Congr. Int. Ing. Electrón. Mem. Electro 2012, vol.34, pp. 259-264 Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2012.zip
ISSN 1405-2172

SISTEMA DE POSICIÓN Y ORIENTACIÓN POR MEDIO DE UNA CÁMARA DE VIDEO PARA ROBOTS MÓVILES EN UNA SUPERFICIE ESTABLECIDA 2-D

J. A. Rojas Estrada, P. Quintero Alvarez, H.A. Villarreal Hdez., L.A. Montoya Soto
Posgrado en Ingeniería Mecatrónica, División de Estudios de Posgrado e Investigación
Instituto Tecnológico de Nuevo León
Ave. Eloy Cavazos 2001, Col. Tolteca, Guadalupe, N.L. C.P. 67170, México
Tel. (81) 81570500 ext. 140 Fax (81) 81570505
jarojas2001@yahoo.com.mx, patyqar@yahoo.com

RESUMEN

En este trabajo se presenta un sistema de posición y orientación para robots móviles. Dichos parámetros se obtienen a través de una cámara web, la cual se encuentra conectada a una computadora. Mediante el procesamiento de imágenes en Matlab, es posible reconocer dos cuadros de colores que se encuentran por encima del robot. Se obtienen los centroides de cada cuadro, teniendo así las coordenadas de los dos centroides. Por medio de los puntos captados se obtendrán dos coordenadas en el plano XY, donde, aplicando la ecuación de la pendiente y obteniendo el arco tangente de dicha pendiente, se obtiene el ángulo que la forma y por lo tanto la orientación en la que se encuentra el robot.

Este sistema de localización esta siendo probado e implementado en navegación con obstáculos, logrando resultados exitosos [4].

1. INTRODUCCIÓN

La posición y orientación de un robot móvil es de suma importancia para la navegación y control de este tipo de robots, [1], [2].

Se han desarrollado diferentes métodos para conocer tanto la posición como orientación de robots móviles [3], como son el GPS, la odometría, triangulación de señales, etc.

Se realizaron procedimientos para trabajar con el Sistema de Posicionamiento Global (GPS) para la localización de un robot móvil en un área definida, como en [1],[2],[3] y [4], no obstante, se observó que existía un margen de error elevado en el posicionamiento del robot en un área deseada, con una precisión aproximada de 7 metros a la redonda del robot, siendo una variación muy grande para lo que se buscaba en la implementación de la medición. Por tal motivo se buscó un sistema el cual fuera lo mas preciso posible a la hora de adquirir los datos de localización. Así se llegó a la implementación de un sistema de visión por medio de una cámara web, [5], en donde la precisión de la posición del robot es muy exacta, ya que se toma como referencia los pixeles de la imagen obtenida de la cámara web para determinar la posición. La resolución de la imagen es la que determina la dimensión del plano en el que se mueve el robot y de esta forma se tiene un plano coordenado el cual lo denotaremos plano XY. La resolución de imagen con la se trabaja en este sistema es de 640x480 pixeles. Así mismo, con la ayuda de la resolución, se obtendrán las coordenadas del pixel en el que se encuentra el centroide de cada uno de los cuadros de colores.

2. DESARROLLO

El desarrollo del sistema de posición y orientación se basa en la obtención y procesamiento de imágenes mediante una cámara web de la marca Logitech, modelo c210 mostrada en la figura 1.

Se configura la cámara web en matlab y se realiza un programa para captar la imagen del área en el que se desplaza el robot. Se aplican diferentes tipos de filtros a la imagen obtenida y de esta forma se logra captar el color rojo o verde de los cuadros.

Figura 1. Cámara web Logitech c210

El área en el que se desplaza el robot se determina por la distancia a la que se encuentra situada la cámara por encima del robot. En este caso se tiene situada la cámara a una altura de 2.525 m, teniendo así un área de visión de 2.06x1.55 m^2, área que es utilizada para el desplazamiento del robot y la realización de pruebas de la referencia [4].

2.1. Obtención de posición en matlab

Como se comentó, la forma de obtener los datos necesarios para la localización del robot se realiza por medio de dos cuadros que se colocan encima del robot, con la finalidad de que la cámara este visualizando estos dos cuadros. Estos cuadros de 5x5 cm son de color rojo y verde, para facilitar el proceso de filtrado de la imagen, debido a que la imagen es captada en formato RGB, por lo que este formato como su nombre lo dice por sus siglas en inglés, cuenta con los colores rojo, verde y azul, por lo tanto la extracción de los colores con los que cuentan los cuadros será más fácil.

Usando las funciones y librerías de Matlab, [8] y [9], es posible reconocer la cámara web que se esta usando, así como de seleccionar la cámara deseada en caso de contar con mas de una. Para crear el objeto de la cámara se teclea el código 1 mostrado mas adelante.

```
imaqhwinfo('winvideo',1);
img = videoinput('winvideo', 1,
'RGB24_640x480');
```

Codigo 1. Reconocimiento de camara en Matlab.

Una ves reconocida la cámara y configurarla para la captura de imágenes se enciende la cámara por medio del código 1 y se captura la primer imagen por medio de las funciones del código 2 y la imagen capturada se guarda en una variable llamada foto. La imagen capturada se aprecia en la figura 2.

```
start(img);
foto=getsnapshot(img);
```

Código 2. Obtención de imagen.

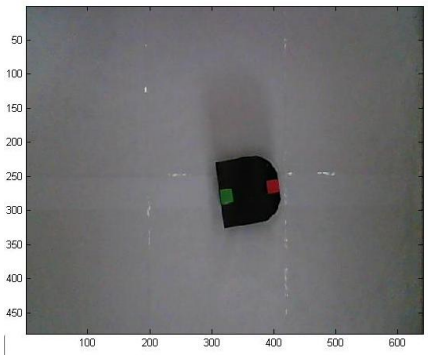

Figura 2. Imagen capturada

A las imágenes capturadas se les aplica un filtro de color, para descomponerlas en RGB (Red, Green, Blue), que es el formato de color con que capta la camara. De esta forma se puede seleccionar el color deseado dentro de este rango (RGB), para así convertirla a escala de grises (codigo 3) y así obtener la figura 3.

```
ColorR= imsubtract(foto(:,:,1),
rgb2gray(foto));
```

Codigo 3. Conversión de RGB a grises.

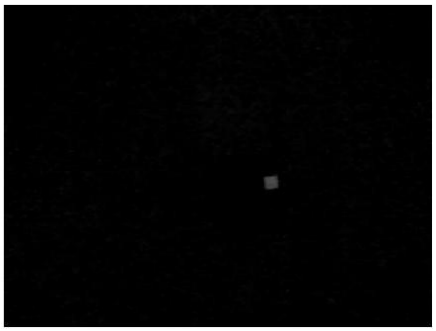

Figura 3. Imagen convertida a escala de grises, mostrando solamente el color rojo en gris.

Teniendo la imagen en escala de grises, ahora se convierte la imagen en grises a blanco y negro por medio del codigo 4. Al convertir la imagen, se genera una matriz binaria, teniendo el cuadro deseado en color blanco y el resto en negro. Las partes con mas luminosidad se toman como 1 (blanco) y las partes mas obscuras se toman como 0 (negro), como se muestra en la figura 4. Se le puede asignar el nivel de blanco o negro, especificando el nivel entre 0 y 1, en dado caso que no se especifique el nivel, automaticamente se establecera 0.5 de nivel, la mitad entre blanco y negro.

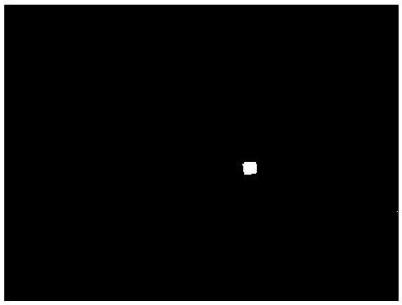

Figura 4. Imagen convertida a blanco y negro.

ISSN 1405-2172

```
ColorR= im2bw(ColorR,0.06);
```

Codigo 4. Conversión de escala de grises a blanco y negro.

Esta conversion ayudará a identificar los objetos que cuentan con números binarios 1. De esta forma se podra conocer la forma del objeto, en este caso el cuadrado mostrado en la figura 4.

Teniendo la imagen binaria se aplica la funcion del codigo 5, que obtiene el centroide y el contorno que encerrará al cuadro de la imagen mostrada en la figura 5. Se muestra el centroide y contorno en la foto original para una mejor presentación.

```
prop =
regionprops(bt,'BoundingBox',
'Centroid');
rect=
prop(objeto2).BoundingBox;
centroid=
prop(objeto2).Centroid;
rectangle('Position',rect,'Edge
Color','r','LineWidth',3)
```

Código 5. Código para Obtener centroide y contorno.

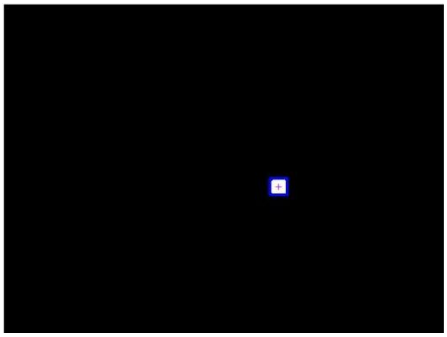

Figura 5. Imagen con centroide y contorno del cuadro.

"Centroid" tendrá la información de las coordenadas del centroide del cuadro, por lo tanto se obtiene la posición del cuadro de color rojo que lleva por encima el robot.

Este mismo procedimiento se aplica para el cuadro de color verde, solamente se cambia el número "1" del código 3 por el número "2", y se escribe el mismo código para localizar el centroide y contorno para el cuadro verde, por consiguiente, se obtiene la posición del cuadro verde.

El código completo estará dentro de un ciclo while, y así visualizar las imágenes en forma de video al estar modificando en cada ciclo la información de los puntos de colores. En la figura 6 se muestran los centroides de los dos cuadros.

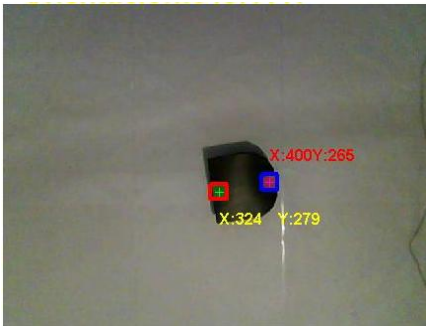

Figura 6. Centroides de los dos cuadros

2.2. Orientación del robot

Conociendo la posición de los dos puntos en el plano de la imagen, es posible conocer la orientación que tiene el robot. El punto verde del robot es el origen del posicionamiento local y el punto rojo es el ángulo que forma respecto al verde.

Llamaremos "p1" y "p2" a los centroides de los cuadros rojo y verde, respectivamente.

Usando la ecuación punto-pendiente

$$y - y_1 = m(x - x_1) \quad (1)$$

Despejando *m*, que denota la pendiente, se obtiene la ecuación de la recta que pasa por dos puntos, obteniendo:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (2)$$

Dado *m*, se calcula el ángulo α entre estos dos puntos aplicando el arco tangente de la pendiente, como se aprecia en la figura 7.

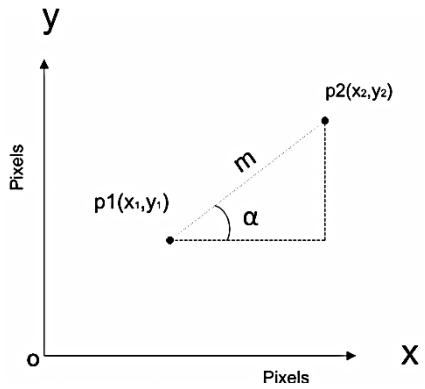

Figura 7. Obtención del ángulo por medio de la pendiente.

De este modo se obtiene el ángulo que existe entre los puntos p1 y p2, obteniendo la orientación que tiene el robot respecto a p1 dentro de un plano definido.

3. RESULTADOS

El robot con el que se esta trabajando (figura 8), es un prototipo de robot armado realizado por alumnos de la institución. Es un robot móvil de tipo diferencial y actualmente se esta aplicando en la determinación del polígono de velocidades admisibles (PVA) posición y orientación en [3].

Figura 8. Robot móvil usado.

Situada la cámara a una altura de 2.525 m, se visualizó un área de trabajo de 2.05x1.55 m^2, lugar donde se realizaron las pruebas contundentes, para verificar el comportamiento del sistema propuesto. En la figura 8 se aprecia el escenario en donde se desplaza el robot, así como la altura de la cámara.

De acuerdo a esta configuración se obtuvo exitosamente la posición y orientación del robot en el área en el que se desplaza. (figura 9.)

.

Figura 8. Escenario donde se desplaza el robot.

Figura 9. Posición y orientación del robot, mostrado en la imagen

La visión real de la cámara respecto a los cuadros colocados en el robot es de 620x460 pixeles, ya que alrededor del plano XY no le es posible captar los primeros y últimos 10 pixeles cuando el robot se encuentra en los limites del área visualizada. Estos 10 pixeles equivalen a aproximadamente 2.5 cm de tamaño real en el plano. En total serian 20 pixeles menos del total de pixeles en el eje de las

ISSN 1405-2172

ordenadas y 20 menos del eje de las abscisas, ya que son 10 pixeles por eje.
Las figuras 10a, 10b,10c, y 10d muestran los limites que la cámara puede visualizar, como se comentó en el párrafo anterior, la cámara puede visualizar los dos cuadros, sobre el robot, casi la totalidad de los 640x480 pixeles.

Figura 10.a Robot en los límites del área visualizada.

Figura 10.b Robot en los límites del área visualizada.

Figura 10.c Robot en los límites del área visualizada.

Figura 10.d Robot en los límites del área visualizada.

4. CONCLUSIONES

Al procesar imágenes en matlab dentro de un ciclo, se tiene la desventaja de que consume demasiada memoria RAM, por lo que si no se cuenta con suficiente memoria en el ordenador, es posible que la memoria RAM se sature y la velocidad de captura de imágenes se hará más lenta. Este sistema se había probado inicialmente en un ordenador con 2 GB en RAM, sin embargo, se hacia lento el proceso de imágenes. Para corregir este problema se implementó este sistema en una computadora que cuenta con 6 GB en RAM, lo cual es suficiente para realizar cualquier prueba que se requiera.
Es importante la iluminación en el área de desplazamiento, ya que si no se cuenta con la iluminación adecuada, la cámara no podrá visualizar los cuadros con los que cuenta el robot. En este caso se colocaron dos lámparas para

ISSN 1405-2172

cuando había poca luz o el ambiente estaba nublado**.**

Una de las cosas más importantes en el control de robots móviles es conocer su posición y orientación. Más importante aún es el conocimiento continuo de tales parámetros mientras se implementa el procedimiento del polígono de velocidades admisibles para conducir el robot hacia un objetivo con determinada orientación. El procedimiento desarrollado en el presente trabajo, representa una alternativa viable en la obtención de la posición y orientación de un robot móvil. El alcance en la técnica presentada, hablando de la medición, entre otros, dependerá de la posición de la cámara y del tamaño de los identificadores en el robot. Actualmente se trabaja en encadenar los resultados calculados en matlab a otra plataforma, visual estudio por ejemplo, con el fin de usar esta técnica de medición de la posición y orientación en la implementación del PVA en un robot móvil para navegación con obstáculos.

5. REFERENCIAS

[1] Quintero Alvarez P., Ramirez, G., Zeghloul, S., ***A collision free path planning method for articulated mobile robot***, Applied Bionics and Biomechanics, 4:2, pp.71-81, 2007

[2] Quintero Alvarez P., Rojas Estrada J. A., Fernández Ramírez A. A., Ramírez Torres J. G. ***Técnicas para evasión de obstáculos en robótica móvil,*** CISCI 2010, Orlando Florida, pp. 75-80, 2010, Vol. I.

[3] Martínez Soto M. A., Quintero Alvarez P., Rojas Estrada J. A., Fernández Ramírez A. A., ***Planificación de trayectorias libres de colisión en robots comerciales***, Electro 2010, Chihuahua, Chihuahua, pp. 226-230, Vol. XXXII.

[4] J. A. Rojas Estrada, P. Quintero Alvarez, M.A. Martínez Soto, A.A. Fernández Ramírez., I. Z. Ochoa ***Medición de distancia para la aplicación del PVA en robots comerciales*** Electro 2011, Chihuahua, Chihuahua, pp. 124-129 Vol. XXXIII.

[5].http://www.mathworks.com/matlabcentral/fileexchange/28757-tracking-red-color-objects-using-matlab

[6].http://www.electron.frba.utn.edu.ar/materias/950454/archivos/matlab_trabajo_con_imagenes.pdf

[7].http://www.slideshare.net/lonely113/procesamiento-digital-de-imgenes-con-matlab

[8] Cuevas, Erik; Zaldívar, Daniel: Pérez, Marco. 2010, ***Procesamiento digital de imágenes con MATLAB y simulación***, Ra-Ma Editorial.

[9] Marques, Oge.***Practical image and video processing using MATLAB®,*** John Wiley & Sons, Inc.