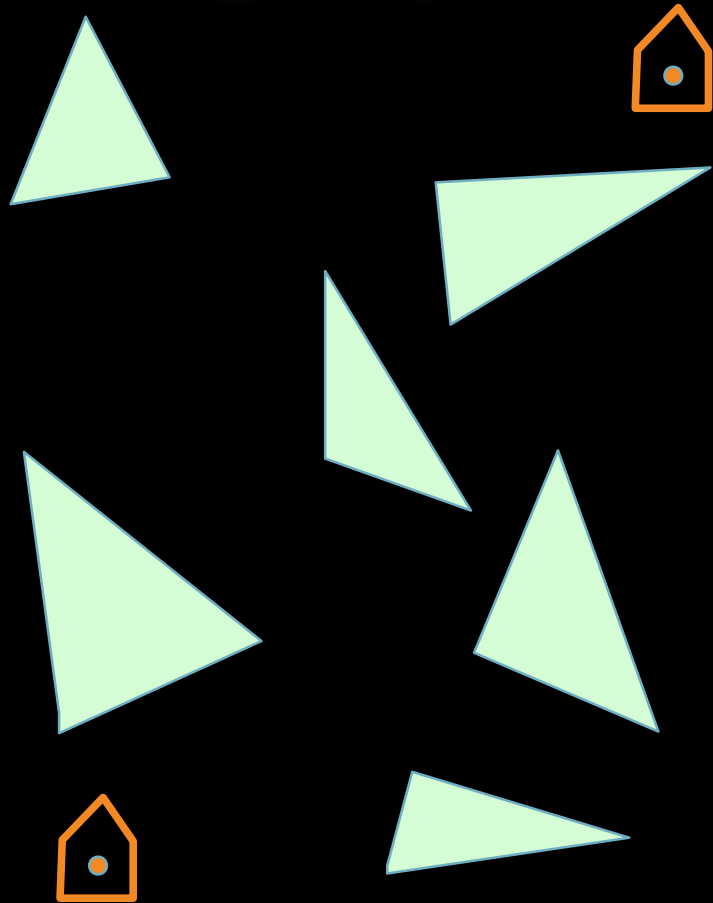
Espaço dos parâmetros de um robô  $R$  onde o robô não colide com o ambiente

## **Espaço de Configuração Obstáculos**

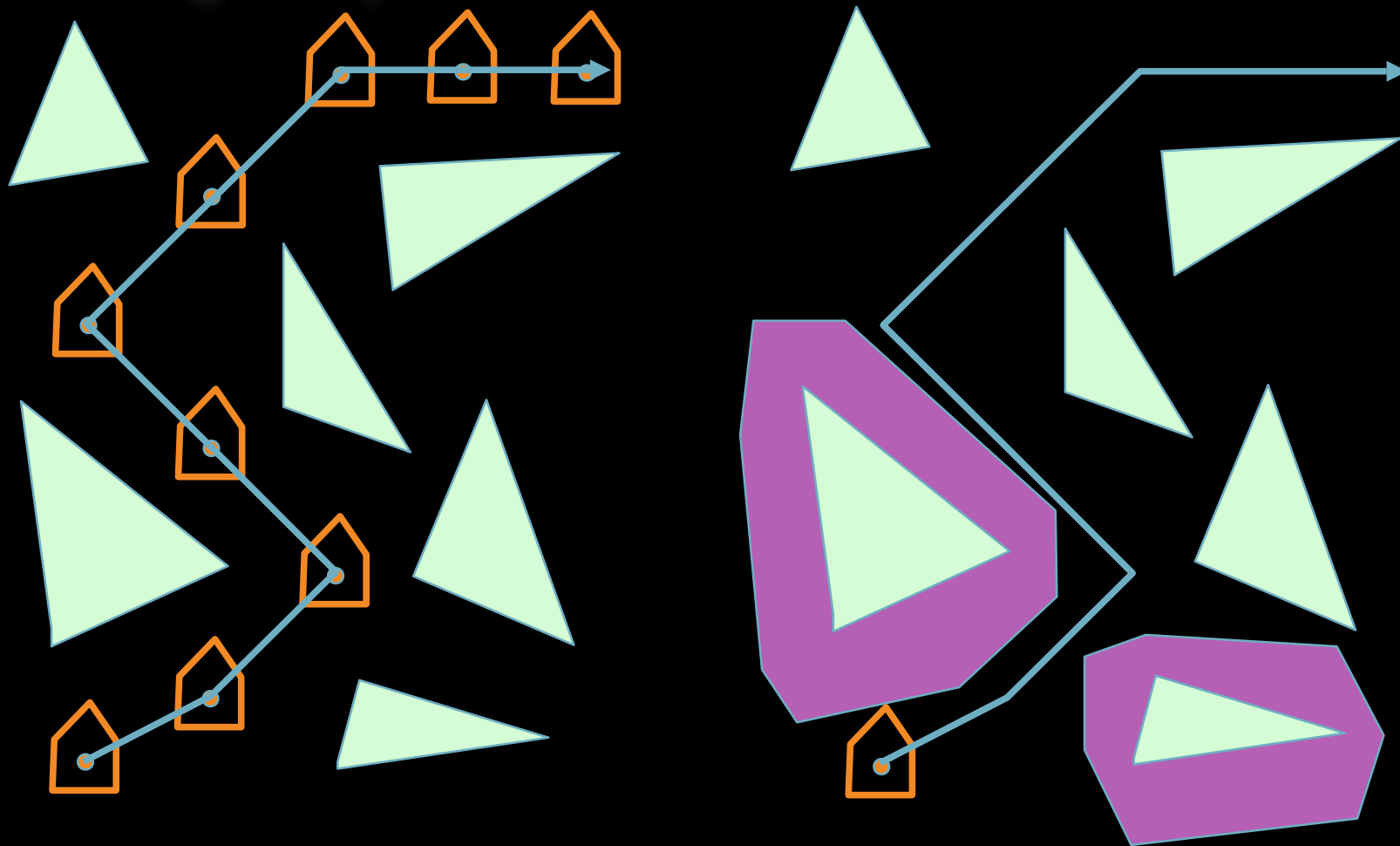
Espaço dos parâmetros dos obstáculos mapeados para o espaço de configuração



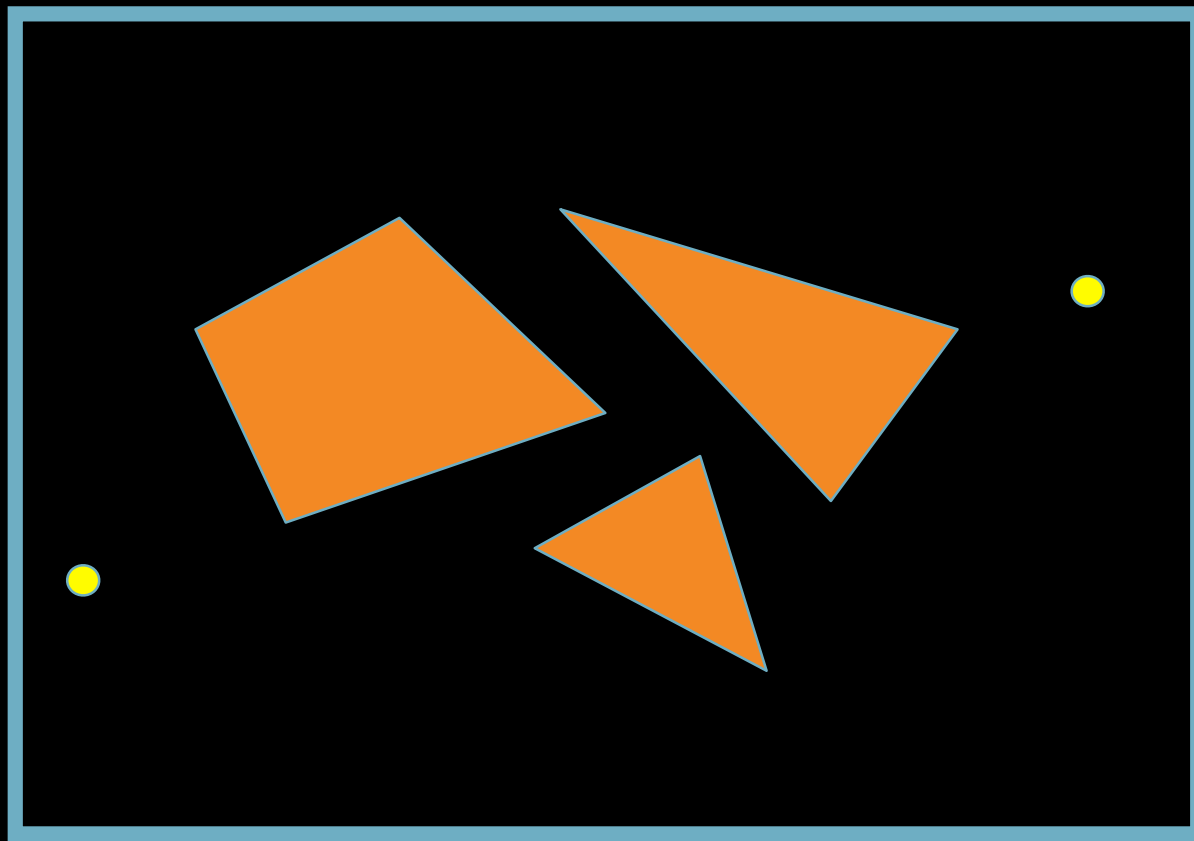
# Espaço de Trabalho e Configuração



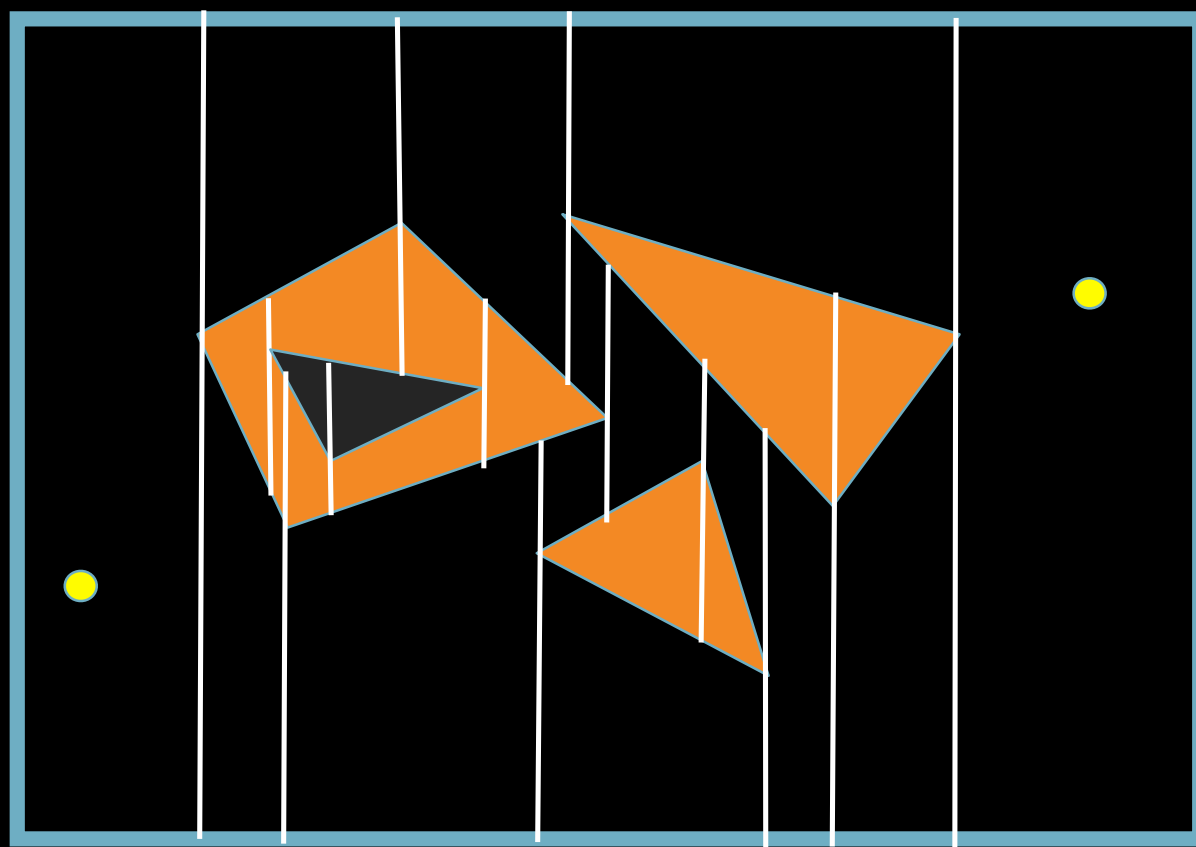
# Espaço de Trabalho e Configuração



## Problema Mais Simples (Robos Pontuais)

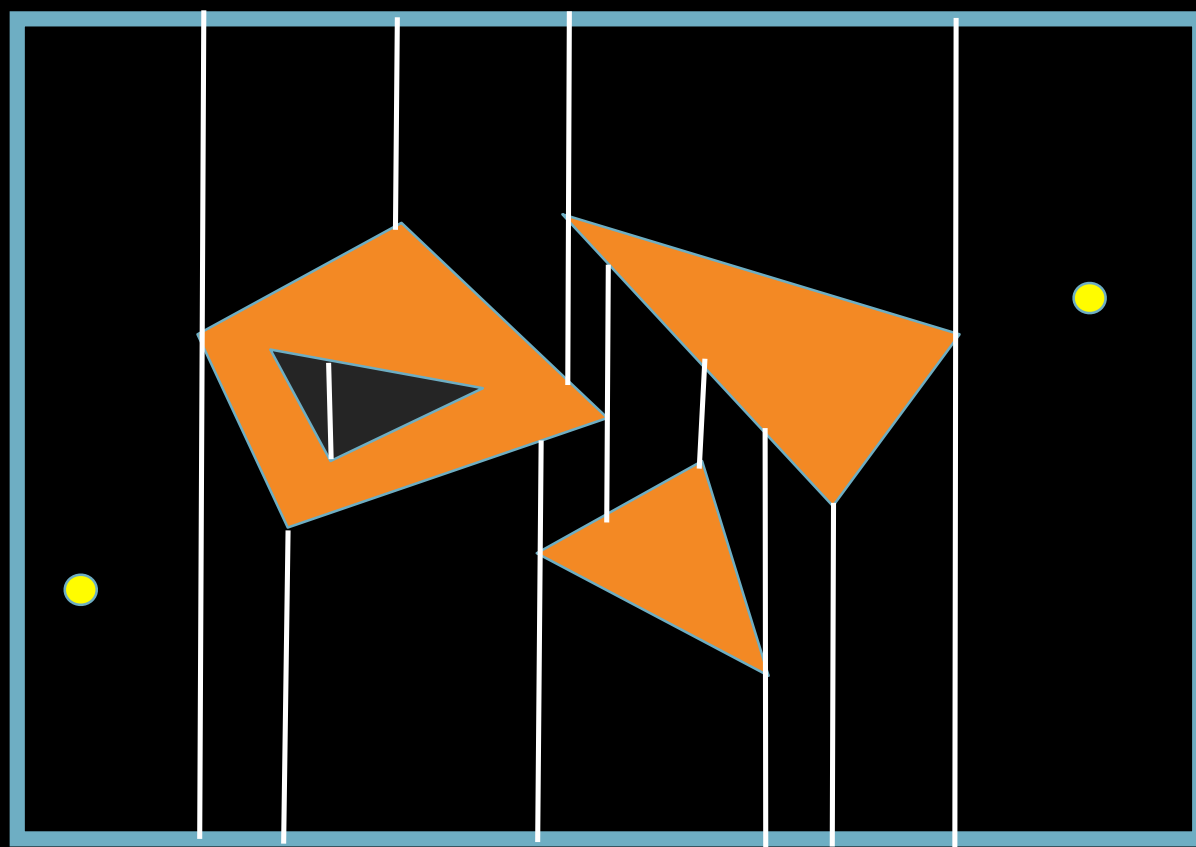


## Problema Mais Simples (Robos Pontuais)



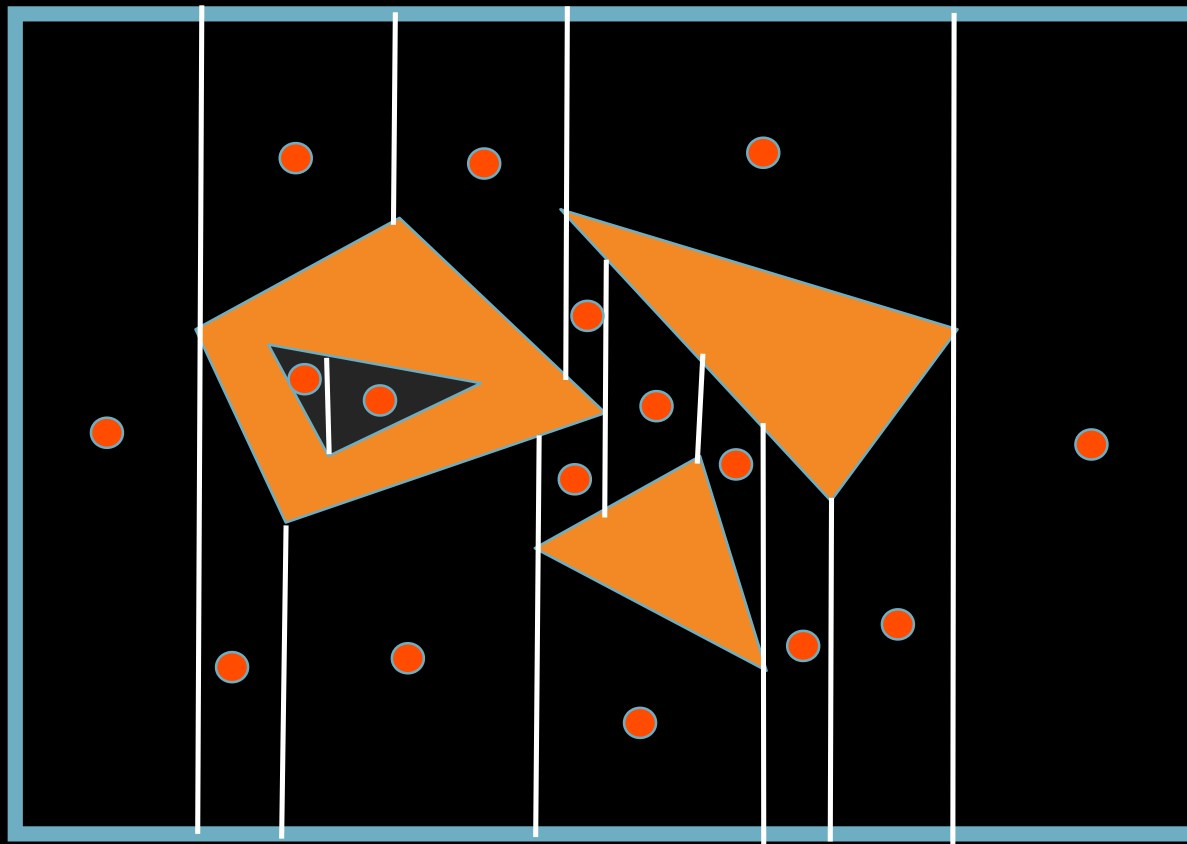
Decomposicao Trapezoidal

## Problema Mais Simples (Robos Pontuais)



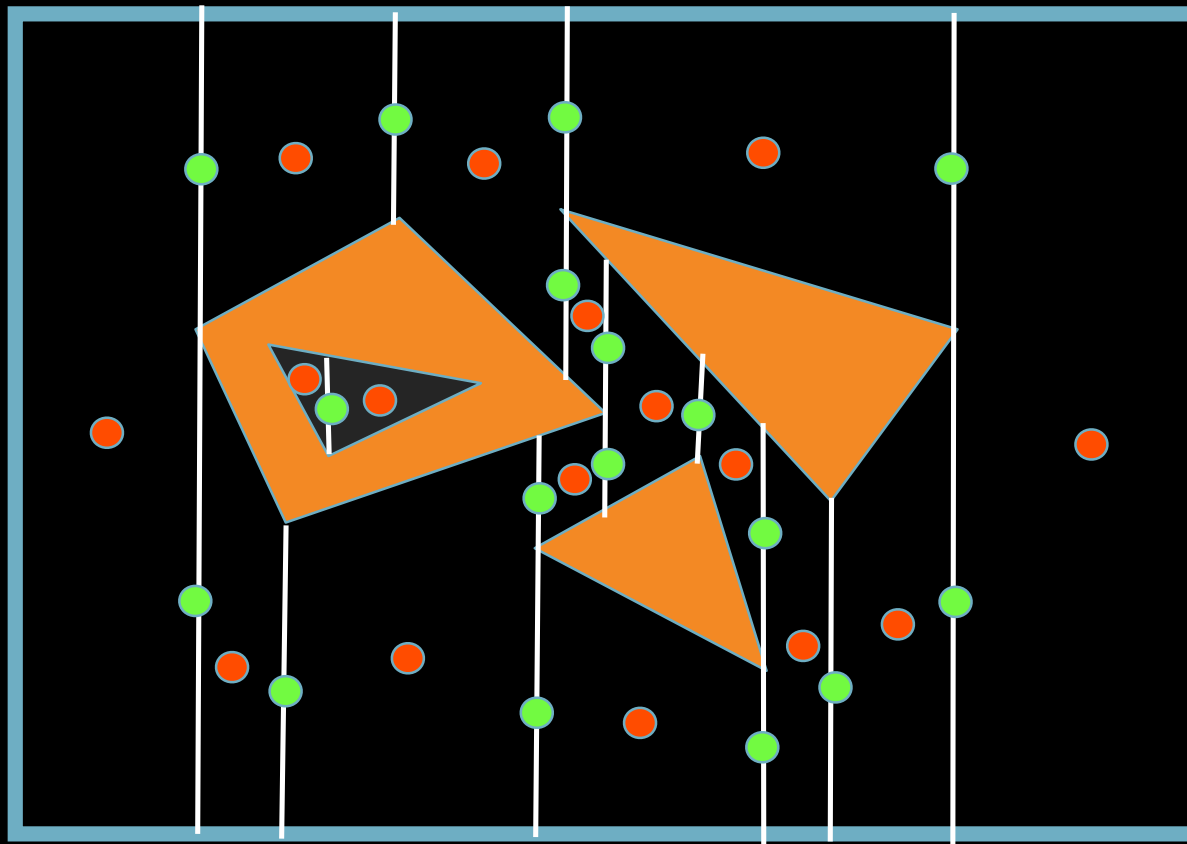
Remover Paredes Internas

Criar um mapa de ruas



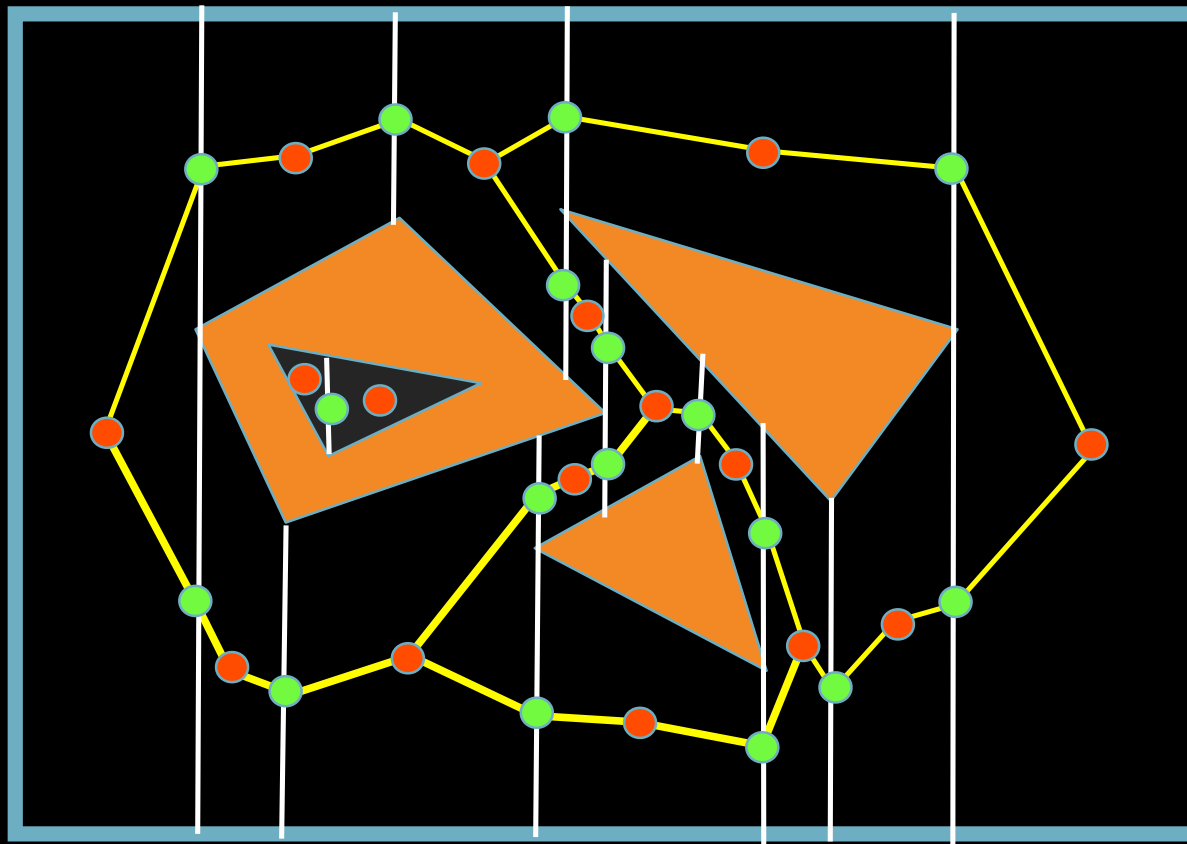
Adicionar o centro de cada trapezoide

Criar um mapa de ruas



Adicionar vertices sobre as extensoes verticais

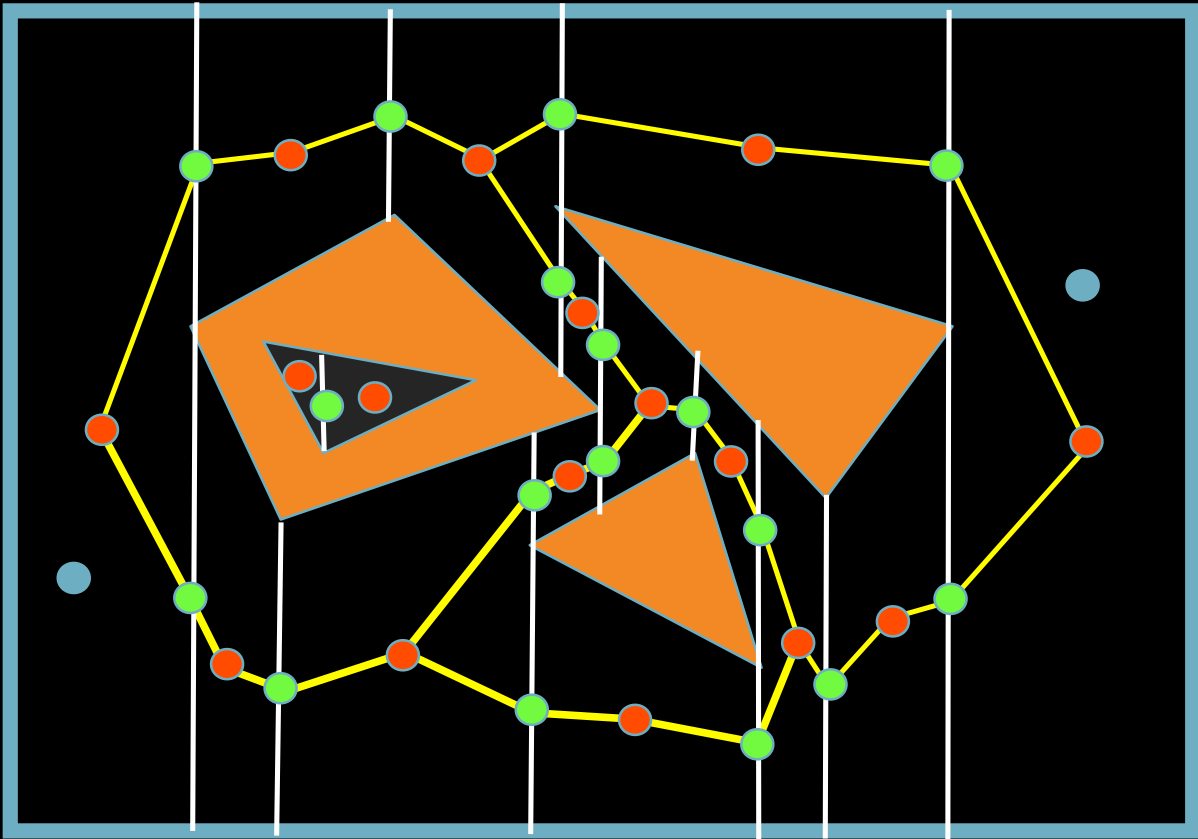
Criar um mapa de ruas



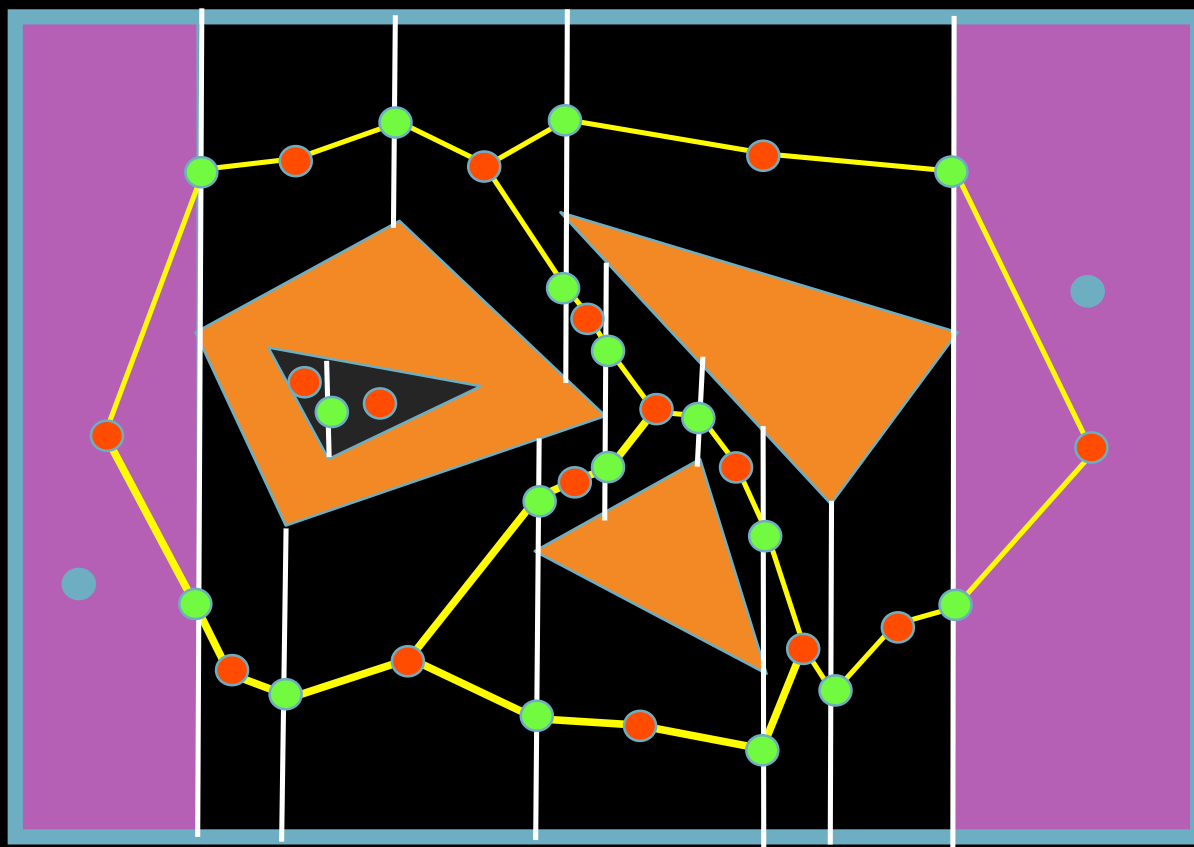
Criar arestas conectando centros-fronteira



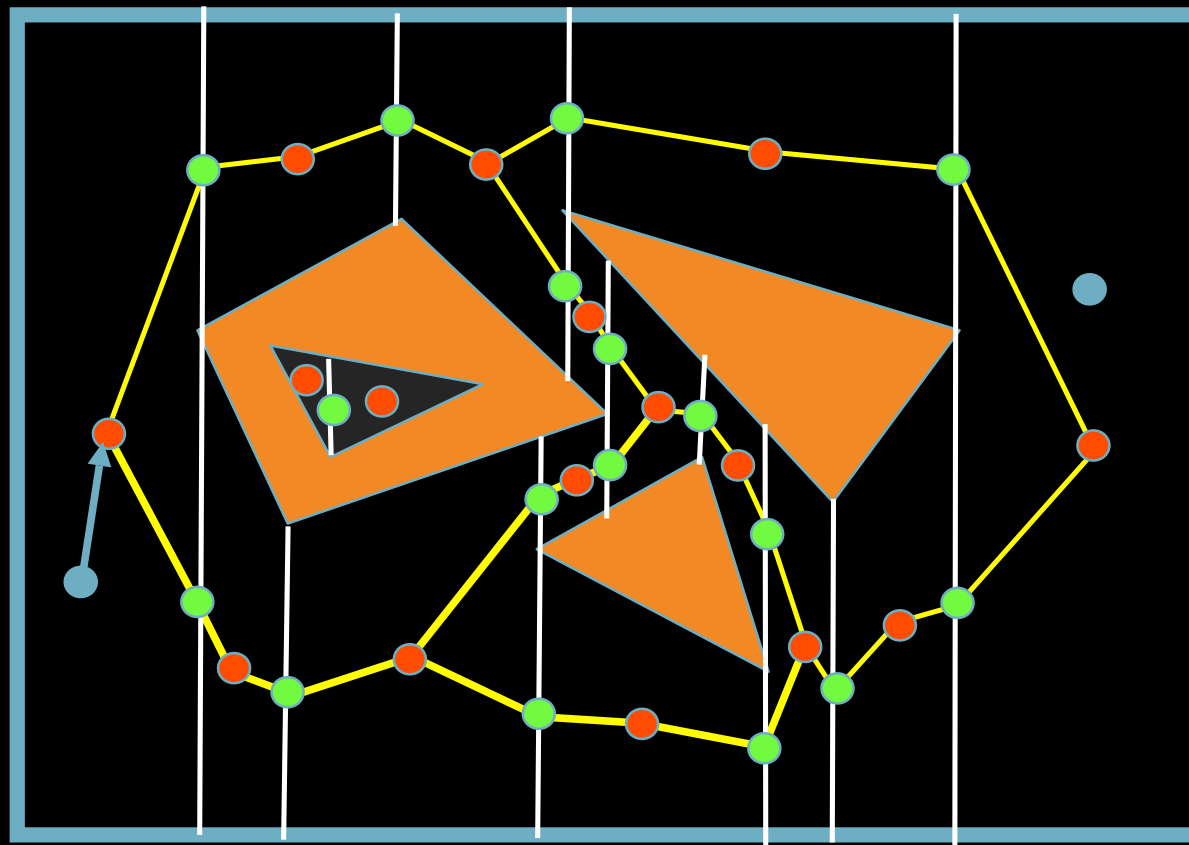
# Restringir o movimento do robo ao mapa de ruas



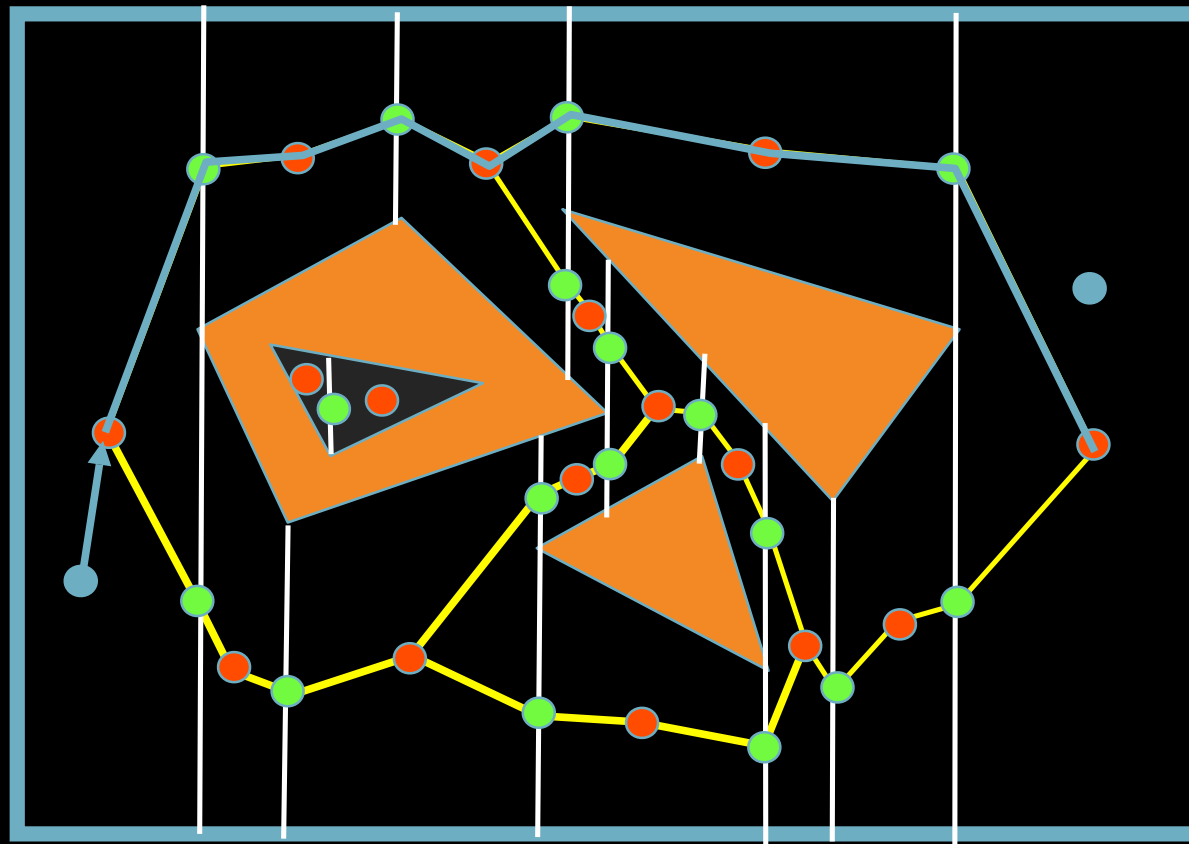
Achar as celulas contendo a posicao inicial e final



Mover o robo para o centro da celula

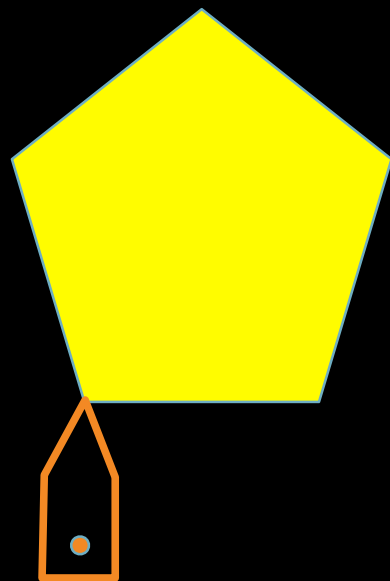


Fazer uma busca em largura até encontrar o vertice contendo a posicao final

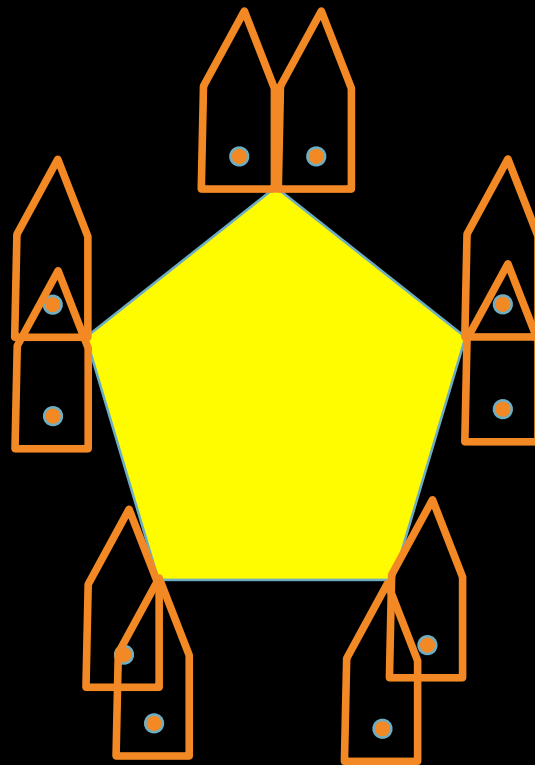




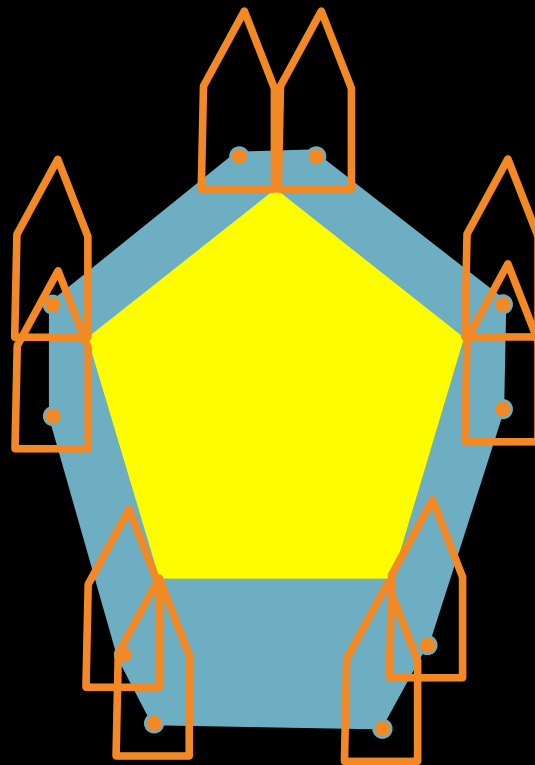
# Robôs Poligonais/Translacionais



# Robots Poligonais/Translacionais

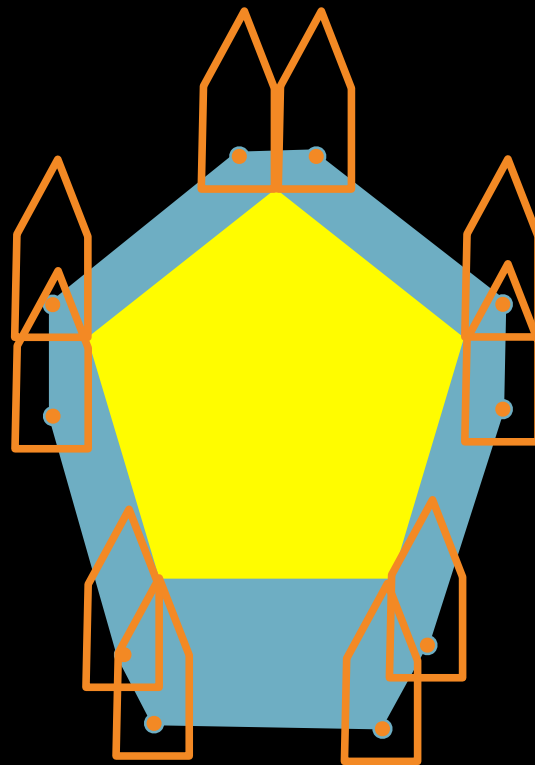


# Robots Poligonais/Translacionais

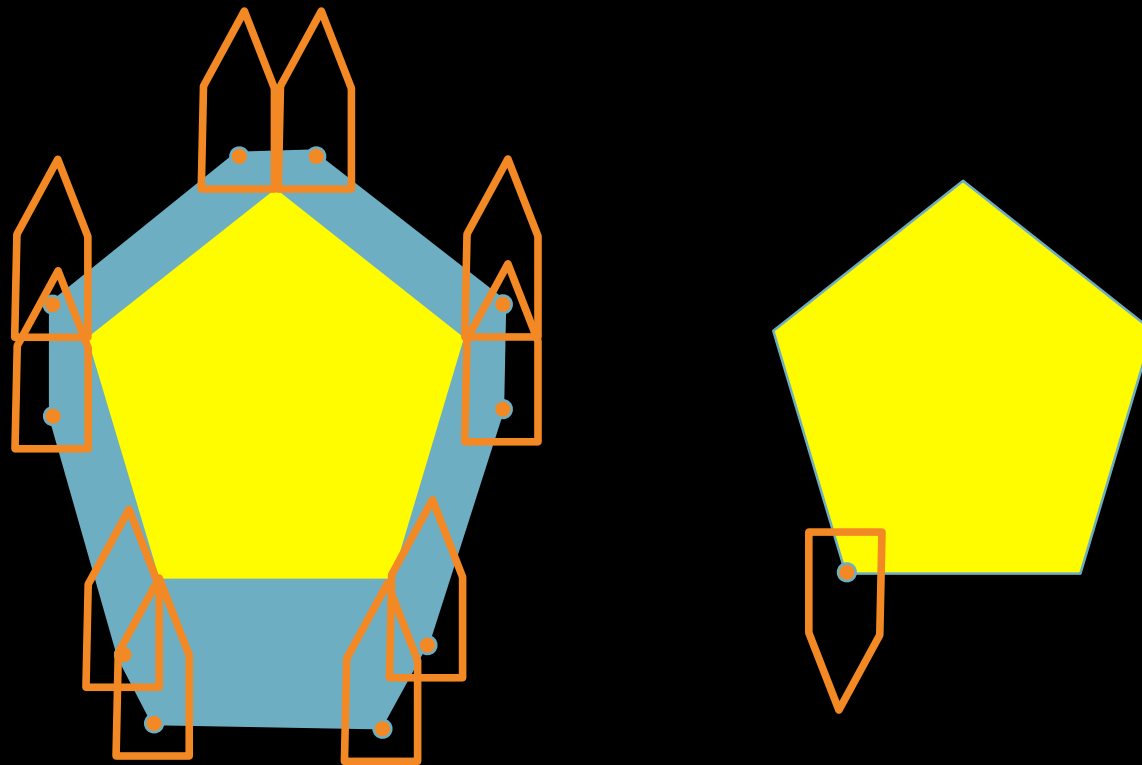




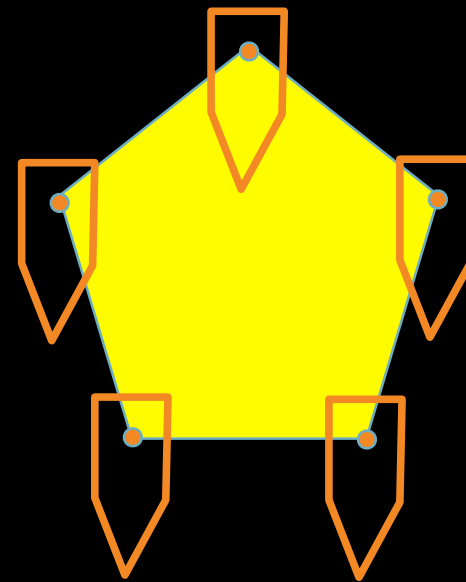
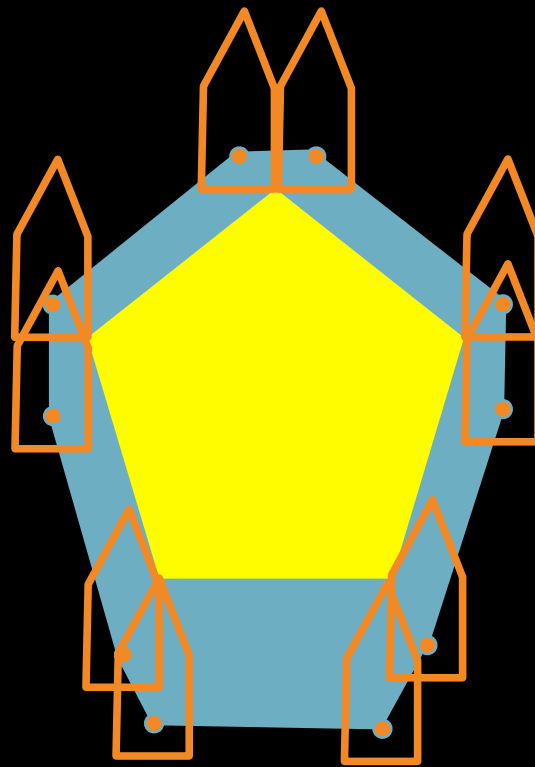
# Soma de Minkowski



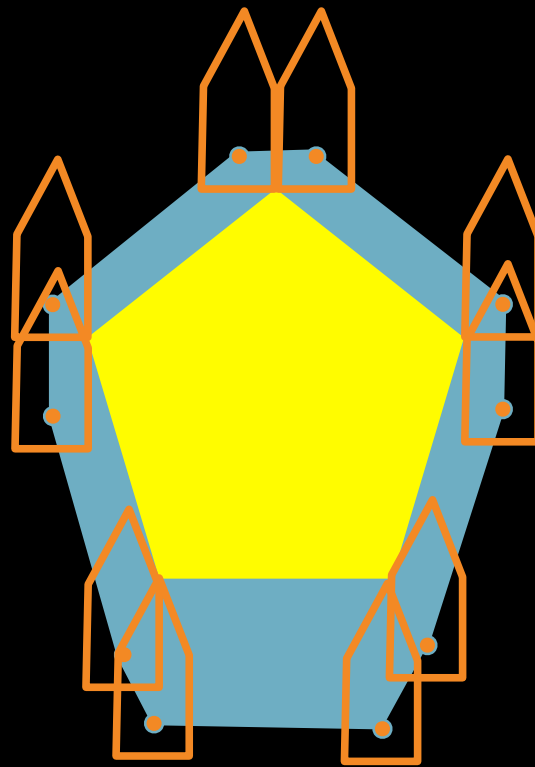
# Soma de Minkowski



# Soma de Minkowski

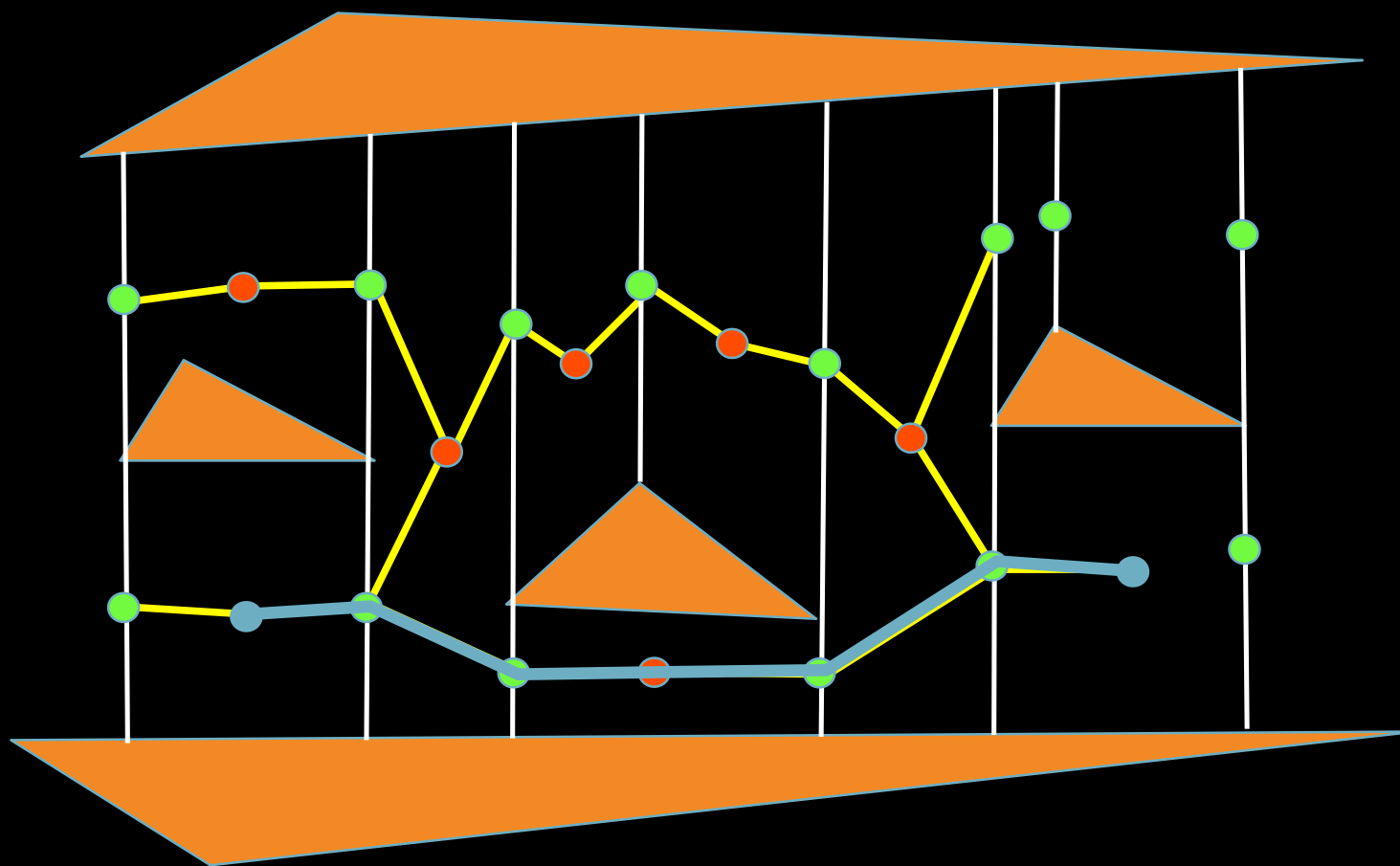


# Soma de Minkowski



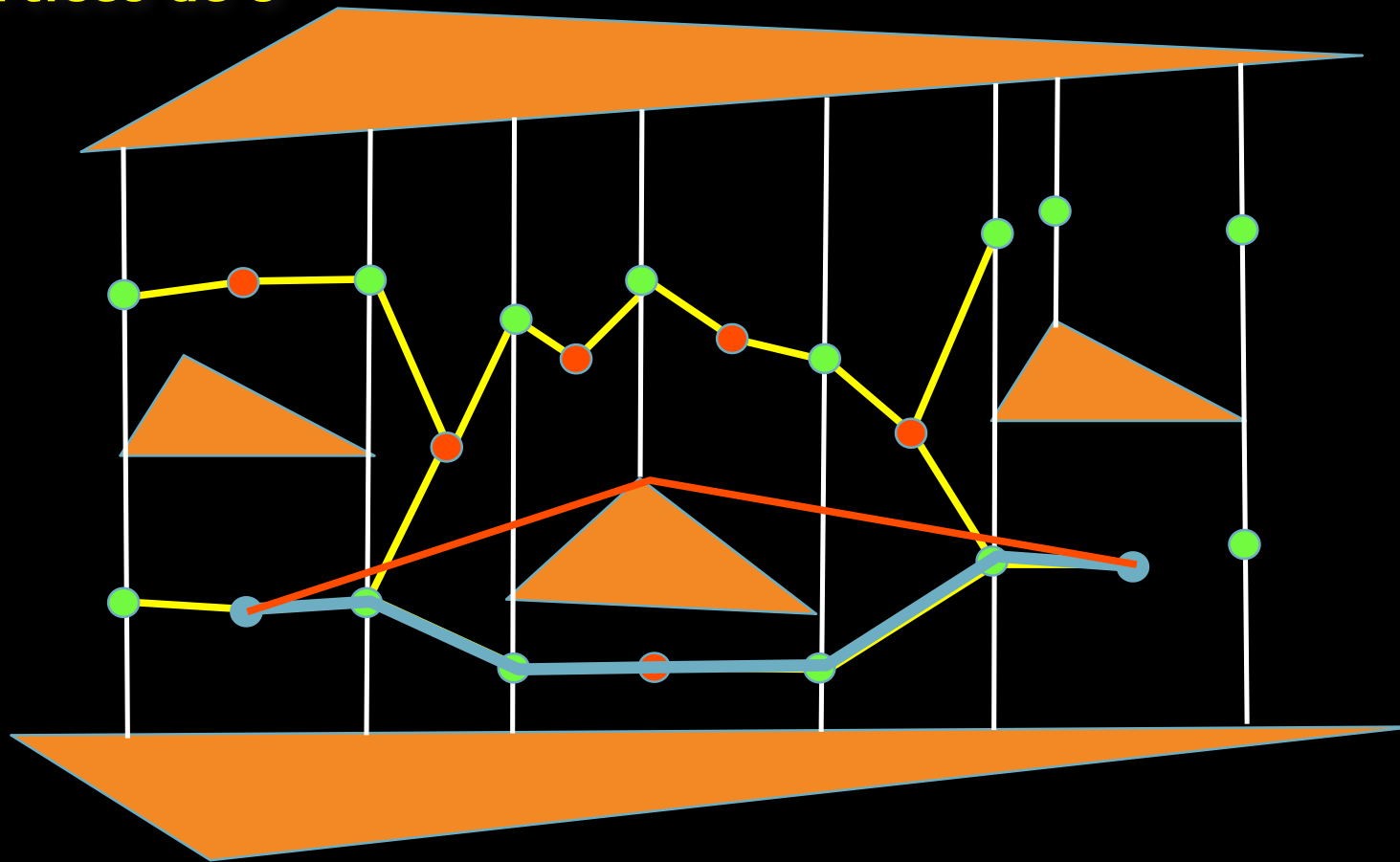


# Caminho mais curto ?



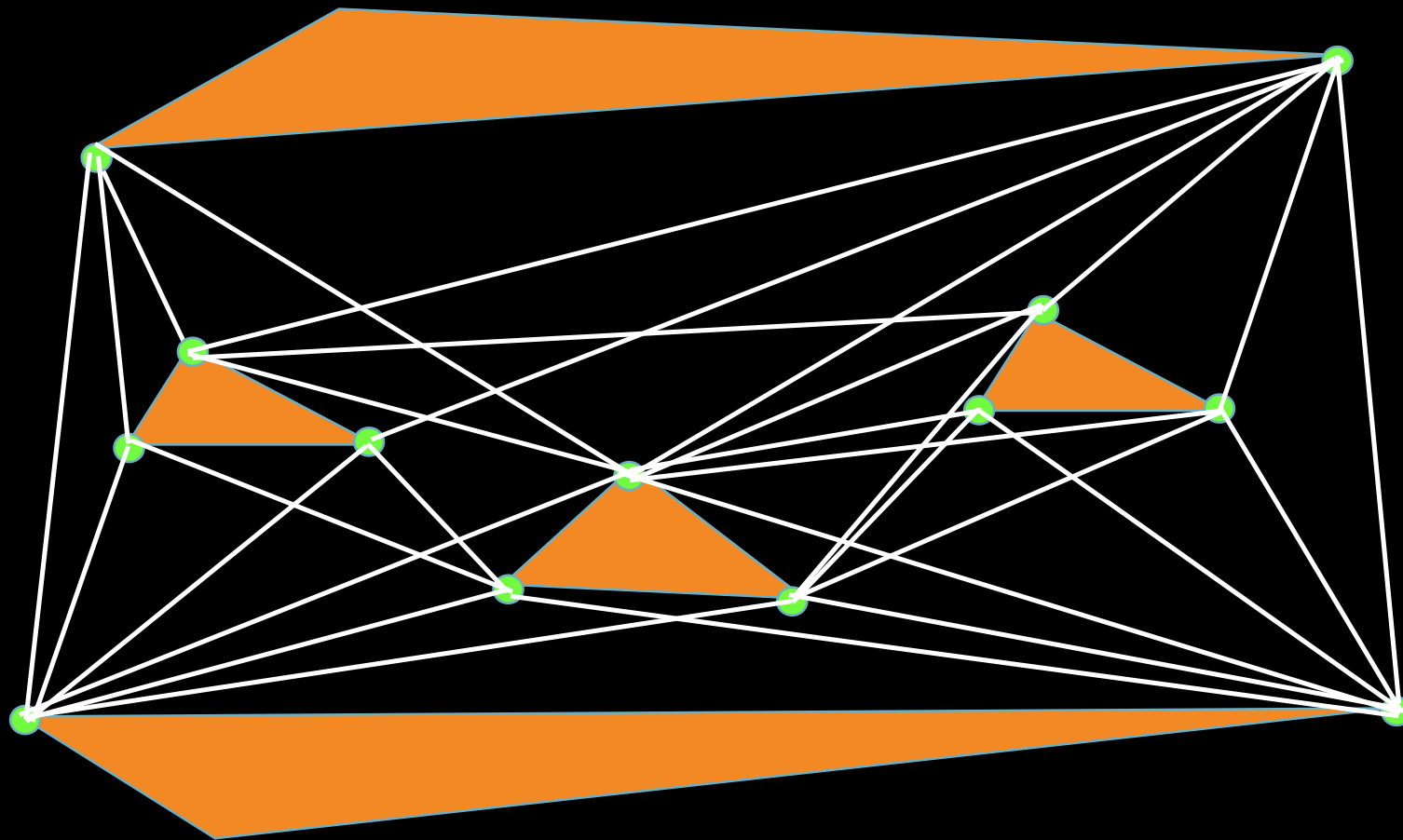


**Lema:** Qualquer caminho mais curto entre  $p_1$  e  $p_2$  entre um conjunto de obstaculos poligonais  $S$  e' um caminho poligonal cujos vertices internos sao vertices de  $S$

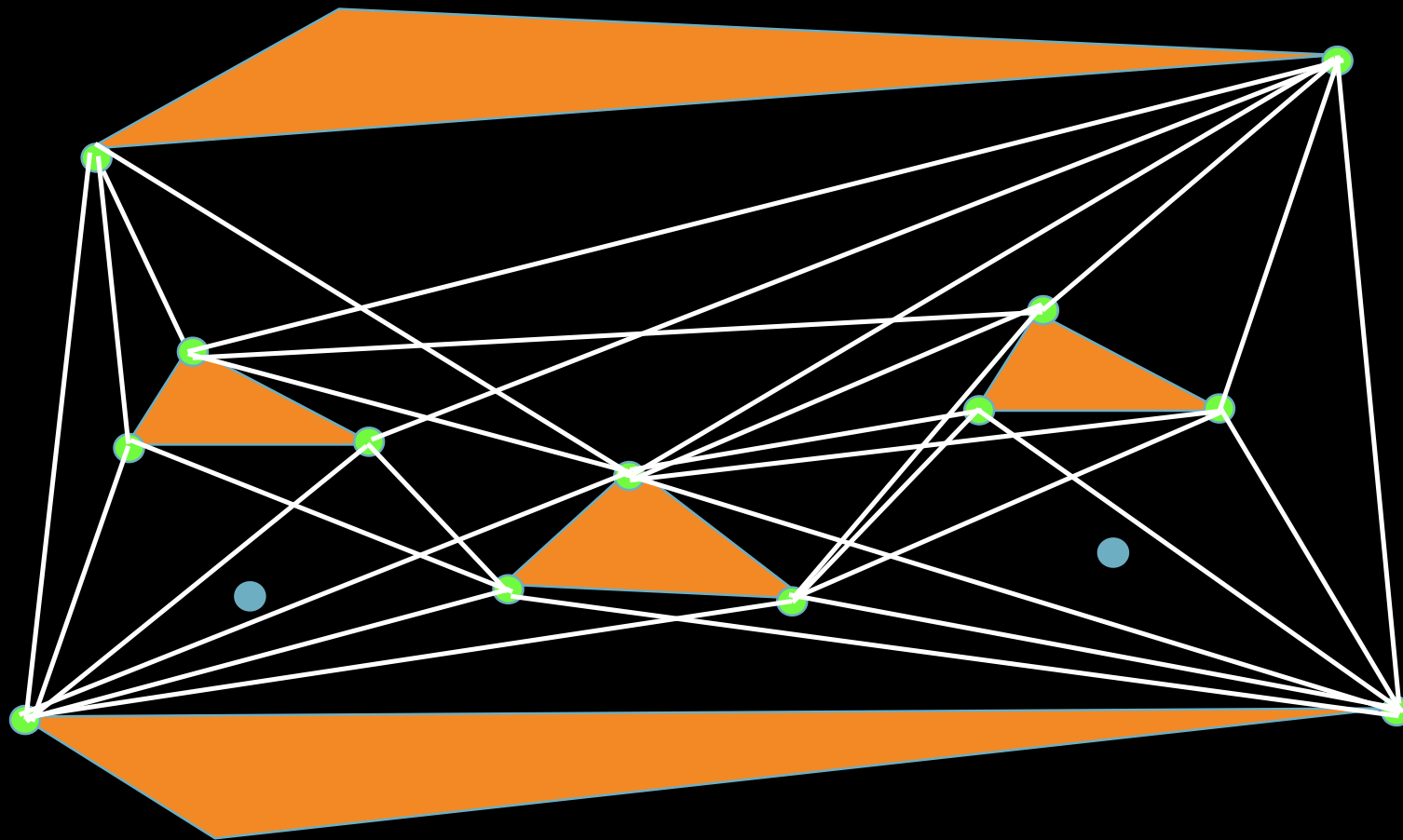




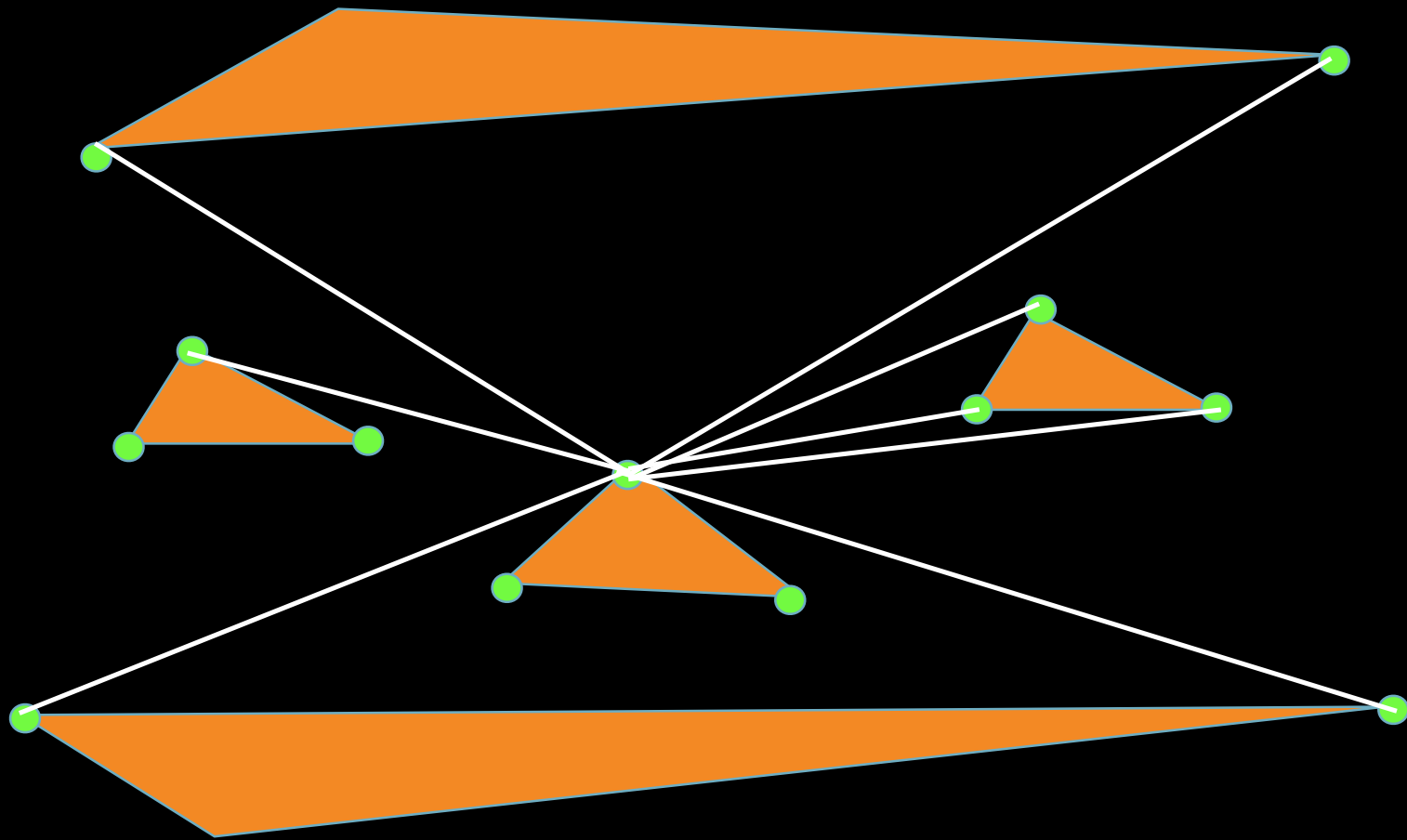
# Grafo de Visibilidade



Usar algoritmo de Dijkstra para  
caminhos mais curtos em grafo



# Calculo do Grafo de Visibilidade



## Lee: rotational plane sweep

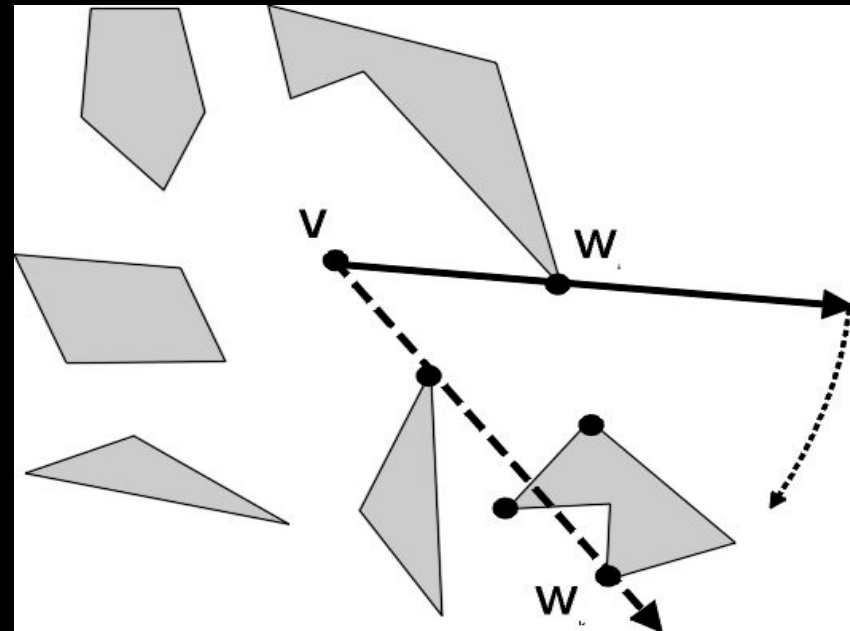
Construct a visibility graph, i.e. a road map based on visibility edges

To do this we perform a rotational plane sweep -- much like a weather radar sweep -- around every vertex in  $S^*$  (obstacle edges and  $p, q$ )

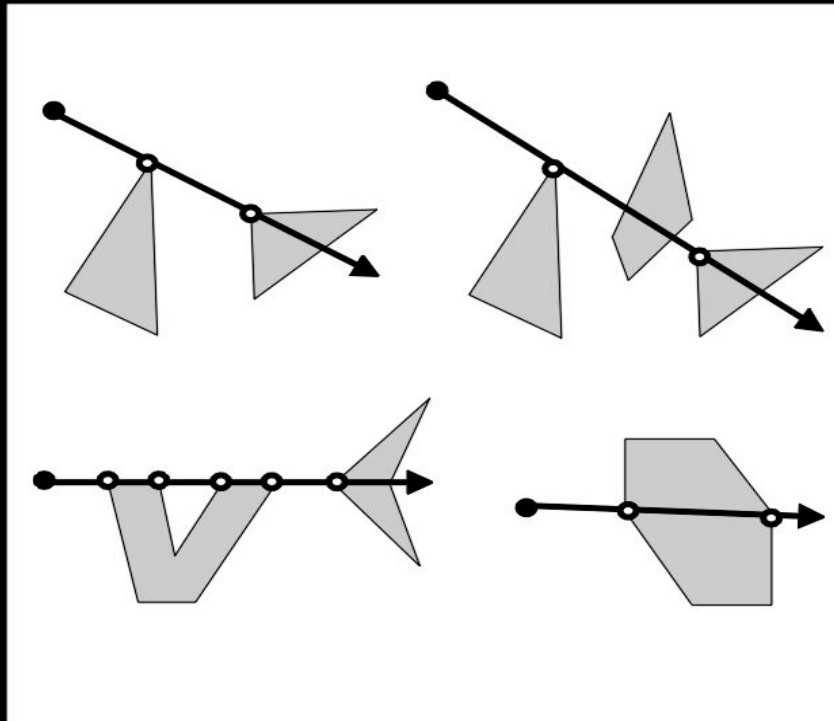
## Lee: rotational plane sweep

In the plane sweep, we consider a vertex  $w$  visible from  $v$  if a ray cast from  $v$  in the direction of  $w$  doesn't intersect the interior of an obstacle before reaching  $w$

- Ray emanating from  $v$  rotating in plane



## Lee: rotational plane sweep



Finding the visible vertices  
for each of  $n$  vertices  
takes  $O(n^2 \log n)$

Running Dijkstra's SP takes  $O(n \log n + k)$ , which is less  
than computing the vertex  
visibility

◀ **Four cases of sweep ray  
intersecting multiple  
vertices**