Autonomous Vehicles:

Research, Design and Implementation of Intelligent Autonomous Vehicles

Autonomous Vehicles Research Group - GPVA
http://www.eletrica.unisinos.br/~autonom

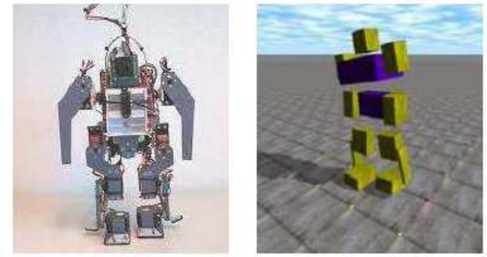
Tutorial page: http://inf.unisinos.br/~osorio/palestras/cerma07.html

Dr. Fernando S. Osório - Applied Computing Post-Grad. Program PIPCA
Dr. Christian R. Kelber - Electrical Engineering / Computer Eng.
Dr. Cláudio R. Jung - Applied Computing M.Sc. Program PIPCA
M.Sc. Farlei Heinen - Computer Engineering B.Sc. (Director)

Veículos Autônomos Inteligentes

- Introduction
- Robotic: Automatons, Mobile Robots and Autonomous Robots
  - Perception, Action, Locomotion e Communication
  - Control and Intelligence
- Intelligent Vehicles
  - Technologies for Vehicle Automation
  - Control pyramid
- Intelligent Control of Autonomous Vehicles
  - *Control: Computational Architectures*
  - *Simulation of Autonomous Vehicles*
- Computer Vision
- Practical Applications

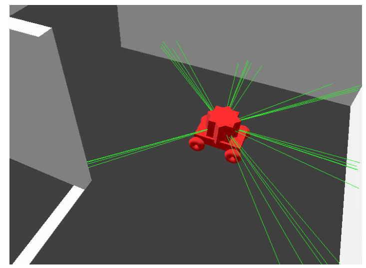
CONTROL: Computational Architectures

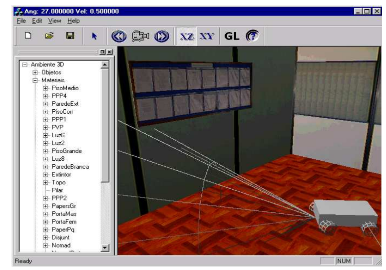


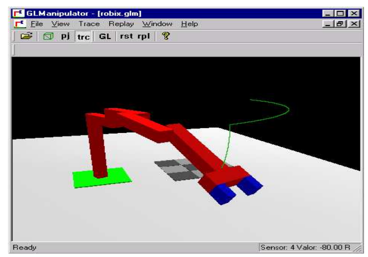
(a) Robô real (b) Robô simulado

=> From where do I start? *Modeling and Simulation*

- Models:
  - Sensorial Models
  - Actuator Models
  - Kinematics Models
  - Environment Models
  - A.I. Models (Path Planning, Agents, ...)



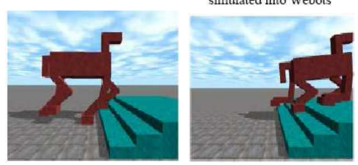


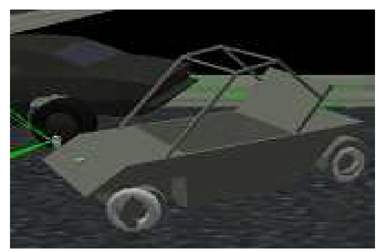


- Simulation:
  - Validate models
  - Test robustness
  - Improve design

The Sony Dream Robot in the real world

The Sony Dream Robot simulated into Webots



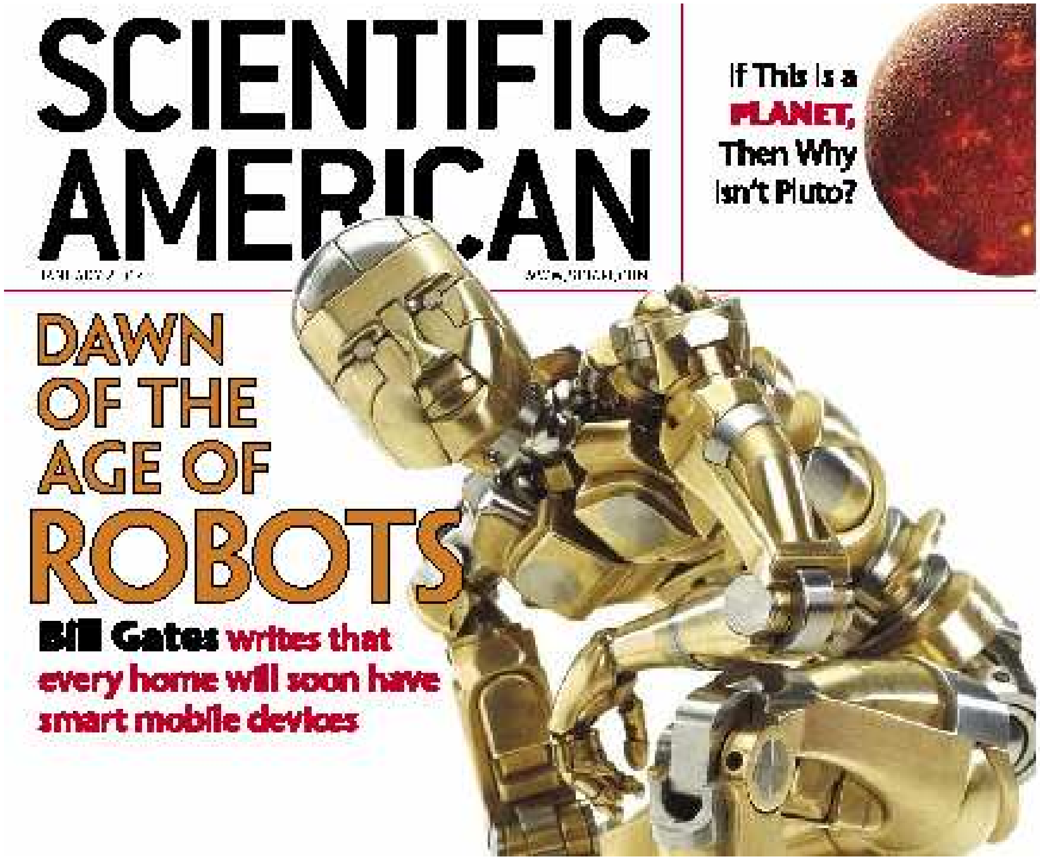


CONTROL: Computational Architectures

A ROBOT IN EVERY HOME

The leader of the PC revolution predicts that the next hot field will be robotics

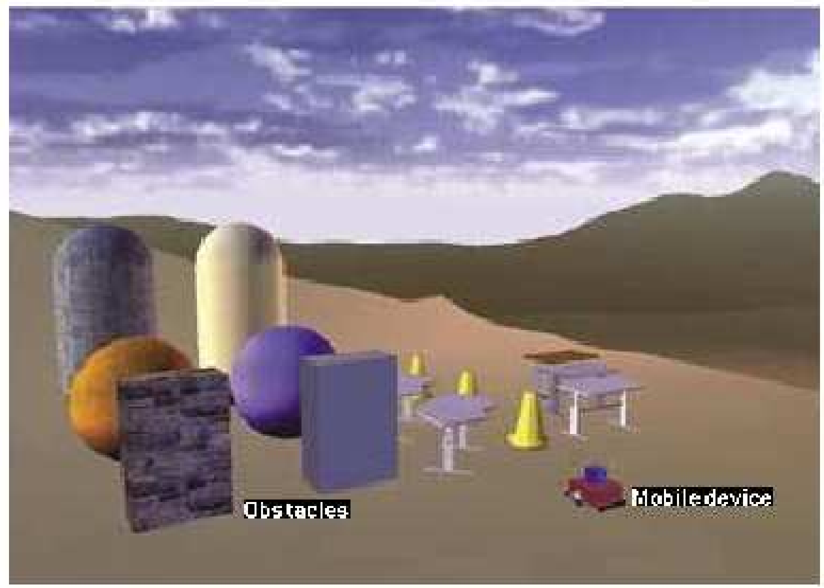
=> From where do I start? ***Modeling and Simulation***



Scientific American January 2007

By Bill Gates

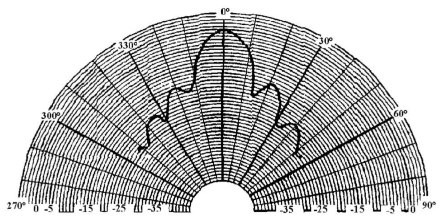
Imagine being present at the birth of a new industry. It is an industry based on groundbreaking new technologies, wherein a handful of well-established corporations sell highly specialized



COMPUTER TEST-DRIVE of a mobile device in a three-dimensional virtual environment helps robot builders analyze and adjust the capabilities of their designs before trying them out in the real world. Part of the Microsoft Robotics Studio software development kit, this tool simulates the effects of forces such as gravity and friction.

CONTROL: Computational Architectures

- Sensorial Models
- Kinematics Models



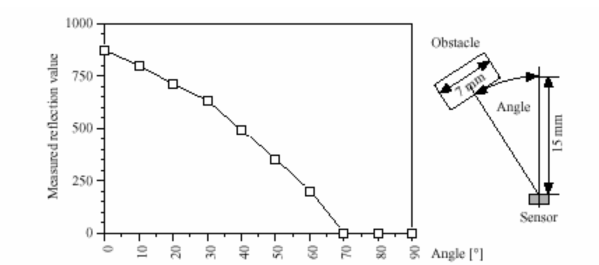
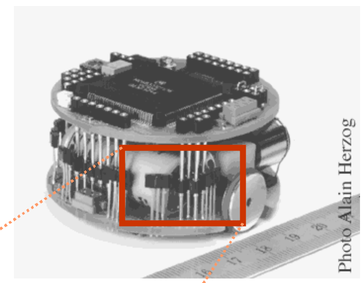
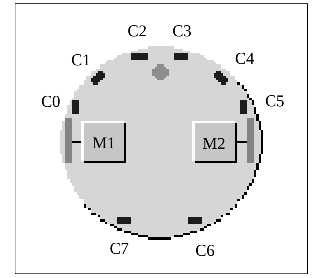


Figure 12: Typical response of a proximity sensor for an obstacle (7 mm in width) at a distance of 15 mm. The measurement is given versus the angle between the forward orientation of the robot and the orientation of the obstacle.

Sensorial Model:

- **Sonar**
- **Infrared**
- **Radar, Compass, Odometer**

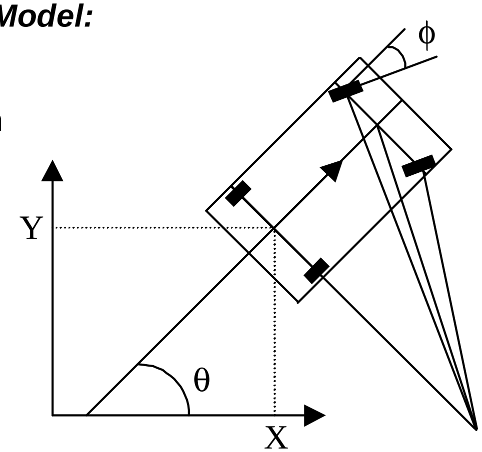




Kinematics Model:

- **Differential**
- **Aeckerman**

Aeckerman



$\theta = V / L * \sin(\Phi)$

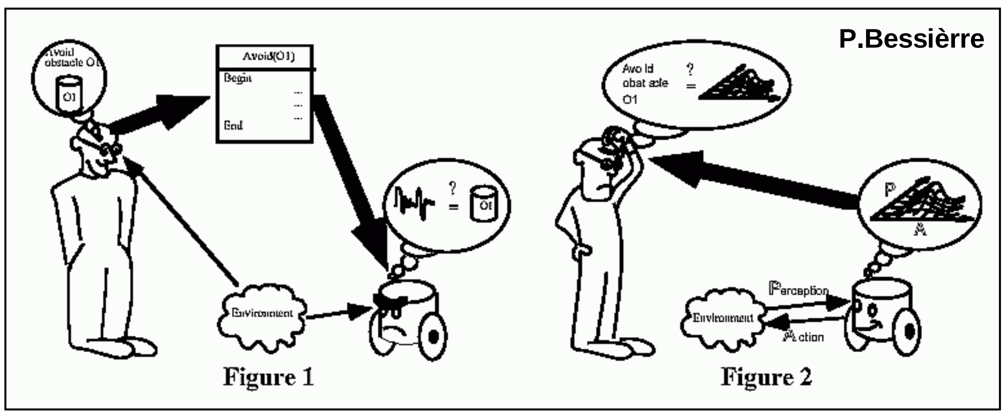
$X = V * \cos(\Phi) * \cos(\theta)$

$Y = V * \cos(\Phi) * \cos(\theta)$

CONTROL: Computational Architectures

- Sensorial Models
- Kinematics Models
- Robotic Control:
  * Reactive
  * Deliberative
  * Hierarchical
  * Hybrid
- Environment Maps
  * Building Maps
  * Path Planning
  * SMPA - *Sense Model Plan Act*
- Problems:
  * Complex tasks
  * Avoid Obstacles: Static / Mobile - **Unexpected obstacles**
  * Robot actual position estimation - **Where am I ?**

CONTROL: Computational Architectures

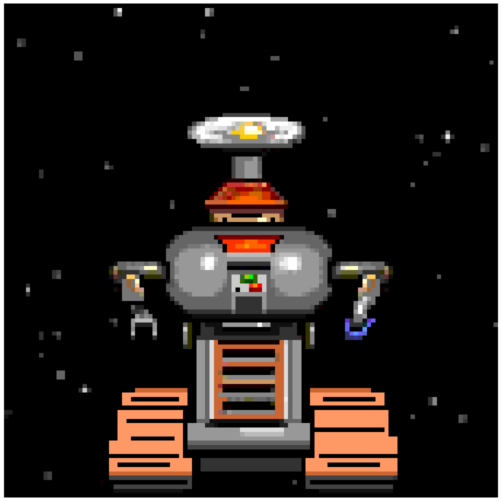


Complexity...

*** Action Planning**
*** Ability to Perceive the Environment**
*** Ability to Decide**
*** Ability to Act**
*** High Level Tasks Planning**
*** Reaction: Sensorial-Motor**
*** Estimate Actual and Future States**
*** Adaptation and Learning**
*** Robustness**
*** Unexpected Situations**

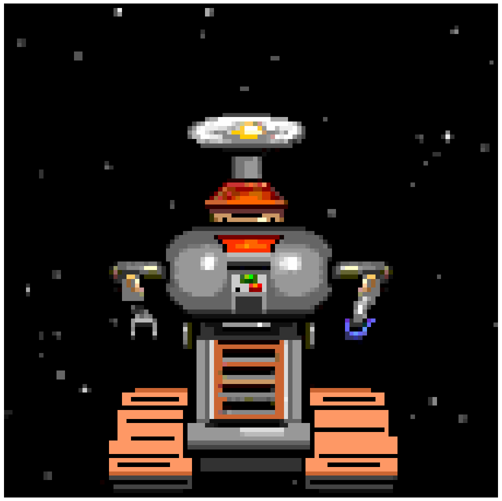
=> From where do I start ???

CONTROL: Computational Architectures

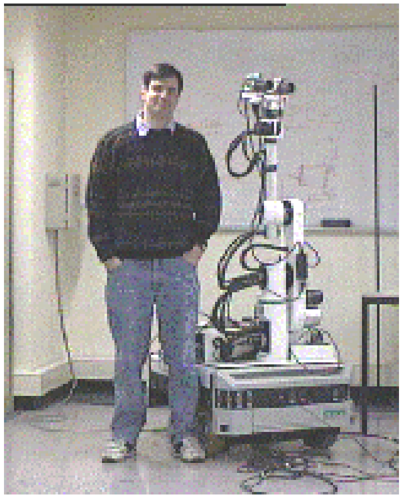


Complexity...

CONTROL: Computational Architectures

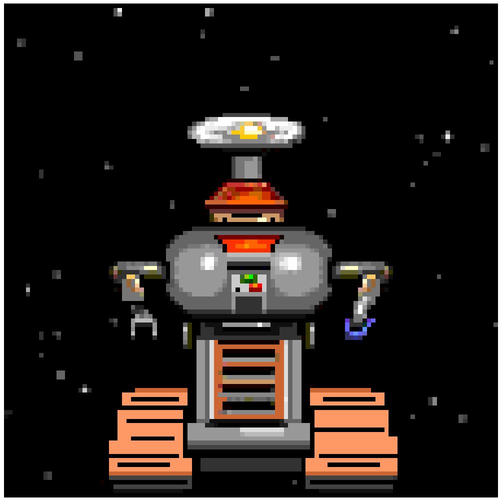


Complexity...

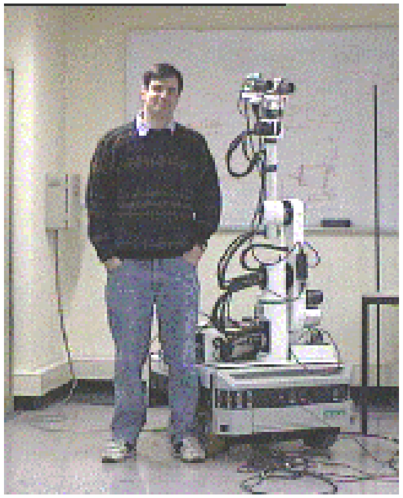


Simplify! How?

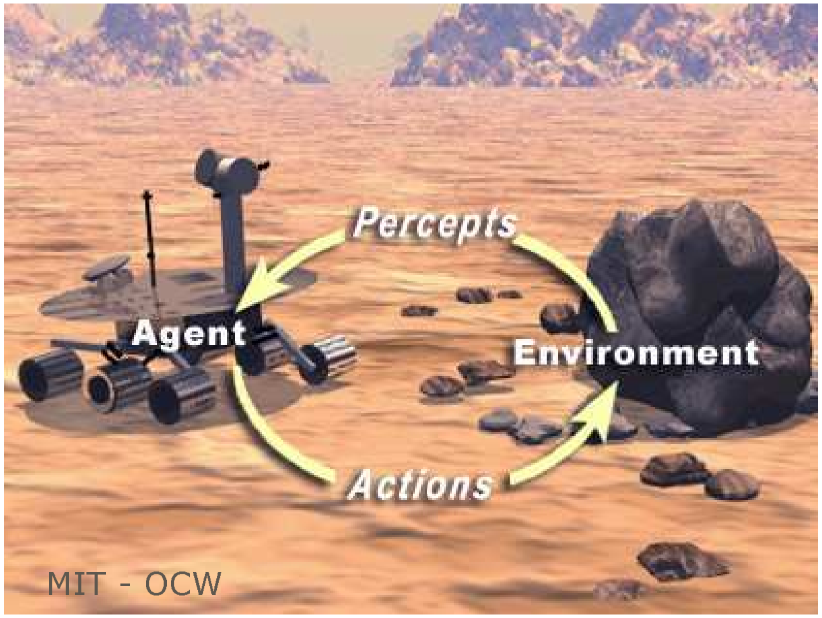
CONTROL: REACTIVE Architecture



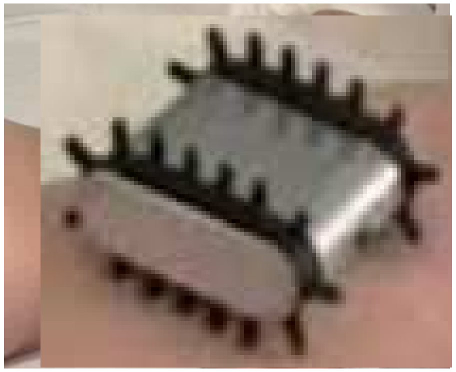
Complexity...



Simplify! How?

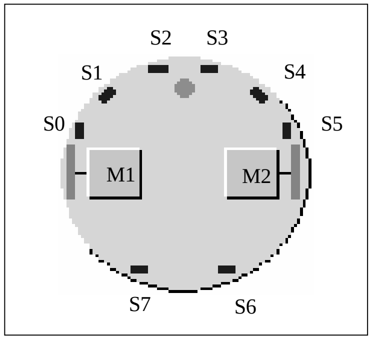


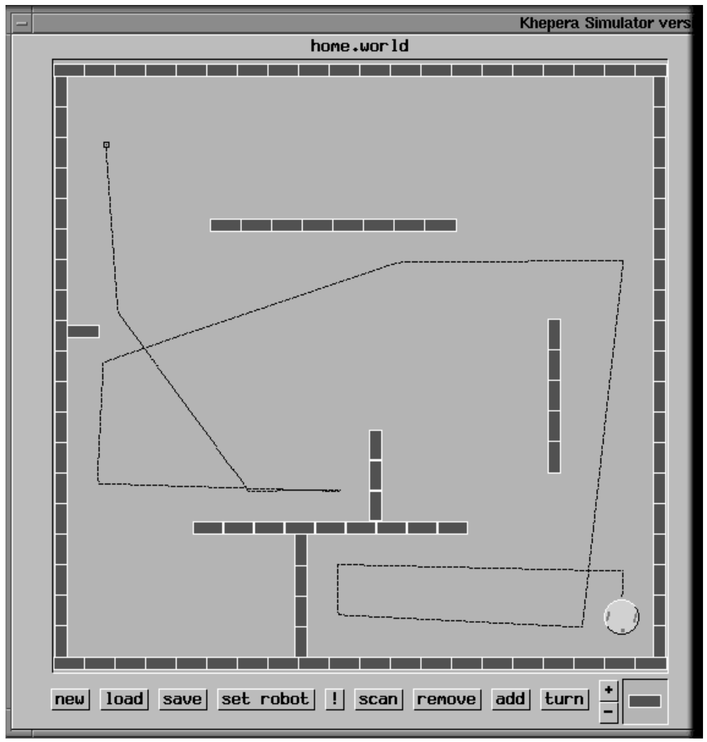
- **Reactive: Sensorial-Motor Integration**
  - **Able to Act**
  - **Able to Perceive the Environment**
    - **Able to React**

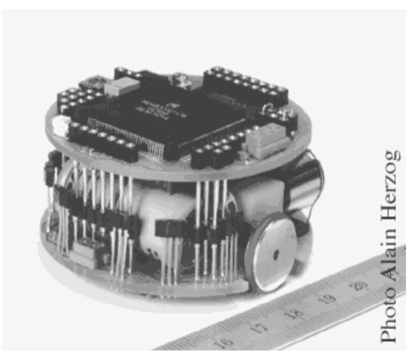


CONTROL: REACTIVE Architecture

- **Reactive: Sensorial-Motor Integration**







Sensorial-Motor: Perceive => Act

Reactive Control

IF S1 < Threshold *and*
S2 < Threshold *and*
S3 < Threshold *and*
S4 < Threshold
THEN Action (Go_Forward)

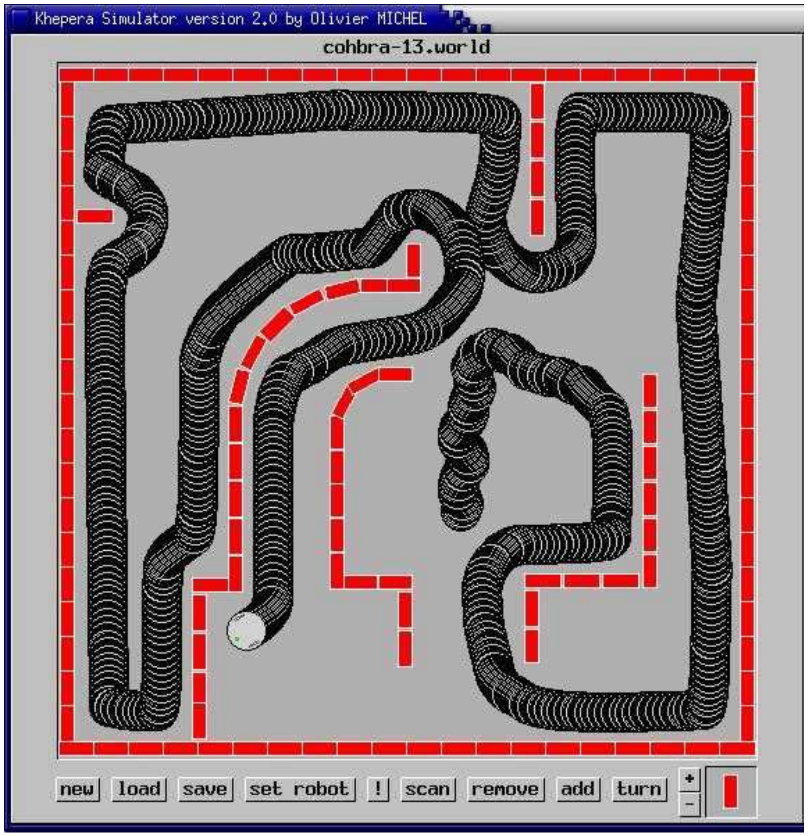
IF S1 < Threshold *and*
S2 < Threshold *and*
S3 > Threshold *and*
S4 > Threshold
THEN Action (Turn_Left)

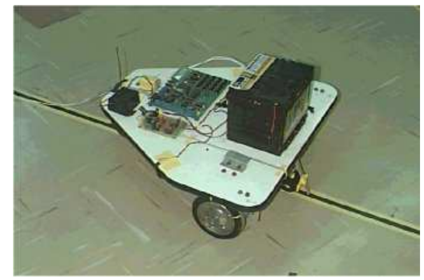
IF S2 > Threshold *and*
S3 > Threshold *and*
S2 > S3 *and*
S1 > S4
THEN Action (Turn_Right)

CONTROL: REACTIVE Architecture

- **Reactive: Sensorial-Motor Integration**

Reactive Control





Robotic Lawn Mowers
- Toro iMow
- **Husqvarna Auto Mower**
- **Automower Electrolux** →

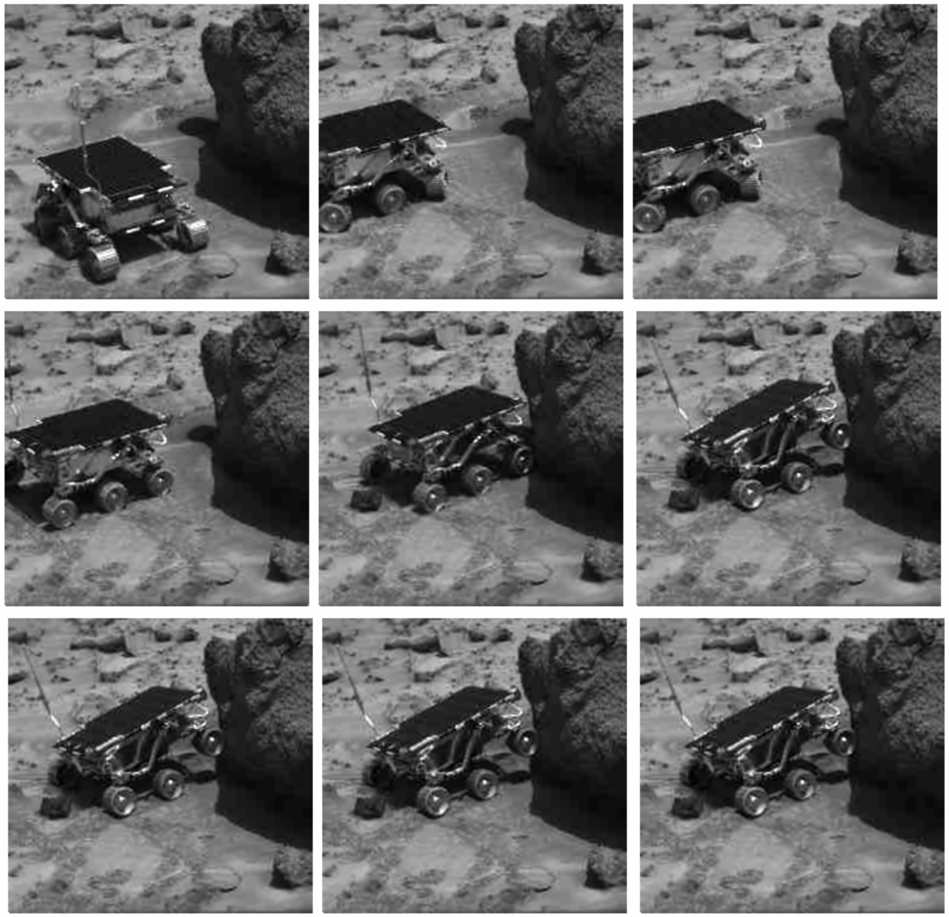
Electrolux Trilobite
Robotic Vacuum Cleaner ZA1

Sensorial-Motor: ***Avoid Obstacles, Wall Following, Wander***

http://www.onrobo.com/reviews/At_Home/Vacuum_Cleaners/

CONTROL: **REACTIVE** Architecture

- **Reactive: Sensorial-Motor Integration**



Reactive Control

Sensorial-Motor:

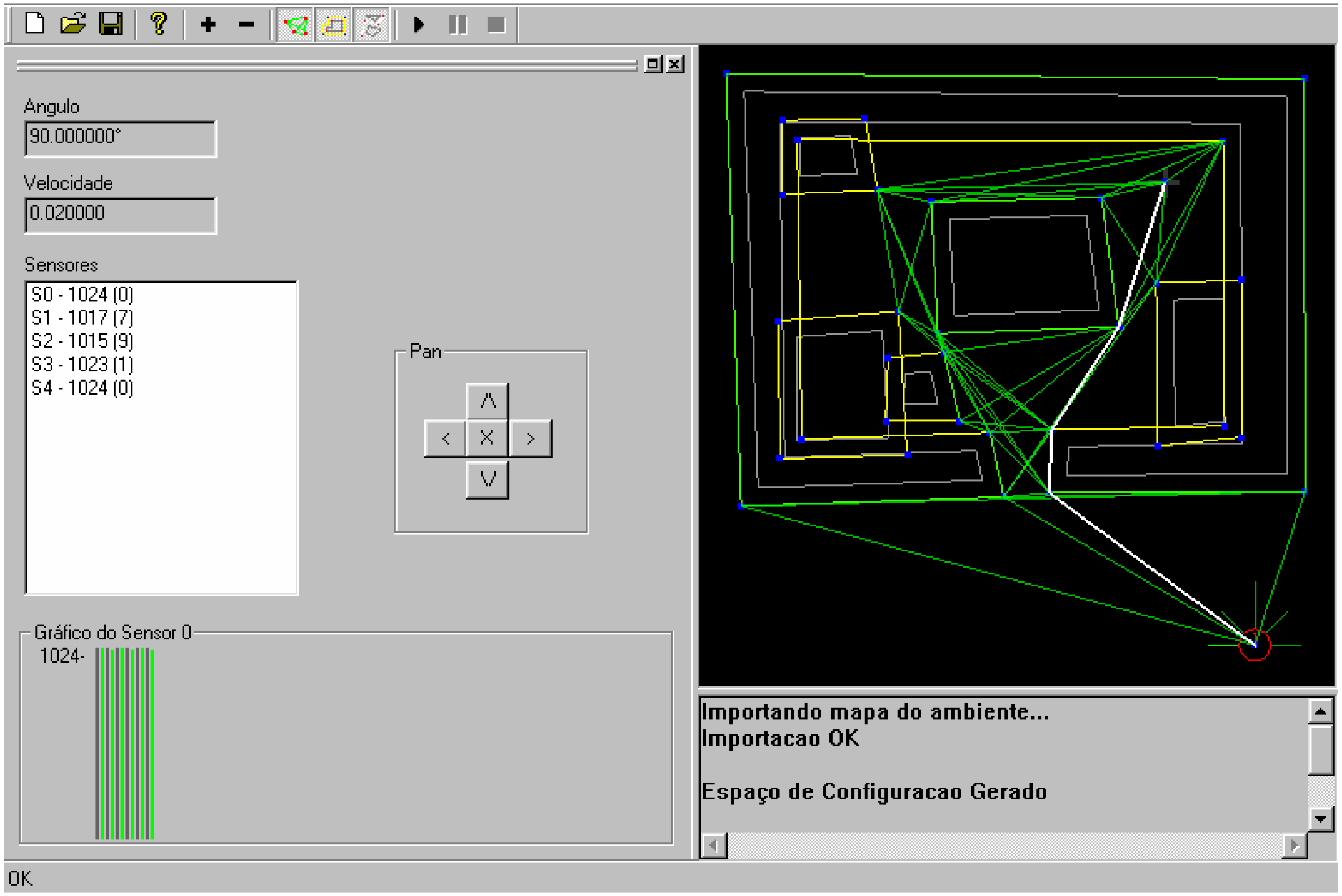
- ***Avoid Obstacles***
- ***Wall Following***
- ***Wander***

Simple behaviors…
Robustness? Complex tasks?

CONTROL: DELIBERATIVE Architectures

Deliberative Control

- **Deliberative: Planning + Action**



SIMROB (2D)

- **Map**
- **Configuration Space**
- **Visibility Graph**
- **Optimized Path (Dijkstra)**

Robotic Arm: Pre-defined paths

Deliberative Control

CONTROL: **DELIBERATIVE** Architectures

- **Deliberative: Planning + Action**

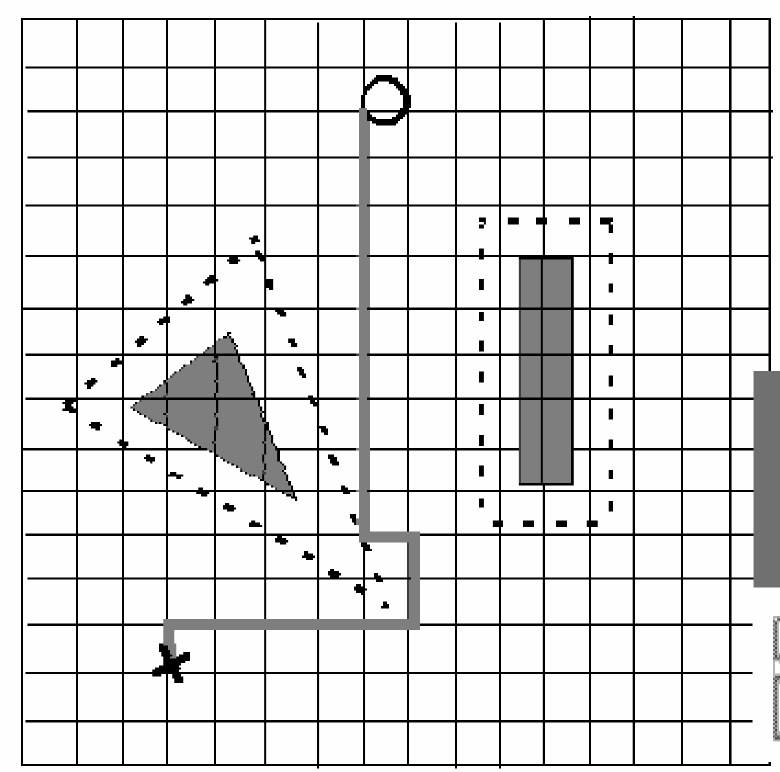


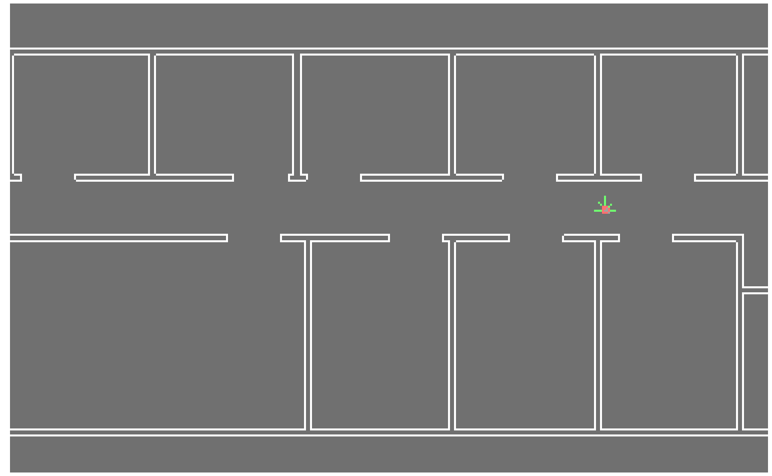
Figura 4.3 Navegação baseada em *Grid*

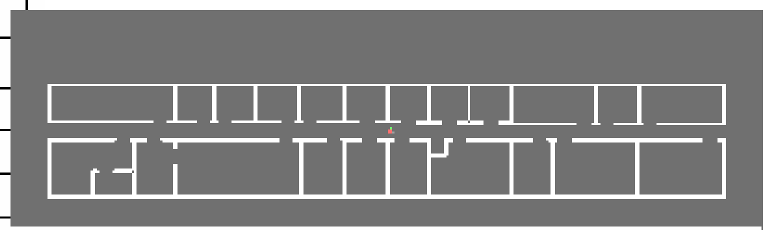
Tarefas Complexas...

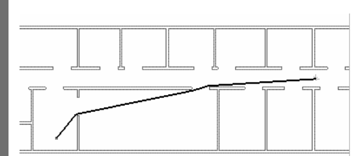
Robustez?

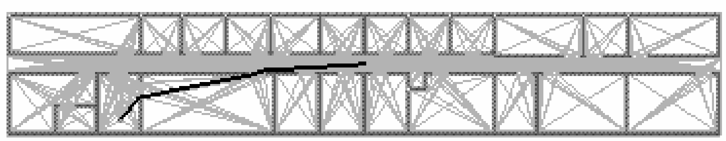
Imprevistos?

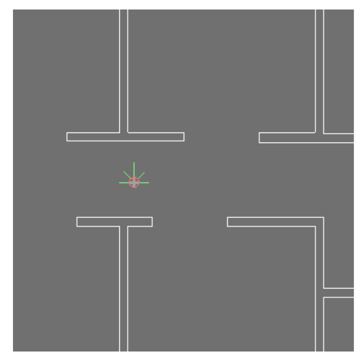
Ambiente pouco conhecido?











Geometric Map based Navigation:
Planning: Graph+Dijkstra, A*
Grid based Navigation:
Planning: A*

Hierarchical and Hybrid Control

CONTROL: HIERARCHICAL and HYBRID Architectures

Combining: Deliberative + Reactive

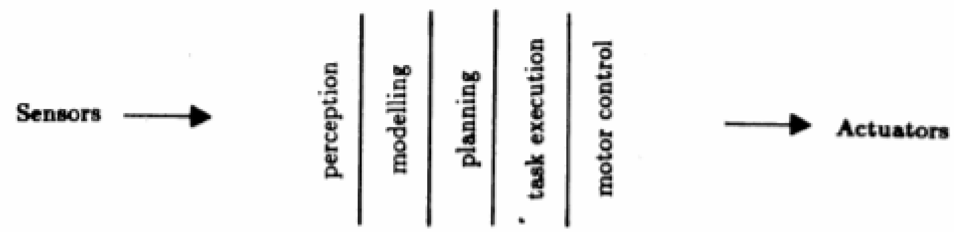


Figure 1. A traditional decomposition of a mobile robot control system into functional modules.

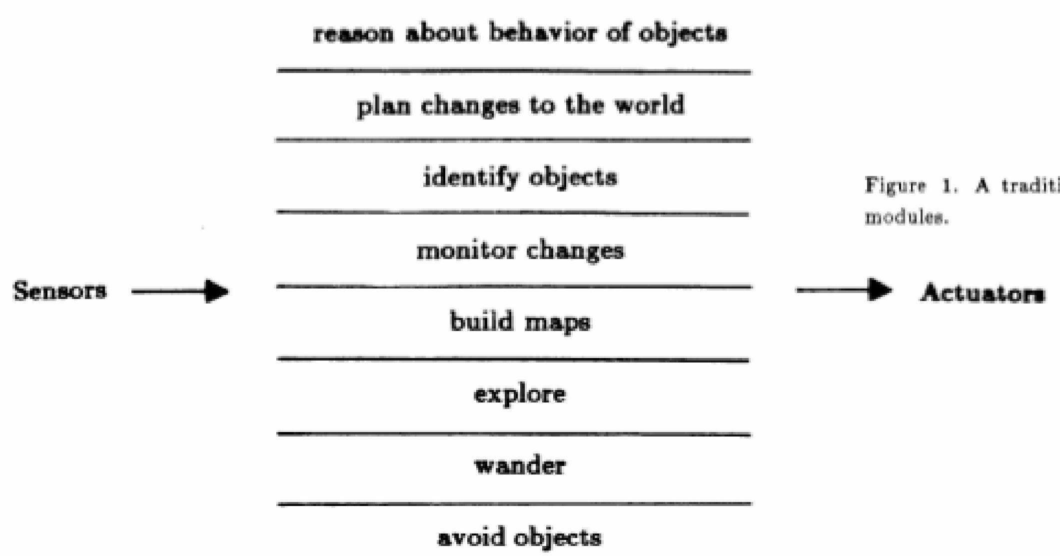


Figure 2. A decomposition of a mobile robot control system based on task achieving behaviors.

Hierarchical Control:

- **Control Layers**
- **Priorities**
- **Information Exchange**

Figures From:
Brooks, R. A.
MIT A.I. Memo 864
Sept. 1985

Brooks - Subsumption Architecture

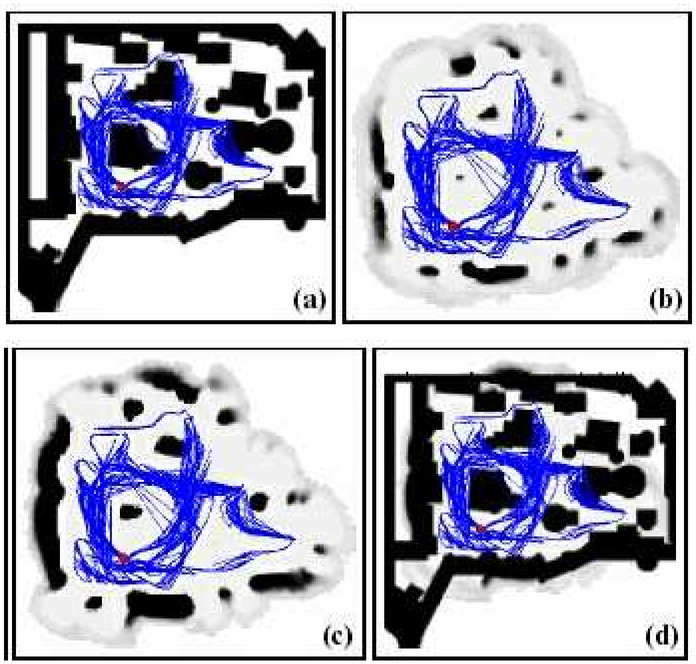
Hierarchical and Hybrid Control

CONTROL: HIERARCHICAL and HYBRID Architectures

Building the Environment Map:

SMPA - SENSE / MODEL / PLAN / ACT

SENSE



MODEL

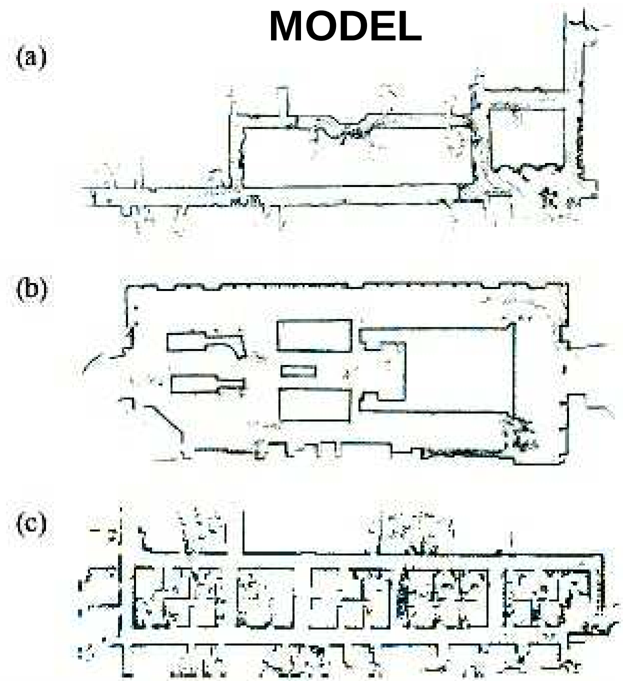


Figure 5: Maps generated in other large-scale environments of sizes (a) 75m, (b) 45m, and (c) 50m. In some of these runs, the cumulative odometric error exceeds 30 meters and 90 degrees.

PLAN:
AStar
Dijkstra

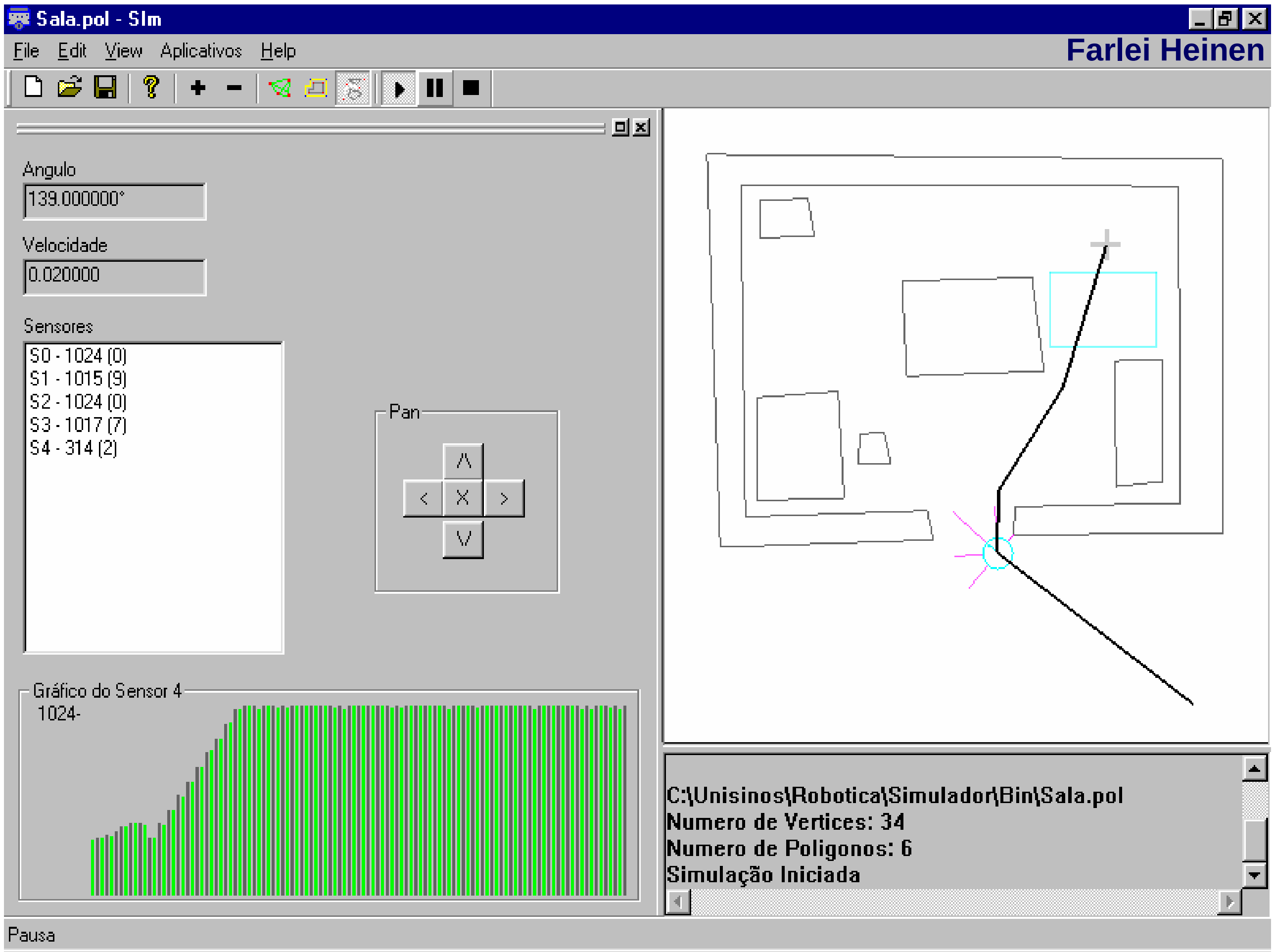
ACT !

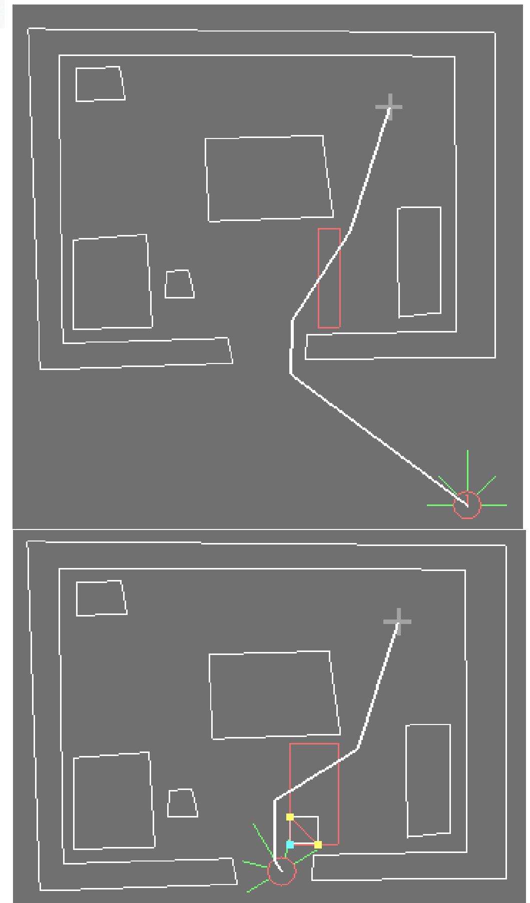
Fig. 9. Integrating multiple maps: (a) CAD map of the museum ($21 \times 20 m^2$) modeling only the static obstacles, (b) laser map, (c) sonar map, and (d) the integrated map used for path planning.

Sebastian Thrun / CMU

CONTROL: Simple HYBRID Architectures

Hybrid Control

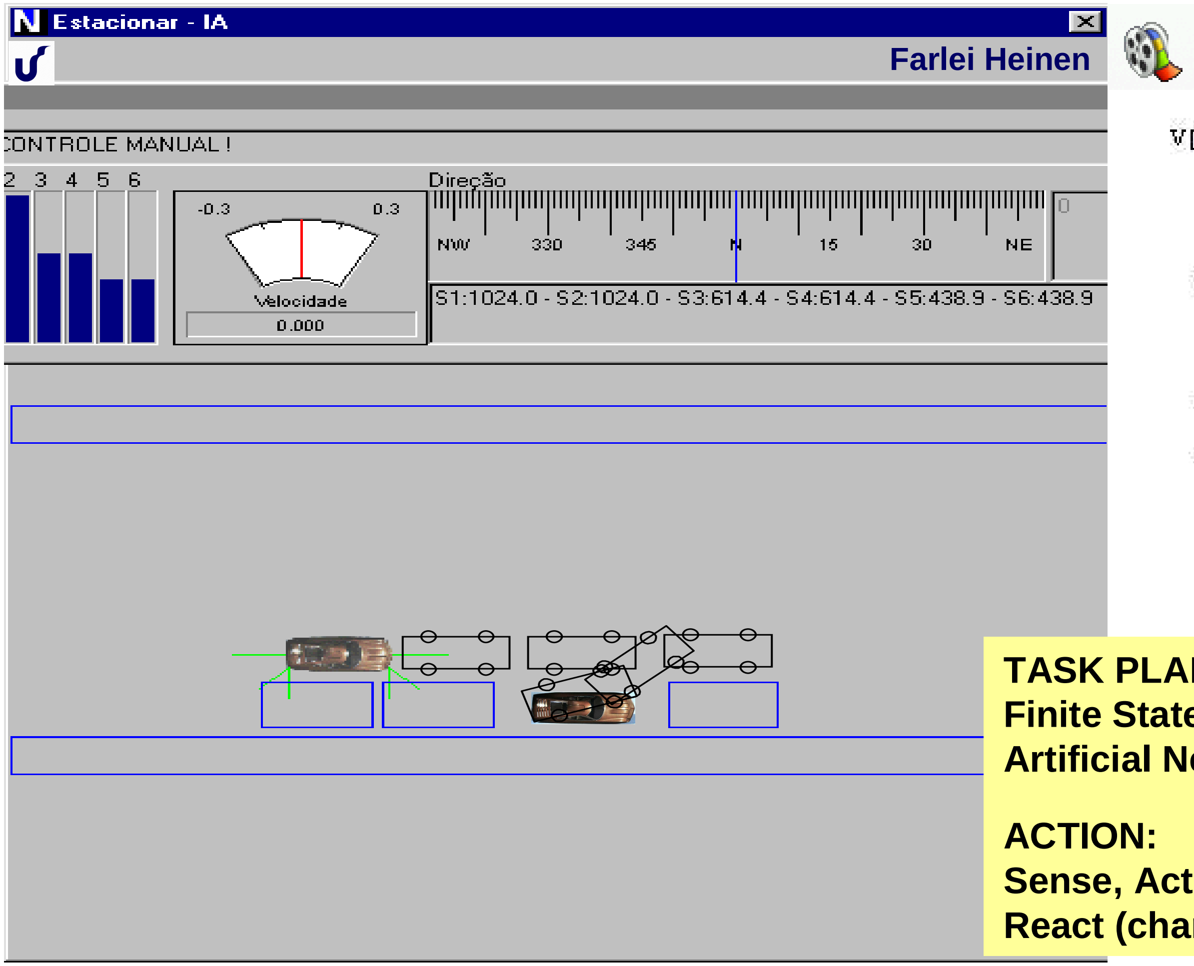


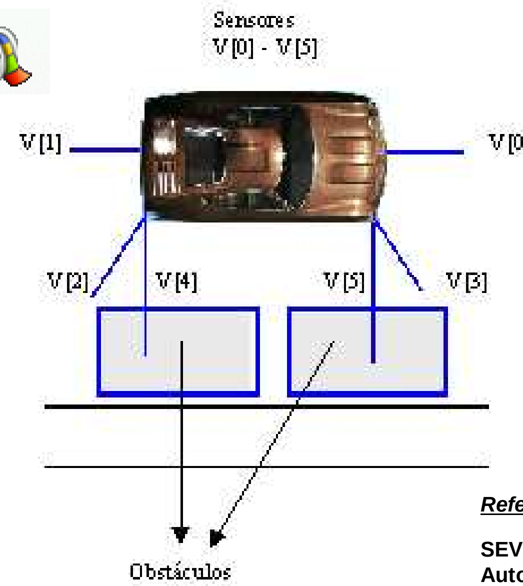


PLAN: Dijkstra
ACT & ReACT

Hybrid Control

CONTROL: Simple HYBRID Architectures





TASK PLANNING & CONTROL:
Finite State Automata (FSA)
Artificial Neural Net (ANN)

ACTION:
Sense, Act
React (change state)

References:

SEVA2D / SEVA3D Autonomous Vehicle Parking

SEVA-A (Automaton)
Farlei Heinen

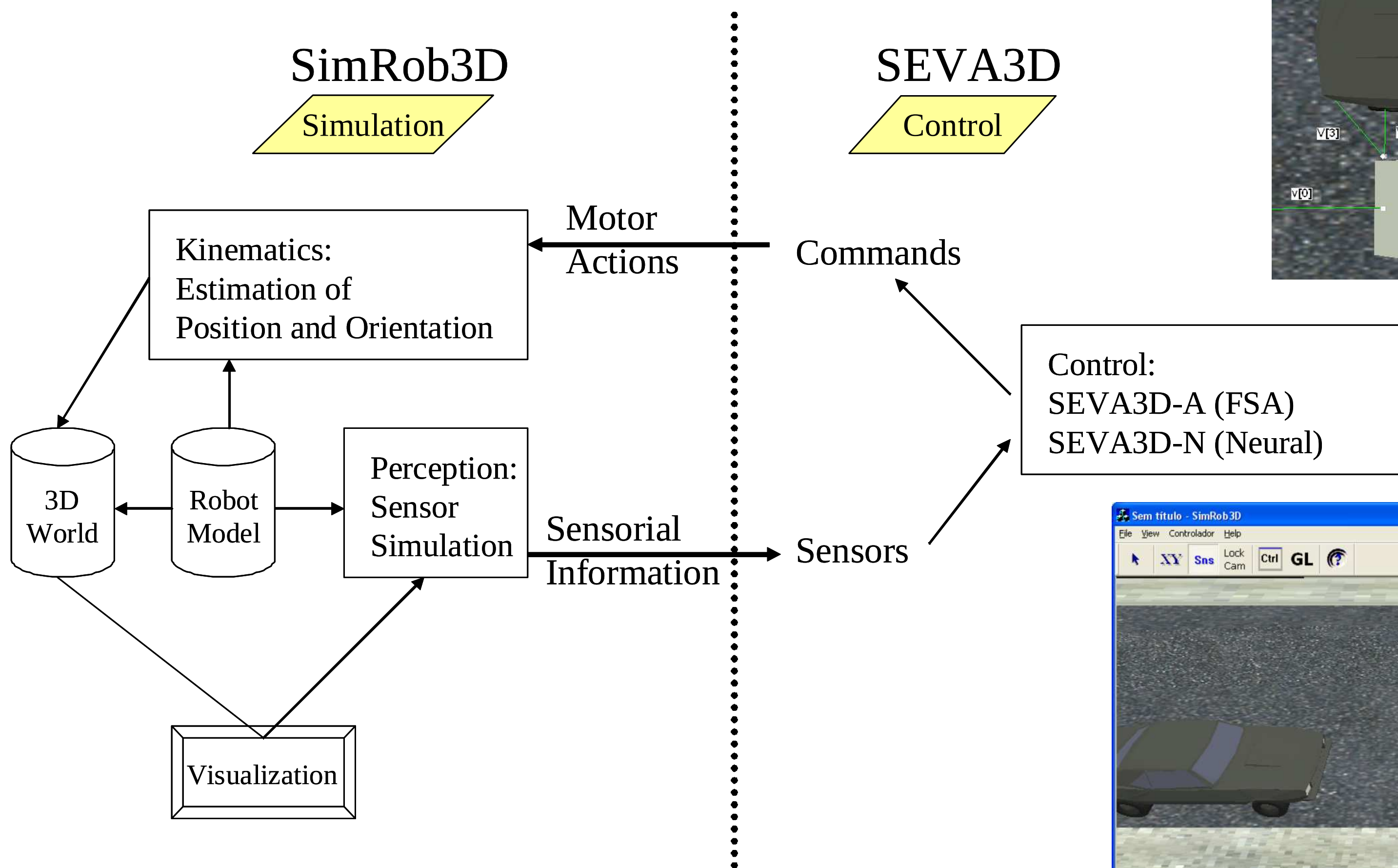
SEVA-N (Neural)
Farlei Heinen
Fernando Osório
Luciane Fortes
Milton Heinen

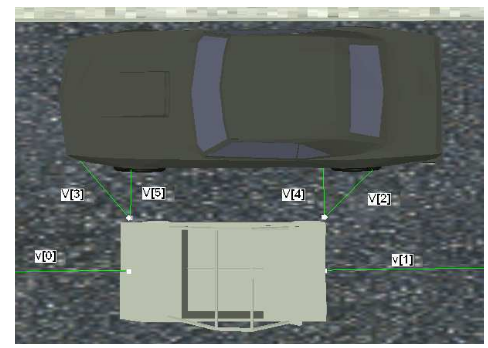
Publications:

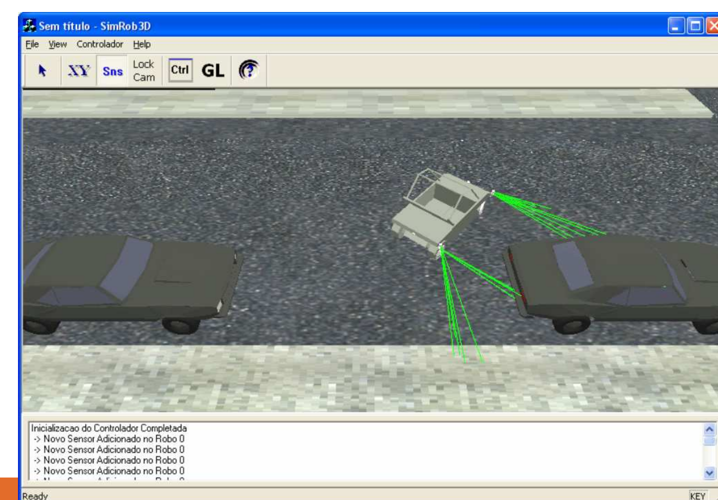
SBRN 2002
WCCI 2006

Hybrid Control

CONTROL: Simple HYBRID Architectures







CONTROL: Computational Architectures

- Sensorial Models
- Kinematics Models
- Robotic Control:
  * Reactive
  * Deliberative
  * Hierarchical
  * Hybrid
- Environment Maps
  * Building Maps
  * Path Planning
  * SMPA - *Sense Model Plan Act*
- Problems:
  * Complex tasks
  * Avoid Obstacles: Static / Mobile - **Unexpected obstacles**
  * Robot actual position estimation - **Where am I ?**

Intelligent Autonomous Vehicles
Control System
Task Execution

PROBLEMS:

*** Avoid Obstacles**

- **Known Obstacles**
- **Unknown Obstacles (static / no movement)**
- **Unknown Obstacles (dynamic / moving objects)**

*** Positioning**

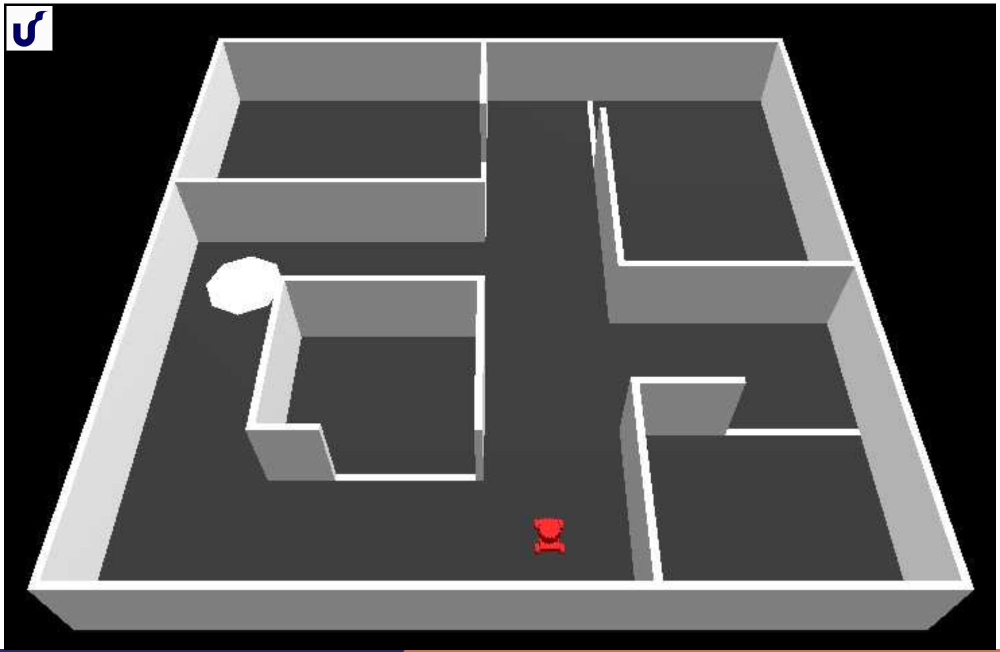
- **How to determine the exact actual position of the robot ?**
- **How to maintain the control of exact position after displacement ?**
- **Error and Imprecision: Move forward / Rotate**

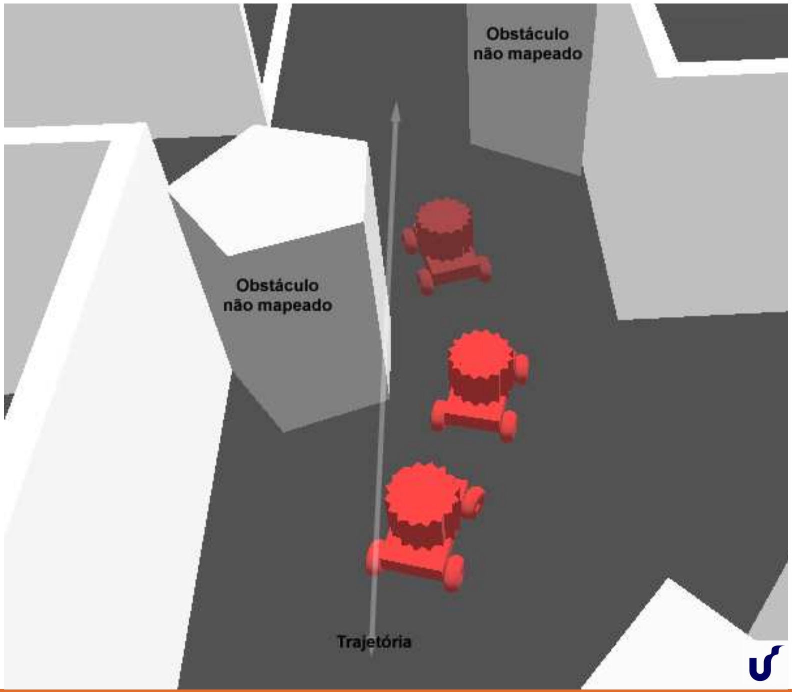
Intelligent Autonomous Vehicles Control System Task Execution

PROBLEMS:

*** Avoid Obstacles**

- **- Known Obstacles**
- **- Unknown Obstacles (static / no movement)**
- **- Unknown Obstacles (dynamic / moving objects)**





Intelligent Autonomous Vehicles
Control System
Task Execution

PROBLEMS:

*** Positioning**

- How to determine the exact actual position of the robot ?
- How to maintain the control of exact position after displacement ?
- Error and Imprecision: Move forward / Rotate

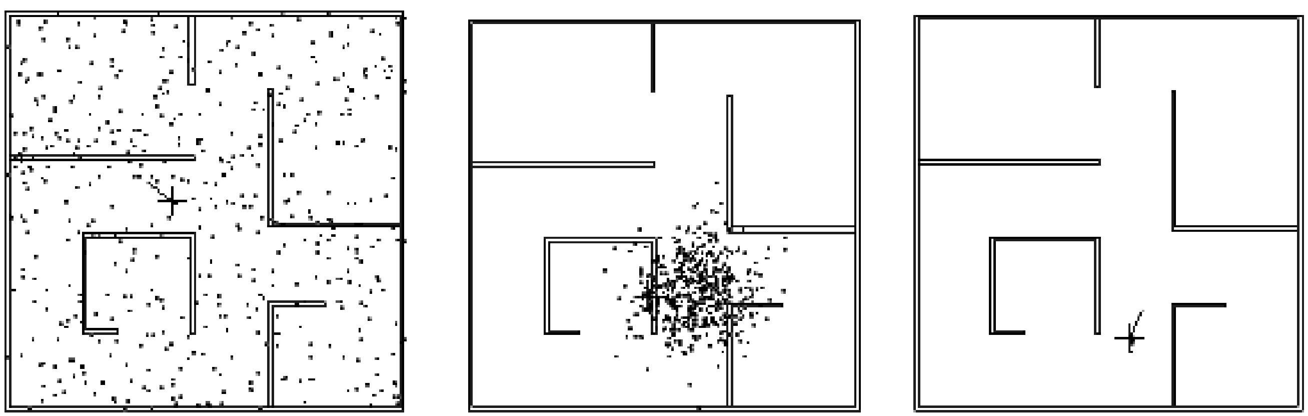
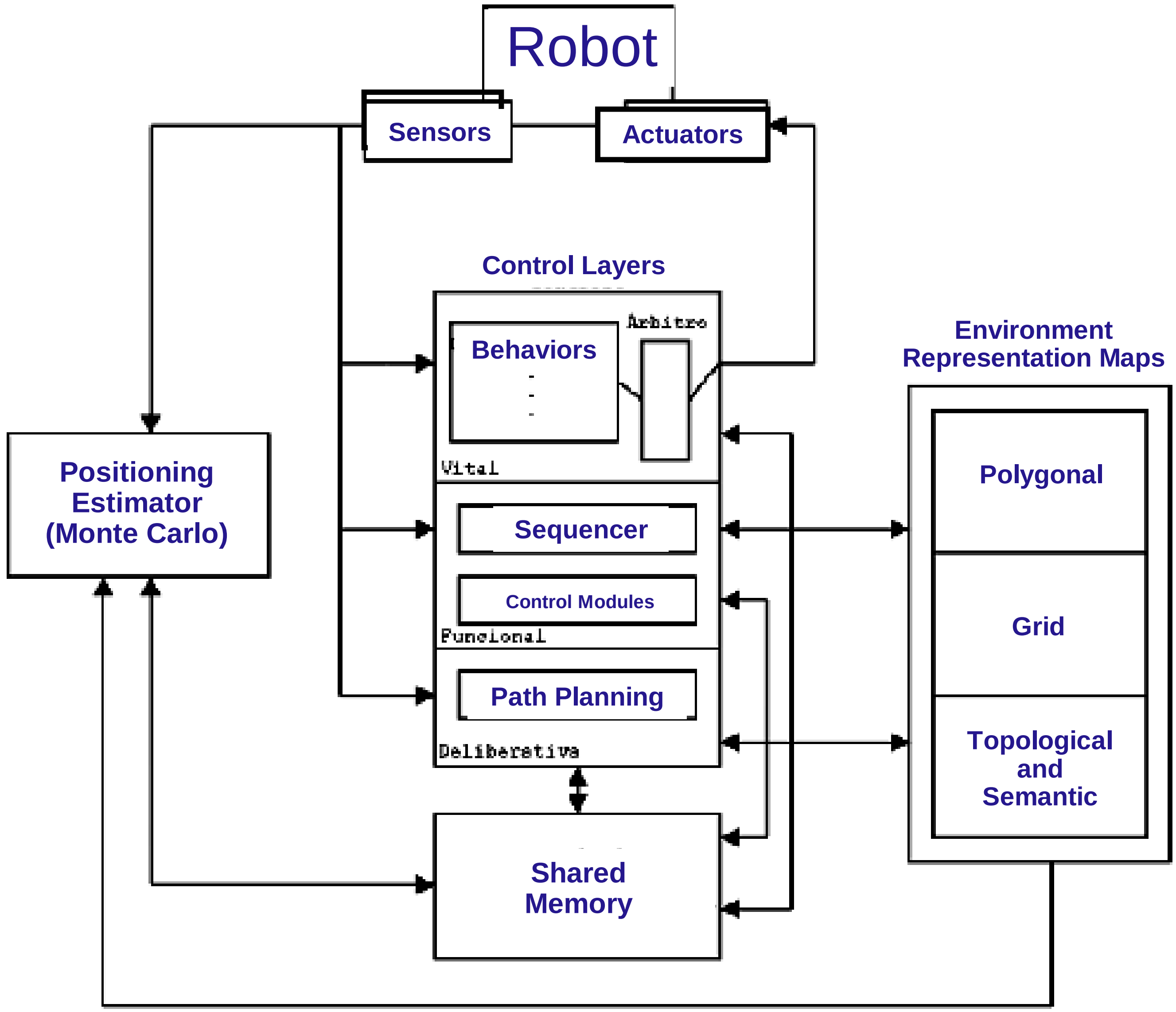


Fig. 2. Seqüência de imagens mostrando a evolução da distribuição das partículas durante a localização do robô móvel utilizando o algoritmo Monte Carlo.

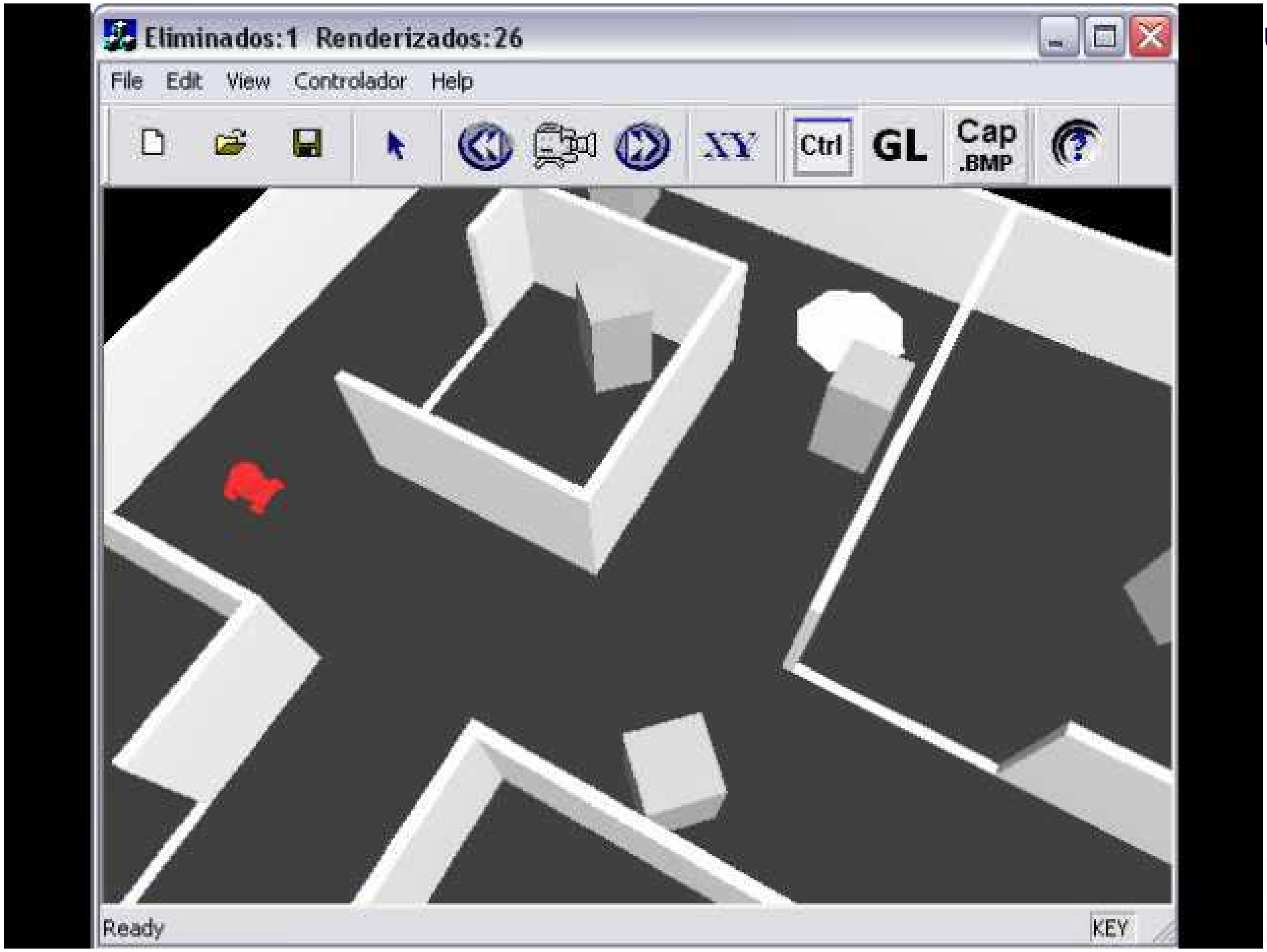
Robust Hybrid Control – COHBRA / HyCAR [SimRob3D]



COHBRA – Controle Híbrido de Robôs Autônomos
HyCAR - Hybrid Control for Autonomous Robots

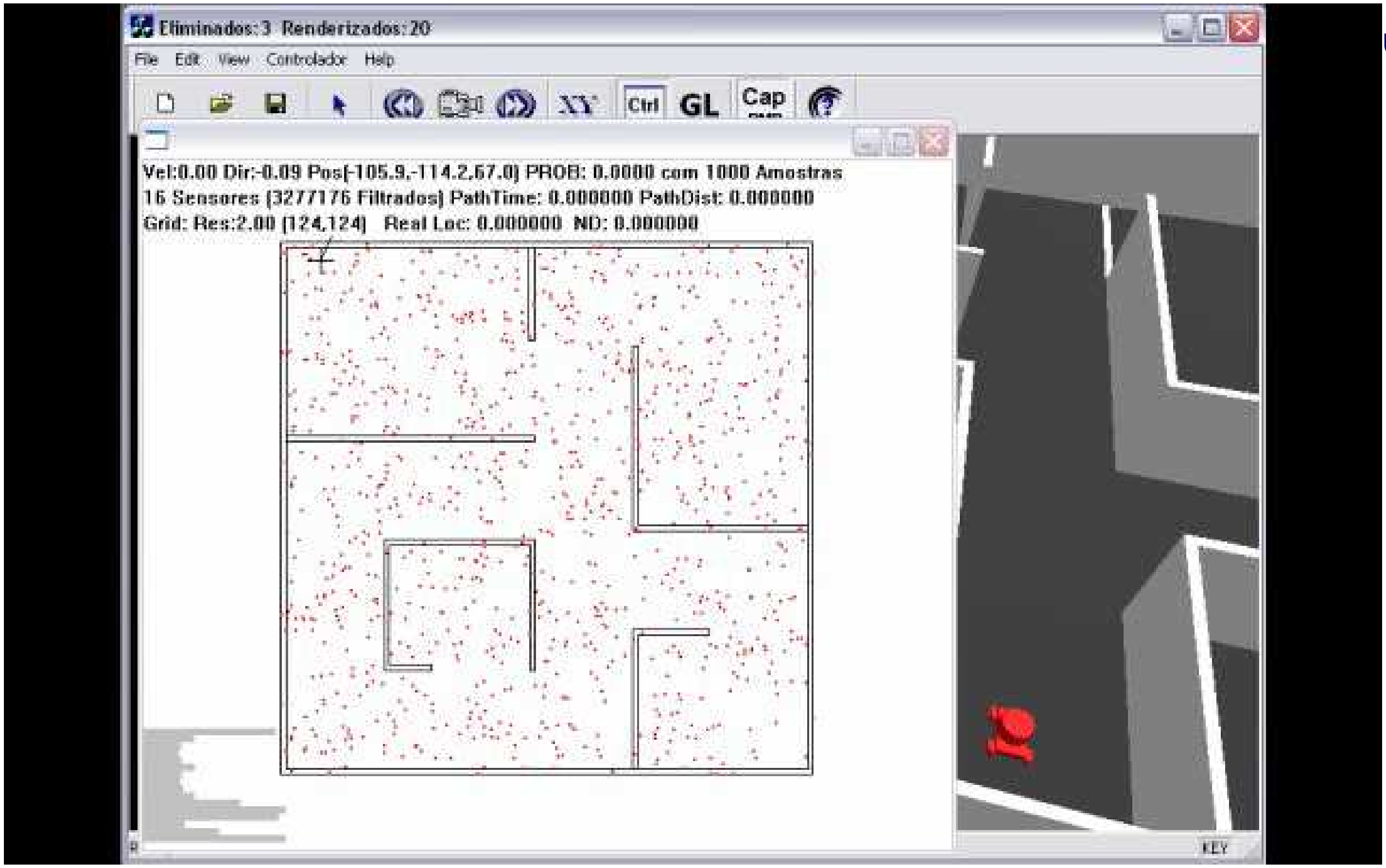
Robust Hybrid Control – COHBRA / HyCAR [SimRob3D]

Simulation using SimRob3D



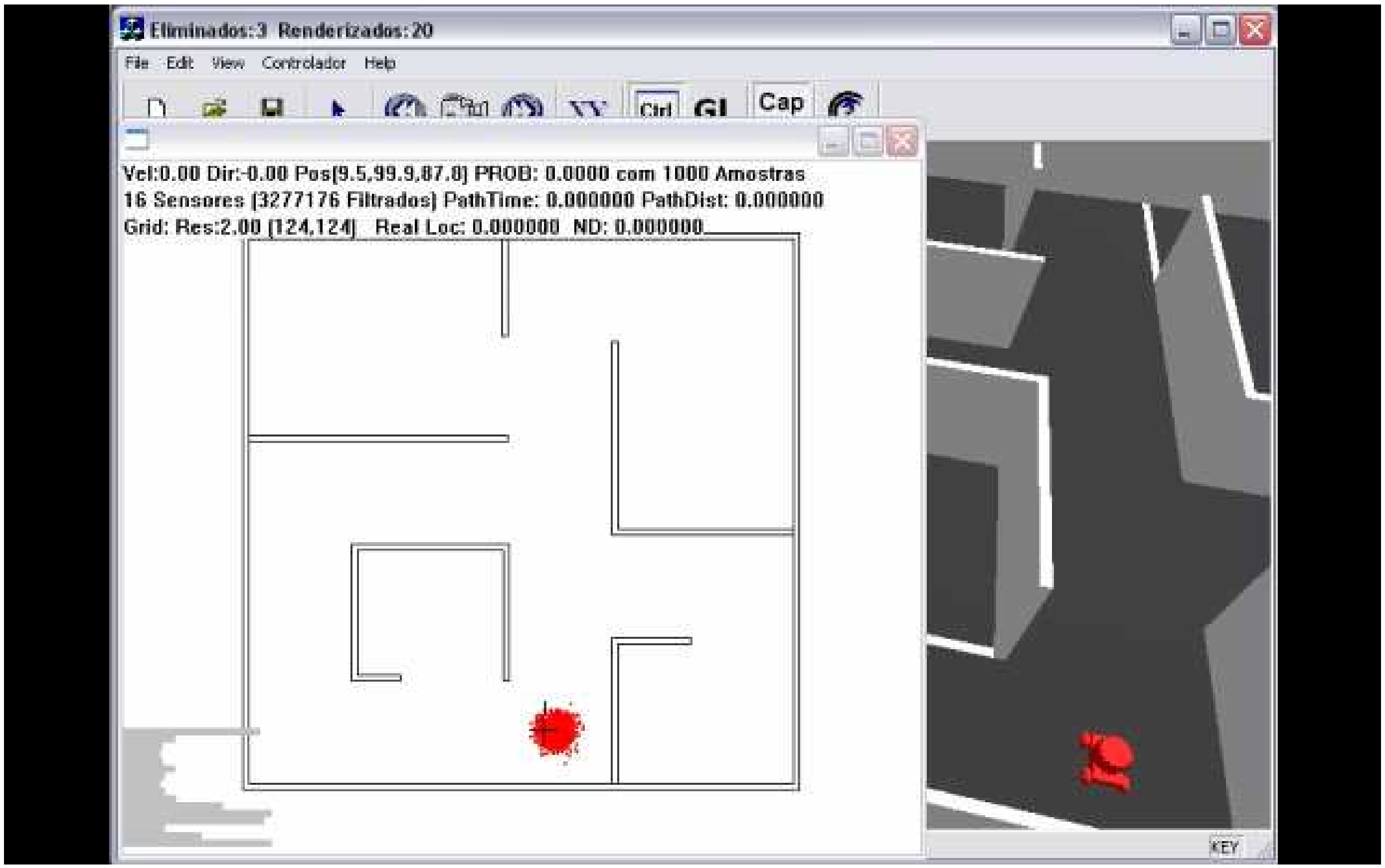
Robust Hybrid Control – COHBRA / HyCAR [SimRob3D]

Simulation using a static environment
Position estimation based on Monte Carlo Localization Method



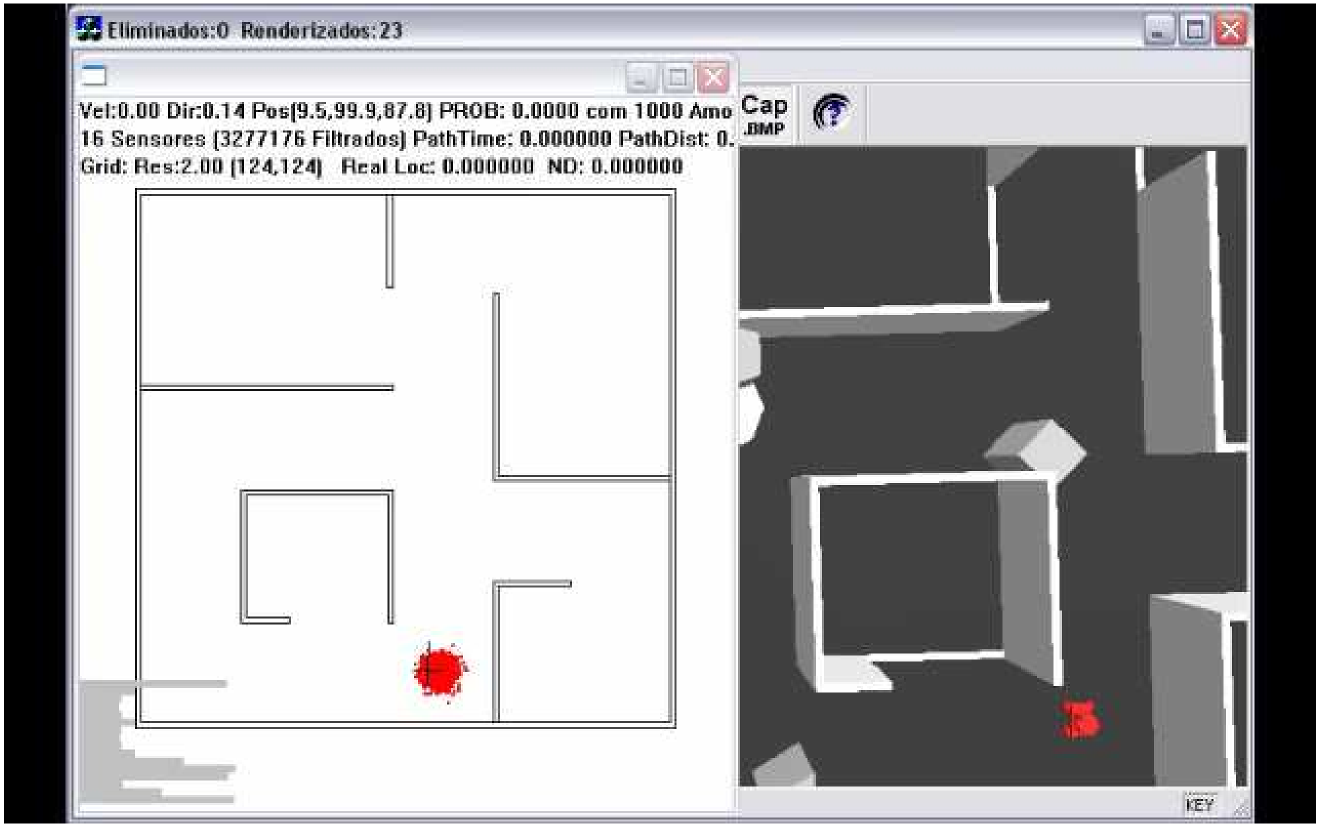
Robust Hybrid Control – COHBRA / HyCAR [SimRob3D]

Simulation using a static environment
Position estimation based on Monte Carlo Localization Method



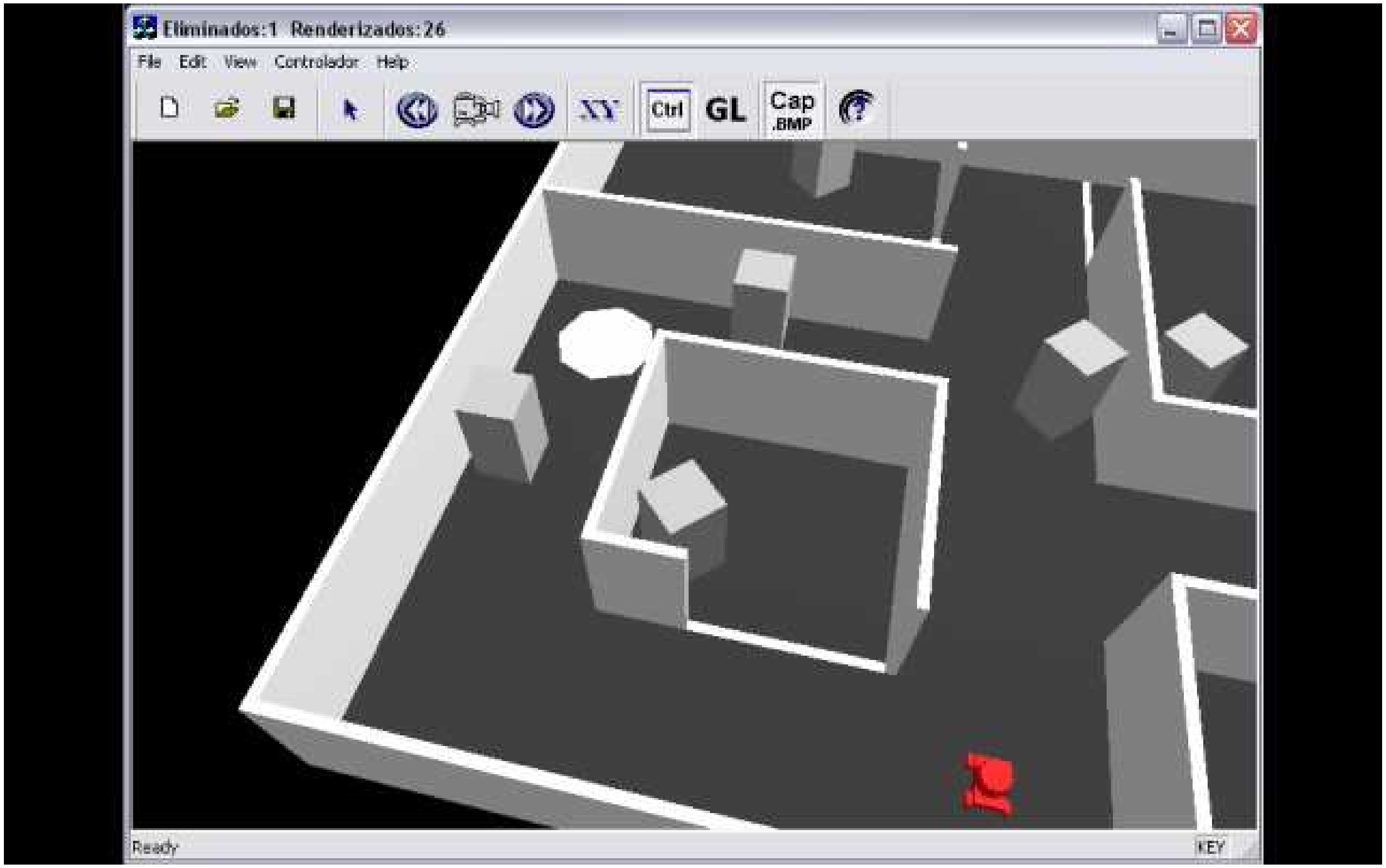
Robust Hybrid Control – COHBRA / HyCAR [SimRob3D]

Simulation using a static environment
Environment was changed related to the original map
Internal robot representation is different from actual world configuration



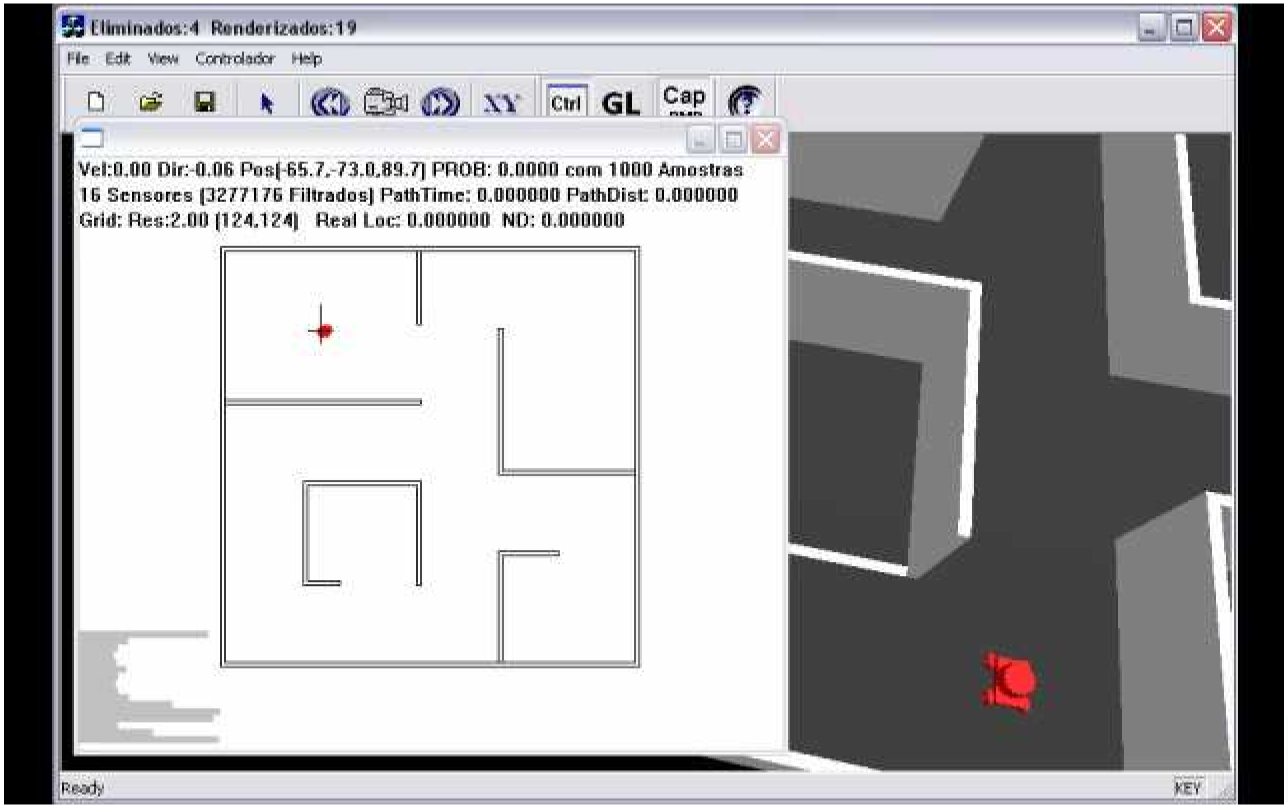
Robust Hybrid Control – COHBRA / HyCAR [SimRob3D]

Simulation using a dynamic environment (mobile obstacles)



Robust Hybrid Control – COHBRA / HyCAR [SimRob3D]

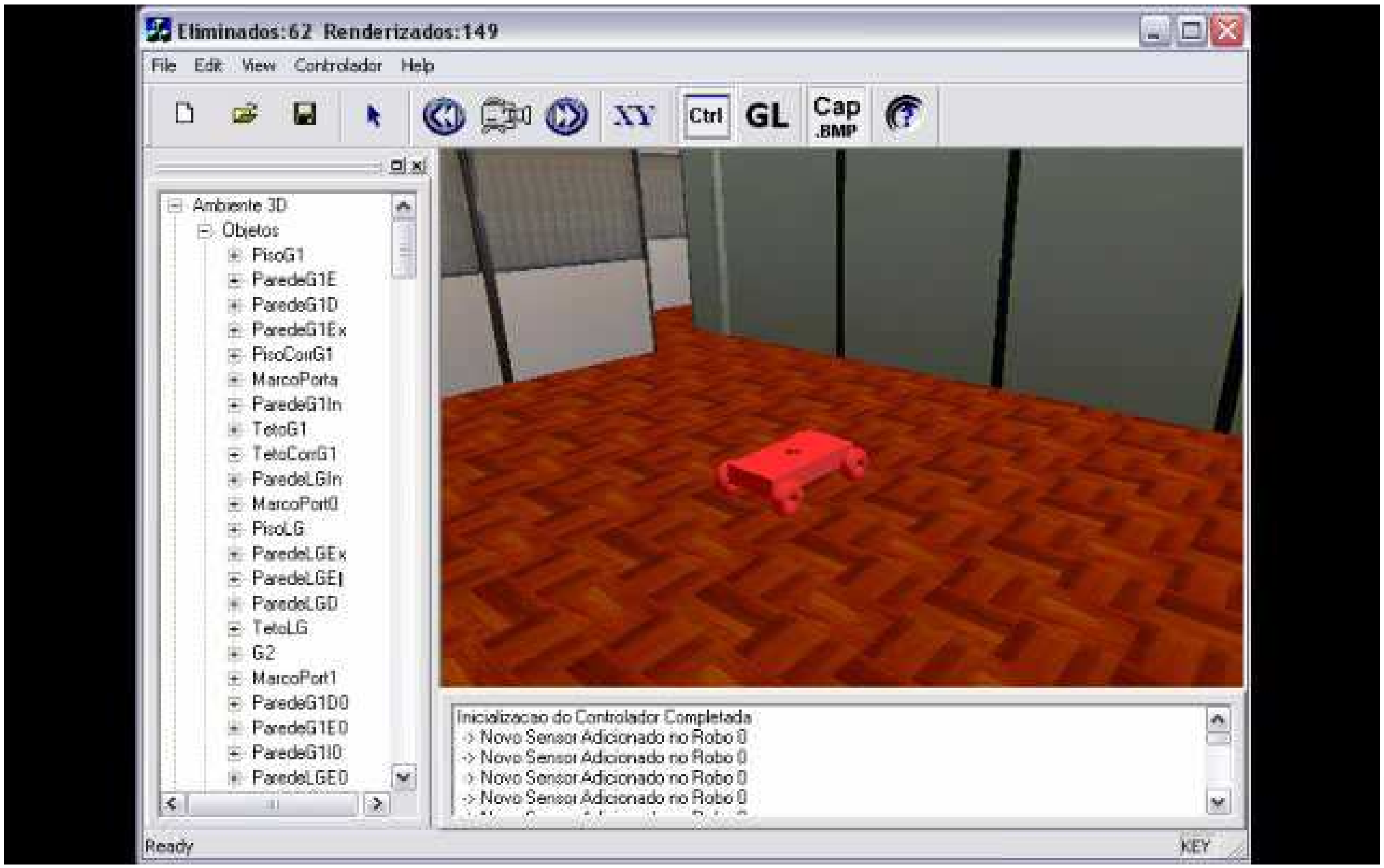
Position estimation based on Monte Carlo Method:
Robot was moved, starting in a new and unknown position



Robust Hybrid Control – COHBRA / HyCAR [SimRob3D]

Virtual Environment: 3D Realistic Environment

SimRob3D Simulation Tool



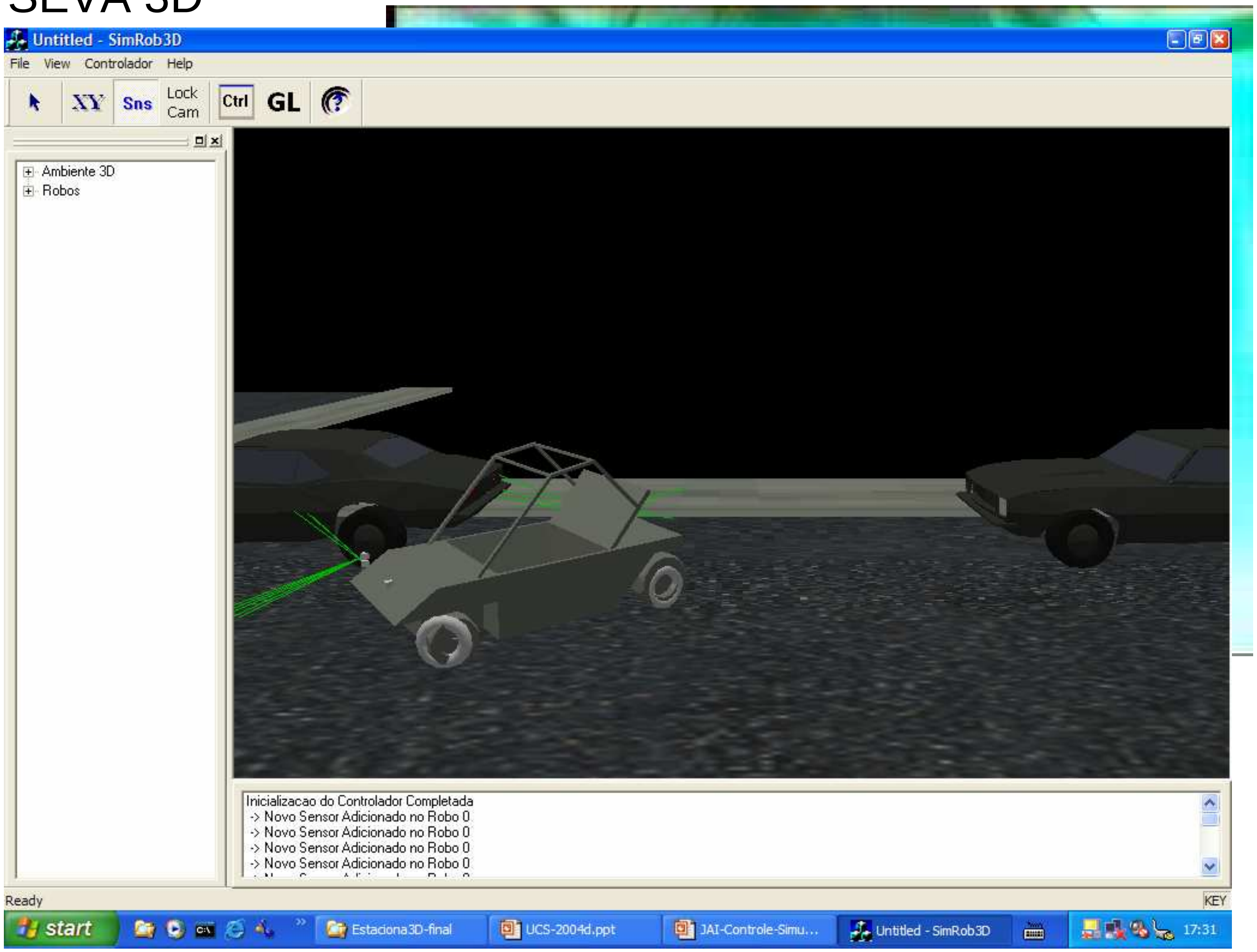
Robust Hybrid Control – COHBRA / HyCAR [SimRob3D]

Virtual Environment: 3D Realistic Environment

SimRob3D Simulation Tool

Robust Hybrid Control – COHBRA / HyCAR [SimRob3D]

SEVA 3D



SimRob3D Simulation Tool

Intelligent Autonomous Robots and Vehicles
<< Intelligence >>

* Task and Actions Planning

* Ability to Perceive the Environment

* Ability to Decide

* Ability to Act

* High Level Tasks Planning

* Reaction: Sensorial-Motor Integration

* Estimate Actual and Future States:
  Environment + Behavior = Interaction

* Adaptation and Learning

* Robustness: Unexpected Situations

Next steps...

Intelligent Autonomous Robots and Vehicles
<< Intelligence >>

* Task and Actions Planning

* Ability to Perceive the Environment

* Ability to Decide

* Ability to Act

* High Level Tasks Planning

* Reaction: Sensorial-Motor Integration

* Estimate Actual and Future States:
  Environment + Behavior = Interaction

* Adaptation and Learning

* Robustness: Unexpected Situations

Next steps...

DARPA Challenge - Desert (2004, 2005)

DARPA Challenge - Urban (2007)

Intelligent Autonomous Robots and Vehicles
<< Intelligence >>

* Task and Actions Planning
* Ability to Perceive the Environment
* Ability to Decide
* Ability to Act
* High Level Tasks Planning
* Reaction: Sensorial-Motor Integration
* Estimate Actual and Future States:
  Environment + Behavior = Interaction
* Adaptation and Learning
* Robustness: Unexpected Situations

Next steps...

Computational Vision

Computational Vision

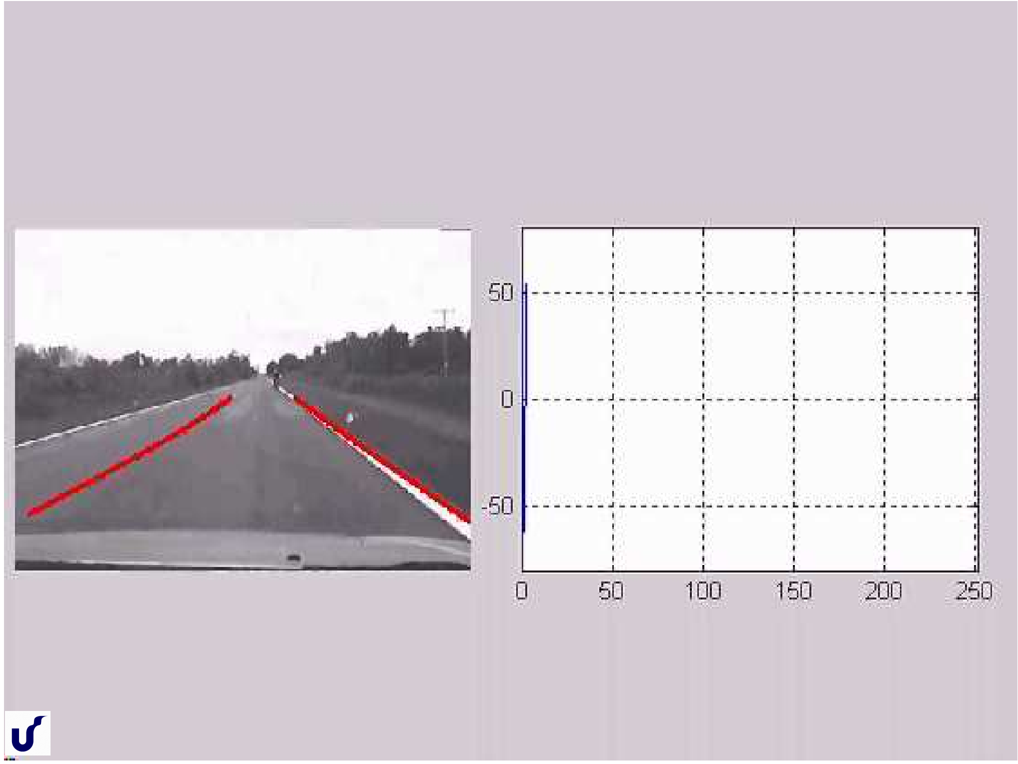
- **Path following:**
  - **Follow Me, Lane Follow**
- **Avoid danger situations: going out of the track**
  - **Lane Detection**
- **Obstacle detection: pedestrians, cars, etc**
- **Traffic signs detection and recognition**
- **Visual Navigation (Based on Images)**

Intelligent Autonomous Robots and Vehicles
<< Intelligence >>

Computational Vision

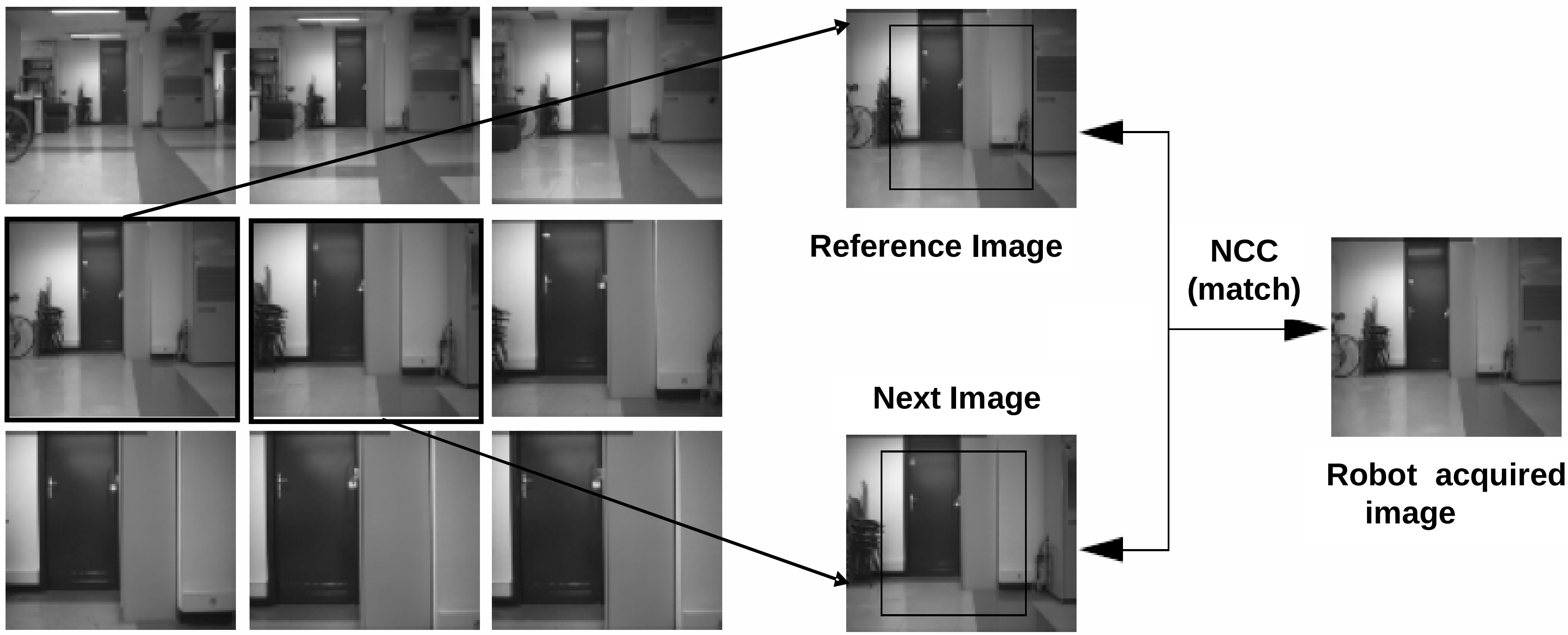
Lane Follow
Lane Departure Detection

Follow Me



Visual Navigation

Image Database:
Path defined by a sequence of image

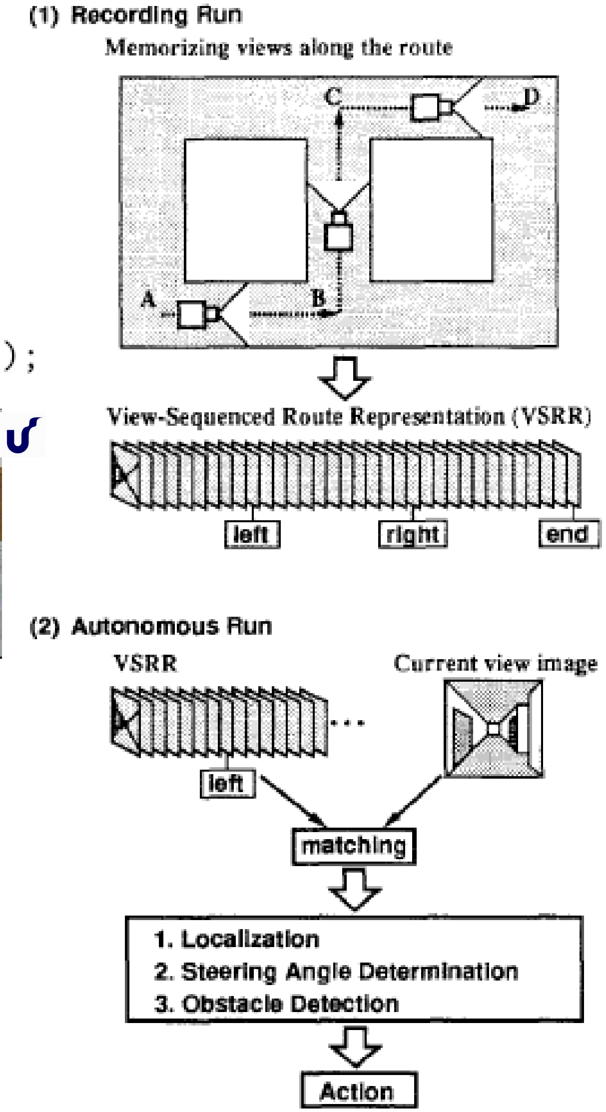


Navigation based on Monochromatic Images [Jones et al. 1997]

Algorithm: NCC – Normalized Cross-Correlation

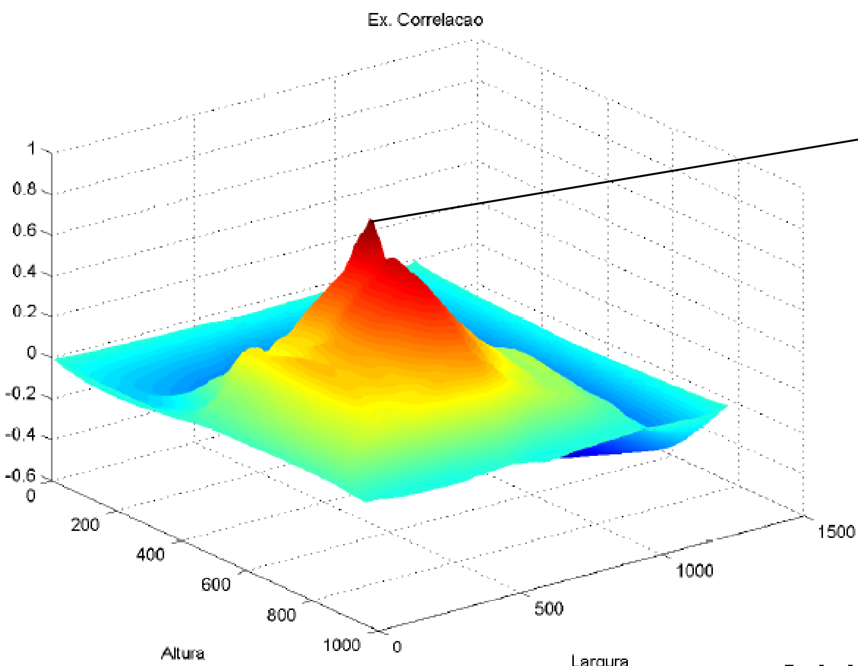
Visual Navigation

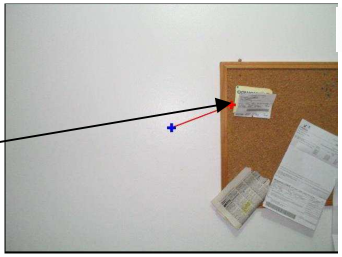
[Matsumoto et al. 1996]



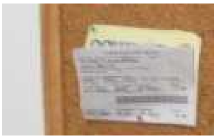
Matlab Code

```
>> img1 = imread('ir\b02ir08.jpg');
>> img2 = imread('icr\b02icr08.jpg');
>> ncc = normxcorr2( img2(:,:,1), img1(:,:,1) );
>> figure,surf(ncc),shading interp,axis ij,view(3);
>> title('Ex. Correlacao'),ylabel('Altura'),xlabel('Largura');
```





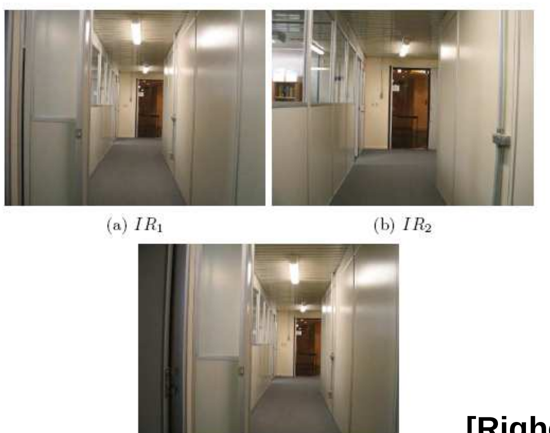
IR:
Reference Image



ICR:
Image Captured by the robot

[Righes 2004, 2005]

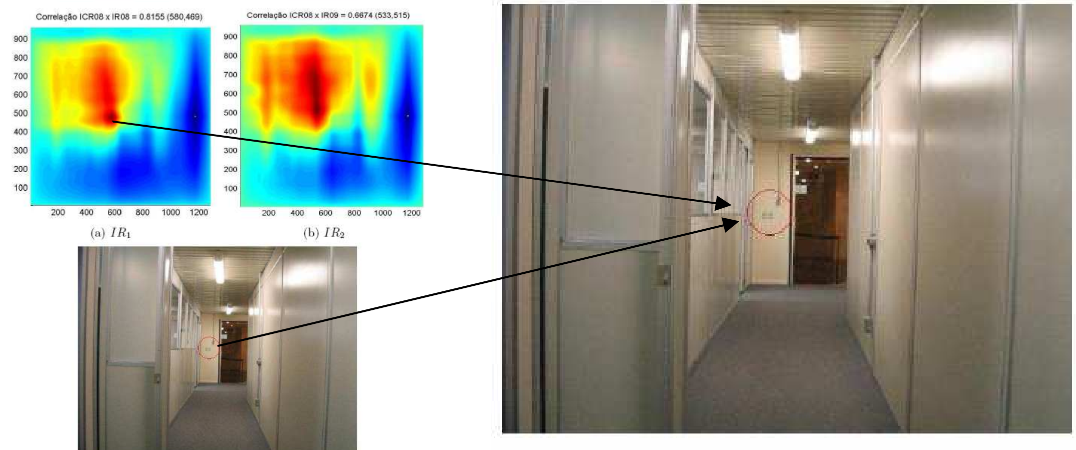
Visual Navigation



(a) IR_1 (b) IR_2

(c) ICR

[Righes 04]



(a) IR_1 (b) IR_2

(c) Região de maior correlação

(c) Região de maior correlação

Visual Navigation

Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks

Stephen Se,
David Lowe,
Jim Little
(UBC, CA)

Algorithm:
SIFT

Reference
Int. Journal of Robotics Research
Vol. 21, No. 8, August 2002,
pp. 735-758,

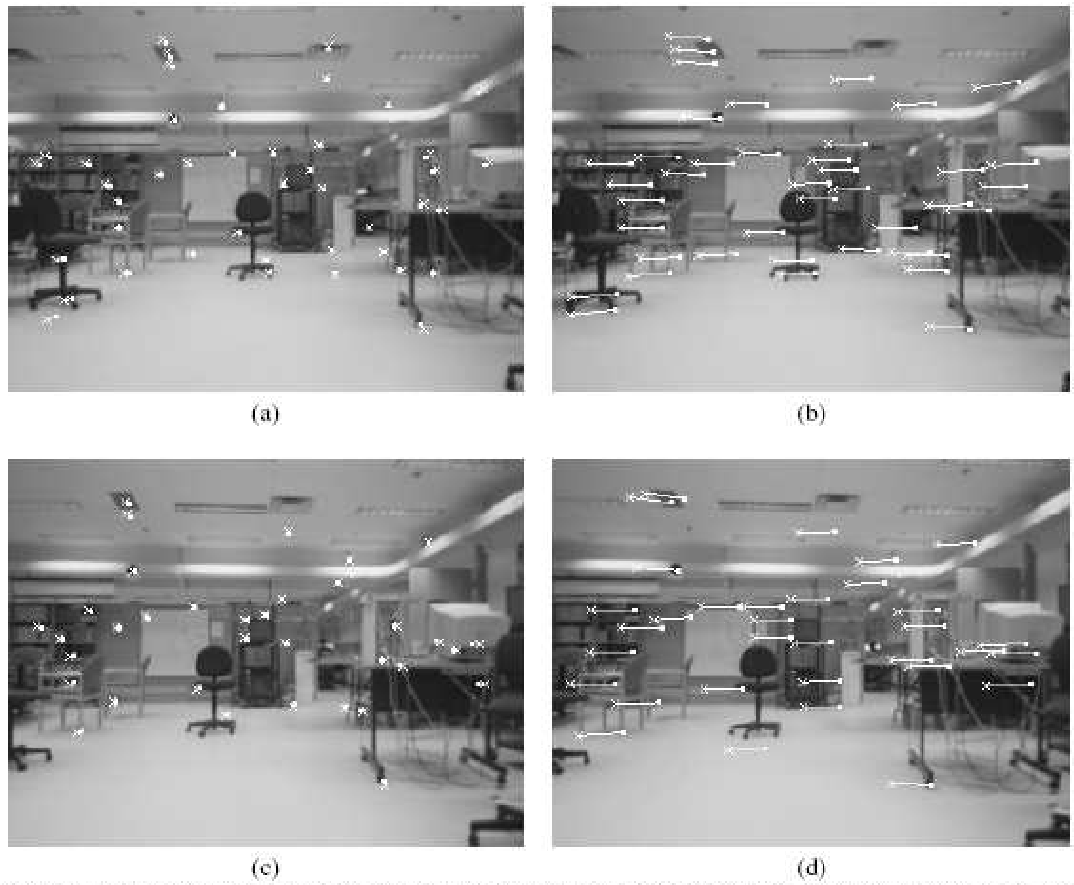
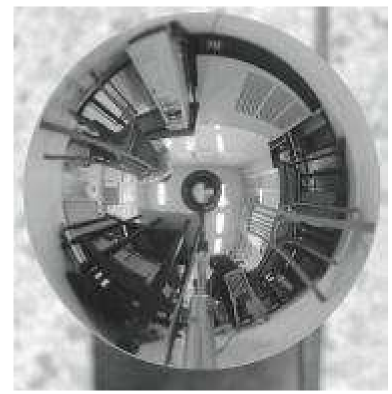


Fig. 3. The SIFT feature matches between consecutive frames: (a) between Figures 2(a) and (b) for a 10 cm forward movement; (b) between Figures 2(b) and (c) for a 5° clockwise rotation; (c) between Figures 2(c) and (d) for a 10 cm forward movement; (d) between Figures 2(d) and (e) for a 5° clockwise rotation.

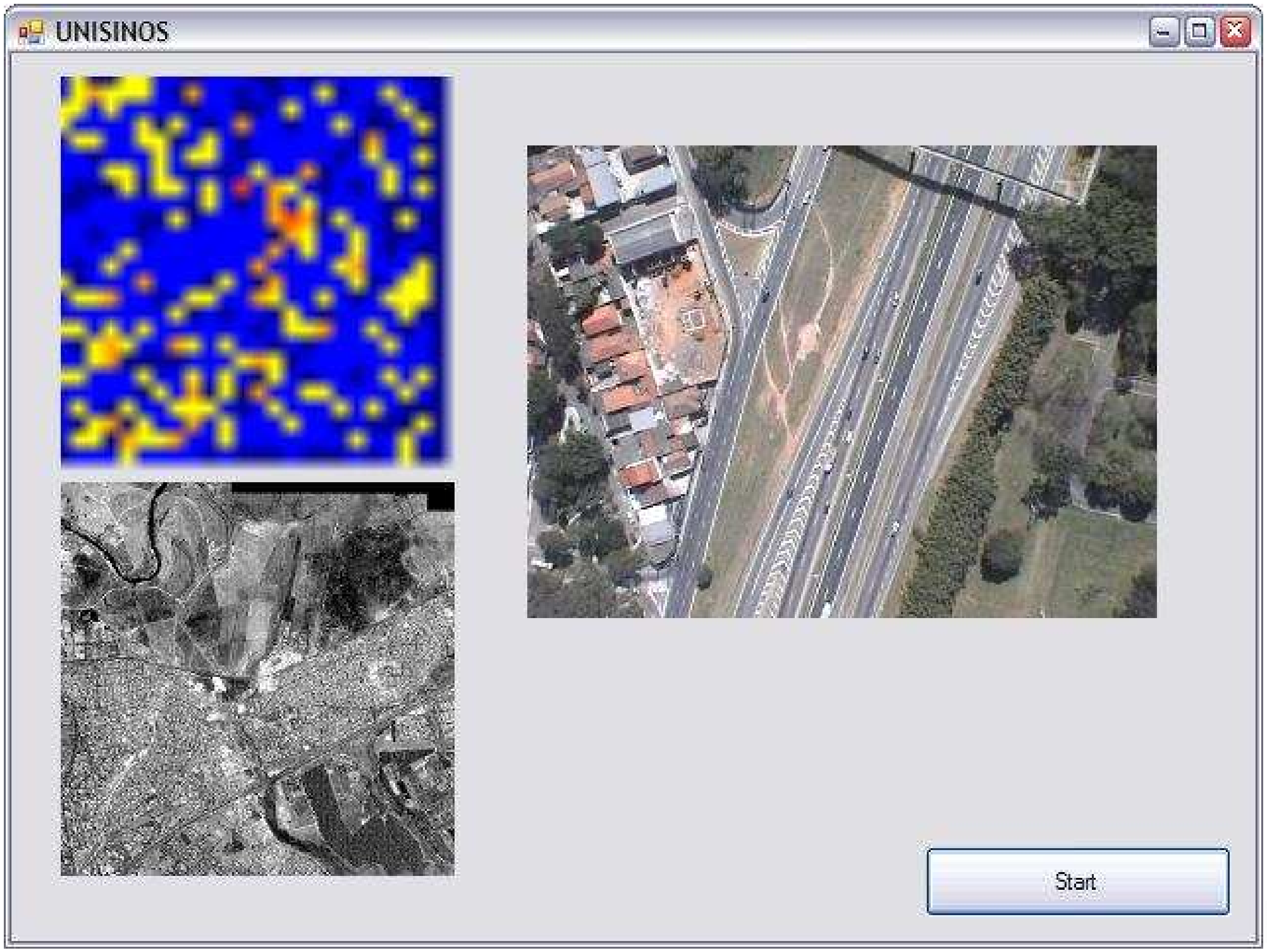
Visual Navigation

Omnidirectional Cameras



Aerial Visual Navigation

Vision System for Unmanned Aerial Vehicles



Correlation:
Satellite image
and Helicopter

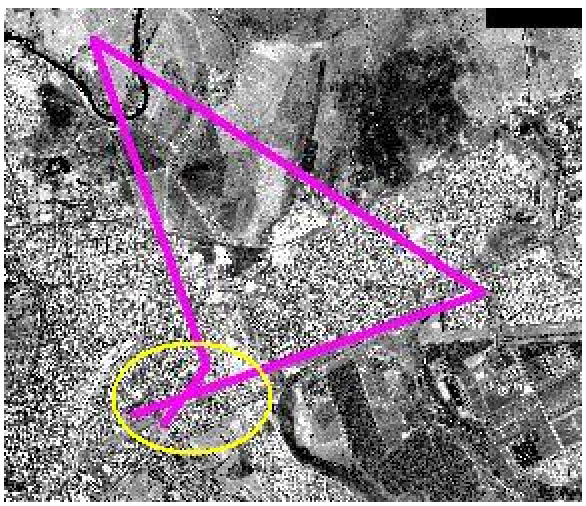
Results...
Not good at all!

Aerial Visual Navigation

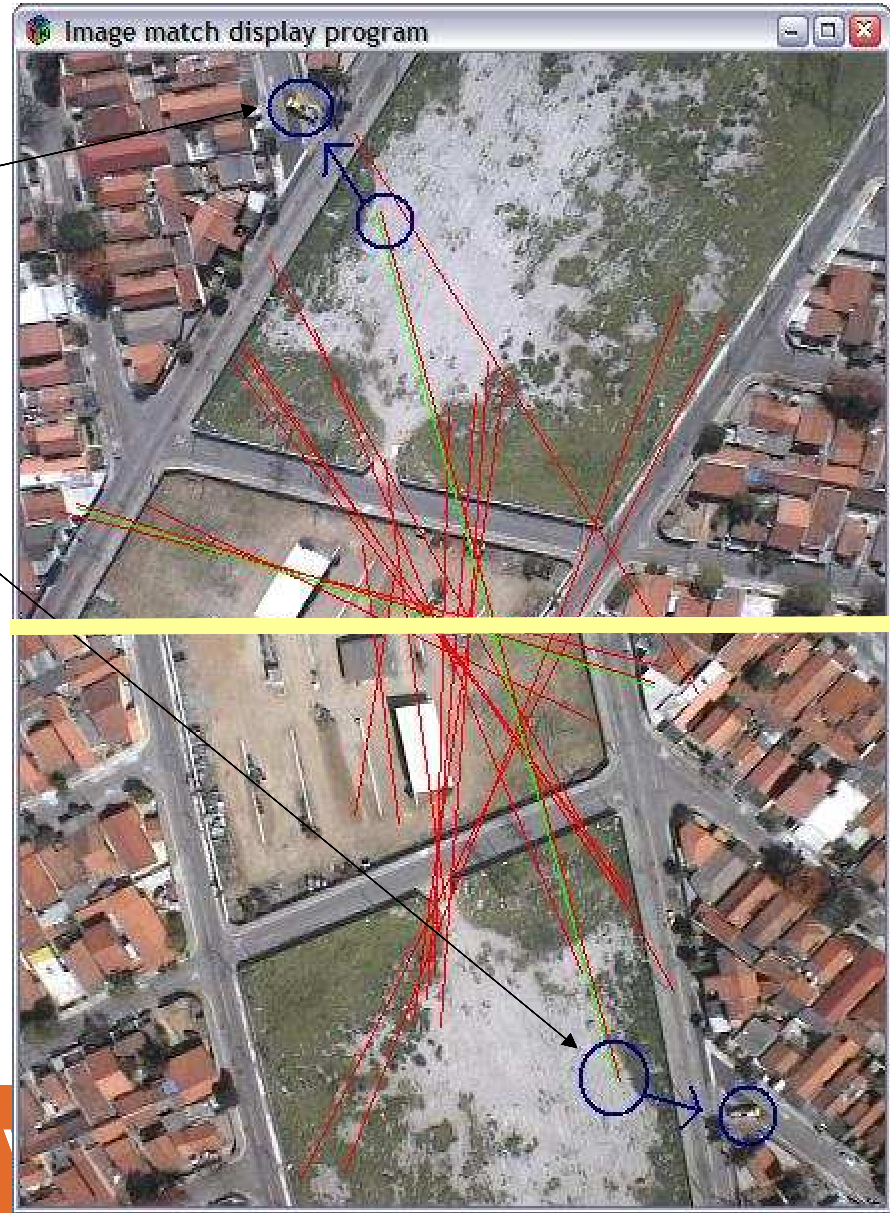
Vision System for Unmanned Aerial Vehicles

Referential

Correlation in the Crossing Point
Using helicopter only images



Very
Good
Match!



Vehicle Visual System

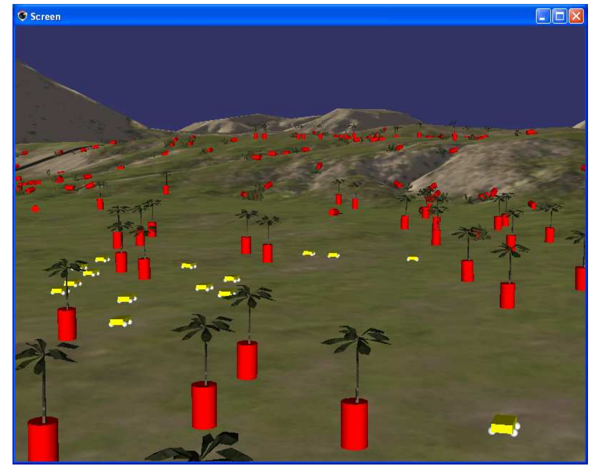
Vision system used to identify traffic signs

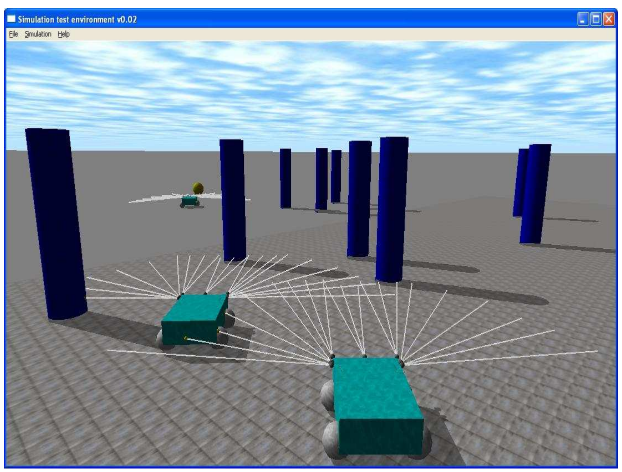
RGB Color
HSV Color

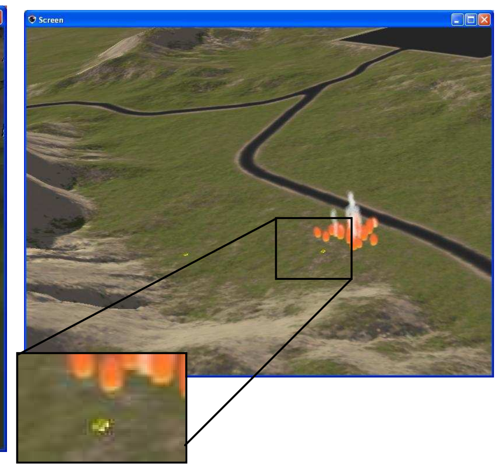
Color based sign segmentation
Artificial Neural Network Recognition

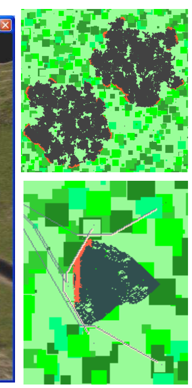
Multiple Vehicles: Fire fighting squad

Planning, Navigation, Control + Strategy, Cooperation

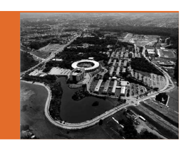








http://www.eletrica.unisinos.br/~autonom



UNISINOS

UNIVERSIDADE DO VALE DO RIO DOS SINOS

Intelligent Autonomous Vehicles