

Visión y Robótica con Software Libre

Vazquez *Raimundo*^a, Cabral *Gabriel*^a, Araguas *Gaston*^b, Canali *Luis*^b, Mason *Leoncio*^a, Cabral *Jorge*^a.

^aGrupo Universitario de Automatización. UTN Facultad Regional Resistencia.
French 414, 3500 Resistencia Argentina.
ray_vazquez_2005@hotmail.com

^bUTN Facultad Reginal Cordoba.
Maestro M. López esq, Cruz Roja Argentina, Ciudad Universitaria, 5016
Córdoba, Argentina.
garaguas@gmail.com

Resumen En el siguiente trabajo se describe una metodología basada en el paradigma estructurado de desarrollo de sistemas del tipo simplificado con la finalidad de resolver una situación problemática conceptual en forma sencilla y eficiente. El procedimiento utilizado es la articulación de librerías de edición de video, software de robótica y un hardware mediante el lenguaje de programación ANSI C, herramienta de compilación para linkeo y depuración en plataforma GNU/Linux. Se logro suavizar la respuesta de control del algoritmo de seguimiento de un objeto mediante la utilización de un controlador proporcional integral derivativo debido a que la implementación del mismo logra acoplar la librería de edición de video y algoritmo de control con la respuesta inercial de la plataforma móvil.

Palabra clave: *plataforma móvil, edición de video, automatismo y control.*

1. Introducción

La robótica es la ciencia y la tecnología de los robots, permite combinar diversas disciplinas como por ejemplo la mecánica, la informática, la electrónica, la matemática, la inteligencia artificial y la ingeniería de control para realizar el diseño, construcción y aplicaciones de los robots. Se pueden destacar en este tema, los siguientes trabajos de investigación Peña et al. [1], Muños et al.[2], Vazquez et al.[3].

Los desarrollos planteados brindan múltiples plataformas de experimentación para los investigadores en las áreas de robótica y sus afines. La compra o adquisición de dispositivos de prueba de éste tipo, trae asociados altos costos, por trámites de importación, adquisición de repuestos, licencias de software propietario, soporte técnico y presupuesto reducido.

En el siguiente trabajo se desarrolla un procedimiento para facilitar tareas de control y prueba de algoritmos de seguimientos, mediante la utilización de un software para robótica llamado stage-player, librerías de edición de video y una plataforma robótica móvil representada en la figura 1.

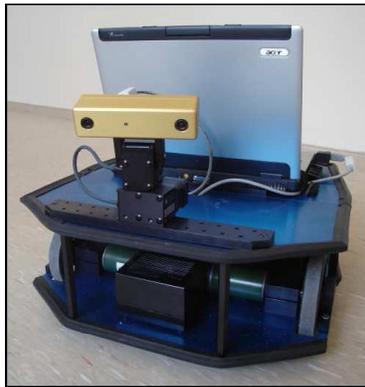


Figura 1. Plataforma móvil de experimentación RoMAA II.

2. Conceptos Fundamentales

El dispositivo denominado RoMAA II fue desarrollado íntegramente en el Centro de Investigación en Informática para la Ingeniería [4], surge como respuesta a la necesidad de disponer de una plataforma sobre la cual ensayar y validar las investigaciones llevadas a cabo en él.

Desde el inicio fue concebido con arquitectura abierta, pudiendo acceder a cada nivel jerárquico de su estructura para modificar o adaptar a los diferentes experimentos. La estructura es de tipo uniciclo con dos ruedas motoras y una

rueda castor atras, de dimensiones adecuandas para ambientes interiores de laboratorio; con posibilidad de montar diferentes tipos de sensores y actuadores de manera sencilla en la parte frontal de robot.

3. Proyecto Player / Satge

El proyecto Player/Stage es un esfuerzo internacional para producir herramientas de software libre para investigación y desarrollo en el mundo de la robótica [5].

Se utiliza el lenguaje de programación C [6] y se elige la plataforma player [7] debido a las siguientes características:

1. Se encarga de gestionar con los controladores de los dispositivo y la comunicación es transparentes.
2. Posee un servidor de dispositivos.
3. Basado en sockets TCP/IP.
4. Estructura multi hilos.
5. Clientes proxy (cualquier lenguaje).
6. Lectura de sensores.
7. Contempla librerías oficiales en C, C++, Python de terceros en Java, Octave, Matlab, etc.
8. Incluye drivers de sensores, actuadores y robots comerciales como por ejemplo: Robots Pioneer, iRobot Roomba, Segway RMP y cámaras de video.

Como complemento del player se utiliza un simulador virtual denominado stage [8] cuyas características principales son:

1. Simulador de múltiples robots.
2. Player los maneja como robots reales.
3. Dispone de muchos modelos de sensores y actuadores comerciales como por ejemplo el Laser Rangefinder.
4. Dispone modelos de cámaras como por ejemplo cámaras con sensores sonares.
5. Los Modelos son simples de utilizar y de bajo costo computacional.
6. Los ambientes se desarrollan en 2D y 2.5D.

El player y el stage estan incluidos en la instalación del sistema operativo GNU/Linux. Un ejemplo del programa stage se visualiza en la figura 2.

Se observa en el esquema un vehículo que se encuentra en un sector de la planta (por ejemplo una industria) y debe dirigirse a otro lugar utilizando algoritmos de navegación. La solución del problema se obtiene mediante programas escrito en language C implementados en el player.

De esta manera, el stage y el player representan una herramienta útil para resolver situaciones problemáticas reales, debido a que los diferentes dispositivos que intervienen en la escena virtual pueden ser representados como un modelo experimental a escala reducida, y las respuestas esperadas se contrastan con las

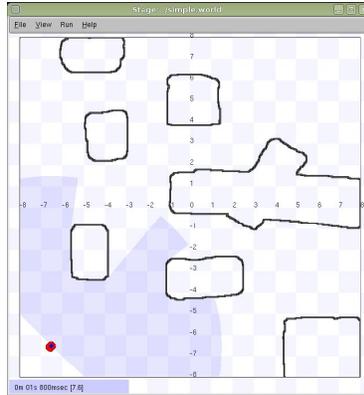


Figura 2. Simulación Stage.

obtenidas en el proceso de prueba y puesta a punto de los algoritmos, permitiendo disminuir el tiempo y esfuerzo en el desarrollo de software para el programador.

3.1. Librerías de Edición de Video OpenCV

OpenCV es una biblioteca libre de visión artificial [9] originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones.

Se lo utiliza desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Su licencia permite su uso en forma libre para propósitos comerciales o de investigación.

OpenCV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos, calibración de cámaras, visión estereo y visión robótica.

El proyecto encargado de desarrollar OpenCV pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado, realizando su programación en código C y C++ optimizados, aprovechando además las capacidades que proveen los procesadores multinúcleo.

4. Hipótesis de Trabajo

La implementación de un procedimiento para relacionar librerías de visión artificial, software de robótica y hardware permite resolver algunas situaciones problemáticas en un entorno de trabajo real o virtual en forma eficiente.

La articulación de las librerías OpenCV, software de robótica y el hardware del RoMAA II generan un nuevo sistema que permite disminuir tiempo y esfuerzo en la resolución de problemas en el área del control automático.

La plataforma móvil al igual que los desarrolladores al ser testeada previamente proporcionan mayor estabilidad en el desempeño y eficiencia que si se lo hubiese creado desde cero.

La utilización de los desarrolladores, hardware y plataforma GNU/Linux son fáciles de implementar y sencillos de utilizar.

5. Metodología

La metodología aplicada se basa en el paradigma estructurado de desarrollo de sistemas [10], que en este trabajo de investigación es del tipo simplificado.

Se plantea al problema en alto nivel, es decir que se abstrae de cualquier tipo de implementación. Luego se define las tareas a realizar, es decir, captura de imagen, detección, procesamiento, elección de alternativas de decisión, transmisión de órdenes al dispositivo físico.

Para desarrollar las tareas descritas se debe realizar los siguiente pasos:

Se analizan las tareas mencionadas, luego se exploran cada una de ellas con un mayor nivel de detalle. Del análisis mencionado se extrae las funciones básicas candidatas para implementar en el software.

Luego se agrupan las tareas en forma conveniente para evitar redundancias. Seguidamente se eligen aquellas candidatas que mejor se adapten a las tareas de control automático. En consecuencia se analiza las posibles implementaciones, seleccionando las librerías para realizar cada función, el lenguaje a utilizar y el software de base. Los criterios de selección evaluados para el desarrollo del algoritmo son, performance, robustez, libertades de la licencia de distribución y compatibilidad.

La captura y detección de imágenes se obtiene de la librería OpenCV, la cual permite una gran flexibilidad para identificar diversos tipos de objetos de variadas formas y colores.

Se implementa las tareas de detección utilizando el lenguaje ANSI C, el toolchain GNU para la compilación y linkeo para la depuración.

El sistema operativo de trabajo es GNU/Linux, más específicamente la distribución denominada Ubuntu.

6. Procedimiento

Se estudia la posibilidad de compatibilidad entre las librerías seleccionadas con el robot. Para ello se realizaron varios ensayos con el RoMAA II para detectar afinidad en la interconexión. Las pruebas fueron satisfactorias, dando comienzo al desarrollo del proyecto.

Empleando el OpenCV se realiza las siguientes tareas para evaluar la funcionalidad en la captura y procesamiento de imágenes. Para ello se evaluó el

performance en la obtención de datos de la cámara web, conversión de imagen al espacio de color necesario, ejecución de operaciones de transformaciones para procesar la imagen, realización de la detección del objeto y obtención del baricentro del objeto.

El funcionamiento y desempeño de la librería OpneCV en las tareas realizadas fueron satisfactorios. Seguidamente se recurre a la heurística para elegir la mejor decisión según los datos enviados en las tareas de obtención del radio y baricentro del objeto.

La decisión efectuada se transmite al robot RoMAA II, los cuales decodifican las instrucciones recibidas para realizar las acciones de control correspondiente. Este proceso se realiza varias veces por segundo, por lo que es necesario mantener el consumo de CPU y ram controlados por el procesamiento intensivo que realiza OpenCV, produciendo una latencia en la respuesta que requiere de heurísticas correctivas adicionales, las cuales no pudieron capturar en primera instancia con una simulación, sino que se hicieron visibles al efectuar las pruebas con el robot.

La naturaleza del proceso iterativo y el control constante del objeto a detectar se asemeja al desarrollo de video juegos, por lo que se elige utilizar una técnica denominada Game-Loop, la cual consiste en realizar el procesamiento como si se tratasen de imágenes estáticas. La capacidad de procesamiento de las computadoras actuales permite realizar este proceso varias veces por segundo, lo que se traduce en una respuesta fluida del proceso.

La principal dificultad reside en que las heurísticas deben adaptarse a las condiciones reales de trabajo, donde variables externas afectan al funcionamiento del modelo ideal del sistema.

La conexión con el robot se realiza mediante el servidor player, el cual se conecta con el software de control por medio de un puerto TCP. El player permite conectarse tanto a stage (para simular un robot en la PC) como también con el RoMAA II, cada uno requiere de un driver distinto pero son totalmente intercambiables.

El RoMAA II posee un driver para player el cual hace que su funcionamiento sea totalmente transparente, todas las órdenes son enviadas desde el software hacia player y mediante un algoritmo de automatización se logra controlar el funcionamiento del hardware en forma más versátil.

La ejecución del programa se realiza mediante el empleo de consolas, seguidamente se visualiza en la pantalla dos ventanas. En la primera, se muestra el objeto resaltado por un contorno de color circunscripto en él, en la segunda ventana se observa el cuerpo extraído del video mediante técnicas de edición de video de las librerías OpenCV.

7. Implementación

En una primera experiencia se logra conectar la librería OpenCV y el simulador stage-palyer. El RoMAA II es representado en forma virtual de color verde como se muestra en la figura 3.

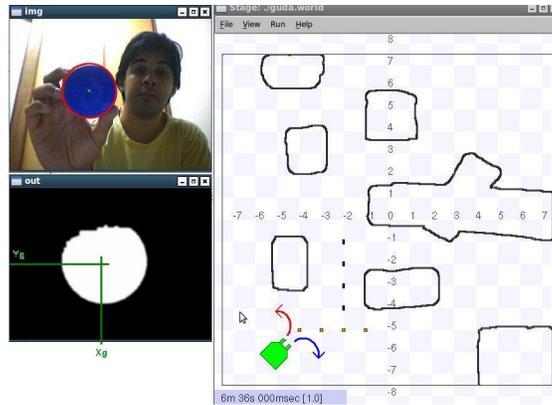


Figura 3. Librería OpenCV y stage-player

Seguidamente se realiza la operación de detección del objeto de color azul y morfología conocida en tiempo real, dicha operación se visualiza en la figura 4. Luego las coordenadas del baricentros son enviadas desde el OpenCV hacia el stage-player y simples rutinas escritas en lenguaje C permiten comparar los



Figura 4. Detección del objeto y su baricentro.

parámetros enviados en relación al centro de referencia de la imagen capturada.

En el primer ejemplo las coordenadas del objeto se encuentran a la izquierda del robot, por ese motivo el movimiento de RoMAA II es hacia la izquierda (flecha roja). Si el objeto se hubiese colocado a la derecha, en ese caso, RoMAA II giraría hacia la derecha (flecha azul). También se observa en la misma figura el objeto segmentado (color blanco fondo negro) y la ubicación de las coordenadas de su baricentro según el eje de la imagen.

En una segunda experiencia se logra conectar la librería OpenCV y el player con el RoMAA II en forma real y el objeto es colocado en el piso de un corredor según se muestra en la figura 5.



Figura 5. RoMAA II persigue a la pelota.

El objeto es reconocido y las coordenadas del baricentro son enviadas desde el OpenCV hacia player. Una rutina de programación analiza los valores obtenidos del baricentro del objeto y se compara con el centro de coordenadas de la imagen.

RoMAA II gira hacia la izquierda según marca la flecha roja. El proceso se detiene cuando coinciden las coordenadas del baricentro del objeto con el centro de la imagen. Seguidamente se establece una condición de stop y luego de un tiempo de espera se ejecuta la condición de avanzar con la finalidad de interceptar al objeto.

El algoritmo utilizado en el seguimiento del objeto en el ejemplo anterior establece un estado de histéresis en las regiones donde el valor de la distancia entre el baricentro y el centro de la imagen no pueda ser establecido mediante un umbral definido (ver figura 6). En ese caso el RoMMA II oscila de izquierda

a derecha en un bucle infinito. También se destaca la influencia de la inercia

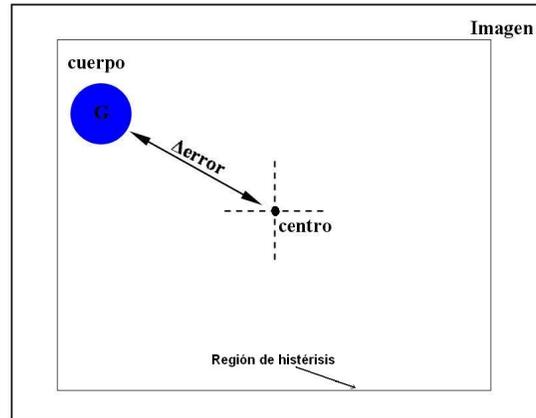


Figura 6. Baricentro centro de la imagen y la zona de histéresis.

mecánica de la plataforma móvil y velocidad de muestro de video. Por ese motivo es conveniente establecer un control de suavidad en el proceso de búsqueda.

Se implementa un controlador proporcional integral derivativo sugerido por Kart at. al.[11]. Permitiendo acoplar la respuesta inercial del RoMAA II con los datos enviados por el sistema de visión artificial.

El controlador proporcional, integral derivativo atenúa el efecto de histéresis en la respuesta del sistema mecánico y disminuye la frecuencia de muestreo de la webcam bajando la exactitud en el acción de detección. Es decir, que en el proceso de búsqueda del objeto el dispositivo móvil se detiene en un entorno del centro de la imagen y el camino de intercepción entre el RoMAA II y el objeto se realiza con un error. Para disminuir dicho inconveniente se debe ajustar las cosntantes del controlador.

8. Resultados

La implementación de un procedimiento para articular librerías de visión artificial, software de robótica y plataforma móvil de experimentación permite facilitar y reducir el tiempo en las tareas de investigación en áreas relacionadas al control automático de proceso.

Las librerías de edición de video OpenCV, los programas de stage y player son considerados como un sistema modular flexible y adaptable donde se prueban algoritmos de control y asociarlos a un hardware en forma transparente.

El software y hardware implementado permita resolver situaciones problemáticas de automatización utilizando un modelo virtual a escala reducida y los

algoritmos de control desarrollados son aplicables al modelo a escala natural permitiendo disminuir tiempo y esfuerzo en el desarrollo de tareas de control.

La simulación en el stage junto con el player puede representar una situación problemática real en la industria proporcionando una herramienta de análisis en la toma de decisiones.

El esfuerzo necesario para manejar los conceptos desarrollados en el presente trabajo se equilibran con los resultados obtenidos , permitiendo resolver problemas de mayor complejidad que de otra forma serían más complicado.

El sistema desarrollado es flexible y sencillo de implementar permitiendo modificar y actualizar en forma parcial o total los programas que articulan la parte de video, simulación y hardware. Este aspecto representa una ventaja sustancial en relación a la plataforma Windows.

9. Conclusión

La situación problemática resuelta en el presente trabajo desarrolla una metodología y procedimientos para definir un sistema constituido por librerías de visión artificial, software de robótica y hardware con la finalidad de facilitar tareas de control automático en forma eficiente.

La reduciendo en tiempo y esfuerzo en implementar la solución en estos tipos de problema es debido a que gran parte de los mismos se realizan con códigos reutilizados de paquetes informáticos como ser el stage player y OpenCV.

El esfuerzo necesario para la implementación de los diferentes conceptos desarrollados se deben realizar una sola vez debido a que el sistema generado es flexible y adaptable a otras situaciones problemáticas.

Existe un tipo de sinergia en el proceso de articulación en los conceptos desarrollado debido a que la estructura que asocia las librerías de visión artificial, software de robótica y el hardware es más que la suma de las partes.

Referencias

- [1] Peña , M., Lopez J., Corona K.:
Visión para Robot en tareas de Ensamble
<http://www.profesaulosuna.com/data/files/ROBOTICA/BRAZO%20ROBOT/AUT059.pdf>
(2011)
- [2] Muñoz, N., Andrés A., Ospina N.:
Diseño y Construcción de un Robot Móvil Orientado a la Enseñanza e Investigación.
INGENIERÍA & DESARROLLO . Vol.19. 2002 p.114-127
- [3] Vazquez, R. Araguas G., Masson L.:
PROYECTO DE INNOVACION EDUCATIVA EN ROBOTICA
Desarrollos e Investigaciones CientíficoTecnológicas en Ingenierías V EnIDI 2009
Capítulo 2, TIS y Electrónica. p.74
- [4] www.ciii.frc.utn.edu.ar/Robotica/RobotsRoboticaWeb
- [5] www.es.wikipedia.org/wiki/Proyecto_player/Stage

- [6] Garay , J., Bronson S.:
C++ para Ingeniería y Ciencia. Editorial Thomsom 2002. Madrid. España.
- [7] www.playerstage.sourceforge.net/
- [8] www.playerstage.sourceforge.net/wiki/Getting_help
- [9] www.es.wikipedia.org/wiki/OpenCV
Wikipedia 2 marzo 2011
- [10] Yourdon D., Edward S.:
Análisis Estructurado Moderno.Ed. Prentice Hall. Mexico 1989 Cap 4. p. 105
- [11] Karl J., Tore H.:
Control PID avanzado. Editorial Pearson Prentice Hall 2009. Madrid. España.
Cap 3. p. 67