

Diseño e Implementación de un Robot Móvil Teleoperado en base al Reconocimiento de Forma y Movimiento de Objetos

Michael Erazo⁽¹⁾, Danilo Matamoros⁽²⁾, Jaime Minchala⁽³⁾, Dennys Paillacho⁽⁴⁾

Facultad de Ingeniería en Electricidad y Computación⁽¹⁾⁽²⁾⁽³⁾⁽⁴⁾

Escuela Superior Politécnica del Litoral (ESPOL)⁽¹⁾⁽²⁾⁽³⁾⁽⁴⁾

Campus Gustavo Galindo, Km 30.5 Vía Perimetral, Apartado 09-01-5863. Guayaquil, Ecuador⁽¹⁾⁽²⁾⁽³⁾⁽⁴⁾

merazo@fiec.espol.edu.ec⁽¹⁾, dmatamor@fiec.espol.edu.ec⁽²⁾, jaimeminchala@hotmail.com⁽³⁾,

dpaila@fiec.espol.edu.ec⁽⁴⁾

Resumen

El presente proyecto tiene como finalidad la construcción de un robot móvil que será teleoperado por una aplicación, la cual servirá como interfaz de usuario para el control del robot. Esta aplicación utilizará técnicas de procesamiento digital de imágenes y visión por computador para reconocer y seguir objetos tangibles que serán manipulados por el usuario. Los objetos físicos que el usuario manipulará serán figuras geométricas de distintos colores, cada una de ellas representará un comando de control para el robot. La aplicación detectará el movimiento de estos objetos utilizando una cámara de video. Luego, enviará la orden al robot de acuerdo al movimiento y objeto detectado.

Para poder realizar lo anterior se implementará un algoritmo de reconocimiento y seguimiento de objetos. Este algoritmo será menos sensible al ruido comúnmente presente en este tipo de aplicaciones, lo cual a su vez, disminuirá el envío de órdenes al robot basadas en información errónea. Dicho algoritmo será de gran utilidad para futuras aplicaciones de visión por computador que necesitan reconocer colores y movimientos. Así mismo, se contribuirá en el área de robótica con el diseño de un robot móvil teleoperado.

Palabras claves: *interfaz de usuario tangible, robot móvil teleoperado, reconocimiento y seguimiento de objetos.*

Abstract

The objective of this project is to build a mobile robot that will be tele-operated by an application that provides a user interface for controlling the robot. This program will use image processing and computer vision techniques to detect and track physical objects that will be manipulated by the user. The physical objects that the user manipulates will be geometrical figures of different colors; every one of each will represent a command for the robot. The application will detect the movement of these objects using a video camera. Then it will send an order to the robot based in the movement and object detected.

For accomplishing the above, we will implement an algorithm of detection and tracking of objects. This algorithm will be less sensible to the noise normally present in this type of application, which as well will reduce the sending of commands to the robot based on incorrect information. This algorithm will be of great use in applications that need to detect and track objects based on color. This project also contributes to the robotics area with the design and implementation of the mobile robot.

Keywords: *tangible user interface, tele-operated mobile robot, object detection and tracking.*

1. Introducción

Actualmente la forma más común de controlar a los robots es por medio de paneles de control con botones, perillas, etc. Sin embargo, la necesidad de tener una comunicación más directa entre el humano y el robot, está llevando al desarrollo de alternativas que incluyen visión por computador y reconocimiento de imágenes.

Este proyecto se implementa con el motivo de contribuir con la investigación y el desarrollo de este

tipo de aplicaciones. Nuestro enfoque va orientado al desarrollo del robot móvil y del algoritmo que servirá para el reconocimiento de objetos utilizando visión por computador.

2. Análisis del problema

Muchas aplicaciones requieren que los robots respondan a acciones que se realizan en el mundo real, mas no a comandos enviados por el usuario a través de

algún panel de control. Es decir, que de cierta manera, el robot reaccione a lo que sucede en su entorno, de tal forma que pueda interactuar óptimamente con el medio. En dichas aplicaciones se utiliza la visión por computador para reconocer las acciones o movimientos que ocurren. Las cuales son interpretadas por un programa que envía los comandos al robot.

Además, existe la necesidad de este tipo de aplicaciones debido al incremento en la complejidad de las acciones y tareas que los robots pueden realizar, además de tener mayor precisión al llevar a cabo dichas acciones. Debido a esto, las órdenes que el usuario necesita enviar al robot se tornan también más complejas, lo que hace que ya no sea suficiente el uso de los medios tradicionales como teclado, palanca de mando o ratón para poderlos controlar [1].

El problema que busca solucionar este proyecto es precisamente la limitación que existe en la forma de controlar a un robot mediante una aplicación que use el teclado o el ratón para ingresar los comandos. Para contribuir a la mejora del problema se diseñará e implementará un prototipo que incluye un robot móvil y una aplicación de PC que utilizará técnicas de procesamiento digital de imágenes y visión por computador para reconocer y seguir objetos tangibles. Los cuales serán manipulados por el usuario para dar las órdenes al robot.

3. Diseño de la solución

Este proyecto consiste en el diseño e implementación de un robot móvil y una aplicación de PC que identifica y sigue el movimiento de objetos predeterminados. Esta aplicación sirve como interfaz de usuario para el control del robot móvil.

En la parte del hardware, el proyecto abarca la construcción del robot en su totalidad. El cual tendrá dos ruedas principales, las cuales serán gobernadas por dos motores independientes.

La aplicación de PC reconocerá objetos previamente definidos, o que el usuario los podrá definir cuando desee. El reconocimiento de los objetos se basa en su color. Debido a esto, es necesario que la luz ambiental sea controlada y que no varíe. La principal ventaja de usar el color para reconocer un objeto es que la alteración de la forma original del objeto producida por la manipulación del usuario, no afectará su reconocimiento por parte de la aplicación.

La comunicación establecida entre el computador que corre la aplicación y el robot móvil controlado es de una vía. Un circuito transmisor se encarga de enviar los datos desde la PC hacia el robot. El sistema no cuenta con retroalimentación del estado del robot hacia la PC.

A continuación se presenta un diagrama de bloques del proyecto.

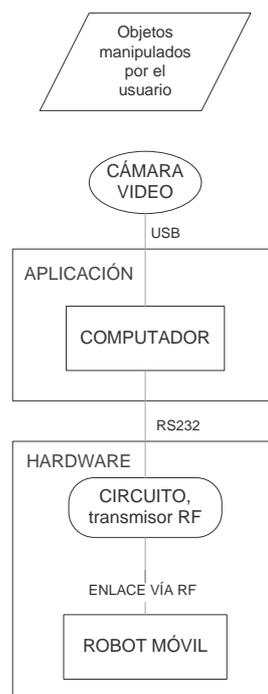


Figura 3.1: Diagrama de bloques del proyecto

3.1. Algoritmo de reconocimiento y seguimiento de objetos

El algoritmo de reconocimiento y seguimiento de objetos deberá ser implementado pensando en que esta es una aplicación en tiempo real. El usuario necesita ver una respuesta casi inmediata en las acciones del robot cuando manipule algunos de los objetos de interés. Es decir, necesita ser un algoritmo lo suficientemente rápido.

Otra característica importante es que sea preciso, en vista de que las acciones que el robot realice dependerán del resultado de este algoritmo. Para lograr este objetivo, se necesita hacer verdaderos esfuerzos en la reducción del ruido presente en toda señal de video y en el análisis de los datos. De esta forma se podrá reconocer los objetos de interés. Como este algoritmo también deberá realizar seguimiento a los objetos, la detección inicial debe ser libre de fallos para que estos no sean propagados en las siguientes iteraciones.

Estas dos características que son requeridas, podrían estar en conflicto. Implementar un algoritmo más preciso involucraría realizar más cálculos en general. Cálculos que podrían consumir muchos recursos del computador y que afectarían a su rapidez de manera significativa. Es decir un algoritmo sumamente preciso podría tener graves problemas de velocidad. Por esta razón es necesario encontrar un balance que se ajuste a las necesidades particulares de este proyecto.

3.2. Robot Móvil

Para el diseño de la parte de hardware del proyecto se consideró que se necesita hacer un robot móvil de ruedas. Para esto se tiene en cuenta que aunque hay varias configuraciones de robots móviles, la más versátil es la de control diferencial, la misma que usa dos ruedas principales laterales. Esta configuración permite mayor facilidad para realizar los giros, ya que incluso puede rotar sobre su propio eje [2].

Para poder controlar estas dos ruedas principales es necesario usar un motor por rueda. Es decir van a ser necesario usar dos motores independientes. Gracias a esto el robot podrá moverse hábilmente en todas las direcciones.

Aunque el control del robot se lo pudo haber realizado desde el computador, se prefirió usar un microcontrolador para esta tarea. Se decidió el uso de un microcontrolador debido a que este se encuentra dentro del robot y el tiempo que se tomará decidir lo que debe hacer será menor. Además en caso de que se pierda la comunicación por un instante no se afectará la ejecución de la tarea ordenada al robot. En este caso el robot continuará realizando la acción que estaba haciendo anteriormente.

Debido a que el robot requiere moverse sin estar conectado a nada fijo, la fuente de poder necesita ser portátil y ser llevada dentro del robot. Para esto es necesario usar una batería para energizar al robot. Esta batería se encargará de alimentar tanto a los componentes electrónicos, como a los motores del robot.

Además, es necesario incluir un módulo de recepción inalámbrico dentro del robot. Este módulo se encargará de recibir las señales que envía el transmisor de la aplicación y hacérselas llegar al microcontrolador.

4. Implementación del proyecto

4.1. Implementación de la aplicación

La aplicación fue desarrollada con el lenguaje de programación C++. Utilizamos la librería OpenCV para las tareas de procesamiento digital de imágenes y visión por computador y para la interfaz gráfica utilizamos la librería Winapi.

El programa está compuesto básicamente por tres pantallas. La pantalla principal muestra las órdenes que se están enviando al robot, de acuerdo a los cálculos realizados para la detección y seguimiento de los objetos de interés. La segunda pantalla brinda una retroalimentación al usuario de los objetos que se han detectado, mostrando la imagen capturada por la cámara de video con los objetos detectados marcados. Por último tenemos la pantalla de configuración en donde se configuran las características del color de los objetos de interés.



Figura 4.1: Pantalla principal

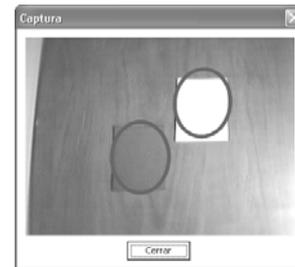


Figura 4.2: Pantalla de resultados

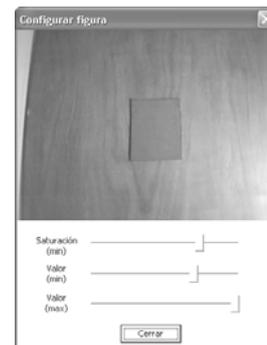


Figura 4.3: Pantalla de configuración de colores

Para la detección y seguimiento de objetos utilizamos el algoritmo de desplazamiento medio continuamente adaptativo. El cual consiste en los siguientes pasos [3]:

1. Ajustar la región de interés a toda la imagen.
2. Seleccionar la región a ser rastreada.
3. Calcular la distribución de probabilidad del color.
4. Iterar el algoritmo de desplazamiento medio para encontrar la centroide de la imagen de probabilidad.
5. Para las siguientes imágenes centrar la ventana de búsqueda a la posición encontrada en el paso 4. Ir al paso 3.

El algoritmo de desplazamiento medio toma como entrada una imagen en niveles de gris, cuyos valores representan la presencia del color a buscar. Esto quiere decir que si en un píxel está presente el color a buscar,

su valor de intensidad será mayor, caso contrario menor o cero si no presenta ninguna característica del color de interés.

La información del color a considerar son las componentes del modelo HSV: matiz, saturación y valor o intensidad. Cada uno de estos valores se puede configurar en la pantalla de configuración de cada objeto de interés. Esto se debe a que la información del color varía de acuerdo a la calidad de la cámara de video a utilizar y a las condiciones de iluminación en donde se ejecute la aplicación.

A continuación presentamos el algoritmo utilizado para realizar esta tarea.

```
void Figura::Buscar(IplImage *image, IplImage *hsv, IplImage *hue) {
    if (!_mask) {
        _mask = cvCreateImage( cvGetSize(image), 8, 1 );
        _backproject = cvCreateImage( cvGetSize(image), 8, 1 );
        _track_window = cvRect(0,0,image->width,image->height);
    }
    cvInRangeS( hsv, cvScalar(0, _smin, _vmin, 0), cvScalar(180,256,_vmax,0), _mask);
    cvCalcBackProject( &hue, _backproject, _hist);
    cvAnd( _backproject, _mask, _backproject, 0 );
    cvCanny( _backproject, _track_window,
            cvTermCriteria( CV_TERMCRIT_EPS | CV_TERMCRIT_ITER, 10, 1 ),
            &_track_comp, &_track_box );
    _track_window = _track_comp.rect;
    if (image->origin)
        _track_box.angle = - _track_box.angle;
    if ( _track_box.size.width > 10 && _track_box.size.height > 10 ) {
        cvEllipseBox( image, _track_box, CV_RGB(255,0,0), 3, CV_AA, 0 );
    }
    _detectado = 1;
}
else {
    _track_window = cvRect(0,0,image->width,image->height);
    _detectado = 0;
}
_valor = (int)(_track_box.center.y / 16);
}
```

Figura 4.4: Algoritmo de detección y seguimiento de objetos

4.2. Implementación del robot móvil

Dentro del robot, el dispositivo que recibe la información enviada por radiofrecuencia, es el receptor RF (RLP-434). Este módulo viene listo para ser conectado directamente a la placa. Una vez que es energizado, empieza a receptar las señales RF. Los datos que recibe este módulo, los cuales provienen de la aplicación, a su vez son enviados por medio de un cable, de manera serial, al microcontrolador.

Dentro del robot, el dispositivo principal que controla los motores y periféricos es el PIC, el cual tiene varias entradas y salidas. Una entrada recibe de manera serial los comandos provenientes de la aplicación, por medio del módulo receptor RF. Como salidas del microcontrolador, tenemos las señales que disparan los MOSFETs, además de la señal que controla al zumbador y una señal que enciende un led que indica cuando el robot está detenido.

El PIC ejecuta las órdenes que se le envía desde la aplicación a través del módulo RF. Para esto tiene grabado en su memoria de programa, algoritmos y subrutinas que se encargarán de operar el robot, según la orden que reciba. Estos algoritmos de control se encargarán de generar la secuencia de disparo que necesitan los MOSFETs para mover los motores de pasos a la velocidad deseada.

El arreglo de transistores MOSFET (IRF530) conectados al PIC, está compuesto por ocho transistores en total. Se usan cuatro por cada motor. Estos a su vez se conectan a los motores. Estos transistores son necesarios porque el PIC no puede proveer suficiente corriente para que trabajen los motores. Más información con respecto al controlador de los motores de pasos a base de transistores MOSFETs puede ser encontrada en [4].

Entre las órdenes que puede enviar la aplicación al robot, está la de activar o desactivar un zumbador. Esta acción se la realiza por medio de un transistor, cuya base está conectada a una salida del PIC.

Para la alimentación del robot se usa una batería y un regulador de voltaje. Los transistores de los motores, así como la placa electrónica del controlador del robot son energizados por el regulador de voltaje (LM2575-5), el cual recibe el voltaje de la batería y entrega un voltaje de 5V.

A continuación se muestra un diagrama de bloques del robot móvil. El cual consta de: El módulo receptor RF (RLP-434), el microcontrolador (PIC16F877A), los ocho transistores MOSFET (IRF530) que controlan a los motores y los dos motores de pasos que le dan movimiento al robot. La alimentación del robot consta de la batería de Litio-Ion y el regulador de voltaje (LM2575-5)

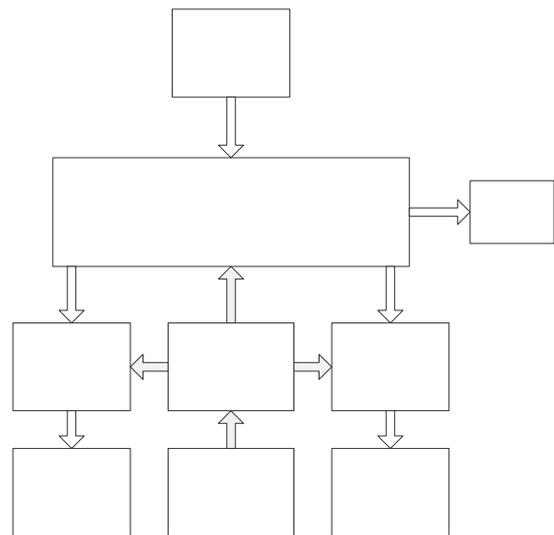


Figura 4.5: Diagrama de bloques del robot móvil

5. Pruebas y resultados experimentales

5.1. Pruebas en el reconocimiento y seguimiento de objetos

En esta prueba se comprobará la eficiencia del algoritmo en la detección de las figuras basándose en información del color. Para esto vamos a detectar 3 figuras de distinto color para verificar si el algoritmo puede discriminar cualquier otro color presente en la imagen.

A continuación mostramos los resultados obtenidos.

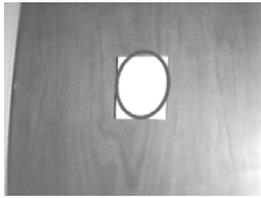


Figura 5.1: Detección de la figura de color amarillo.



Figura 5.2: Detección de la figura de color azul.



Figura 5.3: Detección de la figura de color verde.

5.2. Tiempo de respuesta del robot a órdenes del usuario

Esta prueba determina el tiempo que transcurre desde que el usuario ordena una acción por medio de la aplicación, hasta que el robot ejecuta dicha acción. Para esto consideramos el retardo que hay en la transmisión RF, así como también el tiempo que se toma el computador y el microcontrolador en procesar dichas órdenes.

El retardo total es la suma de tres tiempos: $T=T_c+T_t+T_m$, donde T es el retardo total desde que el usuario realiza una orden hasta que el robot la ejecuta. T_c es el tiempo que se toma el computador en procesar dicha acción y enviar el comando correspondiente por el puerto serie. T_t es el retardo que hay en la transmisión RF, desde que la señal sale del emisor hasta que llega al receptor. T_m el tiempo que se toma el microcontrolador desde que recibe el comando del módulo RF, hasta que finalmente ejecuta la acción correspondiente.

Experimentalmente se comprobó que el tiempo que se toma el computador y el microcontrolador en procesar las órdenes del usuario es del orden de los microsegundos. Por lo tanto el retardo estará exclusivamente en el tiempo que se toma en llegar la trama de datos completa en la transmisión RF.

Para determinar este tiempo se observó en el osciloscopio la trama de datos, la cual se muestra en la siguiente figura. Según esta gráfica el tiempo total que dura la trama de datos es de 33ms aproximadamente.

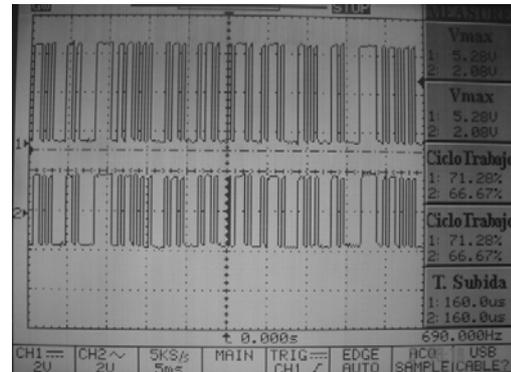


Figura 5.4: Duración de la trama de datos enviada

6. Conclusiones

Después de realizar todas las pruebas necesarias podemos concluir que hemos cumplido con los objetivos que nos hemos propuesto en esta investigación. El algoritmo de detección y seguimiento de objetos es lo suficientemente preciso y rápido para cumplir con las exigentes demandas de una aplicación en tiempo real.

El tiempo de respuesta del robot a las órdenes enviadas por el usuario puede considerarse como instantáneo para este tipo de aplicaciones. Esto es válido porque el retardo que existe entre la orden del usuario y la ejecución de dicha orden por el robot es sumamente pequeño, del orden de los milisegundos.

El uso de motores de pasos en la implementación del robot incrementa la precisión del mismo. Esto se debe a que este tipo de motores funciona con pulsos, cada pulso equivale a una cantidad determinada de grados de rotación.

7. Referencias

- [1] Casals, A., "Robots de servicios y personales: Aspectos tecnológicos que dificultan su progreso" Artículo Técnico, Universidad Politécnica de Cataluña, Centro de Investigación en Ingeniería Biomédica, 2005.
- [2] Liu, K., "Mobile, Flexible-Link, and Parallel-Link Robots", *Robotics, Mechanical Engineering Handbook*, Crc Press LLC, 1999, pp.14/192.
- [3] John G. Allen, Richard Y. D. Xu, Jesse S. Jin, "Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces" University of Sydney, School of Information Technologies, 2004
- [4] Rashid, M., *Power Electronics Handbook*, Academic Press, 2001, pp. 714-715.

