

SISTEMA DE NAVEGACIÓN DE ROBOT MÓVIL BASADO EN RECONOCIMIENTO DE IMÁGENES

Juan Carlos Pomier Salas.
Empresa Industrial Textiles Copacabana
La Paz, Bolivia
jcpomierss@uab.cl

MOBILE ROBOT NAVIGATION SYSTEM BASED ON IMAGE RECOGNITION

Resumen— En el siguiente proyecto, un robot móvil se podrá guiar en una cierta ruta reconociendo imágenes que se le vayan presentando en el camino. El problema se lo enfrenta mediante el uso de Visión Artificial también conocido como Visión por computador.

Se implementa técnicas de clasificación automática para este sistema de visión. Existen muchos métodos de clasificación con las cuales se podía afrontar este problema, pero en este caso se utiliza el método Template Matching el mismo método encuentra objetos dentro una imagen utilizando un emparejamiento de plantillas. La comparación de plantillas es un método para buscar y encontrar una imagen de la plantilla dentro de una imagen más grande.

Luego de procesar la imagen se dará una señal de mando al procesador del robot móvil para que este realice una serie de movimientos según indique la imagen reconocida.

Palabras Claves— *OpenCV, Robot Móvil, Threshold*

Abstract- In the following project, a mobile robot will be able to guide in a certain route recognizing images that are presented to him on the way. The problem is confronted by the use of Artificial Vision also known as Computer Vision.

Automatic classification techniques are implemented for this vision system. There are many classification methods with which this problem could be addressed, but in this case the Template Matching method is used; the same

method finds objects within an image using a template matching. Template comparison is a method to search and find an image of the template within a larger image.

After processing the image, a control signal will be given to the processor of the mobile robot so that it performs a series of movements as indicated by the recognized image.

Keywords- OpenCV, Mobile Robot, Threshold

I. INTRODUCCIÓN

Estos últimos años la automatización se volvió una prioridad en muchas empresas debido a muchos factores ya sea el aumento de producción y en cuanto a la calidad de trabajo que estas desempeñan, haciendo el uso efectivo de la maquinaria para transformar la materia prima. Por lo cual, está la necesidad de controlar cada aspecto del sistema de una empresa.

La visión artificial o visión por computador se define como un área multidisciplinar que pretende, en cierta medida, reproducir artificialmente el sentido de la vista mediante el procesamiento e interpretación de imágenes usando cámaras y software especializado para manejar procesos y verificar que este se desenvuelva de manera correcta y de acuerdo a los parámetros requeridos por el usuario [1].

Las ventajas de la visión artificial son muchas, por ejemplo, se pueden destacar al momento de reemplazar los medios tradicionales de un proceso por una nueva alternativa. Entre estos se encuentra los sistemas de control de automatizados, los cuales puedan generar mayor desempeño en la producción, repetitividad, consistencia, reducción de costos, etc. Es decir que permite mejorar las actividades realizadas manualmente, aquellas que puedan ser de riesgo vital para el ser humano, haciendo factible el reemplazo de los operarios [1][2].

Por todas estas razones mencionadas anteriormente, esta investigación busca crear un robot móvil basado en reconocimiento de imágenes, mediante el uso de visión artificial que nos ayude a evaluar señales que se le vayan presentando en cierto camino y así el robot haga un recorrido autónomamente.

Se define navegación como la metodología (o arte) que permite guiar el curso de un robot móvil a través de un entorno con obstáculos. Existen diversos esquemas, pero todos ellos poseen en común el afán por llevar el vehículo a su destino de forma segura. La capacidad de reacción ante situaciones inesperadas debe ser la principal cualidad para desenvolverse, de modo eficaz, en entornos [3]. Las tareas involucradas en la navegación de un robot móvil son: la percepción del entorno a través de sus sensores, de modo que le permita crear una abstracción del mundo; la planificación de una trayectoria libre de obstáculos, para alcanzar el punto destino seleccionado; y el guiado del vehículo a través de la referencia construida. De forma simultánea, el vehículo puede interactuar con ciertos elementos del entorno. Así, se define el concepto de operación como la programación de las herramientas de a bordo que le permiten realizar la tarea especificada. Un ejemplo: Sistema de visión de apoyo a la navegación de un robot móvil en invernaderos. (Gonzales)

II. ESTADO DEL ARTE

Se define navegación como la metodología

(o arte) que permite guiar el curso de un robot móvil a través de un entorno con obstáculos. Existen diversos esquemas, pero todos ellos poseen en común el afán por llevar el vehículo a su destino de forma segura. La capacidad de reacción ante situaciones inesperadas debe ser la principal cualidad para desenvolverse, de modo eficaz, en entornos. Las tareas involucradas en la navegación de un robot móvil son: la percepción del entorno a través de sus sensores, de modo que le permita crear una abstracción del mundo; la planificación de una trayectoria libre de obstáculos, para alcanzar el punto destino seleccionado; y el guiado del vehículo a través de la referencia construida. De forma simultánea, el vehículo puede interactuar con ciertos elementos del entorno. Así, se define el concepto de operación como la programación de las herramientas de a bordo que le permiten realizar la tarea especificada. Un ejemplo: Sistema de visión de apoyo a la navegación de un robot móvil en invernaderos. (Gonzales, 2010)



Fig. 1. . Fitorobot (Gonzales, 2010)

III. FUNDAMENTACIÓN TEÓRICA

Para el control de la navegación del robot móvil se desarrollaron tres bloques con funciones definidas las cuales son: adquisición de la imagen de la escena utilizando una cámara digital, procesamiento de imagen en raspberry pi3 usando OpenCV 3.0, comunicación USB maestro esclavo (master

slave) entre raspberry pi 3 como maestro y arduino como esclavo para control del robot móvil [5].



Fig. 2. Etapas del sistema de navegación con visión artificial

3.1 Adquisición de imágenes

La adquisición de una imagen digital de una cámara digital está limitada por el sensor de esta cámara los cuales son sensor CCD y sensor CMOS. Este sensor tiene millones de de celdas, estas celdas responden a ciertas longitudes según el filtro que tengan cada una. A estas celdas se las llama pixel con tres canales RGB.

3.2 Filtrado de imagen

Para procesar estas imágenes adquiridas se aplican filtros que ayudan a poder eliminar ruidos dentro de la escena para poder ser procesadas de mejor manera. De las funciones de OpenCV se aplica los siguientes filtros:

- Una función básica para cambio de escala de color a escala de gris es el siguiente comando: `cvtColor(m1,dst1,CV_BGR2GRAY)`, donde en m1 se selecciona la imagen a ser filtrada.
- Otra función para filtrar imágenes es la función blur : `cv2.blur`. Este filtro tiene la finalidad de difuminar la imagen para poder realizar de mejor manera un filtro de umbral(threshold).
- El ultimo filtro que se aplica a la imagen es el de umbral adaptativo (Adaptive Thresholding): `cv2.adaptiveThreshold`, el algoritmo de calcular el umbral para una pequeña región de la imagen. Por lo que tenemos diferentes umbrales para las diferentes regiones de la misma imagen y nos da mejores resultados

para las imágenes con mayor o menor iluminación. Es importante que para el uso de este algoritmo previamente deben estar aplicados los anteriores filtros [9].

3.3 Aislamiento y corrección de objetos de la imagen

Para aislar de una imagen el segmento que es de importancia en este caso un letrero, se aplican funciones que están dentro de OpenCV que ayudan a que se filtren los letreros los cuales están encerrados en un cuadrado y que además se corrijan aquellos que están en perspectiva y no así de frente a robot.

3.3.1 Perspective Transform

El comando: `cv2 . GetPerspectiveTransform`. Es una función que requiere los 4 puntos de retorno de inversión de la imagen original y la lista de puntos transformados. Esta función devuelve una matriz de transformación real [9].

Esta matriz de transformación se aplica la función `cv2 . WarpPerspective` la cual corrige la imagen a nuestra visión de arriba hacia abajo.

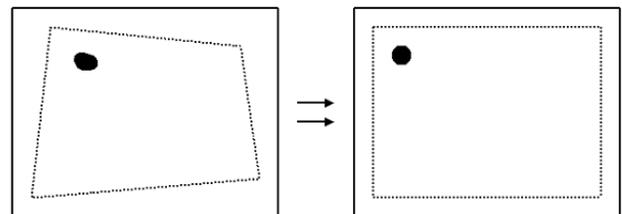


Fig. 3. –Orientaciones aceptadas por Template Machine

3.3.1 Template Matching

Este método es empleado para encontrar objetos en una imagen, utilizando el emparejamiento de plantillas.

La comparación de plantillas es un método para buscar y encontrar la ubicación de la imagen de una plantilla en una imagen más grande. OpenCV viene con una función `cv2.matchTemplate()` para este propósito.

Es simplemente se desliza la imagen de plantilla sobre la imagen de entrada como la Convolución 2D) y compara la plantilla y el parche de la imagen de entrada de la imagen de la plantilla bajo. (Mordvintsev, 2016)[12] Si la imagen de entrada es de tamaño (W+H) “W= Ancho de la imagen de entrada, H= alto de la imagen de entrada” y la imagen de la plantilla es de un tamaño (w+h), la imagen de salida tendrá un tamaño de (W-w+1,H-h+1).

Después de tener el resultado se utiliza el comando `cv2.minMaxLoc()` función para encontrar donde está el valor máximo/mínimo. Donde tomamos (w,h) como la anchura de la altura del rectángulo que serán las medidas de la plantilla.

Para hacer un matching de varias plantillas `cv2.minMaxLoc()` no nos entrega todos matching dentro de la imagen. En este caso, utilizaremos umbral [6].



Fig. 4. Imagen de detección Giro a la Derecha

3.4 Comunicación Raspberry Pi3 y Arduino

3.4.1 Hardware

La conexión entre maestro esclavo se efectua mediante usb Raspberry Pi3 y el robot móvil a nivel de hardware el Arduino Uno, el cual cuenta con un ATmega328 ofreciendo una comunicación serial utilizando una UART TTL este puerto serial. El control del robot se realiza por medio de los pines de entrada salida digital. Realizando una comunicación serial a 9600 baudios.

Para la etapa de potencia en el control de los motores D.C. Del robot móvil se utilizó un puente H [2][4].

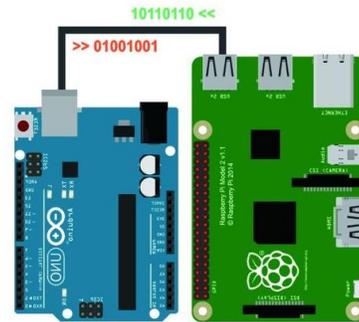


Fig. 5. Comunicación Arduino - Raspberry

3.4.2 Software

Para la comunicación nivel software se utiliza PySerial este software permite el acceso al puerto serial desde Python para así poder entablar una comunicación con arduino.

El Sistema Operativo oficial para este dispositivo es *Raspbian*, el que es una distribución exclusiva para el uso en Raspberry Pi. Está basada en la distribución de GNU/LINUX, Debian y para ser instalada debe grabarse la imagen de Raspbian en la tarjeta SD. Ofrece acceso tanto a través de línea de comandos o *Secure Shell (SSH)*, como de un escritorio gráfico basado en LXDE (Entorno de Escritorio X11 Liviano).

Arduino opera con un lenguaje de programación llamado Wiring (Kushner, 2011), el cual está basado en la plataforma Processing - C/C++. El entorno de compilación que utiliza Arduino es el Ambiente de Desarrollo Integrado (en inglés: IDE, Integrated Development Environment) homónimo, Arduino IDE. En este entorno de programación se escribe el programa que se desea ejecutar en la tarjeta.

IV. METODOLGÍA

La estructura del proyecto se compone por dos procesadores conectados por un modelo de comunicación denominado maestro-esclavo. En este caso el maestro llega ser el Raspberry Pi 3 y el esclavo es el Arduino.

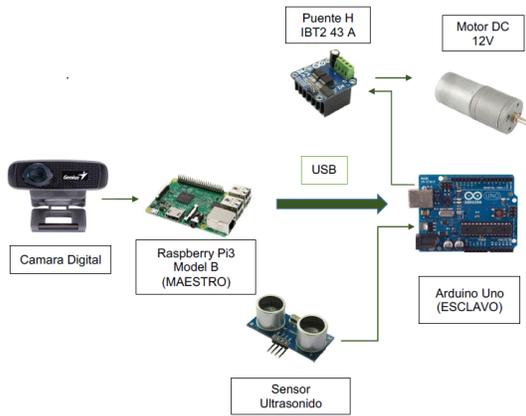


Fig. 6. Diagrama de bloques del Robot

La estructura del proyecto se basa en la adquisición de datos del entorno mediante el uso de una cámara digital. Con el propósito de realizar la detección de una imagen, que en este caso es una señal de tránsito la cual llamaremos “template”. Cuando el sistema detecte una de estas imágenes dentro su recorrido, el robot al que va unido a este sistema es capaz de tomar una decisión según la imagen que se detectó [2].

Para realizar dicho proceso se debe procesar previamente las imágenes es decir que estas deben pasar por un previo pre-procesamiento los cuales son: conversión a escala de grises, filtrado de imagen, threshold (umbral), canny y detección de contornos los cuales son descritos desde la anterior sección dicha imagen será utilizada para la etapa de detección de la imagen que en este caso es una señal encerrada en un rectángulo, esta imagen que está dentro de un rectángulo debe pasar por una corrección geométrica para finalmente poder realizar un match e identificar la imagen dentro del rectángulo.

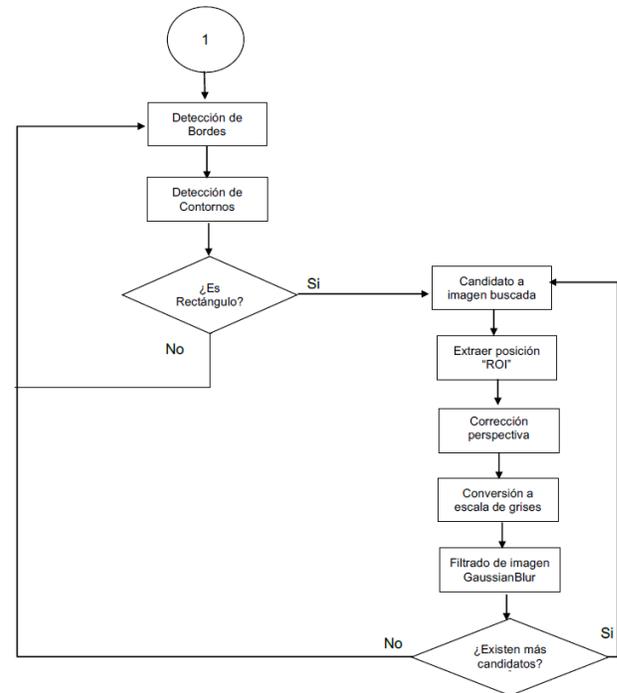


Fig. 7. Diagrama de reconocimiento de candidatos

Una vez recibida la imagen esta será procesada hasta detectar dentro de esta el área de interés en este caso la imagen deseada a identificar que se encuentra dentro de un cuadrado. Se muestra los módulos que llevan este proceso.

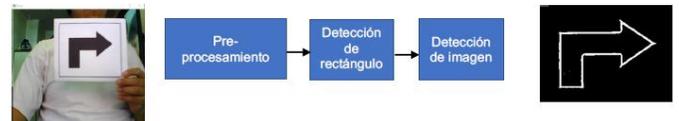


Fig. 8. Diagrama de reconocimiento de candidatos

La detección de la imagen comprende tres módulos:

1. Detección de contornos.
2. Selección de rectángulos.
3. Corrección geométrica

La Fig. nos muestra los módulos del proceso de detección del template.

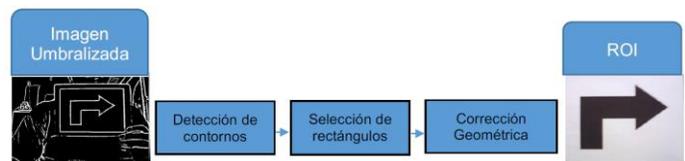


Fig. 9. Diagrama de bloques de detección de imágenes

Una vez encontrado el conjunto de rectángulos aproximados se procede a realizar la corrección geométrica de los polígonos usando una función de perspectiva. Esto debido a que en ciertas ocasiones la imagen del template tiene cierta inclinación o perspectiva.

Cabe recalcar que la corrección geométrica la realizamos sobre la imagen original, no sobre los anteriores procesos. Esto es debido a que los procesos para la detección del área de interés se puede perder información como por ejemplo el color ya que la imagen anterior a este proceso es una imagen umbralizada y para el siguiente proceso de segmentación se necesita la imagen original para realizar el reconocimiento. Las ecuaciones se expresarán dejando líneas de separación entre el texto anterior y el posterior. Si van numeradas, dar el número al final de la línea y cuidar la numeración consecutiva. Si superan la longitud de línea se facilitarán aparte colocando solamente el número en el lugar deseado, tal como se indica en el apartado anterior [6].

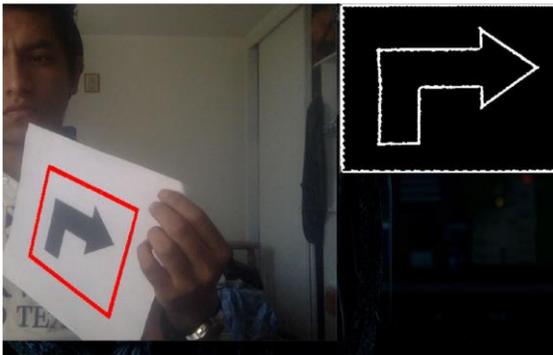


Fig. 9. Corrección Geométrica

V. CONCLUSIONES

- En el proyecto de sistema de navegación basado de robot móvil basado en reconocimiento de imágenes pudimos observar un nuevo método alternativo a los demás sistemas de navegación, en el cual usa programas de software libre como Python y OpenCV

con los que se pudo obtener un reconocimiento de patrones dentro de un espacio determinado.

- La arquitectura modular que posee este proyecto proporciona una gran versatilidad de usos. Dado a que si bien se puede utilizar para la navegación se puede utilizar también la adquisición de datos de otras maneras.
- Una vez finalizado el montado del robot con el sistema de navegación desarrollado, se concluye que no es completamente eficiente la navegación del robot móvil ya que las condiciones del entorno donde este se desenvuelve deben ser optimas en cuanto a luminosidad, estado del terreno donde se navegara, entre otros. Es por eso que el algoritmo debe ser ajustado.

RECONOCIMIENTOS

Agradezco a la Escuela Militar de Ingeniería, en especial al Ing. Marcelo Saavedra por la invitación para publicar uno de mis trabajos para la carrera de Ingeniería Mecatrónica. Espero poder seguir aportando a esta gran Casa de Estudios Superiores.

REFERENCIAS

- [1] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, C Dhaval, C., Darde, D., & Chitalia, S.
- [2] Pomares, J. (2009). Manual Arduino. Alicante España.
- [3] Ryuji Funayama, H. Y. (Noviembre de 2009). ROBUST INTEREST POINT DETECTOR AND DESCRIPTOR.
- [4] Hess, R. (2011). An Open-Source SIFT Library. Oregon USA.
- [5] Gonzales. (2010). NAVEGACIÓN DE UN ROBOT MÓVIL MEDIANTE VISIÓN.
- [6] Mordvintsev. (2016). OpenCV-Python Tutorials.
- [7] Universidad Politecnica de Valencia. (2013). Raspberry pi. (L. Contreras, Ed.) Obtenido de Blog de la asignatura Historia de la Informática de la Escuela Técnica Superior de Ingeniería Informática de la Universidad Politécnica de Valencia.
- [8] Raspberry Pi Server Admin. (2012). IRC Server on the Raspberry Pi. Obtenido de Raspberry Pi Server Admin.

RECEPCION: 26/02//2019

APROBACION: 14/03/2019