



**FUNDAÇÃO EDSON QUEIROZ**  
**UNIVERSIDADE DE FORTALEZA**  
ENSINANDO E APRENDENDO

# Aula 11

## Planejamento de Trajetória

**Disciplina: Robótica**

Prof. MSc. Ítalo Jáder Loiola Batista  
**Universidade de Fortaleza - UNIFOR**  
**Centro de Ciências Tecnológicas - CCT**  
*E-mail: [italoloiola@unifor.br](mailto:italoloiola@unifor.br)*

# Introdução

- ❑ Capacidade de **planejar seus próprios movimentos**, sem a necessidade de interferência direta de seres humanos.
- ❑ Atualmente, uma das **áreas mais importantes** da robótica móvel.
- ❑ A elaboração de um plano de movimentação é uma **tarefa extremamente complexa**.
- ❑ Salvo quando o **ambiente de atuação** do robô é limitado e cuidadosamente controlado.

# O Problema Básico

- ❑ O **problema básico** do planejamento de trajetória pode ser encarado como uma **simplificação**.
  - Robô rígido  $A$ , modelado como um ponto;
  - Ambiente  $W$  é estático e conhecido;
  - O domínio do robô é um espaço Euclidiano:  $R^2$  ou  $R^3$ ;
  - Conjunto de obstáculos conhecidos  $B_1, B_2, \dots, B_q$  em  $W$ ;
  - O robô se movimenta em segmentos de reta perfeitos;

# O Problema Básico

- ❑ O objetivo é transformar o problema do planejamento de um problema “físico” para um problema “puramente geométrico”.
- ❑ Assume-se que o robô é um único objeto rígido:
  - ❑ Ou seja, o robô não apresenta partes móveis, como braços, pernas, ou outros apêndices móveis).
  - ❑ Os movimentos deste objeto são restringidos unicamente pelos obstáculos dispostos no espaço de trabalho.

# O Problema Básico

- ❑ O problema básico pode então ser resumido:
  - ❑ A partir de uma **posição inicial** e uma **orientação inicial**;
  - ❑ **Gera-se um caminho** composto por uma sequência contínua de posições e orientações do robô;
  - ❑ Que deve **evitar o contato** com os outros objetos no espaço de trabalho;
  - ❑ E que termina numa **posição meta pré-estabelecida**.

# Implementação

- ❑ Um método de busca é um algoritmo que controla a exploração do espaço de estados de modo a encontrar um caminho do estado inicial ao final.
- Realização de uma busca num espaço de estados.
- Duas grandes classes:
  - Busca em espaços discretos
  - Busca em espaços contínuos

# Espaço de Estados Discreto

- ❑ Estes métodos consistem em capturar a conectividade do espaço livre do robô em uma rede de curvas chamada **Roadmap**.
- ❑ Uma vez que o roadmap é construído, ele é utilizado como um conjunto de caminhos padronizados.

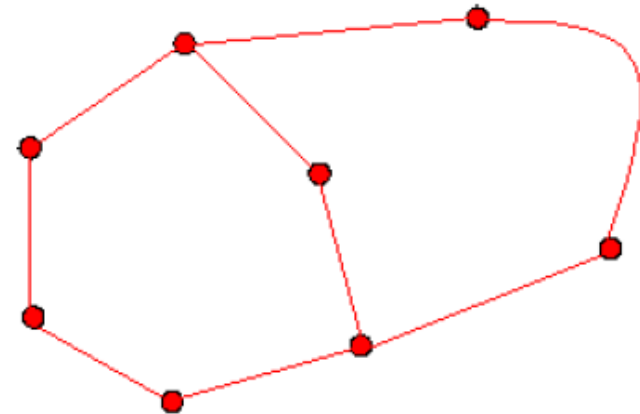
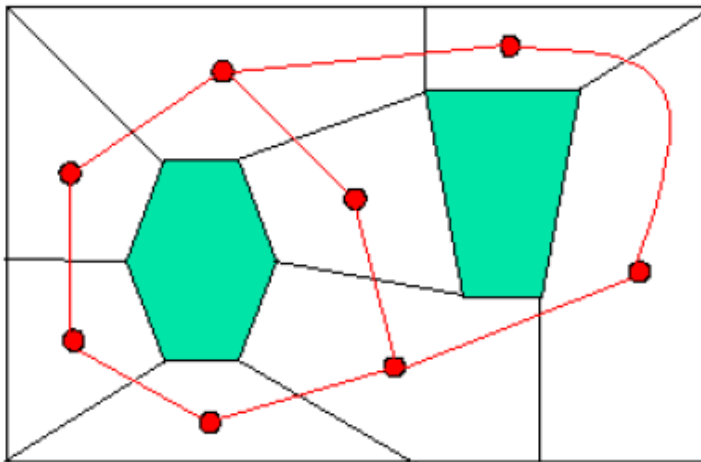
Diferentes técnicas:

- Discretização do C-Space livre.
- V-Graph
- Diagramas de Voronoi

# Discretização do C-Space livre

1

- Abordagem mais direta.
- Construção de grafo por vizinhança de células
- **Desvantagem:** pode resultar em um grande espaço de busca!

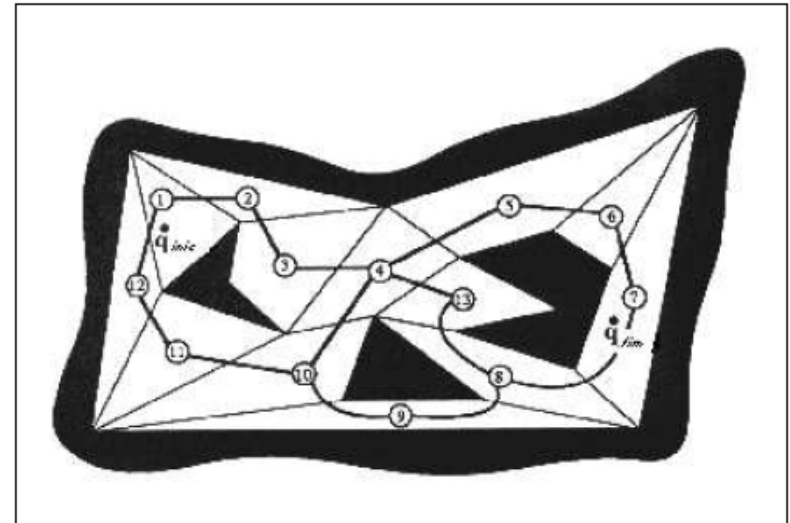
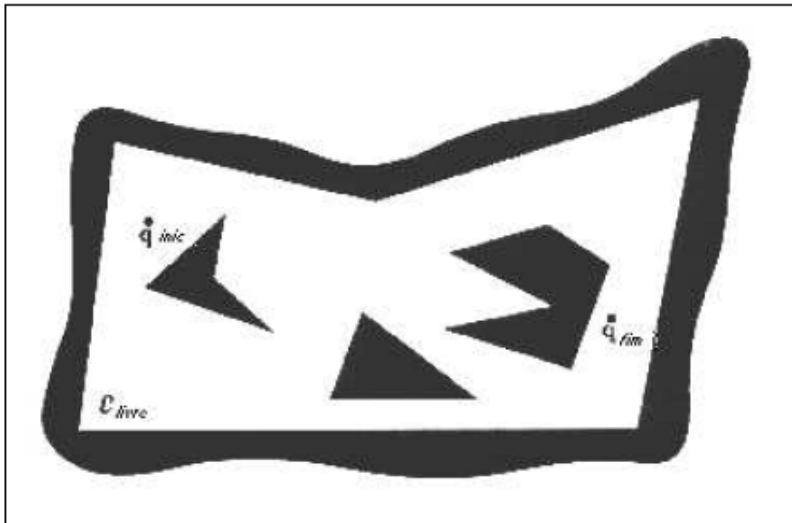




# Decomposição em Células

1

## Exemplo



# V-Graph (Visibility Graph)

2

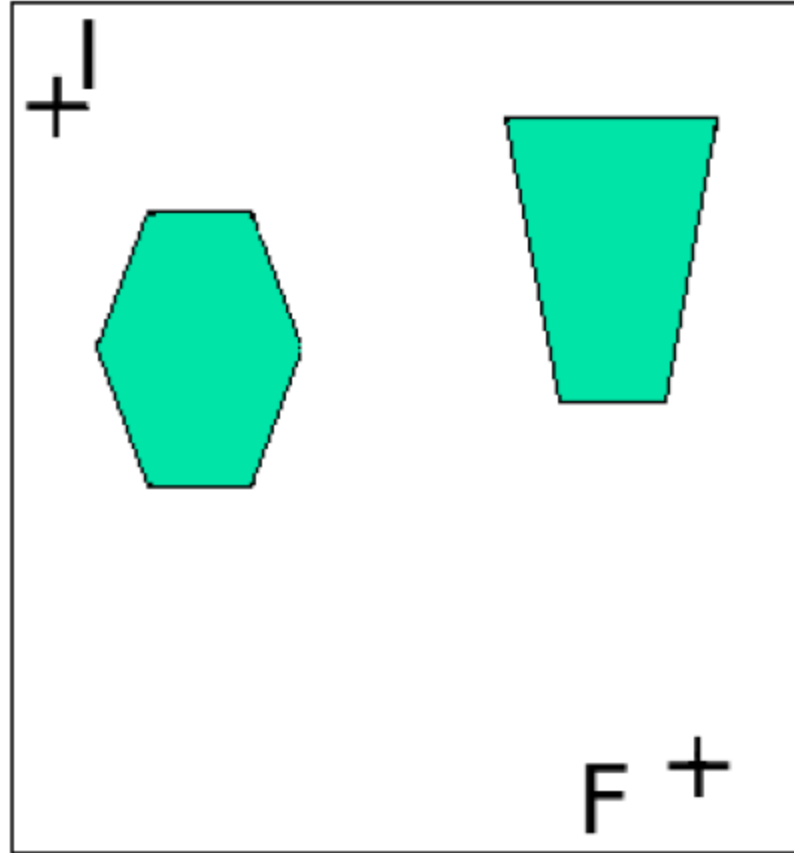


Figura: Esquema de criação de um V-Grafo

# V-Graph (Visibility Graph)

2

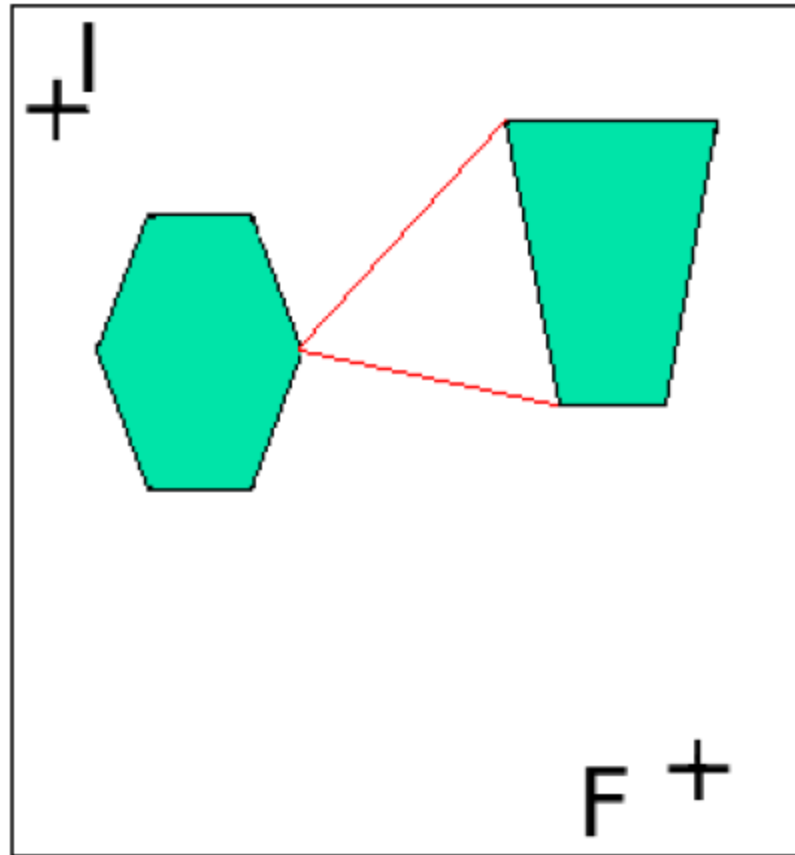


Figura: Esquema de criação de um V-Grafo

# V-Graph (Visibility Graph)

2

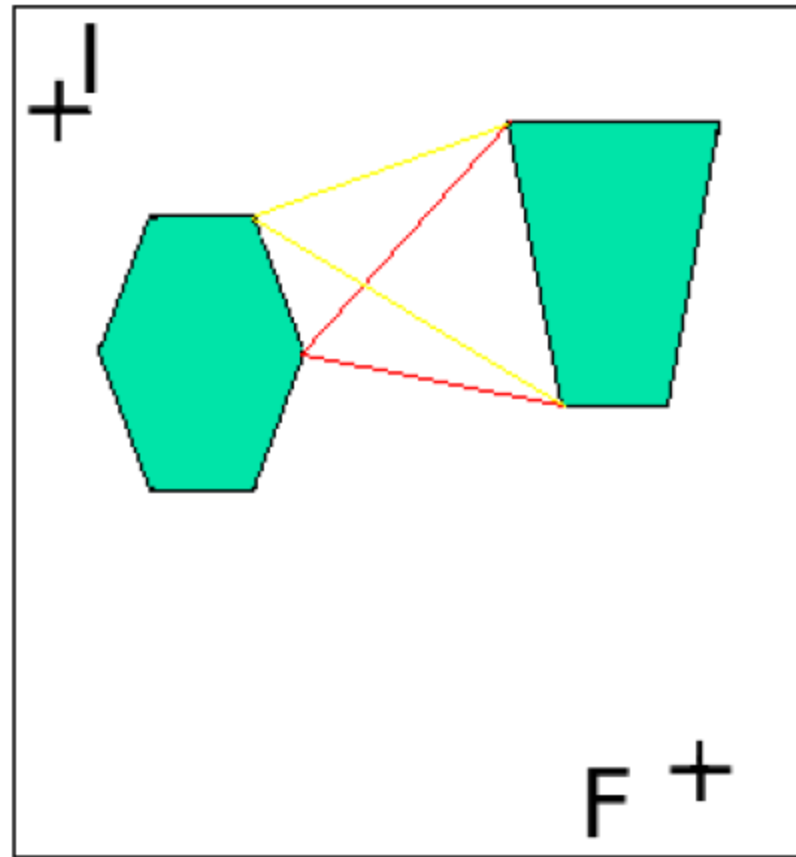


Figura: Esquema de criação de um V-Grafo

# V-Graph (Visibility Graph)

2

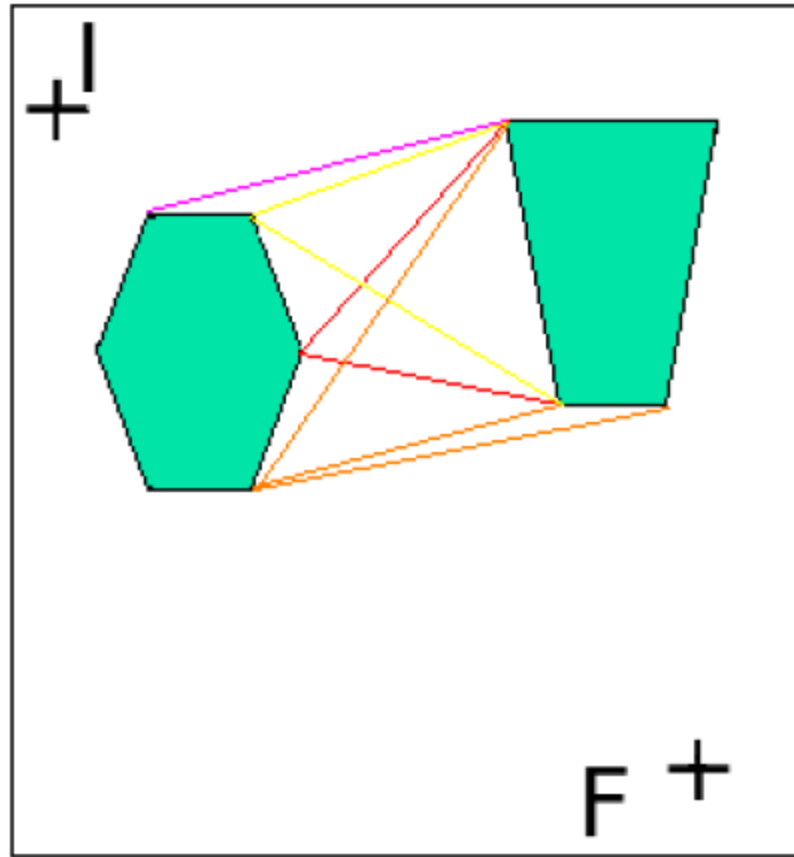


Figura: Esquema de criação de um V-Grafo

# V-Graph (Visibility Graph)

2

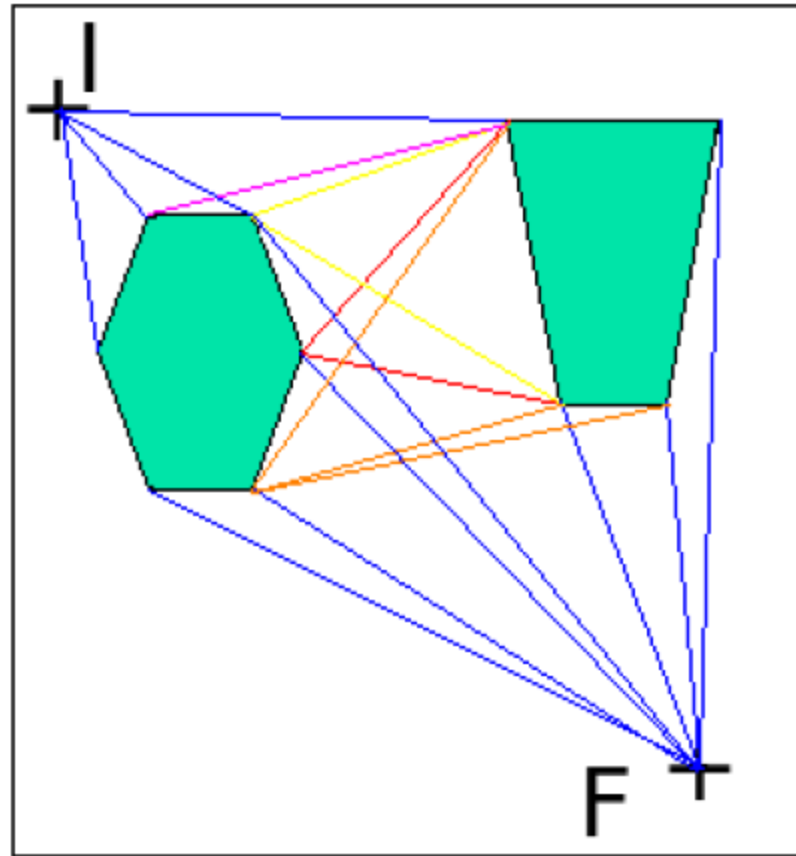
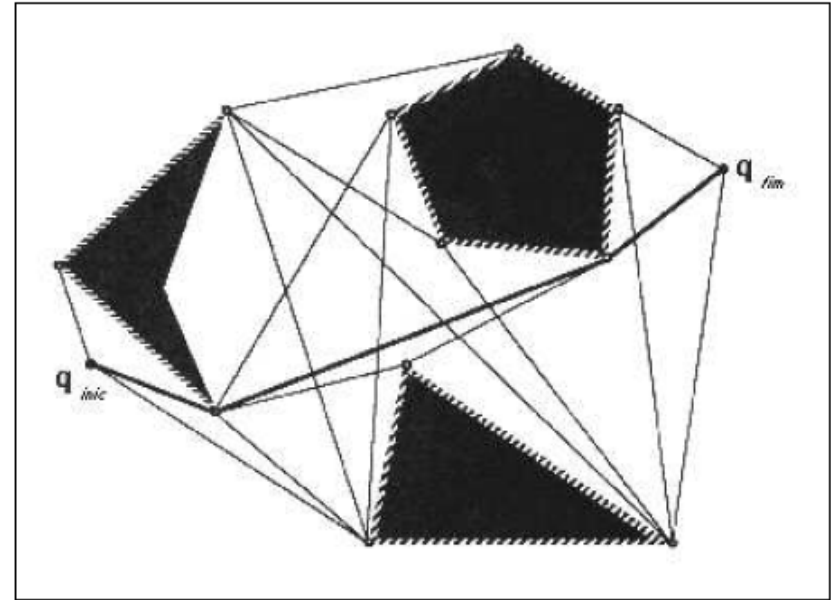
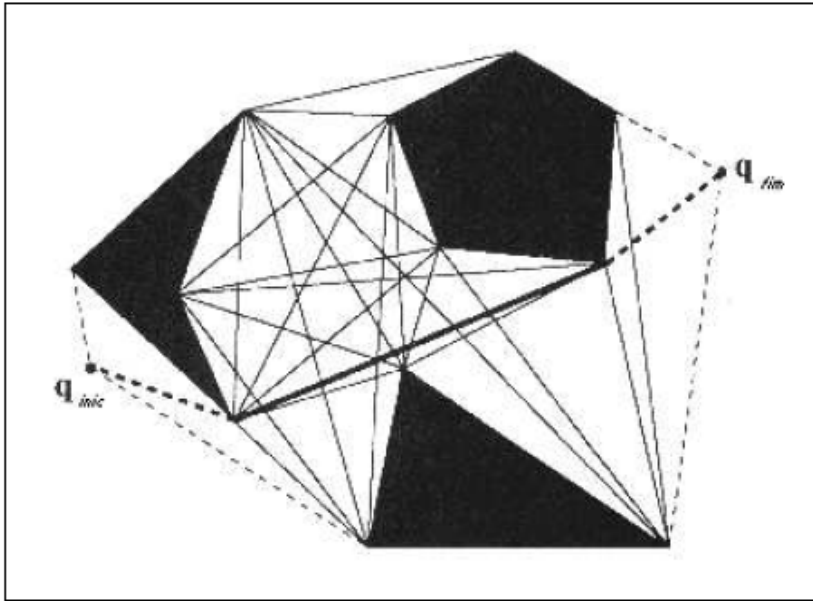


Figura: Esquema de criação de um V-Grafo

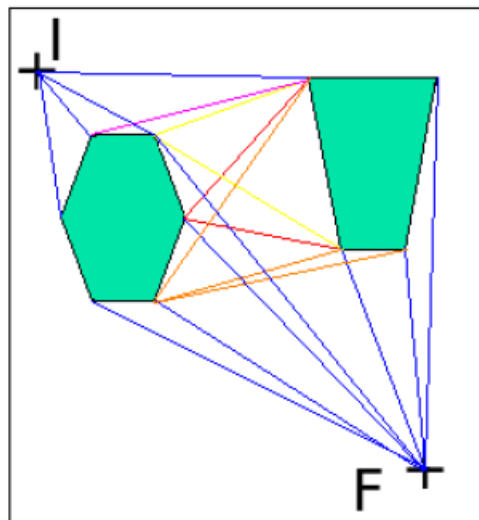
# V-Graph (Visibility Graph)



# V-Graph (Visibility Graph)

2

- V-grafo  $G=(V,E)$ , com:
  - V: conjunto de todos os vértices dos polígonos que definem obstáculos no ambiente  $\cup I \cup F$
  - E: conecta todos os vértices que são visíveis entre si (uma reta entre eles não cruza obstáculos; as arestas dos polígonos  $\cup E$ )
- Produz um caminho de menor comprimento.
- **Desvantagem:** Produz um caminho muito próximo os obstáculos.
  - *Solução:* Dilatação dos obstáculos antes da construção do V-Grafo.

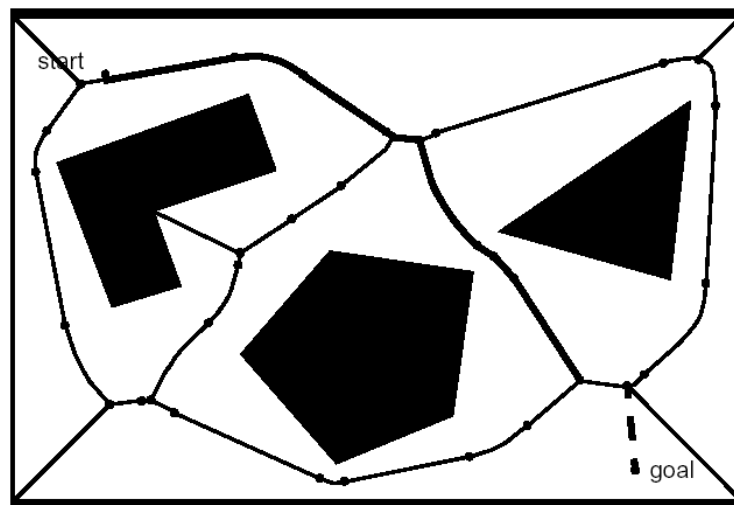




# Diagramas de Voronoi

3 Lugar geométrico dos pontos equidistantes das duas ou mais bordas mais próximas dos obstáculos e dos limites da área de trabalho do robô.

- **Objetivo:** Manter o robô o mais afastado possível dos obstáculos.
- Para espaços livres fechados com obstáculos compactos, DV é feito de curvas contínuas.
- **Desvantagem:** Pode gerar caminhos relativamente longos.



# Diagramas de Voronoi

- 3 • A partir de um ponto inicial  $I$ , andar desviando de obstáculos até entrar no DV. Seguir o DV até que possa perceber o ponto final  $F$  livre de obstáculos (reta até alcançar  $F$ ).

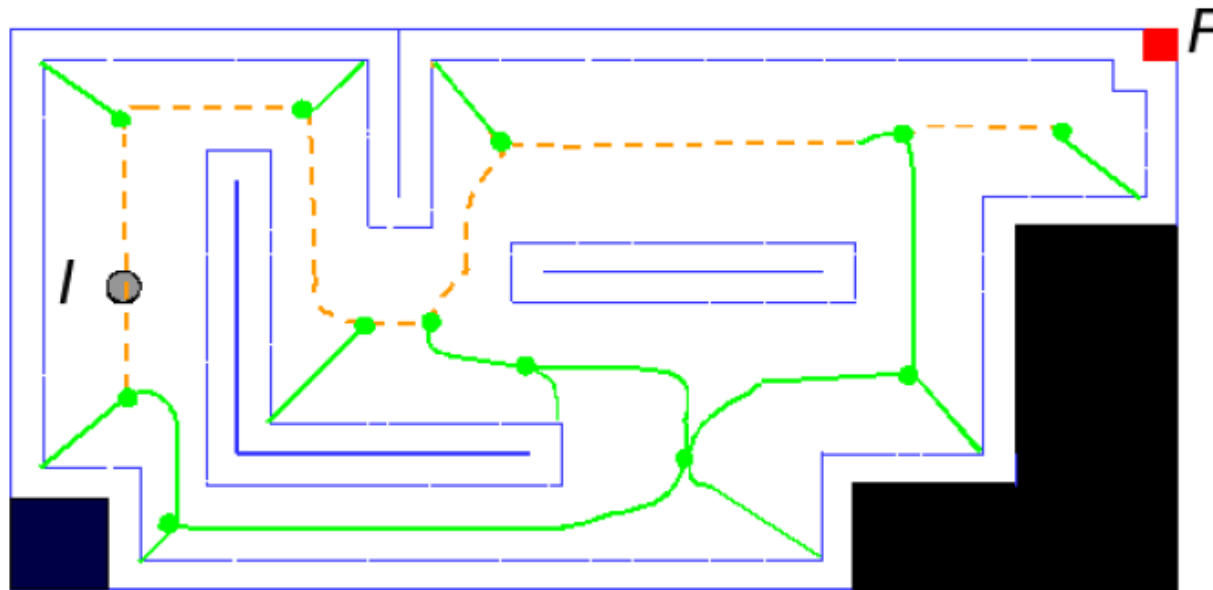


Figura: Exemplo de Diagrama de Voronoi

# Diagramas de Voronoi

3

## Exemplo



# Espaço de Estados Contínuo

- ❑ Planejamento em uma representação contínua dos espaço de estados:
  - ❑ Decomposição em Células
  - ❑ Campo Potencial
  - ❑ VFF - Virtual Field Force
  - ❑ VFH - Vector Field Histogram
  - ❑ Bug Algorithm

# Decomposição em Células

- ❑ São os métodos mais **extensamente estudados**;
- ❑ Consistem na **decomposição do espaço livre** do robô em regiões simples chamadas células;
- ❑ Os métodos de Decomposição em Células podem ser divididos em duas categorias:
  - ❑ Métodos Exatos
  - ❑ Métodos Aproximados

# Decomposição em Células

## ❑ Métodos Exatos

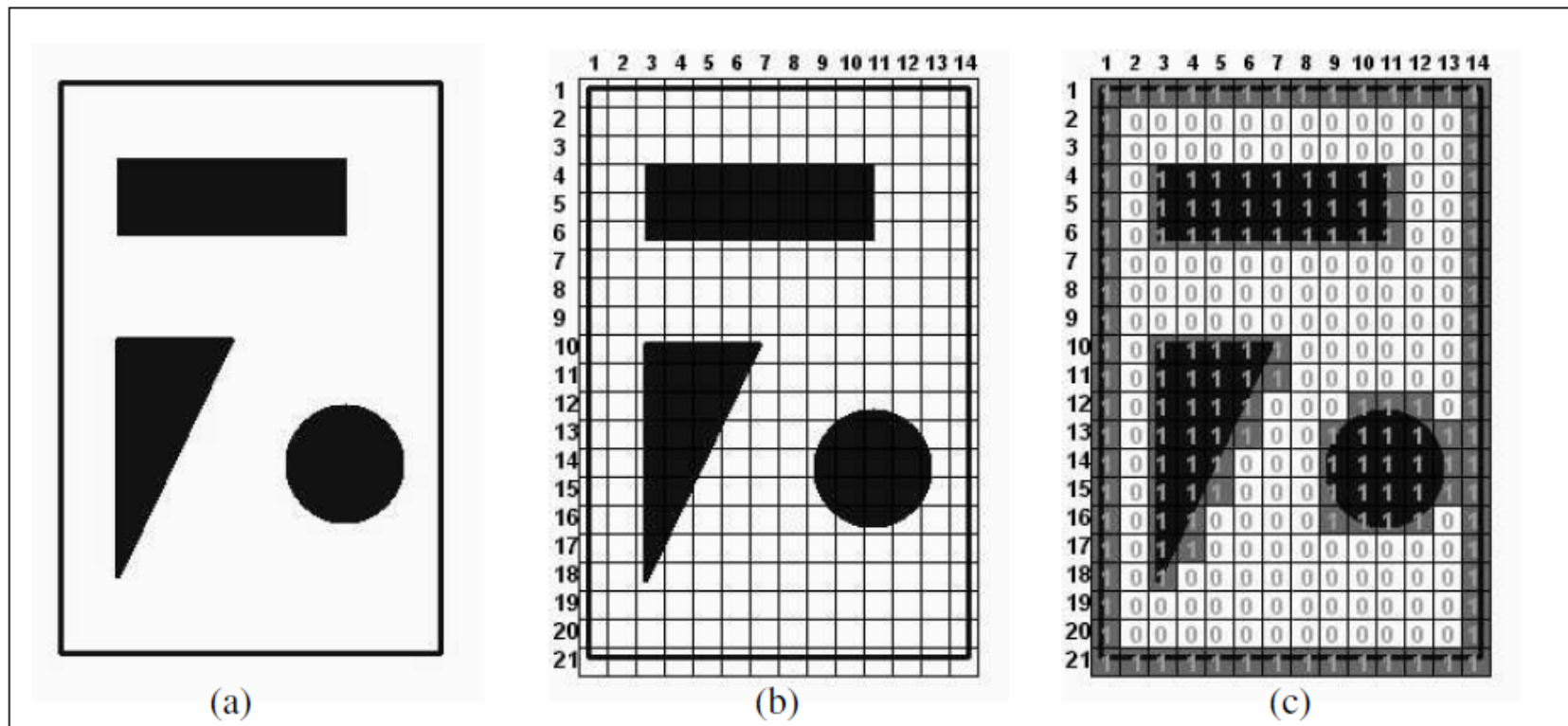
- ❑ Aplica-se a problemas nos quais o espaço de configurações  $C$  é *bidimensional* e os *obstáculos- $C$*  são *poligonais*;
- ❑ O *método* considera o robô como um único ponto, baseado na decomposição poligonal do espaço livre.

# Decomposição em Células

- ❑ Métodos Aproximados
  - ❑ Consiste em representar o espaço livre do robô como um conjunto de células;
  - ❑ A diferença é que o método aproximado as células possuem uma forma simples, definida previamente;
  - ❑ Sendo necessário algumas aproximações de maneira que garanta que o robô não colida com os obstáculos;
  - ❑ O espaço livre modelado tem que estar estritamente contido no espaço livre real do robô.
  - ❑ Talvez não seja encontrado algum caminho entre duas configurações, embora exista

# Decomposição em Células

## ❑ Métodos Aproximados





# Campo Potencial

O robô é modelado como uma partícula atuando sob influência de um campo potencial, onde:

- Obstáculos: campo repulsivo;
- Meta: campo atrativo.

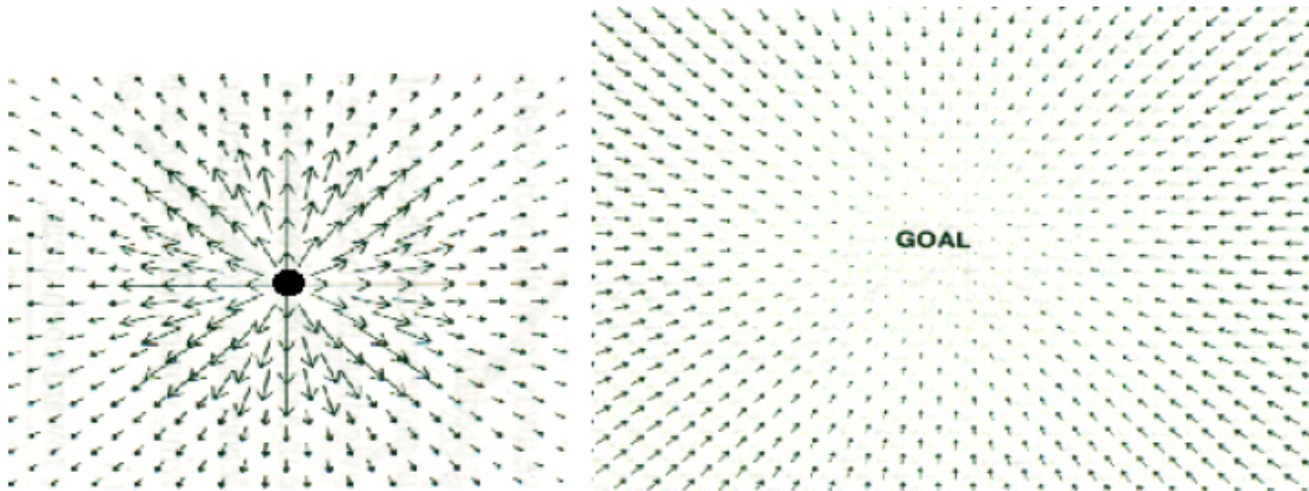
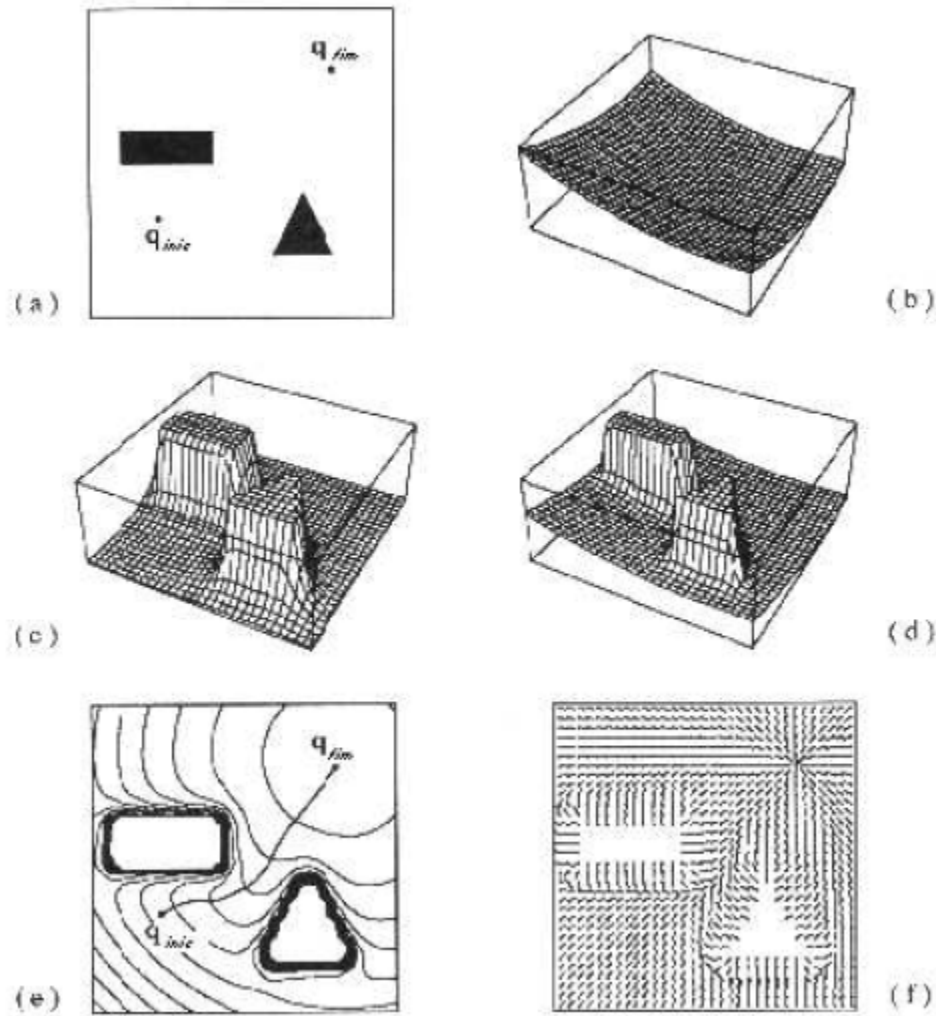


Figura: Campos repulsivo e atrativo

# Campo Potencial

- ❑ É uma **abordagem mais direta**;
- ❑ Consiste em discretizar o espaço de configuração em um fino grid regular de configurações e realizar a busca por um caminho livre neste grid.
- ❑ Esta abordagem necessita de **heurísticas adequadas** para direcionar a busca, pois o grid, em geral, tem um tamanho consideravelmente grande.
  - ❑ Vários tipos de heurísticas têm sido propostas, entre elas as de maior sucesso são aquelas que se tornaram conhecidas como **Campos Potenciais**.
- ❑ Baseados em métodos rápidos de **otimização**;

# Campo Potencial



# Campo Potencial

- O robô se movimenta de acordo com o campo resultante:

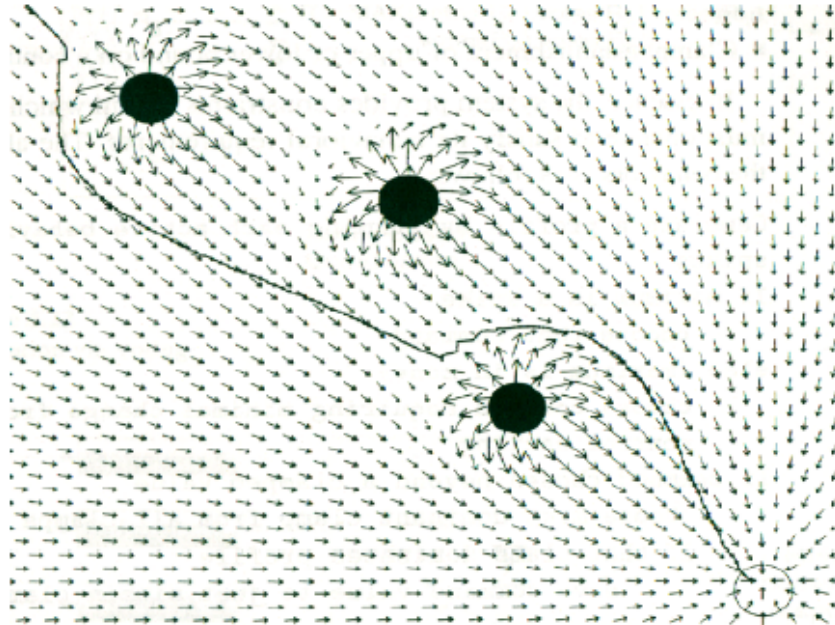


Figura: Campo potencial resultante

- Utilizado principalmente como planejamento local
  - Campo só é calculado na posição atual do robô!

# Campo Potencial

## Vantagens:

- Caminhos não precisam ser pré-planejados, podem ser gerados em tempo-real
- Planejamento e controle são integrados
- Geração de caminhos suaves

## Desvantagens:

- Problemas com mínimos locais.

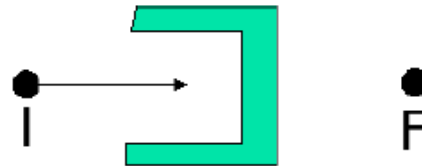


Figura: Configuração que apresenta um mínimo local

# Campo Potencial

Existem algumas soluções para este problema:

- Incorporar um planejador global (camada deliberativa);
- Retroceder quando cai em mínimo local (exige armazenamento de passos anteriores);
- Aumentar o potencial após passar pela região (assim, o robô não pára em um mínimo local, pois o campo muda);
- Introduzir passos aleatórios ao entrar em um mínimo local.

# VFF - Virtual Field Force

- Idéia similar ao uso de campos potenciais;
- Virtual Field Force (VFF) constrói uma grade de ocupação on-line e faz com que os obstáculos e a meta exerçam forças sobre o robô.
- Este algoritmo, ao contrário dos Campos Potenciais, só considera forças em uma pequena vizinhança ao redor do robô.

**Desvantagem:** Redução da distribuição espacial das forças em um único vetor (perde informações detalhadas sobre os obstáculos).



# VFF - Virtual Field Force

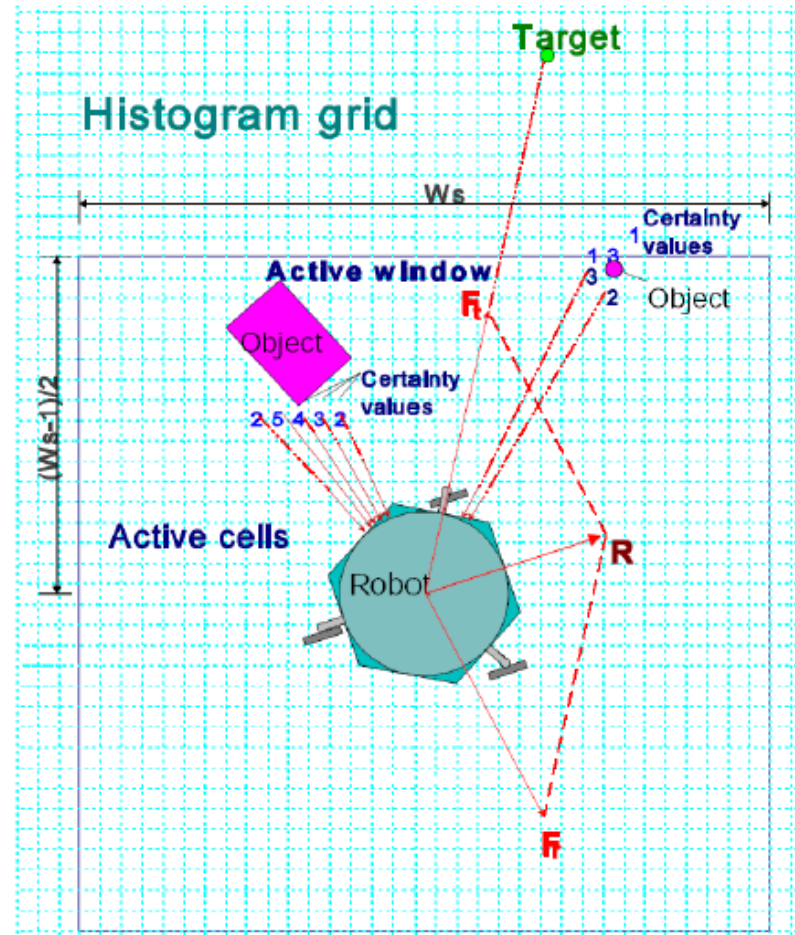


Figura: Esquema do VFF



# VFH - Vector Field Histogram

- VFH constrói uma representação polar local da densidade de obstáculos:
  - Usa uma janela local de  $W_s \times W_s$ , dividida em setores angulares;
  - Os obstáculos detectados em cada setor são combinados para indicar a navegabilidade no setor;
  - O setor livre mais próximo da direção preferencial de navegação do robô é escolhido (caso não haja obstáculos).

# VFH - Vector Field Histogram

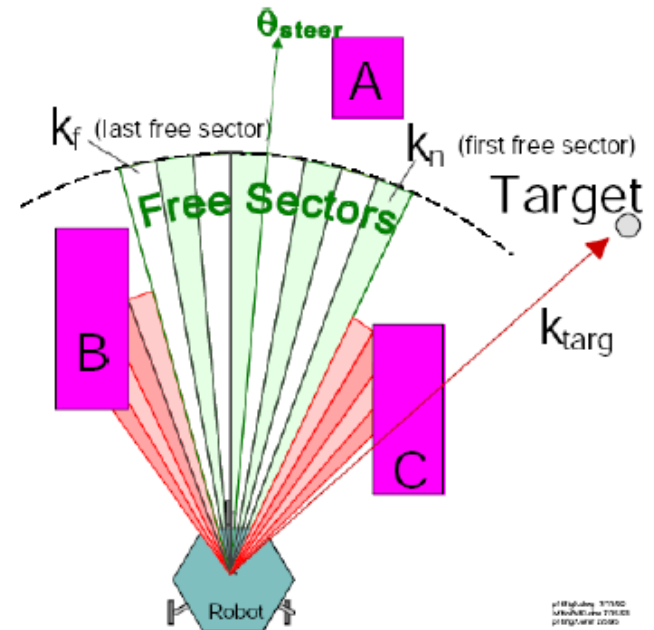
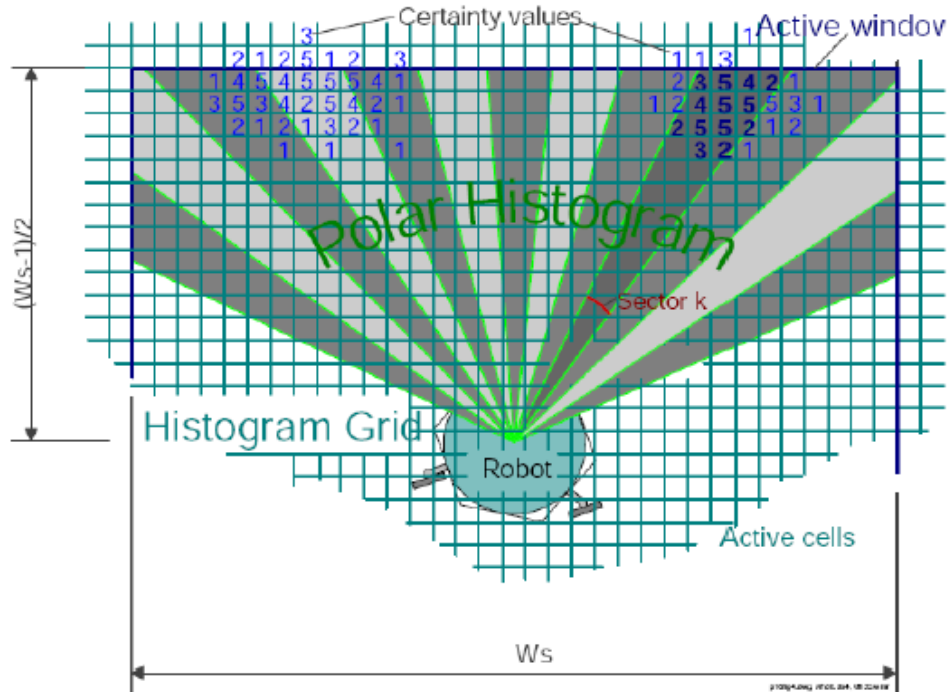


Figura: Esquema do VFH

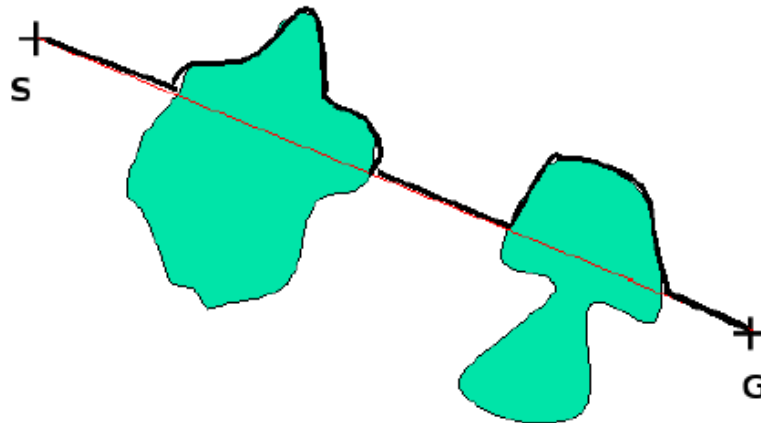
# Bug Algorithm (Princípio de Funcionamento)

- Utilizado onde não há mapas do ambiente;
- Algoritmo é completo (encontra uma trajetória caso exista, mas não necessariamente a ótima);
- Baseia-se em:
  - robô holonômico
  - pontos inicial e final conhecidos
  - robô sabe se localizar perfeitamente
  - robô tem sensores ideais de contato
  - não existe restrição de memória.

# Bug Algorithm (Princípio de Funcionamento)

Alternância entre dois comportamentos:

- 1 Mover em direção à meta
- 2 Contornar um obstáculo



Defina a reta (S,G)

While não chegar em F do{

Vá em direção a F pela reta (S,G).

If caminho obstruído then{

Marque sua localização atual L.

Circunde o obstáculo até:

a) chegar na reta (S,G) e  
seguir em direção a F.

b) retornar a L – neste caso,  
F é inatingível.} }

Figura: Exemplo de Bug Algorithm

# Incerteza Espacial

Em geral o planejamento simplifica os problemas, principalmente relacionados à localização:

- despreza a incerteza espacial.
- considera que o robô detecta que parou em um mínimo local (Potencial Field); e,
- que sabe encontrar novamente a reta (S,G) depois de circundar um obstáculo ou localizar o ponto L. (bug)

*Solução:* Representar o robô não apenas pela sua pose  $[x, y, \theta]$ , mas também pela sua matriz de covariância

$$\begin{matrix} \theta_{xx} & \theta_{xy} & \theta_{x\theta} \\ \theta_{xy} & \theta_{yy} & \theta_{y\theta} \\ \theta_{x\theta} & \theta_{y\theta} & \theta_{\theta\theta} \end{matrix}$$

# Ambientes Complexos

Algoritmos de planejamento de trajetória fazem uma série de simplificações, não lidando diretamente com:

- Ambientes Dinâmicos:
  - Objetos se movem de uma maneira conhecida;
  - Alteração de prioridades ou probabilidades de acordo com a situação;
- Ambientes Externos:
  - diferentes terrenos, caminhos físicos, etc;
  - detecção de buracos
- Ambientes Desconhecidos
  - Planejadores assumem ambientes conhecidos;
  - Algoritmos *online*: podem reavaliar o caminho enquanto estão sendo executados.
    - Exemplo simples: bug Algorithm.

# Planejamento para Múltiplos Robôs

- Possui características distintas daquelas envolvidas para um só robô:
  - Robôs devem evitar colisão entre eles;
  - Devem coordenar suas ações;
  - Podem comunicar entre si, etc.
- Dois tipos:
  - *Planejamento centralizado*: mais fácil, mas torna os robôs semi-autônomos pois dependem do planejador central.
  - *Planejamento distribuído*: maior dificuldade na garantia de cooperação e coordenação adequada entre os robôs, uso intensivo de comunicação.

# Exercícios

1. Discuta em que aspectos o planejamento de trajetória difere de:  
(a) verificação de colisões;  
(b) evitar colisões
2. Escreva um algoritmo para verificar se dois polígonos se chocam, dadas as coordenadas de todos os seus vértices e uma lista das arestas que interconectam pares de vértices.
3. Foi dito que a representação do robô como um ponto no seu espaço de configuração transforma o problema de planejamento de trajetória de um objeto dimensionado em um problema “mais simples”. Você concorda ou discorda com este ponto de vista? Explique sua resposta.



# Exercícios

4. Discuta as suposições sob as quais é realístico considerar um robô com rodas como um objeto bidimensional que se move livremente em um plano.
5. Identifique e discuta brevemente alguns problemas típicos de planejamento de trajetória para:
  - (a) um robô móvel carregando objetos no ambiente de um escritório;
  - (b) um robô de uma plataforma espacial movido por foguetes e equipado com dois braços para montar plataformas orbitais.

# Próxima Aula

## Aula 12

# Controle de Robôs Móveis