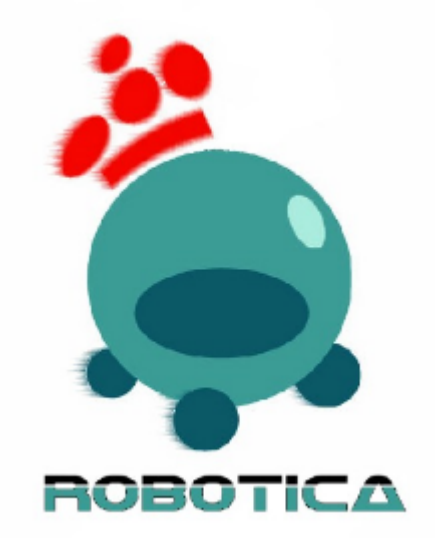
Navegación y autolocalización de un robot guía de visitantes

Julio M. Vega Pérez



---

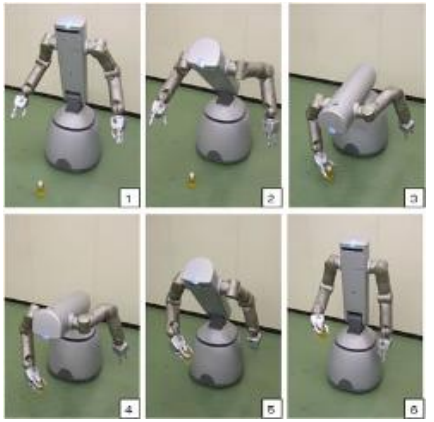
http://jde.gsyc.es/index.php/jmvega_guide_robot

Índice

- Introducción
- Objetivos
- Infraestructura
- Navegación global
- Localización
- Navegación local
- Conclusiones

Introducción

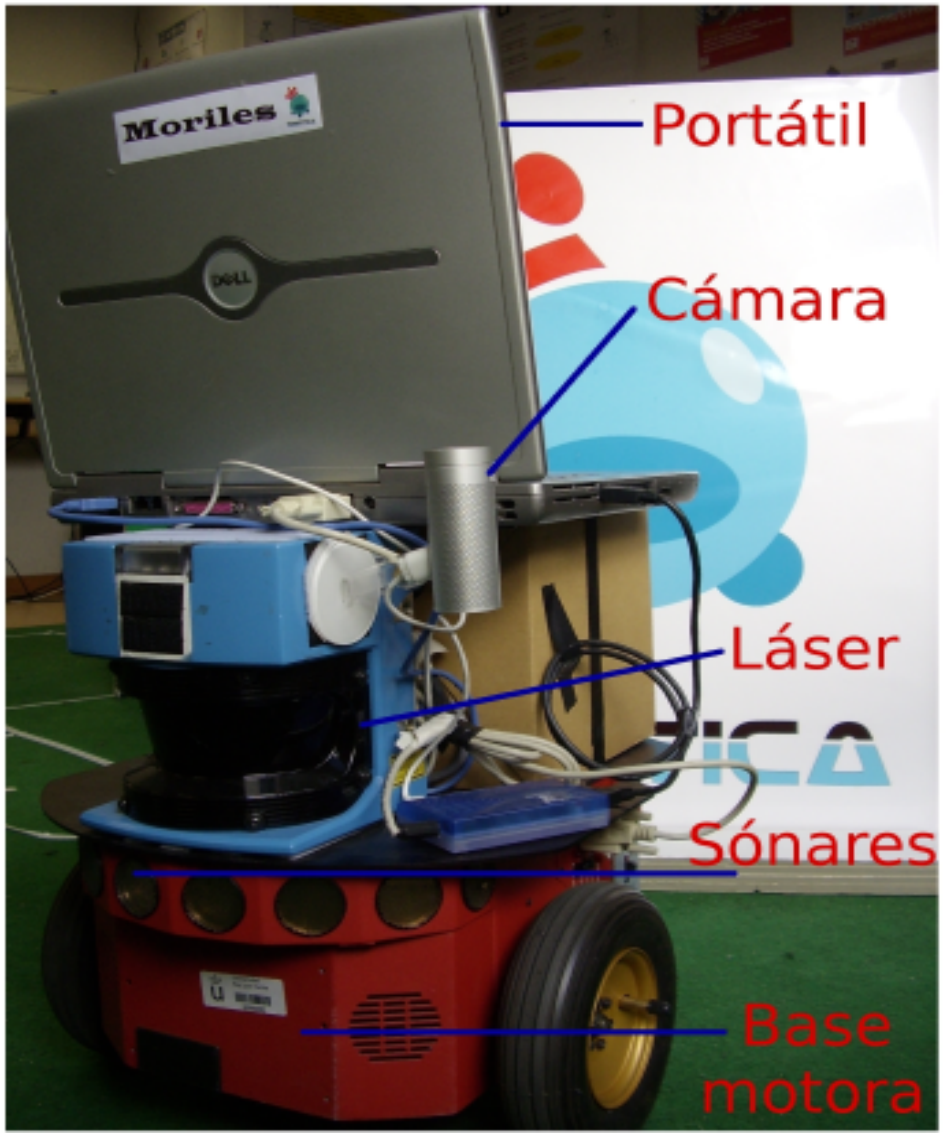
- Robots móviles
- Robots de servicio
- Robots guía
  - Cicerobot (italiano)
  - Enon (de Fujitsu)
  - Smartpal (con articulaciones)
  - Minerva (Carnegie Mellon)



Objetivos

- Desarrollar componentes necesarios para un robot guía
  - Navegación segura con robot físico en tiempo real
  - Cálculo de ruta óptima hacia el destino
  - Sistema de autolocalización del robot

Infraestructura



- Robot Pioneer
- Cámara, para localización
- Láser, para navegación local

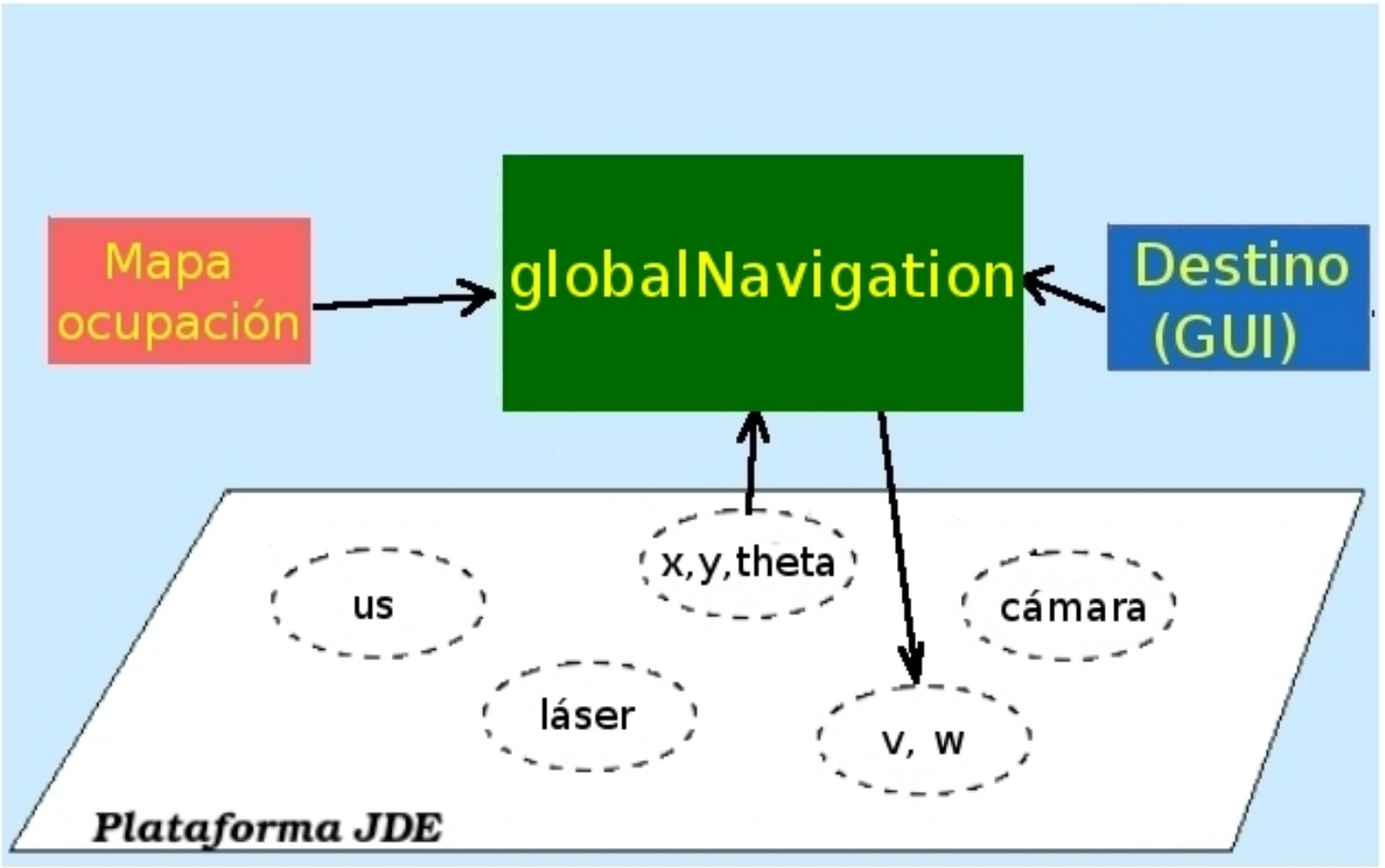
- Plataforma SW Jde
- Simulador Stage
- Bibliotecas auxiliares

Navegación global (1/3)

- Conoce el mapa
- Destino fijado por el usuario
- Origen = posición actual en el mapa
- Necesita planificar la ruta óptima a ese destino
  - Pudiendo éste estar fuera del rango sensorial
  - La ruta se calcula antes de que empiece a moverse
- No tiene en cuenta obstáculos imprevistos

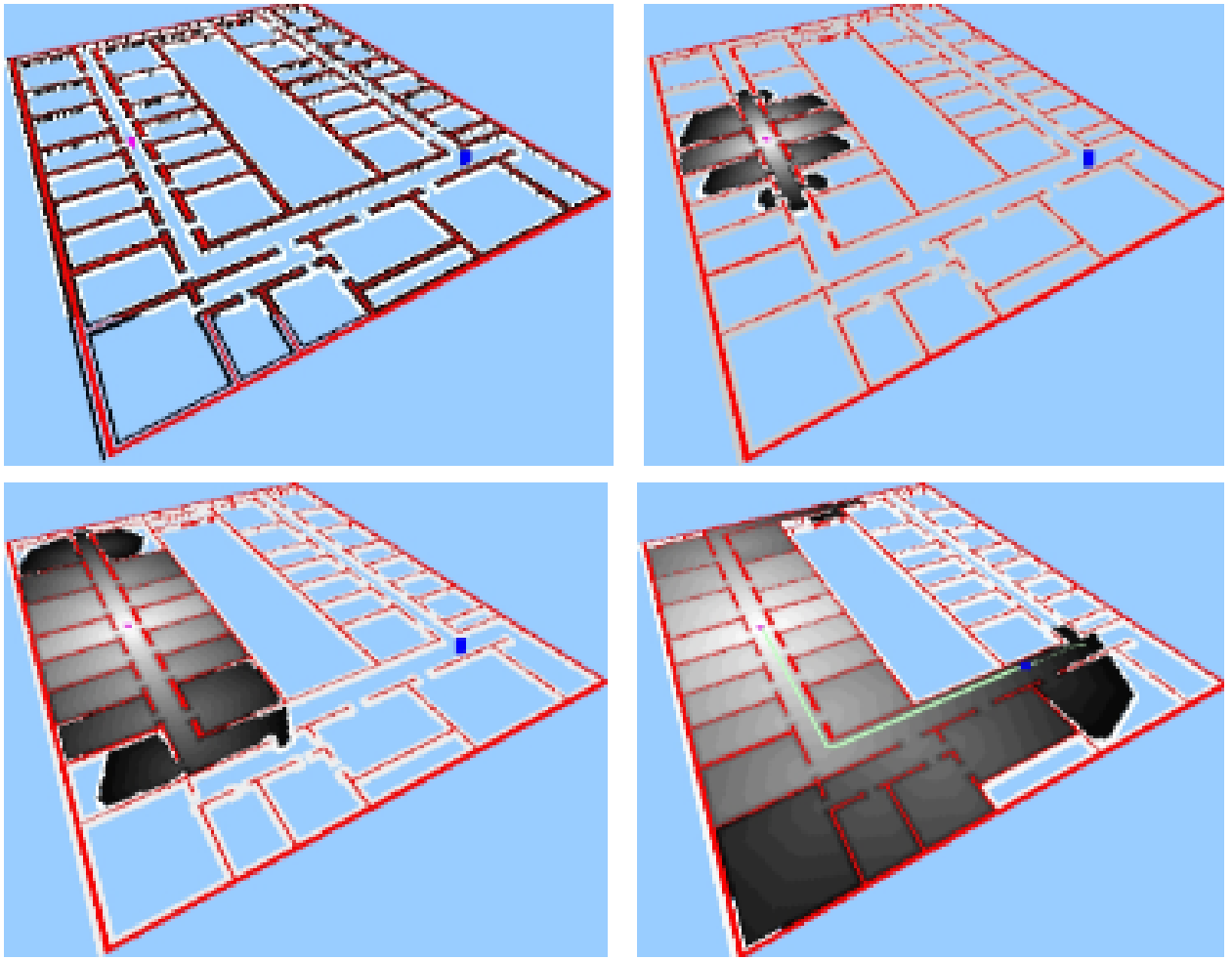
Navegación global (2/3)

- Esquema *globalNavigation*



Navegación global (3/3)

- Experimentos. Navegación global usando *GPP*



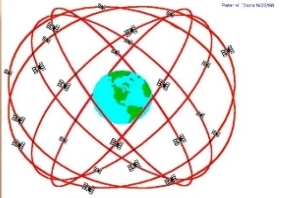
- *Propagación de obstáculos*
- *Propagación de frente de distancias*
- *El robot sigue el gradiente*

(demo)

Localización (1/6)

- Para navegar es necesario estimar la posición del robot
- *Encoders*


(a)



(b)

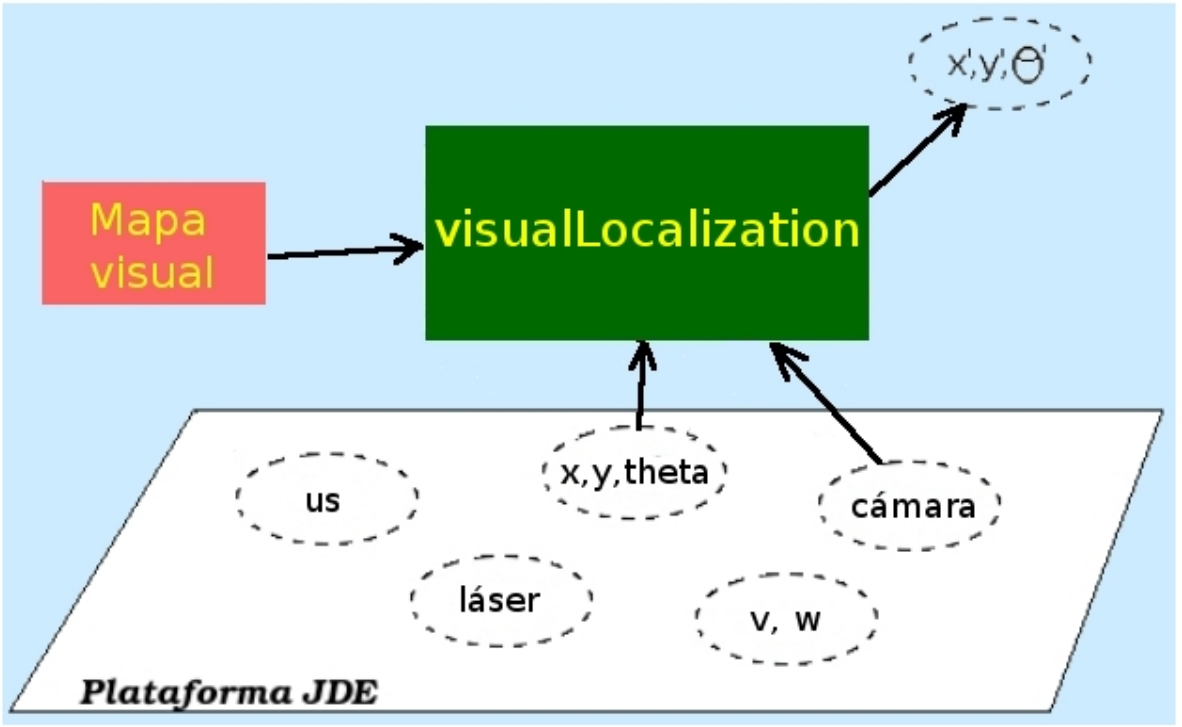
- ¿GPS? Preciso, pero sólo para exteriores
- Sensores no específicos
- Método localización probabilística
  - A cada posible posición asocia una probabilidad
    - Actualizada con nuevas lecturas sensoriales + movimientos

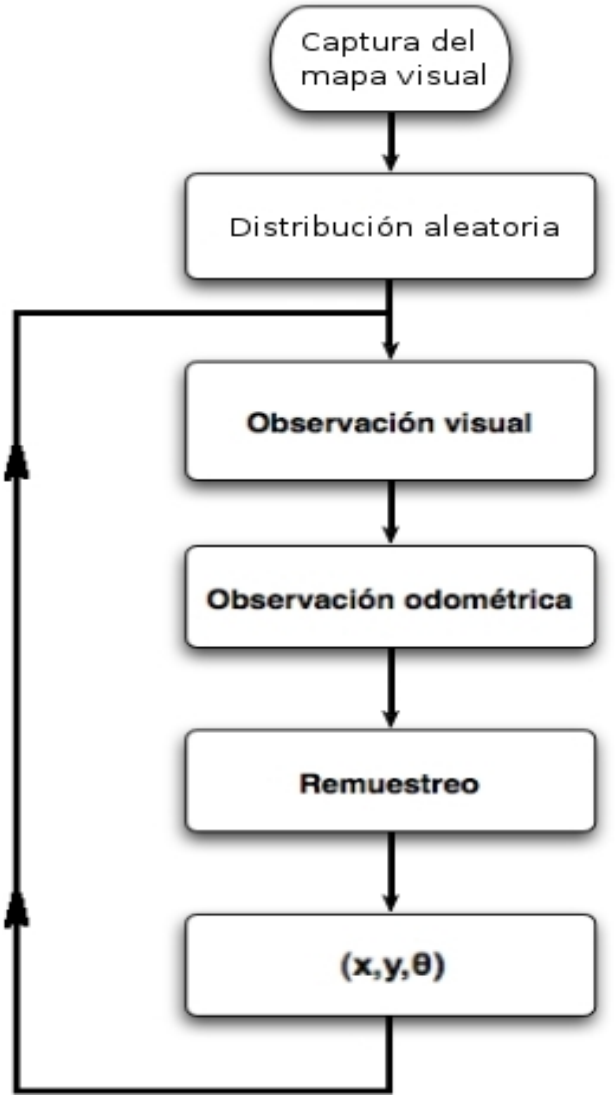
Localización (2/6)

- Localización con muestreo. Filtro de partículas
  - Mejora el tiempo de cómputo. Proceso escalable a grandes entornos
  - Método de Monte Carlo
- Representación muestreada de la distribución de probabilidad
  - Muestra = partícula con posición *(x,y,theta)*
  - Cada partícula tiene asociada una imagen teórica
  - Probabilidad = Imagen teórica *vs.* Imagen real

Localización

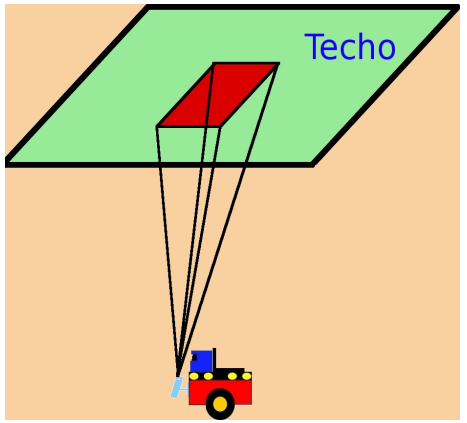
- Esquema *visualLocalization*





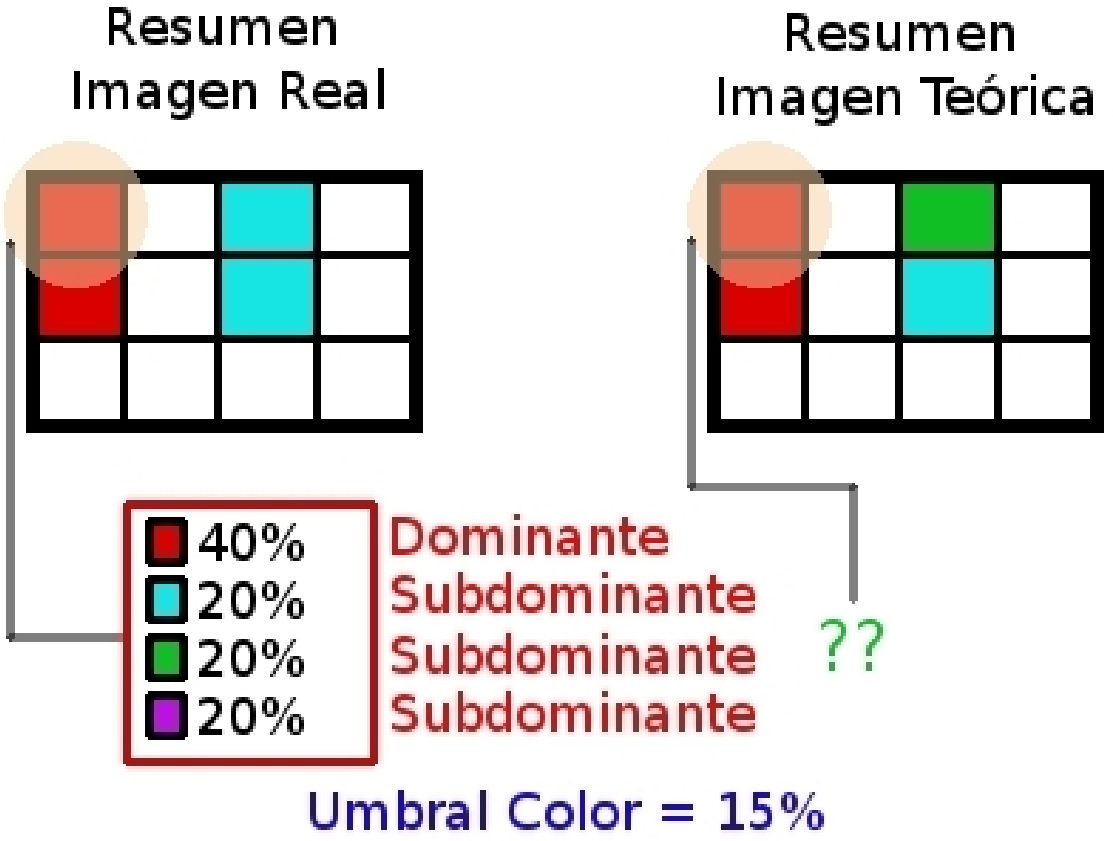
Localización

- Mapa visual del techo



Localización (5/6)

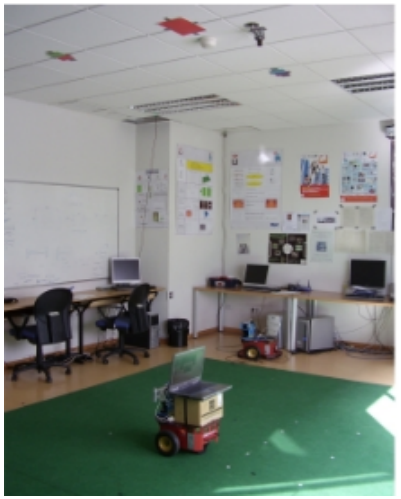
- *Modelo de observación visual*. Cálculo de probabilidad

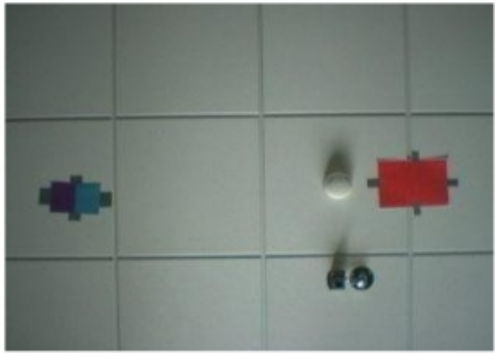


- 12 zonas
- Color dominante por zona (30%)
- Colores subdominantes por zona (70%)

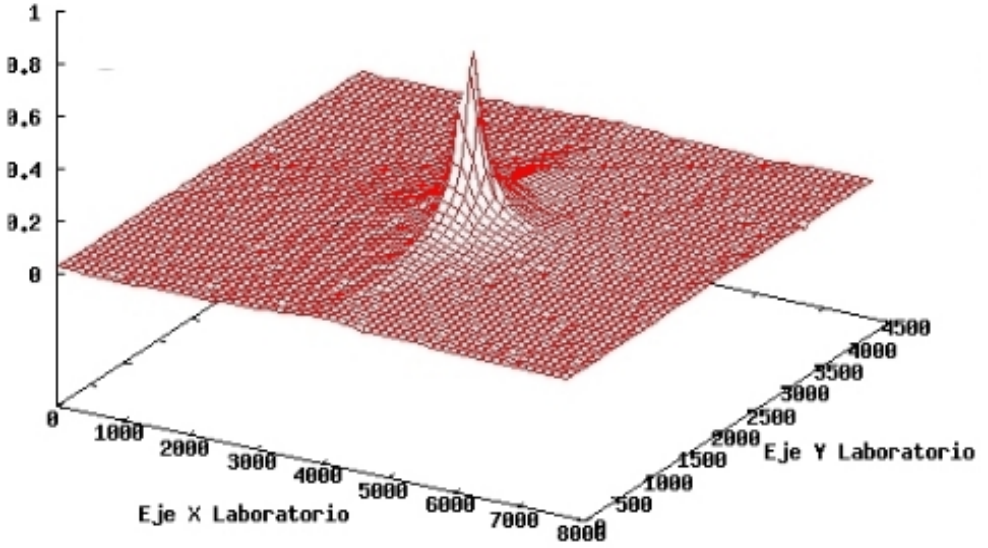
Localización

- *Modelo de observación visual*. Cálculo de probabilidad





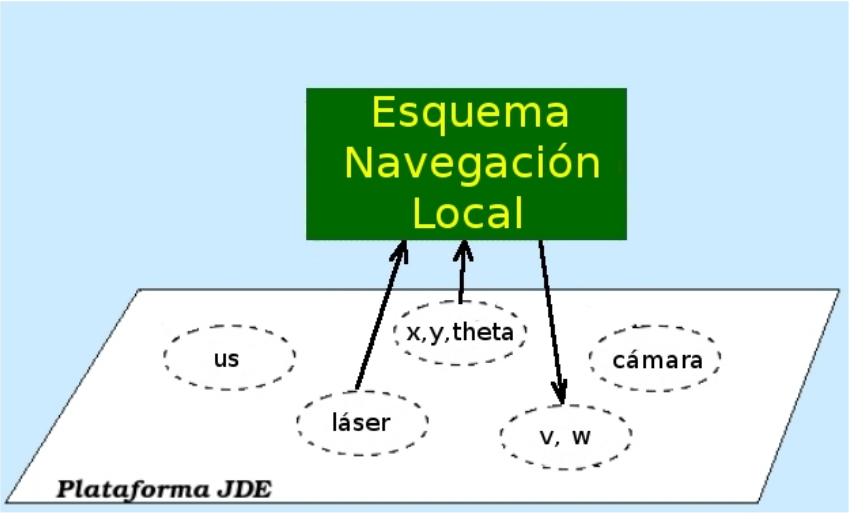
- Alta probabilidad en la zona estimada
- Probabilidad nula en el resto



(demo)

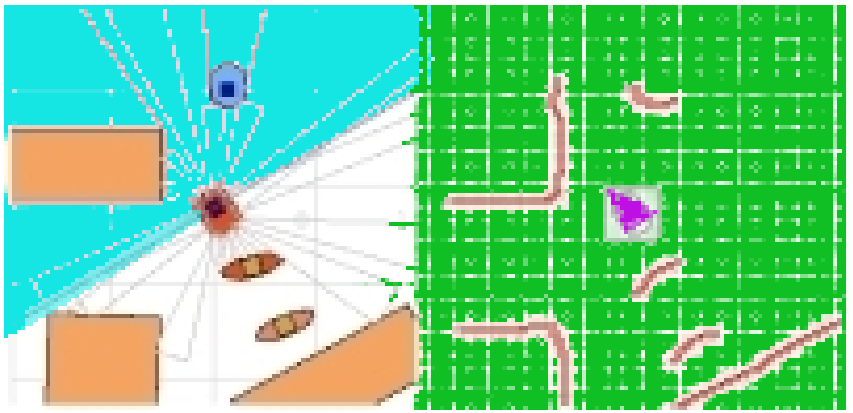
Navegación local (1/7)

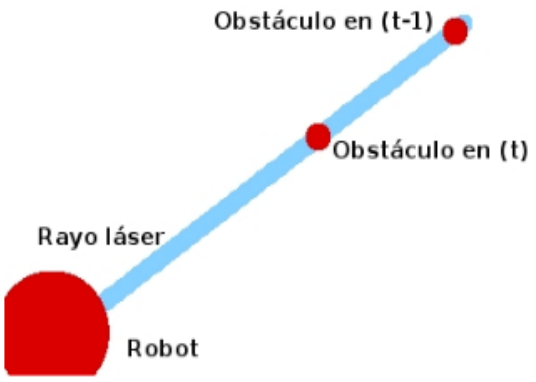
- Avanzar en cierta dirección evitando el choque
  - Con obstáculos, previstos o imprevistos, estáticos o dinámicos, ...
- Necesita que le vayan actualizando destino local
- Se guía según la información dada por el láser
  - Rejilla de ocupación, para acumular lecturas
- Tres técnicas distintas
  - *Gradient Path Planning* (*GPP*)
  - *Virtual Forces Field* (*VFF*)
  - *VFF* + Ventana de Seguridad

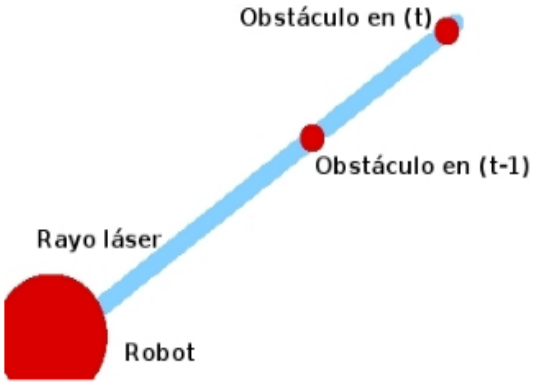


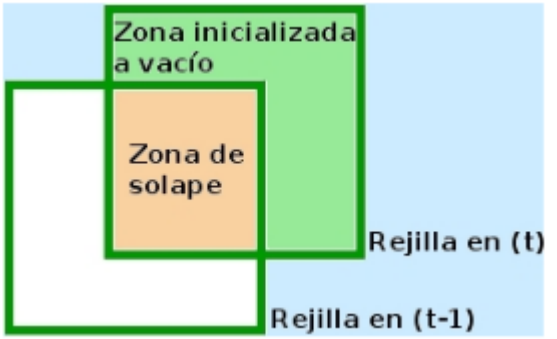
Navegación local

- Rejilla local. Mejor que basarnos en laser *(t)*
  - Memoria del robot. Se acumula información



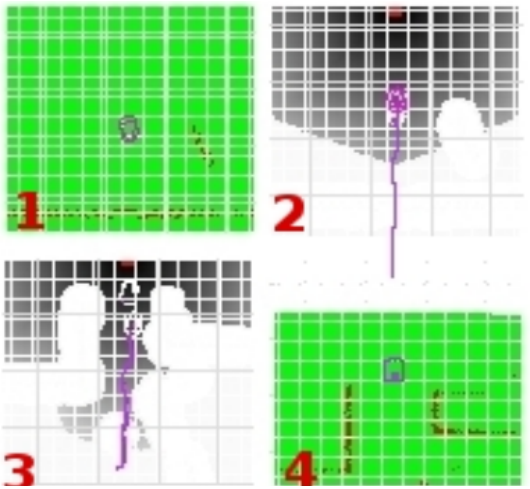


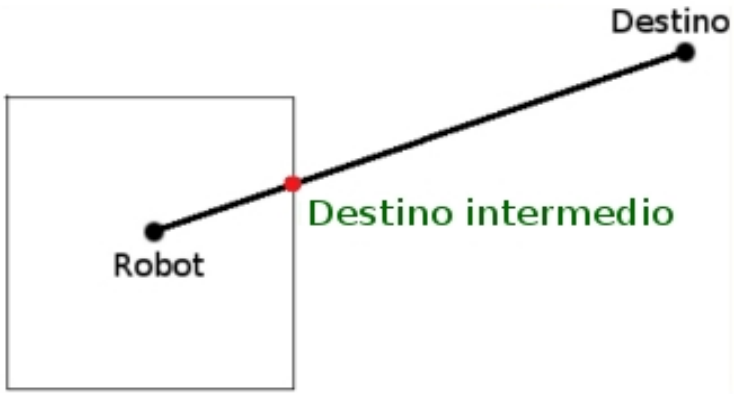




Navegación local (3/7)

- Método *Gradient Path Planning (GPP)*
  - Cálculo del gradiente sobre rejilla de ocupación
    - Recalcular campo con el movimiento del robot
      - ¿Cada cuánto? Solución: novedades de ocupación
  - Tener en cuenta obstáculos dinámicos
  - Destino local. Puede caer fuera de rejilla de ocupación

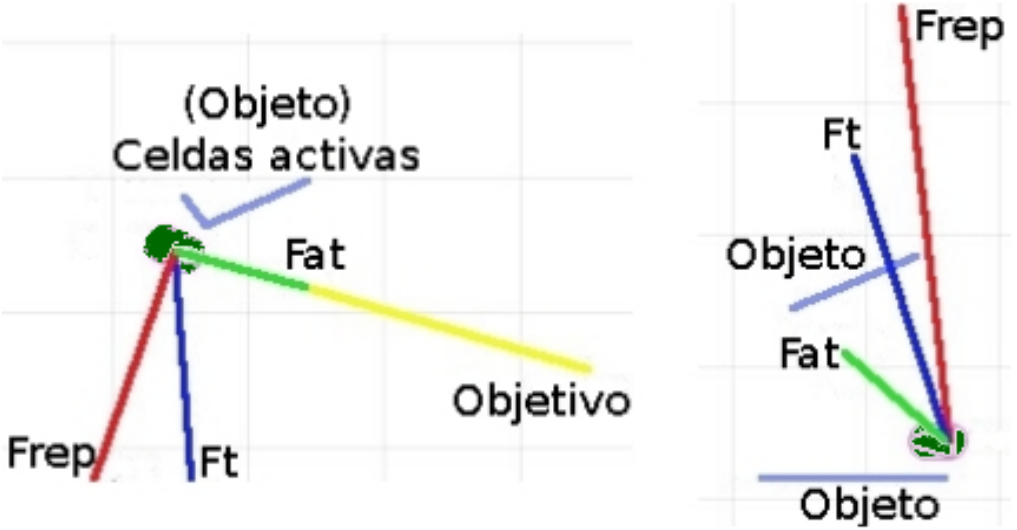




(vídeo)

Navegación local (4/7)

- Método *Virtual Forces Field (VFF)*
- El robot es atraído por la posición del objetivo
- Los obstáculos ejercen fuerzas repulsivas

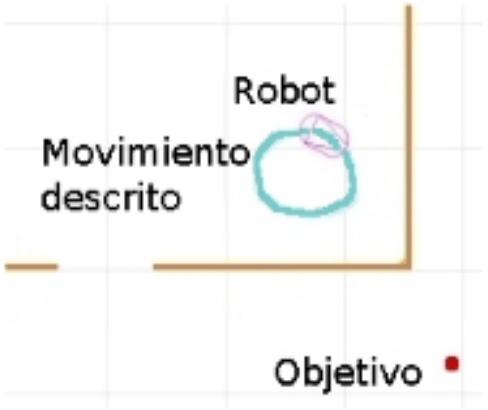


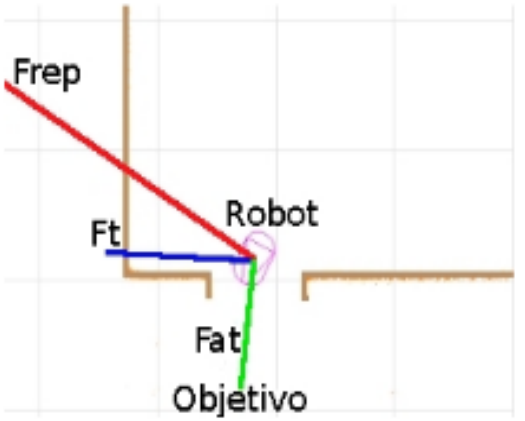
- Traducción de fuerzas a órdenes de velocidad

Navegación local

- Problemas *VFF*

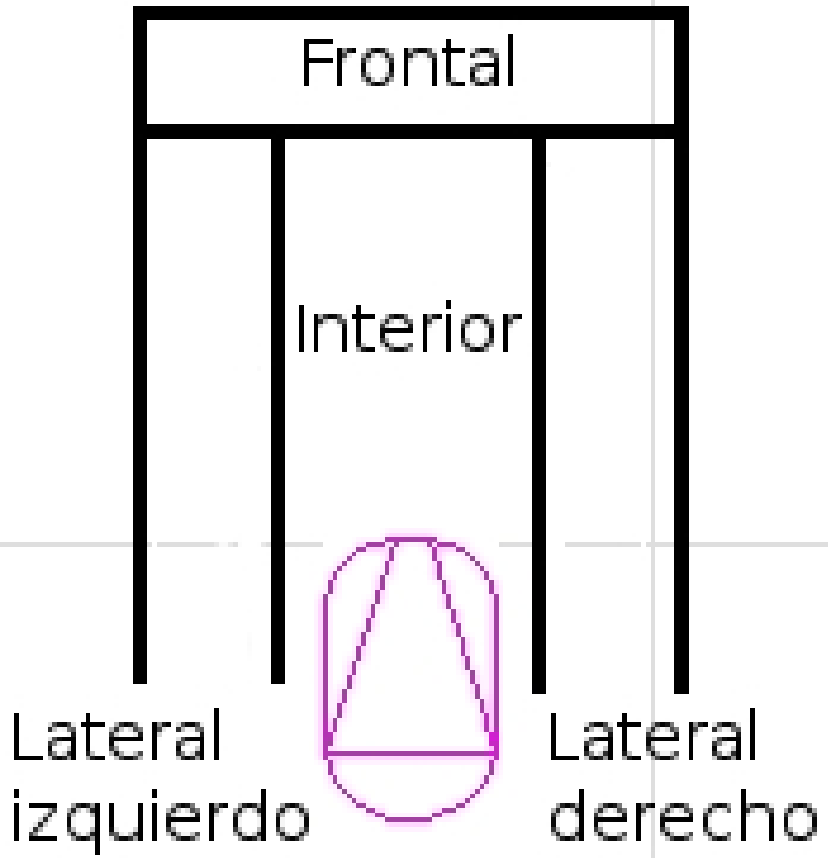
a) Oscilaciones b) Mínimo local c) Sitios estrechos





Navegación local

- Método *VFF* + Ventana de Seguridad



Si ocurre:

1º) No hay obstáculo en el *interior*

2º) No hay obstáculo en el *frontal*

3º) Hay obstáculo en algún *lateral*

Entonces:

$$v(t) = v_{max}$$

$$w(t) = 0$$

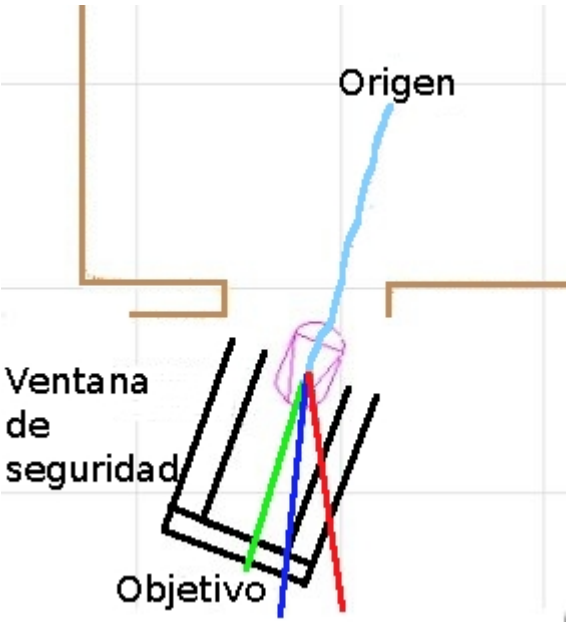
Si no: como *VFF*

Navegación local (7/7)

- Solución a los problemas de *VFF*

a) No hay oscilaciones

b) Salir por sitios estrechos



(vídeo)

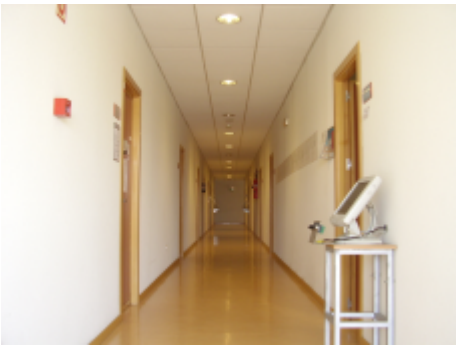
Conclusiones

- Desarrollados componentes básicos necesarios para una navegación autónoma del robot
  - Algoritmo de navegación global: *Gradient Path Planning*
  - Algoritmo de localización: filtro de partículas
  - Algoritmo de navegación local: *GPP* local, *VFF* y *VFF* + Ventana Seg.
- Implementación en 3 esquemas
  - Con más de 11.000 líneas de código
- Diversos experimentos sobre simulador y robot real

Líneas futuras

- Mejorar calidad del algoritmo de localización
  - Tratar con un espacio más amplio
  - Robustez ante espacios simétricos, sin marcas artificiales, etc.

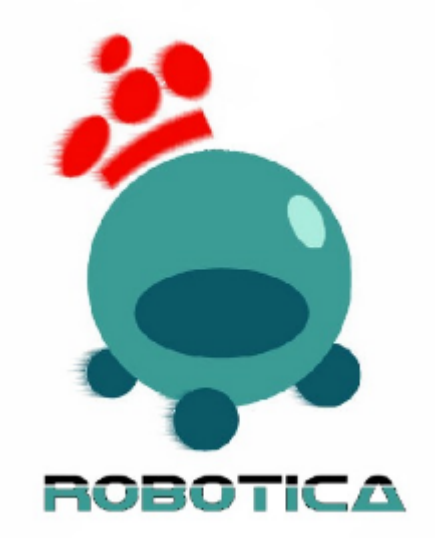





- Integración de los distintos componentes en un prototipo de *robot guía*
  - Robot guía del Dep. II

Navegación y autolocalización de un robot guía de visitantes

Julio M. Vega Pérez

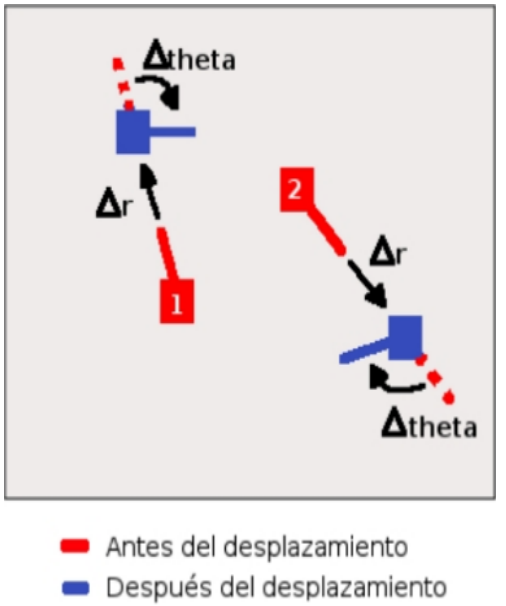


---

http://jde.gsyc.es/index.php/jmvega_guide_robot

Localización

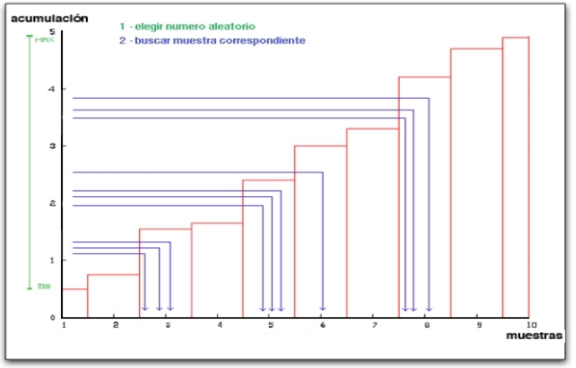
- *Modelo de movimiento*.
  - Este modelo captura información de posición
  - Se usa para medir giros y desplazamientos
  - La nube de partículas se moverá con un movimiento homólogo al medido para el robot
  - Es un modelo aleatorio, al que aplicamos un *ruido gaussiano*
    - Evitando así que partículas repetidas generen partículas en la siguiente población exactamente en la misma población



Localización

- *Remuestreo*.
  - La nube de partículas debe convergir a las posiciones más probables
  - Este paso aporta inteligencia a nuestro algoritmo
    - Generando poblaciones de partículas cada vez más concentradas alrededor de las posiciones compatibles
  - Algoritmo de la Ruleta
    - Muestreo aleatorio y uniforme de la acumulación de probabilidad
    - Seleccionando para la siguiente generación, la partícula correspondiente en el eje X

(a) Ruleta



(b) Acumulación de probabilidad

Localización

- *Modelo de observación*. Probabilidad

$$puntuacion_{dominante} = \frac{\sum_{i=0}^{pixeles_{resumida}} zonasiguales}{12}$$

$$puntuacion_{subdominante} = \frac{\sum_{i=0}^{pixeles_{resumida}} zonasiguales}{36}$$

$$probabilidad = \frac{(k\ puntuacion_{dominante}) + ((1-k)\ puntuacion_{subdominante})}{2}$$

Localización

- *Modelo de observación*. Probabilidad

$$puntuacion_{dominante} = \frac{\sum_{i=0}^{pixeles_{resumida}} zonasiguales}{12}$$

$$puntuacion_{subdominante} = \frac{\sum_{i=0}^{pixeles_{resumida}} zonasiguales}{36}$$

$$probabilidad = \frac{(k\ puntuacion_{dominante}) + ((1-k)\ puntuacion_{subdominante})}{2}$$