

**PIP/CA - Programa Interdisciplinar de Pós-Graduação  
Mestrado em Computação Aplicada da UNISINOS**

2000/2 - 3o. Trimestre - AULA 02 / FSO

**CONTROLE  
&  
ROBÓTICA INTELIGENTE**

• **Professor Responsável:**

**Prof. Dr. Fernando Osório**

E-Mail: [osorio@exatas.unisinos.br](mailto:osorio@exatas.unisinos.br)

Web: <http://www.inf.unisinos.br/~osorio/robi.html>

**TEMAS DE ESTUDO: CONTROLE DELIBERATIVO / PLANIFICADO**

*Controle Deliberativo: Path Planning*

- Mapas Exatos      x Mapas Aproximados
- Representação Local x Representação Global
- Local Planning Strategy    x Global Planning Strategy
- Posicionamento Absoluto    x Posicionamento Relativo

> **Métodos mais usados...**

- Road Map: C-Space & Visibility Graph, Voronoi Diagram, Fuzzy / Probabilistic Maps
- Occupation Grid / Cell Decomposition: Simples ou Valorado
- Potential Fields: Atração e Repulsão, Busca do Mínimo (local x global)
- Máquina de Estados: Finite State Automata, Petri Nets

> **Problemas do controle deliberativo...**

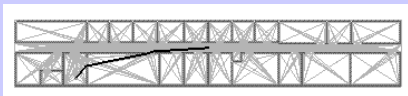
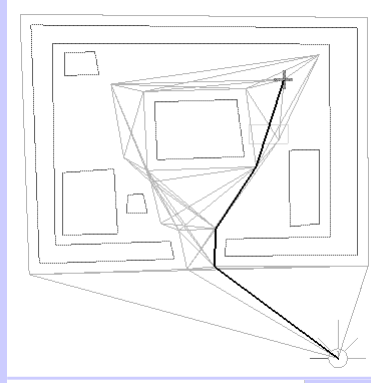
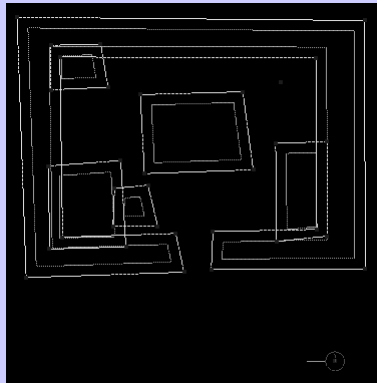
- Posicionamento do robô (recalibragem);
- Alterações do ambiente;
- Não trata: Imprecisão, Incerteza e Adaptação.

**ROAD MAP: CONTROLE DELIBERATIVO / PLANIFICADO**

- Road Map: C-Space & Visibility Graph

Referência: <http://ncg.unisinos.br/robotica/>

Farlei Heinen. Robótica Autônoma: A integração entre planificação e comportamento reativo. Editora Unisinos - 2000.



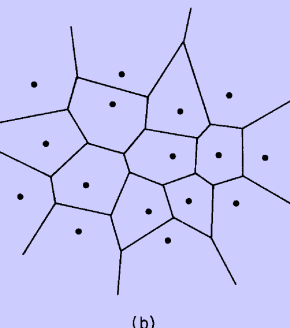
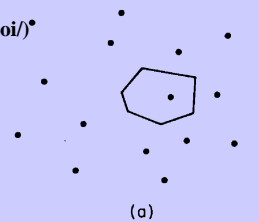
F. OSÓRIO - UNISINOS 2000

**ROAD MAP: CONTROLE DELIBERATIVO / PLANIFICADO**

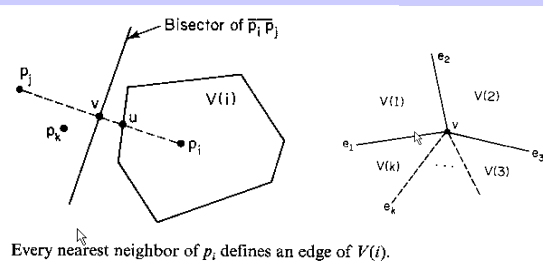
- Road Map: Voronoi Diagram (<http://www.netlib.org/voronoi/>)\*

Referências:

- Preparata, F.; Shamos, M. "Computational Geometry: An Introduction". Springer-Verlag - New York, 1985.
- Roque, W.; Heckler, C.; Dorneles, F. "On a Path Planning System for Autonomous Mobile Robots with Voronoi Diagrams and Visibility Pathways". Workshop de Robótica Inteligente - SBC, Brasília, 1997
- Fortune, Steve. July 1995. Voronoi diagrams and Delaunay triangulations. <http://ftp.eq.uc.pt/software/math/netlib/voronoi/>



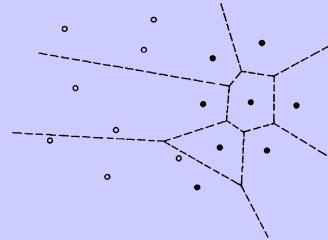
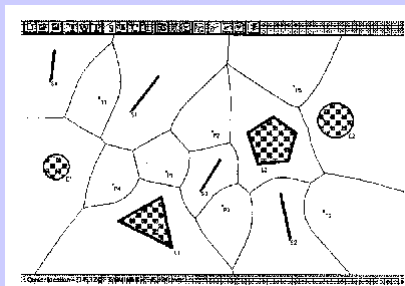
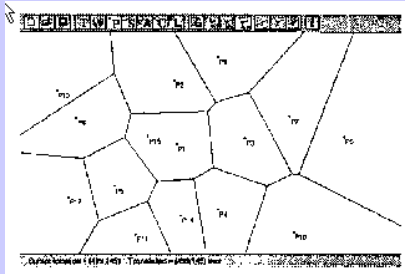
(a) A Voronoi polygon; (b) the Voronoi diagram.



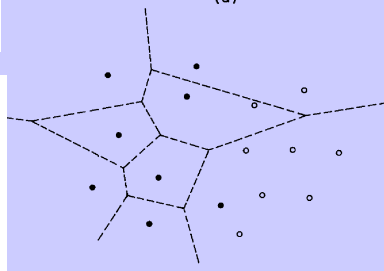
F. OSÓRIO - UNISINOS 2000

**ROAD MAP: CONTROLE DELIBERATIVO / PLANIFICADO**

**• Road Map: Voronoi Diagram - Objetos VD**



(a)



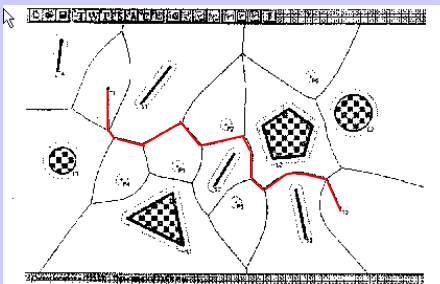
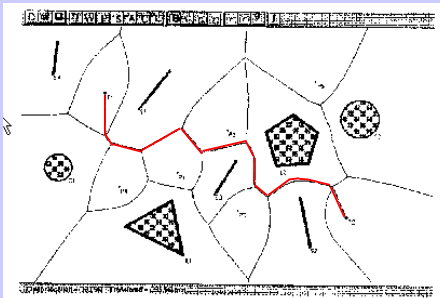
(b)

The Voronoi diagrams of the left set (a) and of the right set (b).

F. OSÓRIO - UNISINOS 2000

**ROAD MAP: CONTROLE DELIBERATIVO / PLANIFICADO**

**• Road Map: Voronoi Diagram**



**Planificação da Trajetória:**  
- Uso do diagrama de Voronoi  
- Objetivo: ficar o mais afastado possível dos objetos, indo de um ponto origem ao ponto destino.

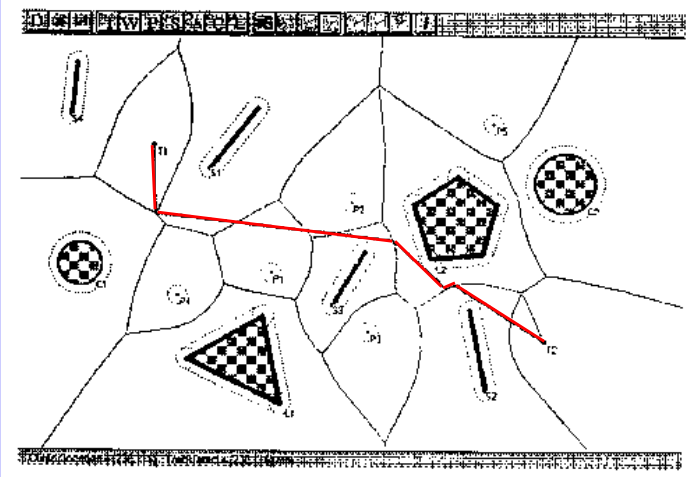
**Planificação de Trajetória:**  
- Adição de uma “margem de segurança” ao redor dos objetos

F. OSÓRIO - UNISINOS 2000

**ROAD MAP: CONTROLE DELIBERATIVO / PLANIFICADO**

- Road Map: Voronoi Diagram x Visibility Pathway

Otimizando a trajetória entre 2 pontos...



Voronoi Demos: <http://compgeom.cs.uiuc.edu/~jeffe/compgeom/demos.html>  
<http://wwwpi6.fernuni-hagen.de/Geometrie-Labor/VoroGlide/>

F. OSÓRIO - UNISINOS 2000

**GRIDS: CONTROLE DELIBERATIVO / PLANIFICADO**

- Occupation Grid: Simples ou Valorado

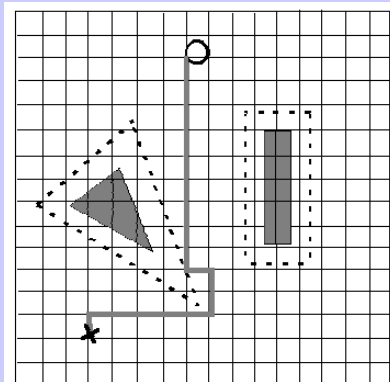


Figura 4.3 Navegação baseada em Grid

16	17	18	19	20	21	20	19	18	19	20	21	22	23	24	25
15	16	17	18	19	20	19	18	17	18	19	20	21	22	23	24
14	15	16	17	18	19	18	17	16	17	18	19	20	21	22	23
13	14	15	16	17	18	17	16	15	16	17	18	19	20	21	22
12	13	14	15	16	17	16	15	14	15	16	17	18	19	20	21
11	12	13	14	15	16	15	14	13	14	15	16	17	18	19	20
10	11	12	13	14	15	14	13	12	13	14	15	16	17	18	19
9	10	11	12	13	14	13	12	11	12	13	14	15	16	17	18
8	9	10	11	12	13	12	11	10	11	12	13	14	15	16	17
7	8	9	10	11	12	11	10	9	10	11	12	13	14	15	16
6	7	8	9	10	11	10	9	8	9	10	11	12	13	14	15
5	6	7	8	9	10	9	8	7	8	9	10	11	12	13	14
4	5	6	7	8	9	8	7	6	7	8	9	10	11	12	13
3	4	5	6	7	8	7	6	5	6	7	8	9	10	11	12
2	3	4	5	6	7	6	5	4	5	6	7	8	9	10	11
1	2	3	4	5	6	5	4	3	4	5	6	7	8	9	10
0	1	2	3	4	5	4	3	2	3	4	5	6	7	8	9

Figura 4.4 Navegação por transformada de distância

© Farlei Heinen 2000

F. OSÓRIO - UNISINOS 2000

**GRIDS: CONTROLE DELIBERATIVO / PLANIFICADO**

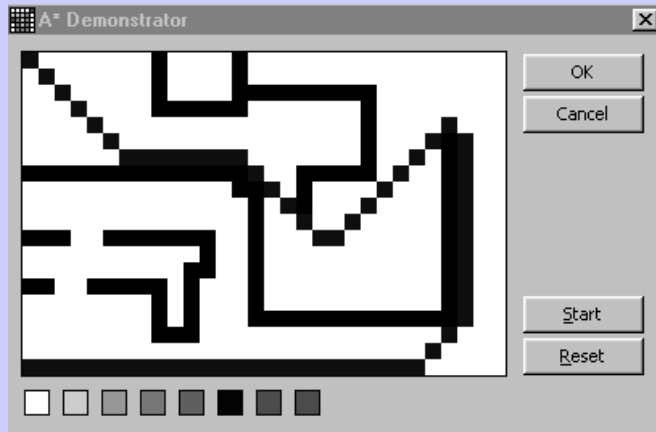
• Occupation Grid: Grid + A\*

“Exploração com Heurística”

Minimizar o custo total =>  $F(X) = G(X) + H(X)$

G(X) ∴ Custo do caminho percorrido

H(X) ∴ Estimativa do custo do caminho a percorrer



F. OSÓRIO - UNISINOS 2000

**POTENTIAL FIELDS: CONTROLE DELIBERATIVO / PLANIFICADO**

• Potential Fields: Campos potenciais de atração e repulsão

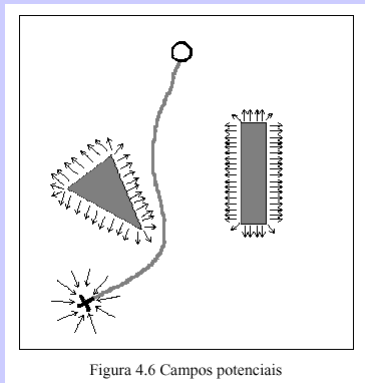


Figura 4.6 Campos potenciais

© Farlei Heinen 2000

**Referência:**  
 Silveira, J.; Aude, E.; Martins, M.  
 Trajectory Planning with Obstacle Avoidance  
 within CONTROLAB. WRI, Brasília, 1997.

O que de certa maneira é muito próximo dos grids com valores...

16	17	18	19	20	21	20	19	18	19	20	21	22	23	24	25
15	16	17	18	19	20	19	18	17	18	19	20	21	22	23	24
14	15	16	17	18	19	18	17	16	17	18	19	20	21	22	23
13	14	15	16	17	18	17	16	15	16	17	18	19	20	21	22
12	13	14	15	16	17	16	15	14	15	16	17	18	19	20	21
11	12	13	14	15	16	15	14	13	14	15	16	17	18	19	20
10	11	12	13	14	15	14	13	12	13	14	15	16	17	18	19
9	10	11	12	13	14	13	12	11	12	13	14	15	16	17	18
8	9	10	11	12	13	12	11	10	11	12	13	14	15	16	17
7	8	9	10	11	12	11	10	9	10	11	12	13	14	15	16
6	7	8	9	10	11	10	9	8	9	10	11	12	13	14	15
5	6	7	8	9	10	9	8	7	8	9	10	11	12	13	14
4	5	6	7	8	9	8	7	6	7	8	9	10	11	12	13
3	4	5	6	7	8	7	6	5	6	7	8	9	10	11	12
2	3	4	5	6	7	6	5	4	5	6	7	8	9	10	11
1	2	3	4	5	6	5	4	3	4	5	6	7	8	9	10
0	1	2	3	4	5	4	3	2	3	4	5	6	7	8	9

Figura 4.4 Navegação por transformada de distância

F. OSÓRIO - UNISINOS 2000

## MÁQUINA DE ESTADOS: *CONTROLE DELIBERATIVO / PLANIFICADO*

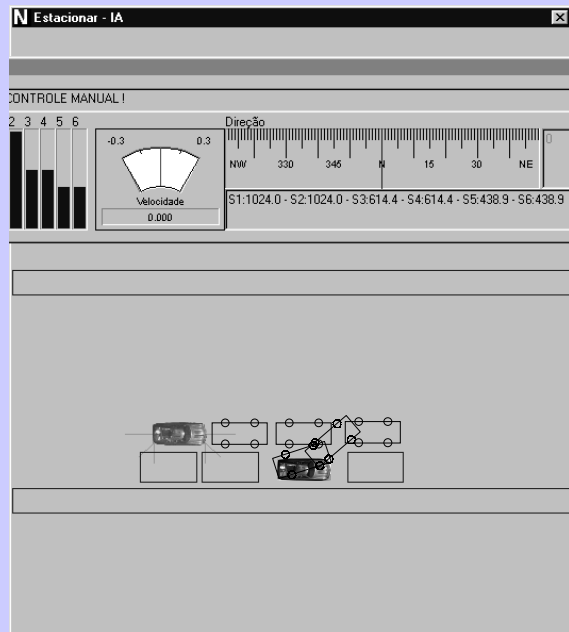
### • Máquina de Estados:

```

void AvaliaEstado()
{
    verifica_sensores();
    switch(estado)
    {
        case PARADO:
            break;
        case PROCURANDO_VAGA:
            speed = 0.04f;
            rotvel = 0.0f;
            // mantém o carro alinhado
            if(phi < 0.0f) phi += 0.1;
            if(phi > 0.0f) phi -= 0.1;
            // evita colisoes frontais
            if(vaisens[0] < 128.0f)
            {
                speed = 0.0f;
                estado = PARADO;
            }
            // evita colisoes traseiras
            if(vaisens[1] < 128.0f)
            {
                speed = 0.0f;
                estado = PARADO;
            }
            ...
        case POSICIONANDO:
            speed = 0.04f;

            // mantém o carro alinhado
            if(phi < 0.0f) phi += 0.1;
            if(phi > 0.0f) phi -= 0.1;
            // se afasta se estiver perto dos obstaculos
            if(vaisens[4] < 300.0f) phi -= 0.3;
            if(vaisens[5] < 300.0f) phi += 0.3;
            // evita colisoes frontais
            if(vaisens[0] < 128.0f)
            {
                speed = 0.0f;
                estado = PARADO;
            }
            ...
    }
}
    
```

® Farlei Heinen 2000



F. OSÓRIO - UNISINOS 2000

## ARIADNE CLEW: *CONTROLE DELIBERATIVO / PLANIFICADO*

### • Planificação usando Algoritmos Genéticos:

#### Referência:

Bessière

Mazer

Ahuactzin

Lab. Leibniz

Eq. Laplace

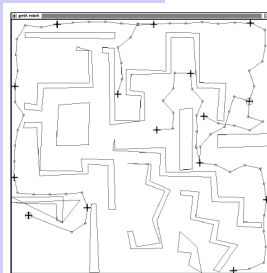
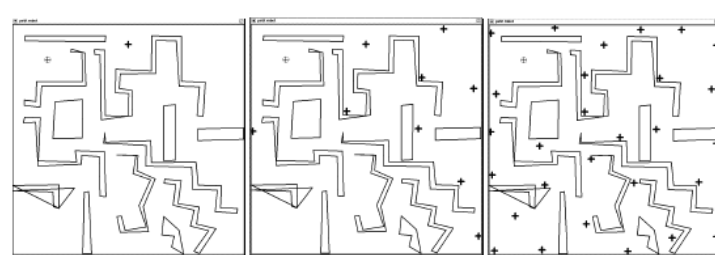
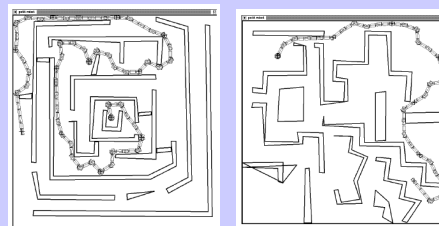


figure 8: The ARIADNE'S CLEW

The first picture represents the initial position and the first landmark. The subsequent landmarks are then uniformly spread over the search space while the method keeps track of all paths joining the landmarks to the initial position.



F. OSÓRIO - UNISINOS 2000

## ROBÓTICA AUTÔNOMA :

### \* Controle Deliberativo:

#### • Vantagens:

- Pré-planejamento de tarefas;
- Uso de informações sobre o ambiente (Indoor Navigation);
- Otimização da trajetória.

#### • Desvantagens:

- Usualmente possuem problemas ligados a complexidade computacional e mínimos locais;
- Posicionamento do robô (recalibragem);
- Alterações do ambiente;
- Não trata: Imprecisão, Incerteza e Adaptação.

### \* Controle Robusto...

Vide: <http://www.ai.mit.edu/people/brooks/papers.html>  
Rodney A. Brooks - Cambrian Intelligence: The Early History of the New AI  
Ronald C. Arkin - Behaviour Based Robotics  
Richard Sutton & Andrew Barto - Reinforcement Learning: Na Introduction

*Controle Reativo!*