

Open IEEE: Robot Limpiador de Playas

Latin American Robotics Competition 2013

Saúl Mijael Ibaven Bueno

Ing. en Sistemas Digitales y Robótica
ITESM Campus Monterrey
Monterrey, Nuevo León
saul.ibaven@gmail.com

Emmanuel Cortés Martínez

Ing. en Sistemas Digitales y Robótica
ITESM Campus Monterrey
Monterrey, Nuevo León
Emm670@hotmail.com

Alicia Elizabeth Hernández Juan

Ing. en Sistemas Digitales y Robótica
ITESM Campus Monterrey
Monterrey, Nuevo León
alis_bentley@live.com.mx

María del Pilar Solórzano Velasco

Ing. en Sistemas Digitales y Robótica
ITESM Campus Monterrey
Monterrey, Nuevo León
pilar.solorzano9@gmail.com

Joel Pérez Orona

Ing. en Sistemas Digitales y Robótica
ITESM Campus Monterrey
Monterrey, Nuevo León
joelp.orona@hotmail.com

Abstract - Este documento tiene como propósito exponer la solución diseñada por estudiantes de ingeniería en Sistemas Digitales y Robótica, para la creación de un robot autónomo que pueda recolectar basura en una playa y depositarla en un contenedor.

Keywords – Robótica; Visión; Basura; Electrónica; Mecánica; Sensores; Control; Ecología;

I. Introducción

La contaminación de playas en América Latina y el Caribe constituye un serio problema debido a que la basura contiene grupos contaminantes como metales traza, plásticos, materia flotante, entre otros. Según el Ocean Conservancy's International Coastal Cleanup [1], quienes recogen basura de playas y las categorizan, entre los 10 principales objetos recolectados de una muestra de once millones de artículos, se puede encontrar las botellas de bebidas en tercera posición con 1, 065,171 unidades y en octavo lugar las latas de bebidas con 338,875 unidades.

Desarrollar robots autónomos recolectores de basura es un buen acercamiento para reducir la contaminación. A lo largo de éste proyecto se ha buscado reducir la basura en la arena, enfocado principalmente en la recolección de latas como primer meta y posteriormente poder ser extrapolado a otros tipos de basura. La implementación de estos robots implica un gran reto debido a los algoritmos y superficie en la que deben trabajar, además del consumo energético que tienen ya que deben trabajar remotamente y sin interacción humana. Se busca que los robots no interfieran con el momento recreacional de los visitantes a

las playas, de tal forma que evitan cualquier colisión con objetos y personas.

II. Desarrollo

A. Diseño Mecánico

La estructura del robot está formada por un chasis central constituido por un perfil de aluminio de 2 X 1.5 pulgadas, sobre este están montados los soportes para los 4 motores que dan la tracción y el movimiento al robot. Los motores están montados en perfiles de aluminio de 1.5 X 1.5 pulgadas y unidos al eje central mediante un perfil de aluminio de 1 X 2 pulgadas que están remachados a la estructura principal en un ángulo cercano a los 45°; el propósito de esto es dar mayor resistencia a la estructura que soportara la electrónica y el contenedor para almacenar las latas, así como dar mayor altura al robot para facilitar el vaciado de las latas al contenedor final.

El robot cuenta con 4 motores, dándole una tracción de 4X4, esto tiene la finalidad de buscar darle un mayor empuje al robot y evitar que el robot se atasque en la arena. Los motores utilizados son motores de corriente directa, de 65 revoluciones por minuto y con una alimentación entre 8 y 12 volts. Los motores están fijados directamente sobre el chasis del robot, cada uno en su propia caja de protección, elaborada con aluminio y fijados a esta mediante el uso de 3 ángulos de aluminio y tornillos genéricos, estos mantienen al motor inmóvil para evitar problemas con la tracción del robot. A estos motores se les acoplaron llantas de caucho de 12 cm de diámetro y un ancho 5 cm.



Fig. 1. Base con motores y llantas.

Para mejorar la tracción del robot sobre la arena, se diseñaron un par de orugas para acoplar sobre la llantas. Las orugas se construyeron utilizando para cada una 2 cadenas de bicicleta de aproximadamente 70 cm de largo a las cuales se unieron mediante trozos de 9cm de PBC de media pulgada de espesor, los cuales fueron partidos a la mitad y fijados de sus extremos mediante tornillos de 1 pulgada de largo a cada una de las cadenas de bicicleta.

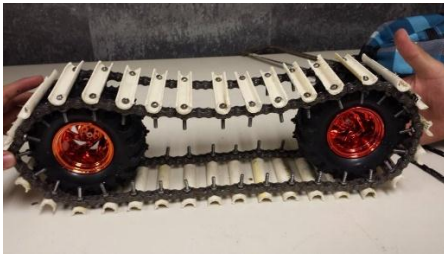


Fig. 2. Oruga del robot

Para la elaboración del gripper para la recolección de la latas, se utilizaron 2 servomotores para el eje principal que levanta el gripper del suelo, y otros dos para completar la acción de apertura y cierre del gripper. Los servomotores del eje principal se montaron directamente sobre la base de aluminio y se fijaron mediante tornillos. Por otra parte los servomotores para la acción de cierre se montaron sobre el eje de los servomotores previamente mencionados y sobre estos se acoplo un eje de aluminio sobre el que se montó una malla metálica para así generar el gripper. El gripper se levanta y baja con los primeros 2 servomotores y se abre y cierra mediante los segundos dos, así se completa la acción de recolección de latas y su posterior depósito en el contenedor del robot.



Fig. 3. Robot ensamblado con gripper

B. Diseño Electrónico

Dentro del área de diseño electrónico nos encontramos con dos vertientes principales, por un lado tenemos el diseño electrónico para el control de los actuadores del sistema, y por el otro el área de control electrónico.

En este proyecto la electrónica consistió en diferentes circuitos que nos permitieron desarrollar y controlar los motores y rendimiento de la luz para la visibilidad del robot.

Control de motores de CD

El control de motores de CD está basado en el Puente H L298 que es capaz de manejar dos motores de una manera eficiente y sencilla a partir de dos entradas por motor proporcionadas por la tarjeta “Raspberry Pi” [2] y que guiará a los motores de una manera conveniente.

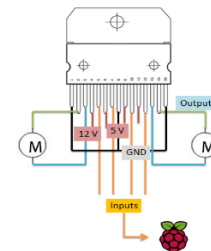


Fig. 4. Esquema de control de motores mediante un puente H

Sistema de iluminación

El amplificador operacional en modo comparador se usó para detectar la presencia o ausencia de luz, censada a través de una fotorresistencia o LDR.

El relay está cerrado con la ausencia de luz y se abre cuando se detecta luz, es decir cuando el voltaje cambia.

La sensibilidad es controlada por el potenciómetro y el diodo previene los picos inversos para el relay.

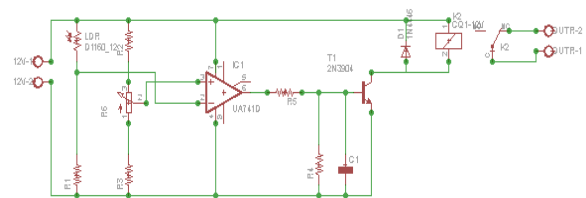


Fig. 5. Esquemático para el sistema de iluminación

Sistema de Control

Para controlar los actuadores usados en el robot se usó una tarjeta “Raspberry Pi” (Figura 6) y diferentes drivers para conectarla con la estructura mecánica.



Fig. 6. "Raspberry Pi"

La "Raspberry Pi" es una tarjeta que permite al usuario usarla para diferentes aplicaciones. Nosotros la usamos como un controlador. La Pi fue programada en Python.

La tarjeta supervisaba motores de DC, servomotores y algunos sensores. También fue conectada via puerto serial a la EPIA-N800, la cual es la encargada de decidir lo que realizarán los motores y sensores conectados a la PI.

Se utilizaron diferentes librerías tanto de python como algunas específicas de la PI para lograr la correcta comunicación entre ella y los dispositivos. También se usaron los GPIOs de la tarjeta que se pueden observar en la figura 7.

PI			P5		
<50mA	3V3	1	2	1	SV
BCM GPIO00/02	SDA0/I	8	3	4	SV
BCM GPIO01/03	SCL0/I	9	5	6	GND
BCM GPIO04		7	7	8	15 TX
	GND	10	9	10	16 RX
BCM GPIO17		0	11	12	1 PWM0
BCM GPIO21/27		2	13	14	GND
BCM GPIO22		3	15	16	4
<50mA	3v3	17	17	18	5
BCM GPIO10	SPIMOSI	12	19	20	GND
BCM GPIO9	SPIMOSO	13	21	22	6
BCM GPIO11	SPI SCLK	14	23	24	10 SPI CE0 N
	GND	25	25	26	11 SPI CE1 N
P5			P6		
<50mA	3V3	2	1	17	SDA0
BCM GPIO29	SCL0	18	4	3	19
BCM GPIO31		20	6	5	GND
	GND	8	7		

Fig. 7. Puertos de Entrada/Salida de la "Raspberry Pi"

A continuación se da una breve explicación de cada módulo conectado a la tarjeta.

- Control de un motor de DC

Para controlar un motor de DC, la "Raspberry Pi" fue conectada a un puente H y éste último fue conectado al motor, de esta manera podían cambiarse tanto la velocidad como la dirección del mismo. La PI también necesitó hacer un llamado especial a sus GPIO (General Purpose I/O) a través de un paquete llamado RPi.GPIO

- Control de un servomotor

Para controlar el servomotor se tuvo que instalar un módulo avanzado de GPIO que contiene un paquete especial el cual es capaz de usar el PWM en diferentes pines, llamada RPIO. Esta librería tuvo que ser usada debido principalmente a que el paquete RPi.GPIO solamente permite usar el PWM en un pin.

- Control de los sensores

Para controlar los sensores se utilizó el paquete RPi.GPIO, donde se hacía la lectura del GPIO al que se conectan los sensores y de esta manera se revisa si hay un cambio de voltaje. Si dicho cambio es detectado entonces se activa una interrupción que manda un comando vía serial a la EPIA para que ella lo maneje.

- Comunicación serial

Esta comunicación se realizó utilizando la librería para usar el puerto serial de python, también se necesitó de un circuito que usa un Max3232 para convertir el voltaje y de esta manera poder comunicar ambas tarjetas sin dañarlas.

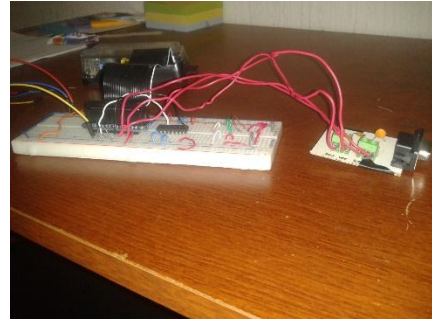


Fig. 8. Circuito MAX3232 usado para la comunicación serial entre la computadora y la "Raspberry Pi".

C. Sistema de Visión

El reconocimiento de objetos es una de las tareas cruciales en el desarrollo del robot, por lo que la implementación de un sistema de visión computarizado es imprescindible.

Para integrar un sistema de visión al robot, tres factores son importantes:

- Computadora:

Es importante contar con un sistema que tenga el poder suficiente para el análisis y procesamiento de imágenes en tiempo real. La computadora debe tener la capacidad de incorporar una cámara, ya sea por medio de usb o algún puerto especial. Para el proyecto se optó por la tarjeta "EPIA-N800" de la compañía "Via Embedded" que puede apreciarse en la Figura 9.



Fig. 9. EPIA-N800. VIA Embedded Corp

- Cámara:

La cámara es un factor importante ya que es por este medio que obtendremos las imágenes que nos interesa analizar. La cámara elegida para el desarrollo del proyecto es de la compañía “Acteck” y el modelo es el “ATW-1200” que puede apreciarse en la Figura 10, esta cual provee de imágenes de alta definición al sistema.



Fig. 10. Acteck Lynx View ATW-1200

- Librerías de Visión:

Ya que se tienen los elementos de hardware es necesario contar con librerías de software que nos permitan extraer, manipular y analizar las imágenes de la cámara. Las librerías elegidas para el desarrollo del proyecto corresponden a “OpenCV” [3], la cual provee un gran número de funciones y algoritmos para la implementación de un sistema de visión computarizado, añadiendo que su licencia es gratuita, de tal manera que cambios pueden hacerse fácilmente para acoplarla al diseño personal. Es importante decir que el desarrollo del sistema de visión se hizo completamente en el lenguaje de programación de alto nivel “C++” en el sistema operativo “Windows”.

Una vez que se integraron estos tres factores, se procede al análisis y procesamiento de los objetos de interés en el área de competencia. Los tres objetos que deben caracterizarse son las latas, el contenedor de basura y el agua circundante.

Los tres objetos tienen colores característicos, por lo que la estrategia de reconocimiento se basa principalmente en el color de los mismos y en sus momentos.

Al estar trabajando con colores, es más fácil hacer uso del sistema de colores “Hue-Saturation-Value” (HSV), el cual nos permite variar sus tres componentes hasta encontrar el color deseado y localizarlo en la imagen. La representación del modelo de color puede verse en la Figura 11 a continuación.

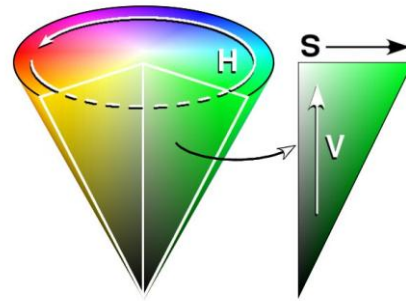


Fig. 11. Modelo de color HSV

Podemos observar que el modelo es representado por medio de un cono, en el cual el matiz (Hue) va en un rango de 0 a 360°, mientras que la saturación (Saturation) y la luminosidad (Value) van de un valor de 0 a 1. Sin embargo estos valores son representados por valores distintos en la librería de OpenCV, en los cuales el matiz va de un valor de 0 a 180, y la saturación y la luminosidad van de un valor de 0 a 255.

La técnica para el reconocimiento se basa en cinco pasos principales:

1. Extraer la imagen de la cámara y guardarla en memoria, luego convertir la imagen del estándar RGB al modelo de color HSV. En la Figura 12 se muestran ambas imágenes.

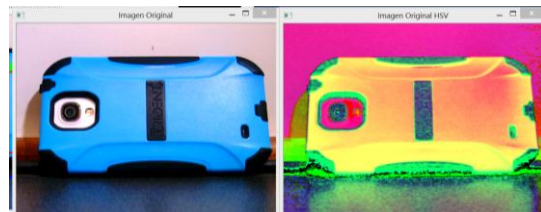


Fig. 12. Imagen Original (Izquierda) e Imagen en HSV (Derecha)

2. Aplicamos un desenfoque gaussiano que nos permita suavizar las características de la imagen.

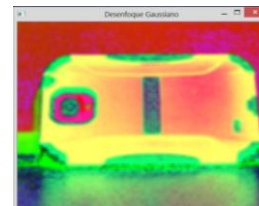


Fig. 13. Imagen con desenfoque gaussiano.

3. Aplicar una operación de binarización con los valores de HSV acorde al color que se desea encontrar. En la Figura 14 podemos ver el resultado de la binarización al seleccionar el color azul.

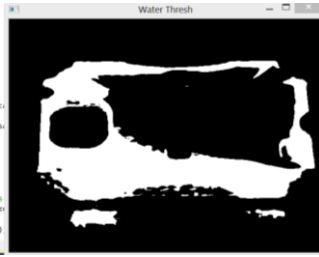


Fig. 15. Imagen HSV binarizada con enfoque en el color azul.

4. Aplicar una operación morfológica de dilatación de tal manera que ensanchemos un poco más los objetos que queremos identificar.

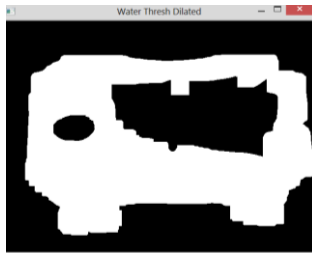


Fig. 16. Imagen binarizada dilatada para hacer más grandes los rasgos del objeto.

5. Encontrar los contornos y áreas de todos los elementos con el color especificado en la binarización.

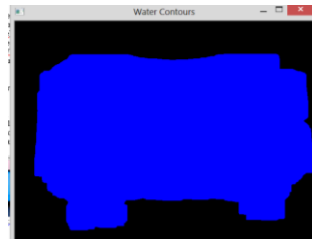


Fig. 17. Área del objeto identificado en base a sus contornos.

Los pasos anteriores nos permiten identificar todos los elementos del color seleccionado en la imagen, que para el caso específico del contenedor y de la región de agua son suficientes. Para el caso específico de las latas es necesario que se agregue un paso más:

6. Comparar las áreas de los objetos identificados, de tal manera que el objeto con mayor área sea el único que sea identificado en la imagen. Con esto aseguramos que en la mayoría de los casos el robot busque el objeto que está más cerca de él. Ejemplo de lo anterior puede observarse en la Figura 18, en donde se puede ver como el color negro de los contenedores es filtrado y binarizado, para luego identificar cuál de los dos está más cercano al robot.

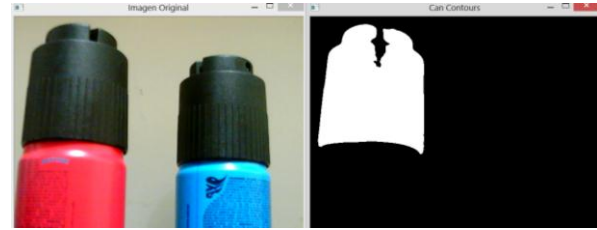


Fig. 18. Imagen original (Izquierda) y objeto más cercano detectado.

Ya que hemos identificado el objeto que está más cerca, tan solo es cuestión de ubicar el centro del mismo y mover el robot de tal manera que el objeto se encuentre dentro de una región importante en el mapa definido para su localización (Figura 19). Dependiendo de la posición del objeto serán las acciones llevadas a cabo por el sistema para poder llegar a él y recolectarlo.

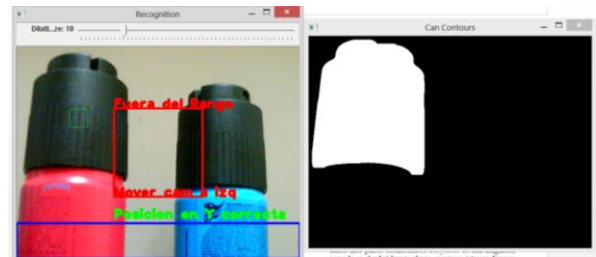


Fig. 19. Localización de centro de objeto y mapa de coordenadas importantes (Izquierdo) y objeto localizado con dilatación (Derecha).

IV. Conclusión

El estado actual de la tecnología ha permitido implementar cada vez dispositivos más complejos y capaces de realizar diferentes actividades de acuerdo a las necesidades presentadas. La robótica es una disciplina que se encuentra aún en la etapa de explotación en las economías emergentes y por lo tanto es importante impulsar a las personas a adoptar los cambios tecnológicos lo más rápido posible para evitar el rezago que existe ante países desarrollados. Proyectos de alta magnitud, como lo es el robot limpia playas, requieren integrar diversas disciplinas para crear un producto exitoso. Se requieren sistemas de visión, diseño, materiales; son proyectos interdisciplinarios y por lo tanto para obtener lo mejor de cada área debemos apoyarnos en la experiencia de distintas especialidades. El desarrollo de robots autónomos es una gran área que aún tiene mucho futuro debido a que permite que éstos realicen tareas que en ciertas ocasiones resultan demasiado desgastantes para los humanos, que se encuentran en zonas de alto riesgo como para exponer a las personas o que requieren una precisión con la que no se cuenta normalmente con los obreros. La robótica se ha ido expandiendo hasta llegar a todas las disciplinas, se puede encontrar desde la fabricación de dispositivos hasta en los hogares con robots limpiadores.

Agradecimientos

Agradecemos al Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Monterrey y a la Dirección de la Escuela de Ingeniería y Tecnologías de la Información, por su apoyo en la elaboración de este proyecto.

Referencias

- [1] Ocean Conservancy. "Top 10 Items Found". [En Línea] Disponible en: <http://www.oceanconservancy.org>
- [2] Raspberry Pi. [En Línea] Disponible en: www.raspberrypi.org
- [3] Laganiere, Robert. "OpenCV2 Computer Vision Application Programming Cookbook". PACKT Publishing. Birmingham, UK. May 2011
- [4] Richardson, Matt and Wallace, Shawn. "Getting started with Raspberry Pi". O'Reilly Media, Inc. First Edition. USA. December 2012.
- [5] Lélis Baggio, Daniel. Millán Escrivá, David. Mahmood, Naureen. Shilkrot, Roy. Emami, Shervin. Ievgen, Khvedchenia. Saragih, Jason. "Mastering OpenCV with Practical Computer Vision Projects". PACKT Publishing. Birmingham, UK. December 2012.
- [6] Dunn, Willam C. "Introduction to Instrumentation, Sensors, and Process Control". ARTECH HOUSE, INC. Norwood, MA.
- [7] Sandler, Ben-Zion. "Robotics: Designing the Mechanisms for Automated Machinery". Second Edition. Academic Press. 1999.
- [8] Shapiro, Linda G. and Stockman, George. "Computer Vision". Prentice Hall. February 2001.