

Robots Móviles

Ejemplos de Aplicación

Otto Colomina Pardo

otto@dccia.ua.es

Mapas topológicos

Planificación de trayectorias

Rejillas de
ocupación

Robótica probabilística

Mapeado con poses conocidas

SLAM

Localización



Índice

**Vehículos autónomos
Robocup
Guías robóticos**

Algunos antecedentes

- Stanford Cart
(1960-1979)
- VaMoRs (Ernst
Dickmanns 1986-2003)



LiveSlides web content

To view

[Download the add-in.](#)
liveslides.com/download

[Start the presentation.](#)

4

Algunos antecedentes

- ALVINN (Autonomous Land Vehicle in a Neural Network) (1989-1992) - CMU



Actualidad: empresas privadas



Apollo

Plataforma abierta (software & hardware) para vehículos autónomos

Info developers: <http://apollo.auto/devcenter/devcenter.html>

apollo

Home

Open Platform

Cooperation

Apollo Fund

Developer Center

中文 | English

Visit Baidu AI

Introducing Apollo 3.0: so much more than before

Promote Open Capability-Share Resources-Accelerate Innovation-Sustain Mutual Benefit

Cooperation

Github



○ ● ○ ○

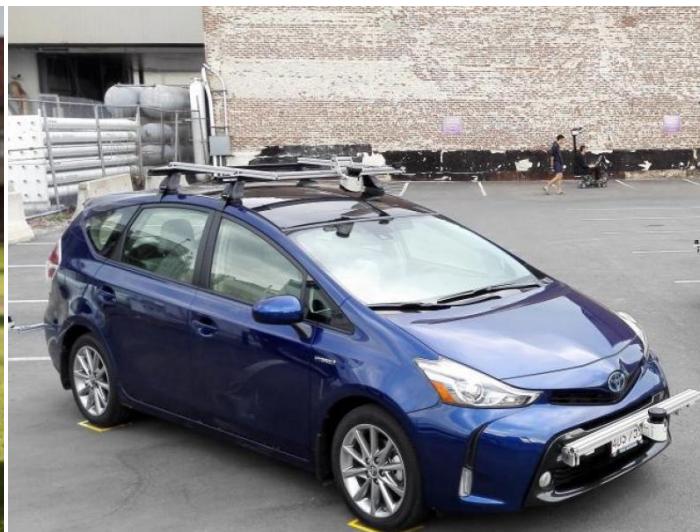


Stanford Autonomous Driving Team



<http://driving.stanford.edu>

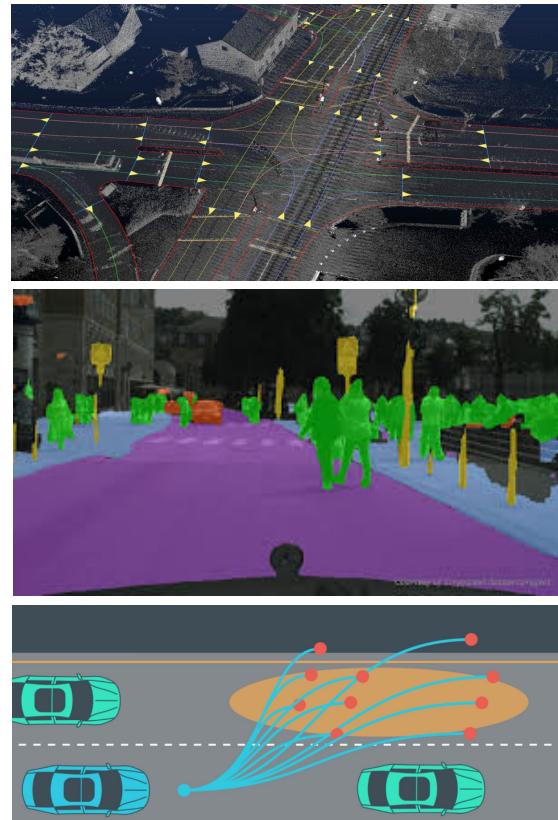
TOYOTA - CSAIL (MIT)



<https://toyota.csail.mit.edu>

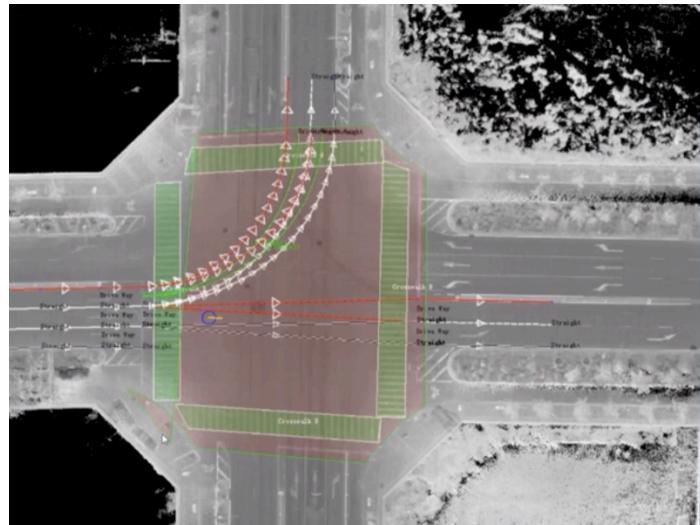
Funcionalidades necesarias

- Mapeado previo
- Localización
- Comprensión de escenas
- Planificación de movimiento



Construcción de mapas

- Se necesita resolución en el orden de cm
- El mapa se “aumenta” con información sobre
 - Carriles
 - Direcciones en la intersección
 - Señales de tráfico
 - Túneles
 - ...
- Hay formatos abiertos:
[OpenDRIVE](#)



Construcción de mapas en Apollo



Map Production

Data Sourcing



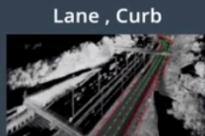
Data Processing



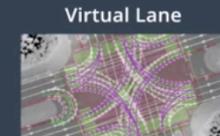
Object Detection



Manual Verification



Map Products



Algoritmos de construcción de mapas

- Algoritmos estándar en SLAM
- Ejemplo: Levinson & Thrun, 2011, SLAM basado en grafos
 - Celdas de 15x15 cm
- Ignorar objetos móviles: las celdas con alta varianza de reflectividad representan objetos móviles

Media



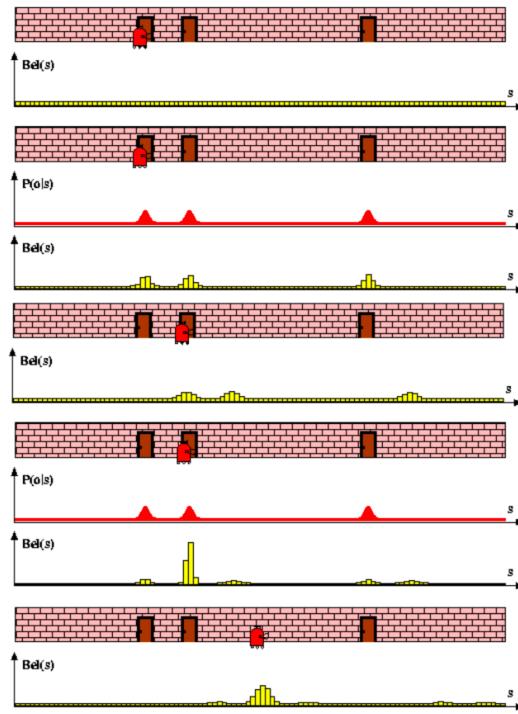
Varianza



Localización

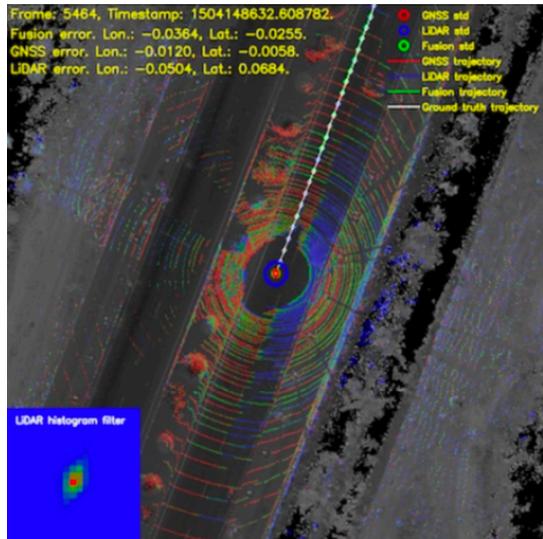
- Uso habitual de filtros bayesianos discretos

Representación
discretizada



Localización

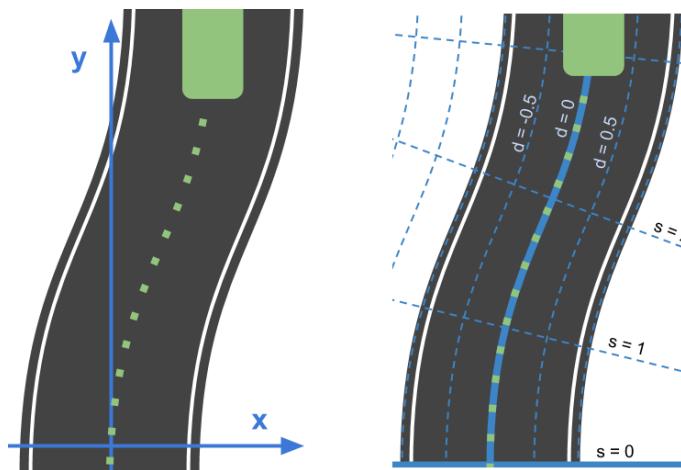
- Uso habitual de filtros bayesianos discretos
 - El GPS proporciona una estimación inicial razonable (error de algunos metros)
 - Basta con una rejilla de localización “pequeña”



Robust and Precise Vehicle Localization based on Multi-sensor Fusion in Diverse City Scenes

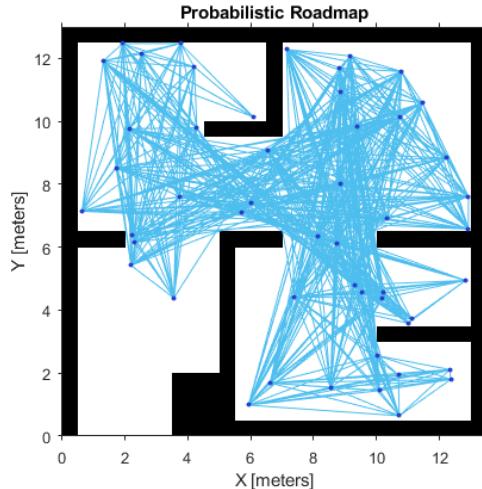
Planificación de movimiento: características

- Algoritmos de muestreo para reducir el espacio de búsqueda
- Evaluación de trayectorias (*al estilo Dynamic Window*)
- Algoritmos especializados p.ej. Apollo
 - No se usa un sistema de coordenadas cartesiano sino con respecto a la carretera
 - Se planifica trayectoria y velocidad por separado



Planificación de movimiento probabilística

- En espacios de búsqueda **muy grandes** no es posible hacer una búsqueda exhaustiva del mejor camino
- Se **muestren aleatoriamente** los caminos posibles. A mayor número de muestras mayor probabilidad de encontrar un mejor camino



PRM (Probabilistic Roadmap Method)

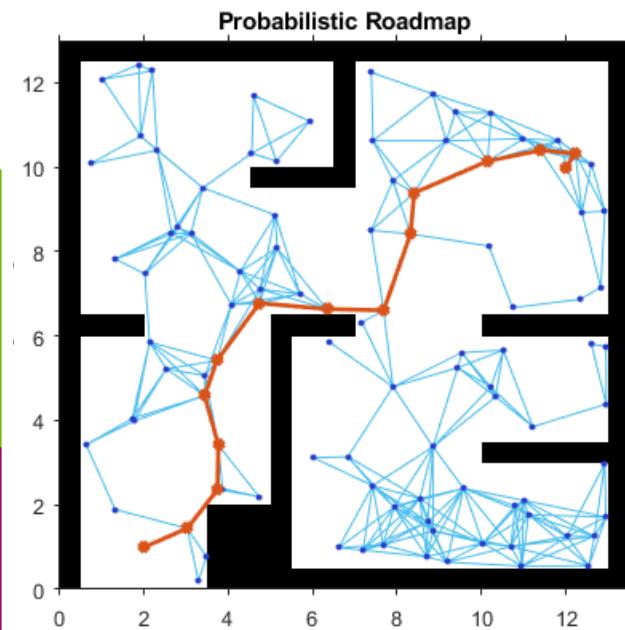
- En vez de considerar todas las celdas, se muestrea aleatoriamente el espacio, obteniendo un **grafo**

Procesamiento

- Generar nodos al azar en el espacio libre
- Conectar cada nodo con los vecinos si el camino no comisiona con obstáculos

Consulta

- Calcular el camino más corto en el grafo mediante algún algoritmo de búsqueda



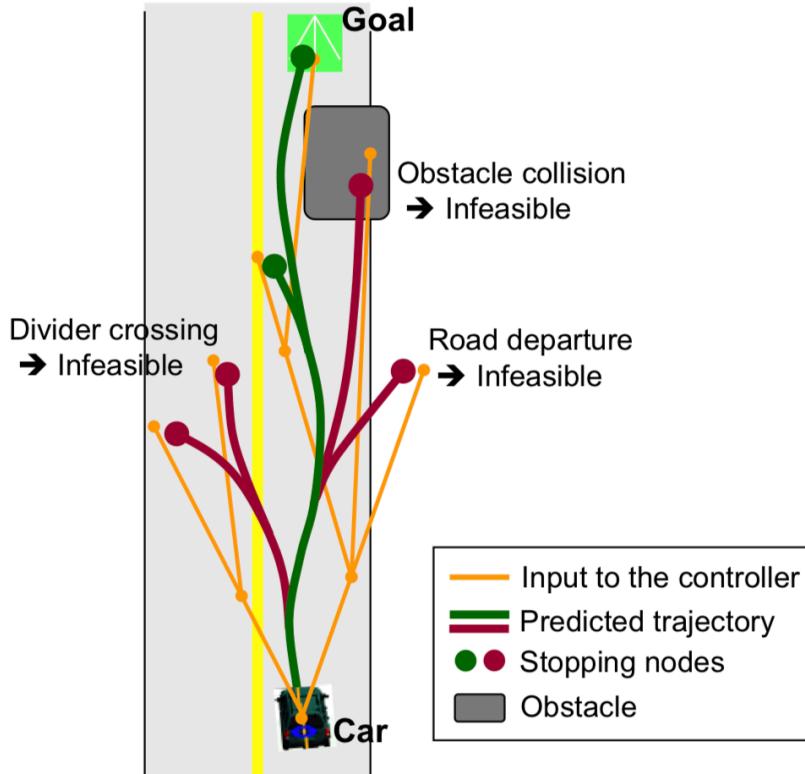
RRT (Rapidly-exploring Random Tree)

- Vamos formando un árbol que va “creciendo” desde el origen hasta el destino

Algorithm 1: RRT

```
1  $G.\text{init}(q_0)$ 
2 repeat
3    $q_{rand} \rightarrow \text{RANDOM\_CONFIG}(\mathcal{C})$ 
4    $q_{near} \leftarrow \text{NEAREST}(G, q_{rand})$ 
5    $G.\text{add\_edge}(q_{near}, q_{rand})$ 
6 until condition
```

Generación de trayectorias



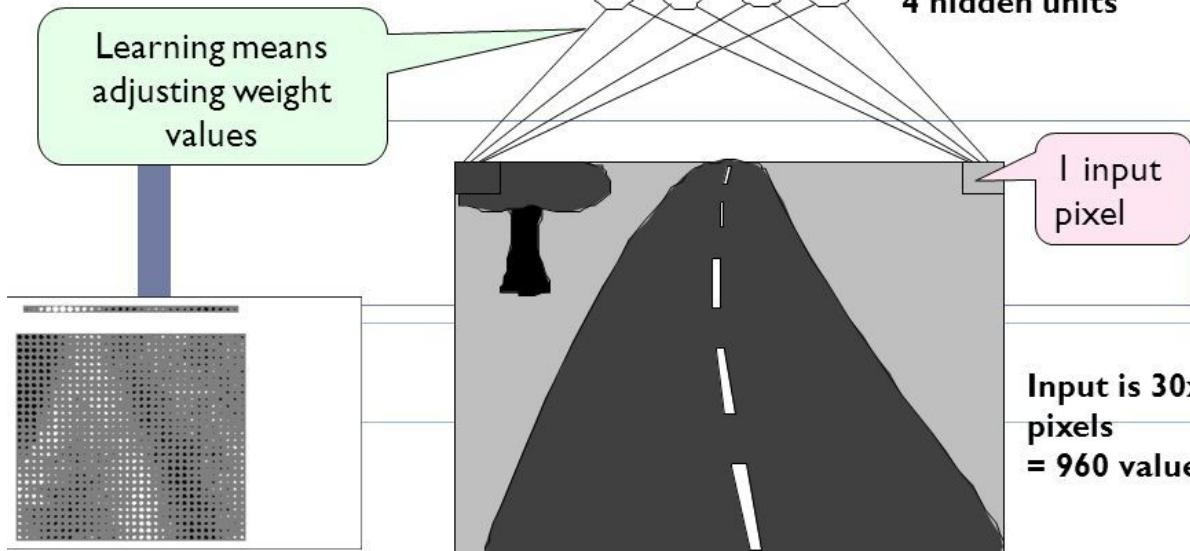
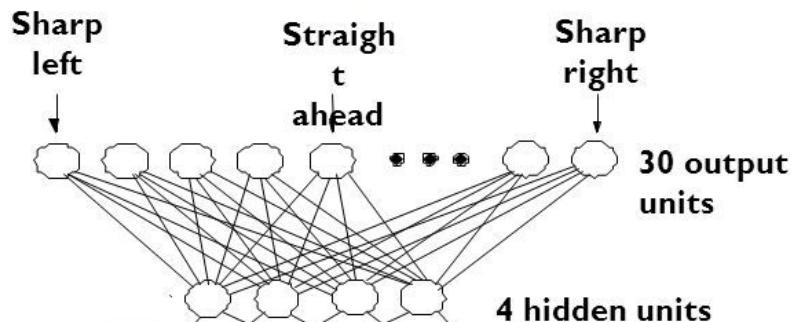
Evaluación de trayectorias

- Idea similar a la ventana dinámica

```
157     double TrajectoryEvaluator::Evaluate(
158         const PlanningTarget& planning_target,
159         const PtrTrajectory1d& lon_trajectory,
160         const PtrTrajectory1d& lat_trajectory,
161         std::vector<double>* cost_components) const {
162     // Costs:
163     // 1. Cost of missing the objective, e.g., cruise, stop, etc.
164     // 2. Cost of longitudinal jerk
165     // 3. Cost of longitudinal collision
166     // 4. Cost of lateral offsets
167     // 5. Cost of lateral comfort
168
169     // Longitudinal costs
170     double lon_objective_cost =
171         LonObjectiveCost(lon_trajectory, planning_target, reference_s_dot_);
172 }
```

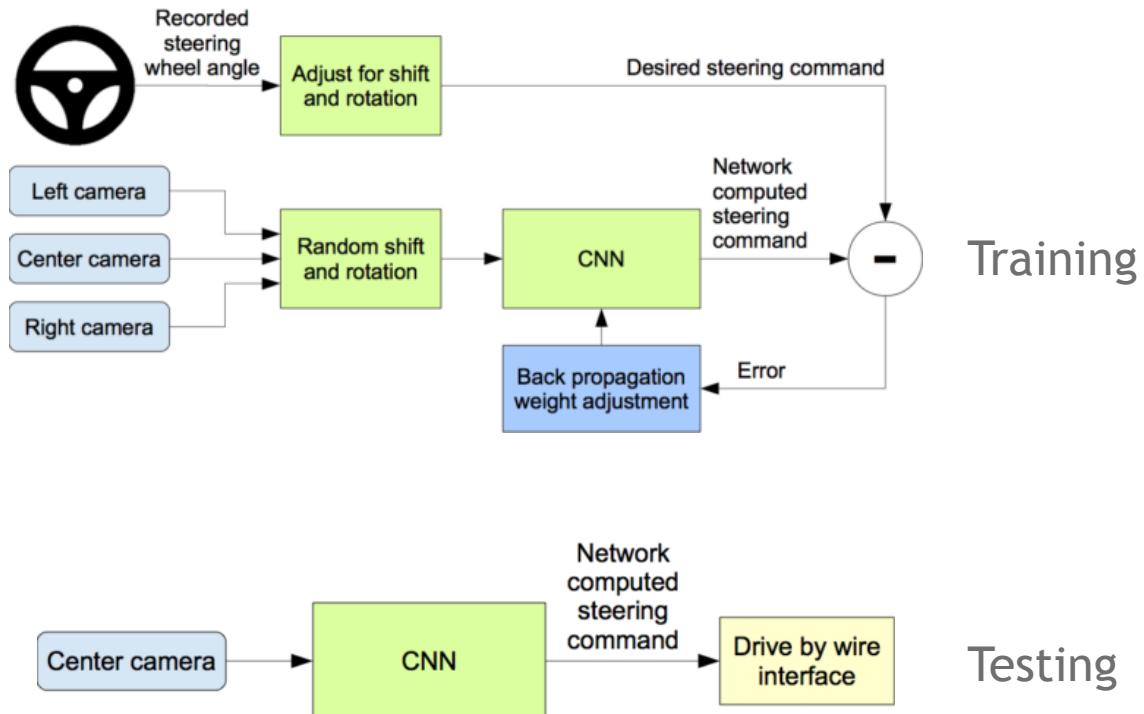
trajectory_evaluator.cc en https://github.com/ApolloAuto/apollo/tree/master/modules/planning/lattice/trajecotry_generation

ALVINN (1989)



NVIDIA end-to-end self driving (2016)

<https://devblogs.nvidia.com/deep-learning-self-driving-cars/>





LiveSlides web content

To view

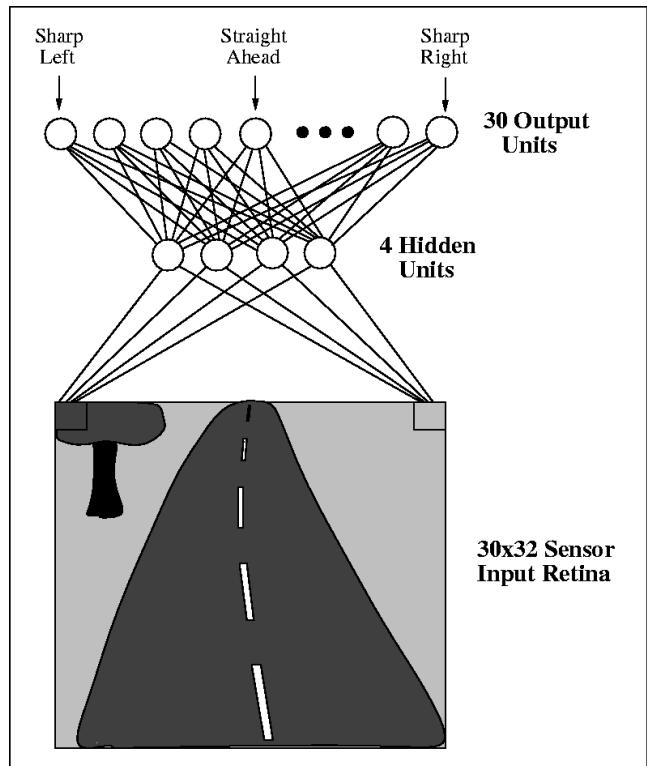
Download the add-in.

liveslides.com/download

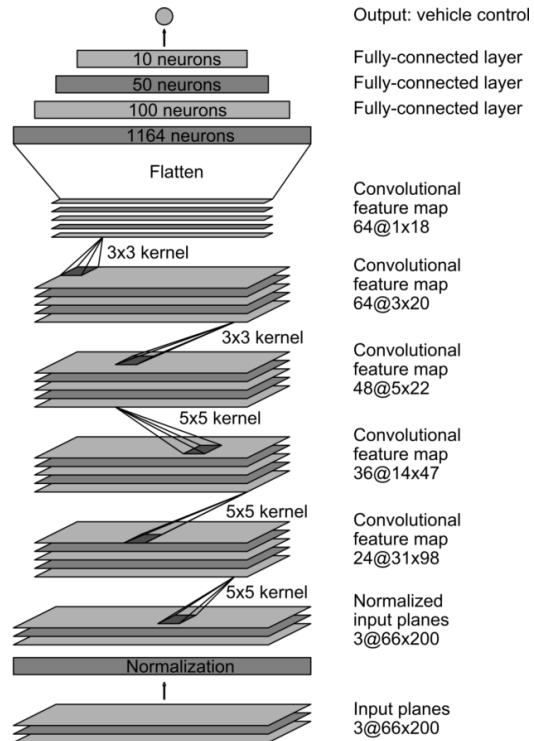
Start the presentation.

<https://youtu.be/NJU9ULQUwng>

ALVINN vs DAVE-2



$$30 \times 32 \times 4 + 30 \times 4 = 3960 \\ \text{conexiones}$$



Aprox. 27 millones
conexiones y 250000
parámetros

Índice

Vehículos autónomos
Robocup
Guías robóticos

By the middle of the 21st century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup

<https://www.robocup.org>





LiveSlides web content

To view

Download the add-in.

liveslides.com/download

Start the presentation.

¿Por qué Robocup?

El fútbol es un buen contexto para comprobar avances en distintas áreas de la robótica

- Locomoción y cinemática
- Coordinación de robots
- Planificación de alto y bajo nivel
- Algoritmos de IA
 - Visión
 - Aprendizaje
 -

Hay diferentes categorías (*leagues*)

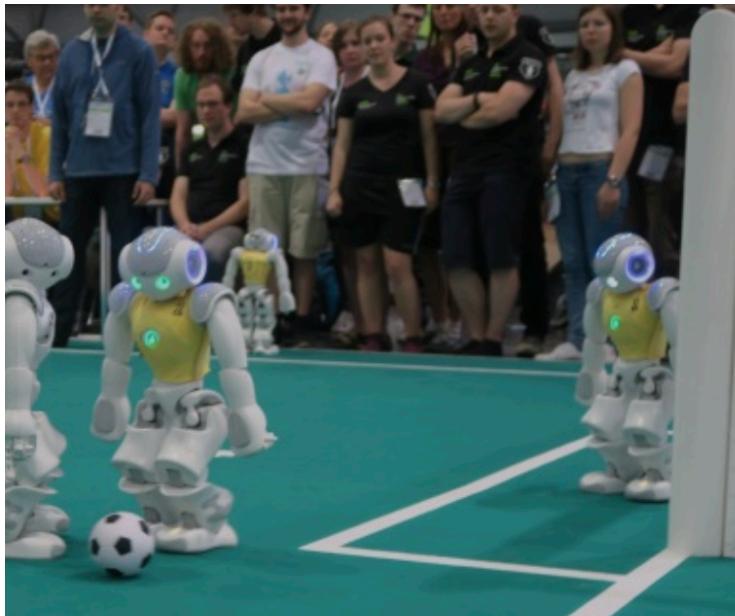
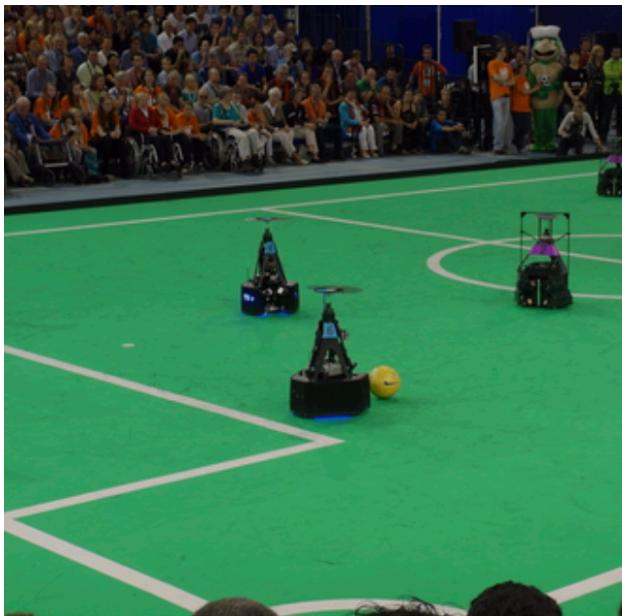


Simulación (hay 2D y 3D)



Small size
máx 18cm diam x 15 alto, no son
autónomos (ordenador externo)

Hay diferentes categorías (*leagues*)



Middle size

Pelota tamaño oficial FIFA,
hardware propio de cada equipo

Standard Platform

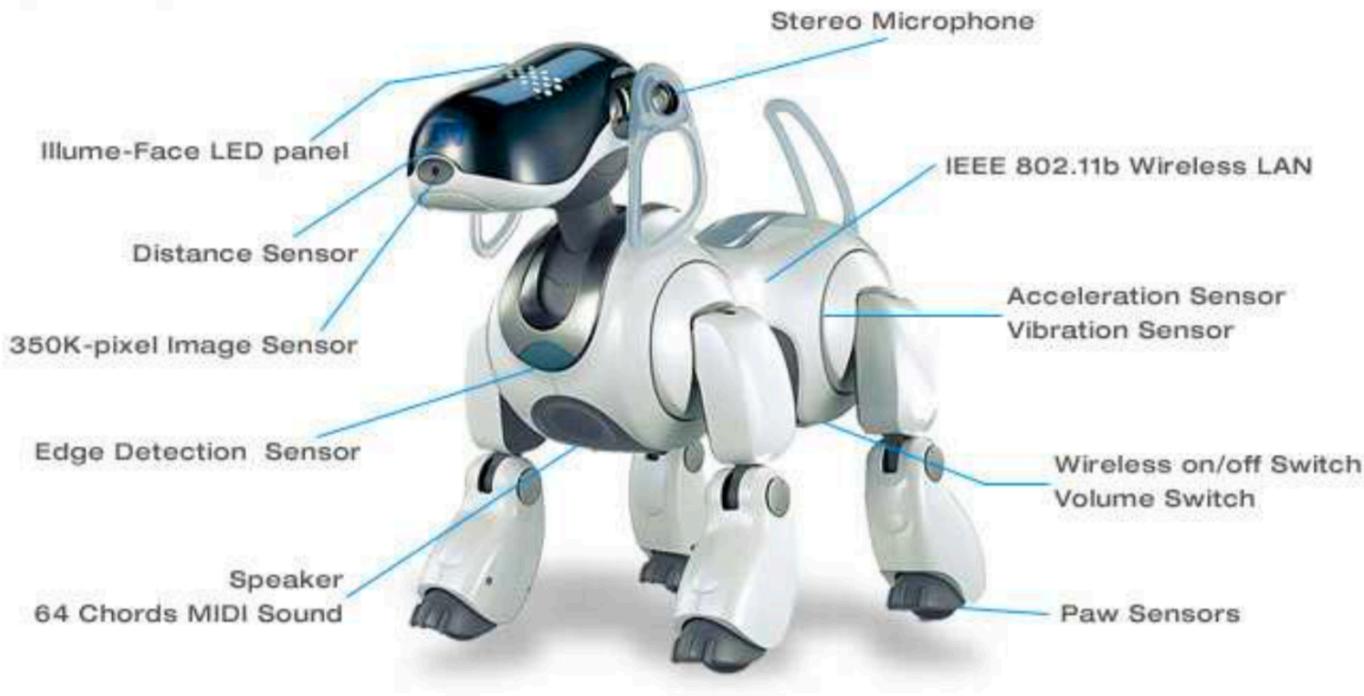
Todos los equipos usan el mismo
robot (Nao)

TeamChaos

Formado en 2005 por Univ. Orebrö (Suecia), U.Rey Juan Carlos, U. de Murcia y U. de Alicante

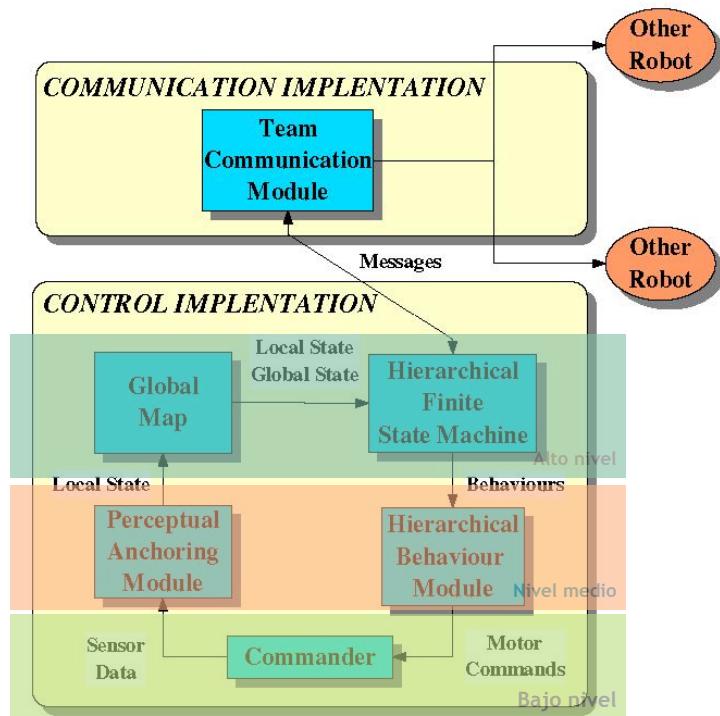


Sony Aibo (1999-2006)



La figura muestra la versión ERS-7

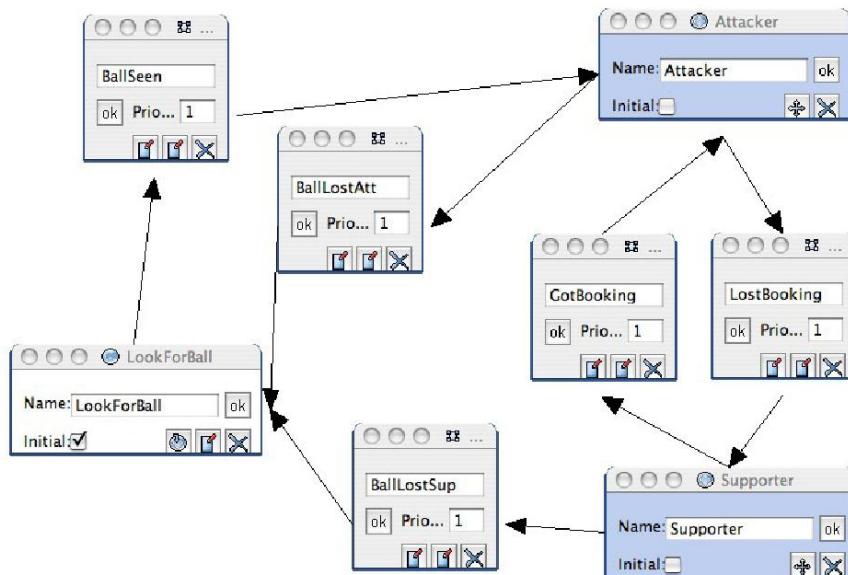
Arquitectura TeamChaos



- Arquitectura basada en conductas
- Dividida en niveles
- Uso de lógica fuzzy

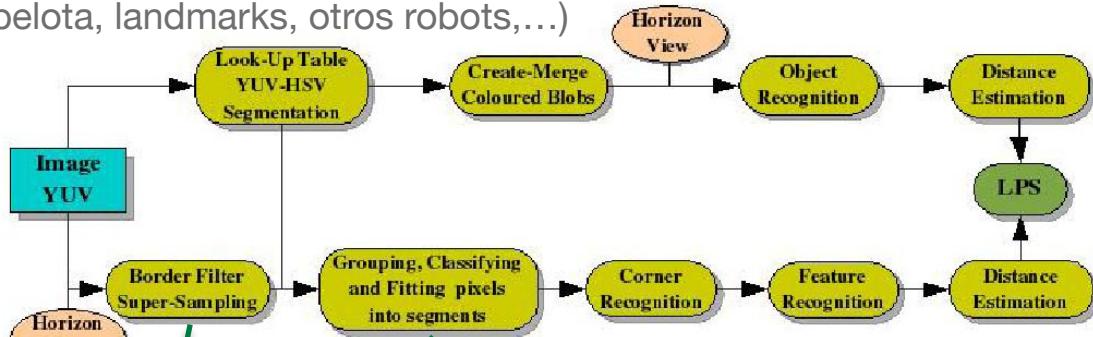
Referencia: TeamChaos 2006

Las conductas de alto nivel se implementan con una máquina de estados finitos jerárquica (HFSM)

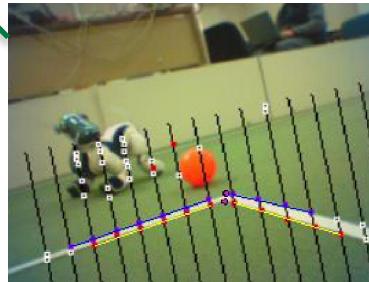
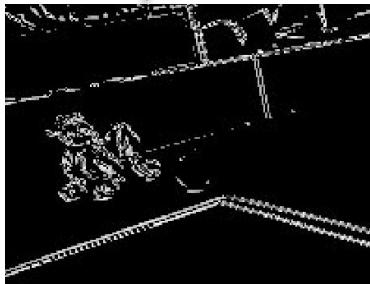


Visión: 2 caminos para obtener información

Detección de marcadores de color
(pelota, landmarks, otros robots,...)



Detección de características del campo
(líneas, esquinas, semicírculo central...)



Localización

- Basada en lógica *fuzzy*
- A partir de las esquinas de las líneas del campo y de los colores de las redes de las porterías

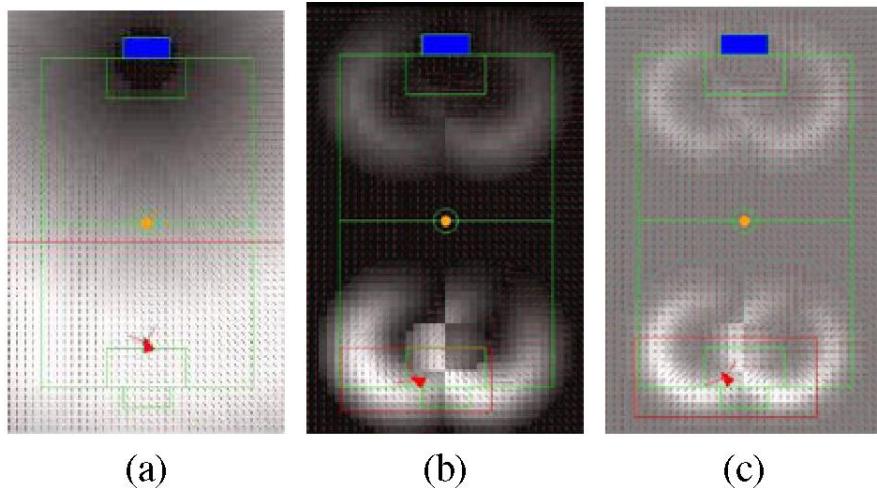
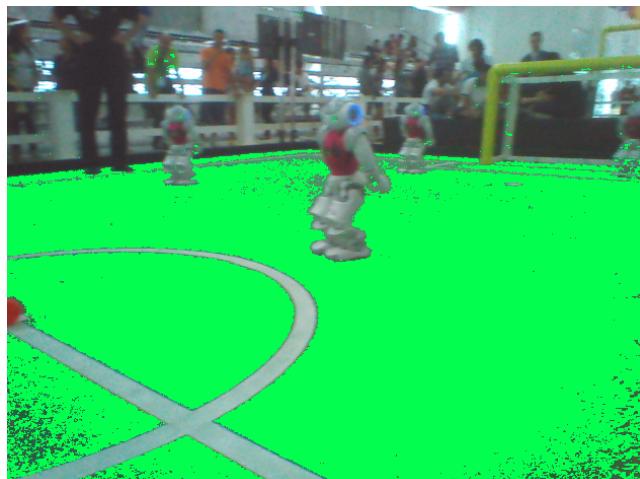


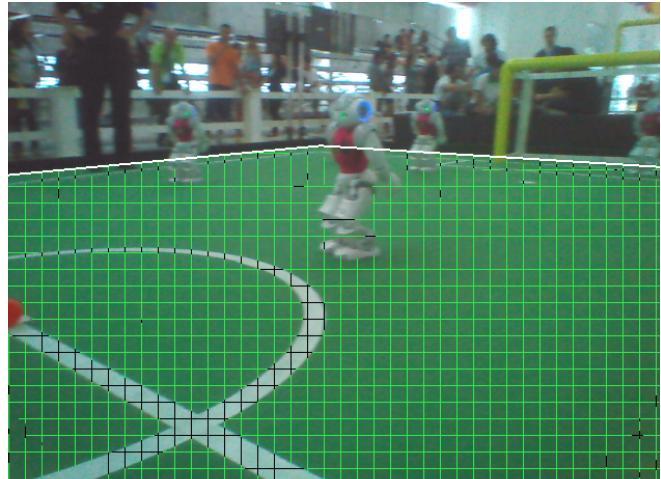
Fig. 9. Belief induced by the observation of (a) a net, (b) the first feature, and (c) the second feature. Initially the position of the robot is unknown (belief distributed along the full field).

Nao-HTWK team (2017)

Detección de campo y sus límites



Segmentación de color
con machine Learning



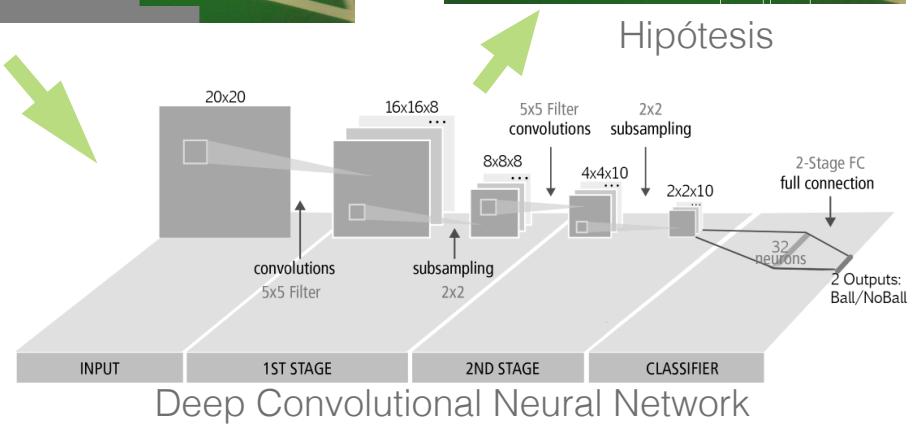
Scanlines +. RANSAC

Nao-HTWK team (2017)

Detección de la pelota (desde 2016 es blanca y negra)



Selección zonas candidatas(dentro del campo, con alto contraste)



Índice

Vehículos autónomos
Robocup
Guías robóticos



Rhino

6 días en el *Deutsches Museum* Bonn
1997



Minerva

2 semanas en el *Smithsonian*
1998

Algunos desafíos

- El robot en ocasiones está totalmente rodeado de gente
- Alguna gente trata de poner “en apuros” al robot, por ejemplo bloqueándole el paso
- Ciertos elementos comunes en los museos son invisibles a los sensores del robot (p.ej. vitrinas)

Algoritmos “convencionales” usados

- **SLAM:** EM (Expectation Maximization) (*EM actualmente no se suele usar tal cual para SLAM por su elevado coste*)
- **Localización:** filtros bayesianos discretos (sustituidos luego por filtros de partículas)
- **Evitación de obstáculos:** ventana dinámica, aumentada con información del mapa para tener en cuenta obstáculos invisibles a los sensores

Algunas ideas ingeniosas

Mapa del techo (Minerva), para evitar el problema de la gran cantidad de gente

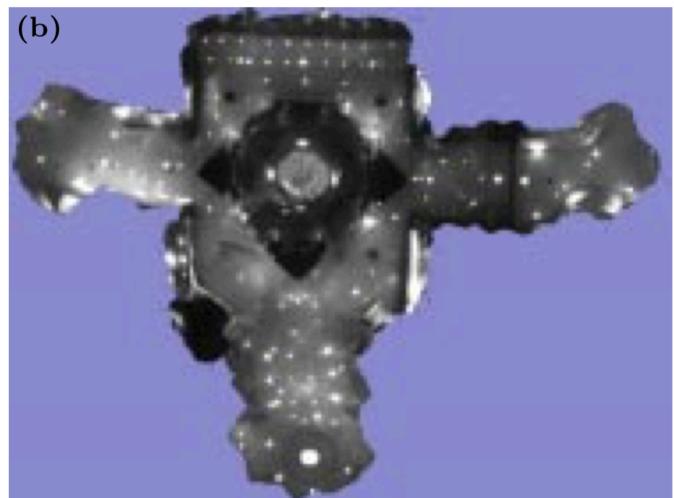
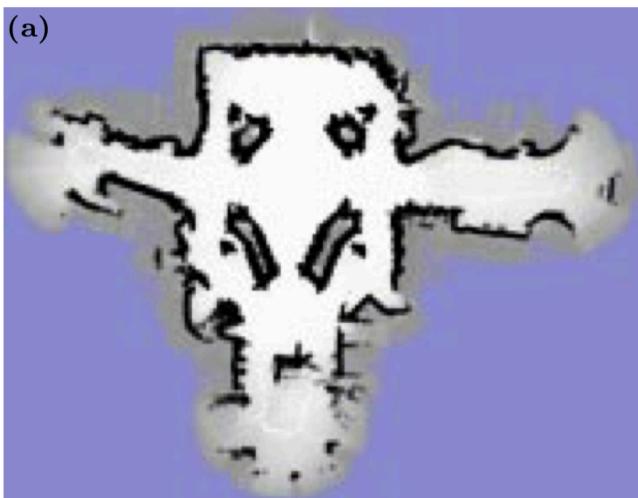
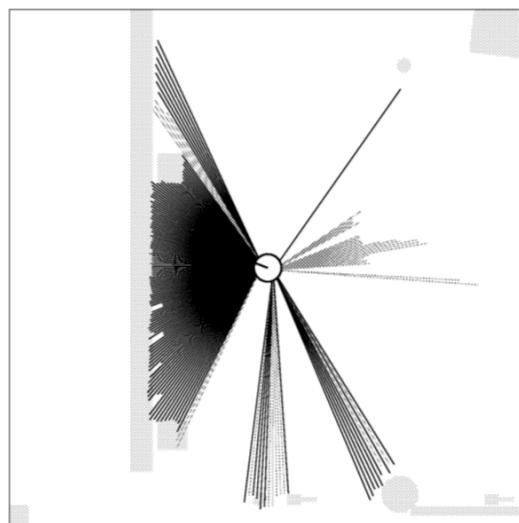


Figure 10: (a) Occupancy map of the center portion of the Smithsonian museum. (b) Mosaic of the museum's ceiling. The various bright spots correspond to lights. The center portion of the ceiling contains an opening—the lights there are approximately 15 meters higher.

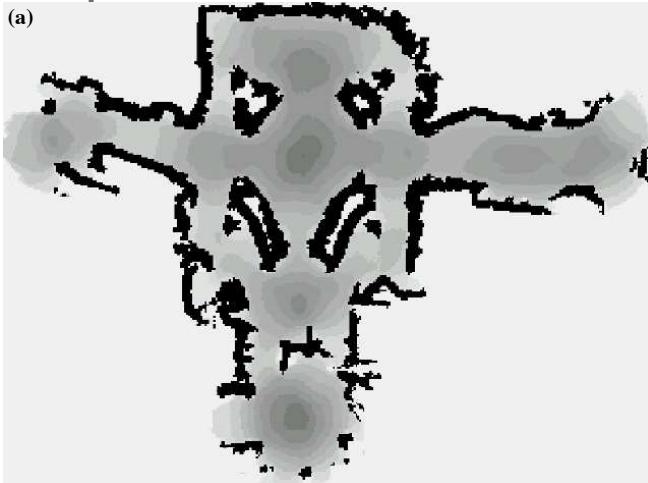
Algunas ideas ingeniosas

Filtro de ganancia de entropía (Rhino/Minerva): solo se hace caso a las medidas que disminuirían la incertidumbre en la localización. El efecto es filtrar las medidas causadas por obstáculos no previstos en el mapa (típicamente los visitantes del museo)

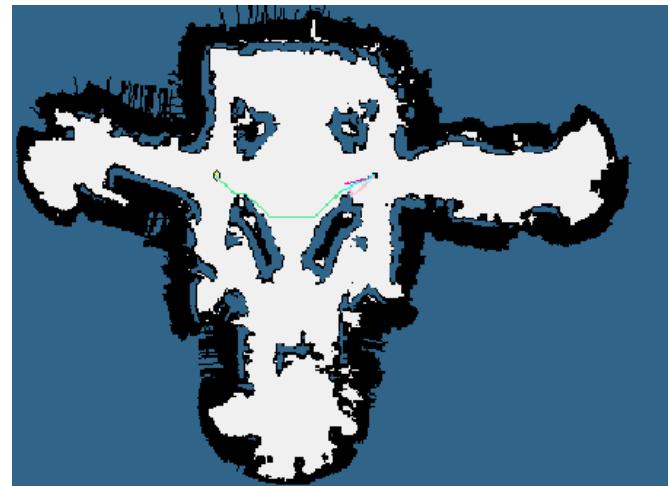


Algunas ideas ingeniosas

“Coastal navigation” (Minerva): En planificación de trayectorias, balance entre el camino más corto y el que proporcione más información a los sensores para no perderse



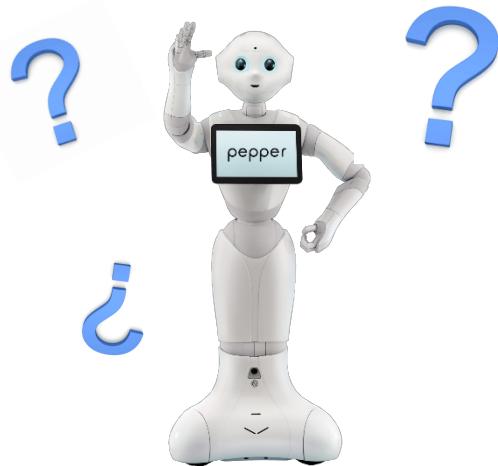
Mapa de entropía, a más oscuro
menos información



Trayectoria planificada por el algoritmo (se
acerca más a los obstáculos que el
camino más corto)

Robots Móviles

Grado en Ingeniería Robótica



otto @ dccia.ua.es

