

Desarrollo de un robot para la recolección de residuos

JAR 2010
Ciudad de Buenos Aires
ITBA

3 de Noviembre de 2010

Temario

1 Introducción

- Por qué un robot recolector de residuos?

2 Arquitectura de procesamiento y comunicaciones

- Soporte físico
- Protocolo de comunicación

3 Arquitectura de comportamientos

- Cómo está compuesta?
- Simulación

4 Sistema de visión

- Cómo está compuesto?

5 Conclusiones

Principales motivaciones y objetivos

Por qué?

- La recolección de basura es un problema logístico de grandes ciudades
- La acumulación de basura alimenta animales transmisores de enfermedades
- Porque implica un desafío tanto físico como lógico
- Porque serviría como base y punto de partida a nuevos proyectos

Objetivos

Construir un robot móvil capaz de identificar y recolectar residuos en un ambiente dinámico como la terraza de la facultad. Desarrollar el soporte físico, protocolo de comunicación, estructura de comportamientos y sistema de visión.

Placas controladoras

Placa controladora de motores DC

Permiten establecer la velocidad y sentido de giro del motor, registrar la información del encoder del motor y conocer su consumo.

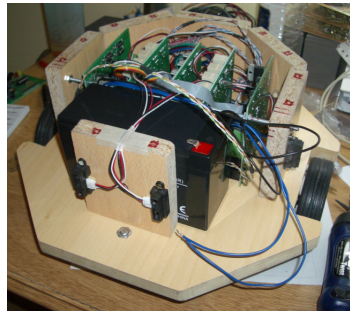
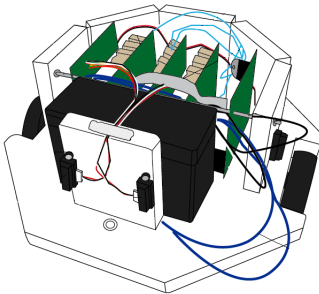
Placa controladora de sensores

Obtener información del entorno mediante distintos tipos de sensores. Utilizamos 8 telémetros GP2D120, 3 sensores de proximidad CNY70 y 1 sensor de distancia ultrasónico SRF05. También tomamos lecturas de la carga de la batería.

Placa controladora de servos

Diseñada pero no implementada. Controla hasta 5 servo motores digitales.

Diagrama interior, placas controladoras de motores y sensores



Estructura de los comportamientos: Subsumption

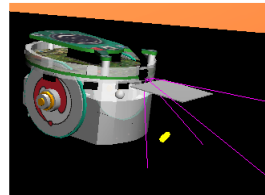
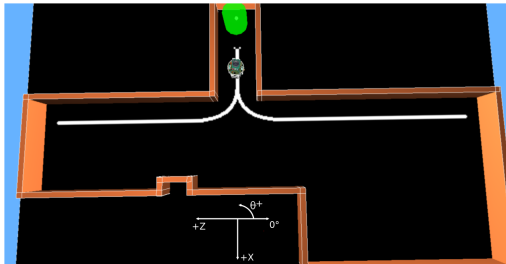
Arquitectura de comportamientos

- **Salir de situaciones no deseadas:** Evita el atascamiento.
- **Evitar obstáculos:** Acciona los motores según información de los sensores.
- **Recalibrar odometría:** Utiliza landmarks para calibrar su posición.
- **Recargar batería:** Mecanismo para recargarse en la base.
- **Ir a zona de recarga:** Navega hacia la línea mas cercana y luego realiza line-following para llegar a la zona de recarga.
- **Descargar basura:** Mecanismo de descarga.
- **Ir a zona de descarga:** Navega hacia la línea mas cercana y luego realiza line-following para llegar a la zona de recarga.
- **Recolectar basura:** Activa el mecanismo de recolección.
- **Ir hacia basura:** Desplaza al robot hacia las cercanías de un residuo.
- **Enfocar basura:** Desplaza al robot hasta que la basura se encuentre centrada.
- **Wander:** Recorre el entorno en bsqueda de residuos.

Simulación del ambiente

Herramienta

Para la simulación del ambiente utilizamos Webots.



Sistema de visión

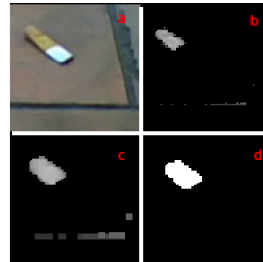
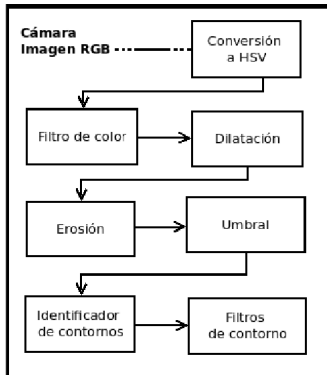
Elementos principales

Las imágenes captadas por la cámara son procesadas por una serie de algoritmos en C++ y la librería de visión computacional OpenCV.

Sistema de filtros

- **Filtro de color:** Elimina píxeles que no posean un color dentro del rango de interés.
- **Dilatación-erosión:** Eliminación de ruido, realce de características.
- **Umbral:** Elimina componentes débiles.
- **Identificador de contornos:** Extracción y representación de bordes con polígonos.
- **Filtros de contorno:** Analiza distintos parámetros (área, perímetro, excentricidad) para determinar si es un objeto de interés.

Estructura del algoritmo de visión y figura explicativa



Resultados de aplicar umbralado binario. (a) Captura original. (b) Salida del detector de color. (c) Salida de la operación de dilatación-erosión. (d) Salida de la operación de umbralado binario con un umbral de 100.

Conclusiones

Conclusiones

El testeo tanto del hardware como del software hasta ahora implementados son alentadores. Queda mucho trabajo por realizar pero creemos que la mayor contribución se encuentra en el proyecto de incorporación progresiva de tecnología para el diseño, construcción, experimentación e investigación en robótica e inteligencia computacional en del CENCOM del Departamento de Ingeniería Informática del ITBA.

SVN del proyecto

Página del proyecto: <http://code.google.com/p/tpf-robotica/>