Capítulo 4 – Exploración del ambiente.

Para explorar el ambiente se tomó como base el vehículo explorador de Braitenberg, la idea es tomar este comportamiento y adaptarlo al uso de una cámara de video, esto con el fin de que funcione como sistema de navegación para el robot construido con el kit LEGO mencionado en el capítulo anterior, y de esta forma otorgarle al robot, la capacidad de explorar un ambiente estático desconocido evitando colisionar con los obstáculos que se encuentran dentro de dicho ambiente.

4.1 Formato .bmp de 24 bits.

Este es el formato en que el controlador QuickCam SDK 1.0 permite guardar imágenes, es por esta razón que se explica a continuación dicho formato, esto nos ayudará a entender mejor como se llevan a cabo los pasos del procesamiento de la imagen.

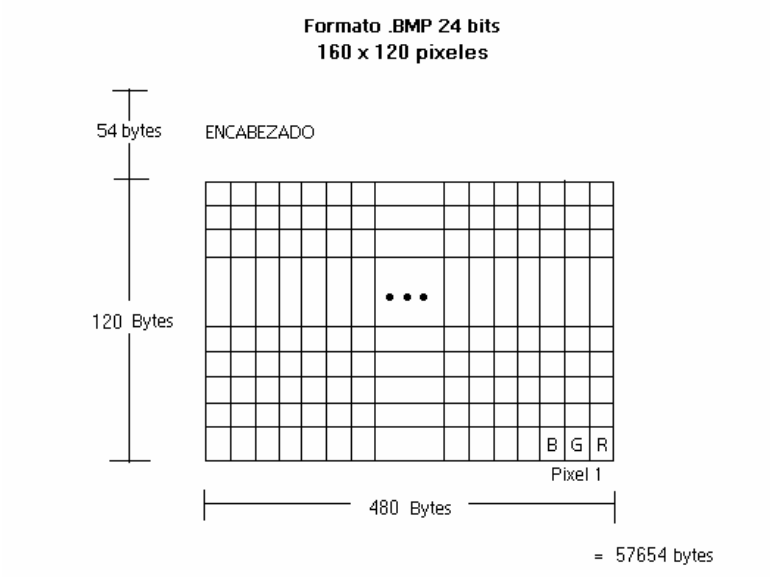

Figura 4.1 Formato .bmp 160 x 120 pixeles.

Como podemos observar en la figura anterior, el formato bmp comienza con un encabezado que tiene un longitud de cincuenta y cuatro bytes, este encabezado tiene información como el tamaño, el ancho, la altura, bits por píxel, entre otros

.

Seguido de los bytes del encabezado se encuentran los bytes que representan las componentes RBG de cada píxel, en total tenemos cincuenta y siete mil seiscientos bytes que nos representan la imagen, pero de estos solamente hasta el byte cincuenta y cinco representan las componentes RGB, estos bytes deben ser leídos de atrás hacia delante, es decir el último byte representa la componente R del primer píxel, el penúltimo byte representa la componente G del primer píxel y el antepenúltimo byte representa la componente B del primer píxel.

4.2 Captación de la imagen.

El primer paso dentro de el procesamiento de la imagen, consiste en captar la imagen a intervalos de tiempo constantes y guardarla en un archivo, para esta tarea se utilizó el controlador Logitech QuickCam SDK 1.0 que permite controlar la cámara del kit Vision Command desde un lenguaje de programación visual como Visual Basic 6.0®, a través de un comando permite guardar la imagen que la cámara está captando en ese momento, estas imágenes fueron guardadas en formato bmp de 24 bits, ya que este es el único formato que soporta el controlador Logitech QuickCam SDK 1.0 y la resolución con que fueron tomadas las imágenes fue de 160 x 120 pixeles. La captación de la imagen a intervalos de tiempo, se hizo a través del controlador Timer que provee Visual Basic 6.0®, y se eligió un

intervalo de tres segundos, ya que al experimentar con diferentes intervalos de tiempos, este presenta buenos resultados en términos generales.

La siguiente figura muestra una imagen típica captada por el robot, en ella se muestra un obstáculo de color rojo, al fondo también se puede apreciar la pared del ambiente que se encuentra en tonalidad negra y el piso del ambiente que es de tonalidad blanca.

Figura 4.2 Imagen Original.

4.3 Segmentación de la imagen.

Después del proceso de guardar la imagen en un archivo, el siguiente paso consiste en leer la imagen guardada en formato bmp y segmentar la imagen original, esto quiere decir dividir los obstáculos que en nuestro caso son de tonalidad roja de los demás objetos que se encuentran en la imagen como la pared y el piso, para segmentar la imagen se hizo uso de la componente ortogonal I2 del sistema I1I2I3 dado que en resultados experimentales, esta componente presenta mejor resultados para segmentar porciones del color rojo.

La formula se presenta a continuación.

$$\mathbf{I2 = (R - B) / 2}.$$

Como se puede observar en la formula anterior, únicamente se consideran las bandas de color rojo y azul, pero da buenos resultados para segmentar porciones rojas de la imagen, esta fórmula se aplica a cada uno de los pixeles que componen la imagen original y el resultado es un imagen en la cual, los obstáculos se encuentran representados en una tonalidad más clara que los demás objetos en la imagen como el piso y las paredes.

El resultado de aplicar esta fórmula a la imagen original se puede observar en la siguiente figura.

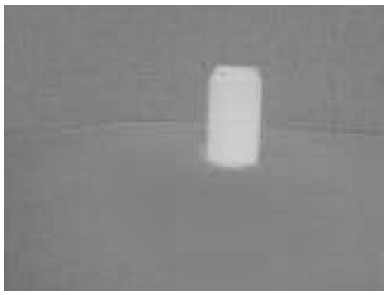

Figura 4.3 Imagen segmentada.

En la figura 4.3 se puede observar que el obstáculo que originalmente es de color rojo, ahora se encuentra en una tonalidad de gris claro y los demás objetos que conforman la imagen original, se encuentran en una tonalidad de gris oscuro.

4.4 Reducción de la imagen.

Ahora que tenemos segmentados los obstáculos de los demás objetos, se procede a reducir la imagen, esto se hace con la finalidad de que los procesos siguientes se hagan con mayor rapidez, para reducir la imagen se tomaron ventanas de cinco pixeles en sentido horizontal, y cinco pixeles en sentido vertical, dándonos un total de veinticinco pixeles por ventana. Estas ventanas se traslapan una con la otra en un píxel en sentido horizontal y un píxel en el sentido vertical. Al aplicar este algoritmo obtenemos una imagen de 40 x 30 pixeles. La figura 4.4 tomada de la tesis de Alberto Chávez muestra esto.[11]

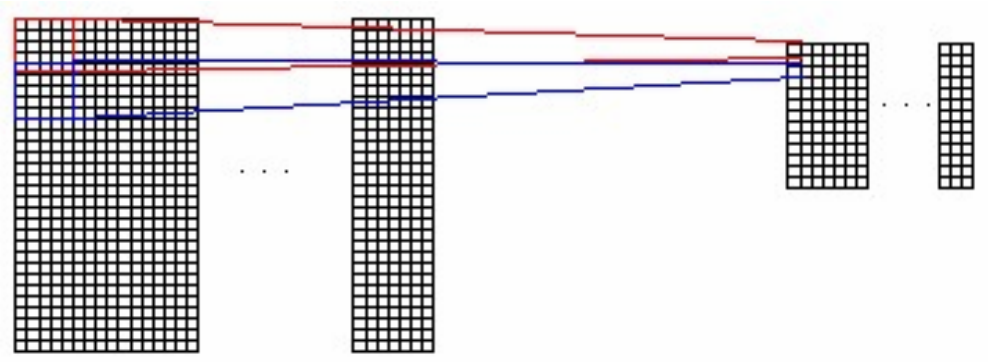

Figura 4.4 Selección de las ventanas.

Al aplicar este algoritmo de reducción, obtenemos la imagen mostrada en la figura 4.5 que es una imagen reducida de la imagen segmentada, en esta imagen también se puede observar que el obstáculo se encuentra en una tonalidad gris clara mientras que los demás objetos se encuentran en una tonalidad de gris oscuro.

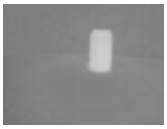

Figura 4.5 Imagen reducida.

4.5 Imagen binaria.

En el siguiente paso se procede a aplicar un umbral a la imagen, esto con la finalidad de convertir la imagen segmentada en una imagen binaria. Se decidió aplicar el umbral 145 ya que al experimentar con diversos valores se llegó a la conclusión de que al umbralizar con este valor, los contornos del obstáculo en la imagen en blanco y negro se encuentran mejor delineados y mas apegados a las proporciones del obstáculo en la imagen original.

Para convertir la imagen en binaria una vez que se ha decidido el umbral, se hace lo siguiente:

1. Todos los valores que sean igual o mayores al umbral, se convierten en el valor RGB doscientos cincuenta y cinco.
2. Todos los valores que sean menores al umbral, se convierten en el valor RGB cero.

El resultado se muestra en la siguiente figura.

Figura 4.6 Imagen binaria.

Como se puede observar en la figura anterior, el obstáculo se encuentra representado en color blanco y los demás objetos dentro de la imagen que no son de nuestro interés se encuentran en negro.

4.6 Adaptación del vehículo explorador de Braitenberg a visión artificial.

Hasta ahora se ha hecho el procesamiento de la imagen para obtener una imagen en la cual se encuentra representado únicamente el objeto o los objetos de nuestro interés según sea el caso, el siguiente paso consiste en adaptar el comportamiento del vehículo explorador de Braitenberg a nuestro robot tipo LEGO, para esto se debe obtener una representación de la localización del o de los obstáculos dentro de la imagen en blanco y negro, esto se lleva a cabo generando un histograma de frecuencia de luminosidad a partir de la imagen en blanco y negro.

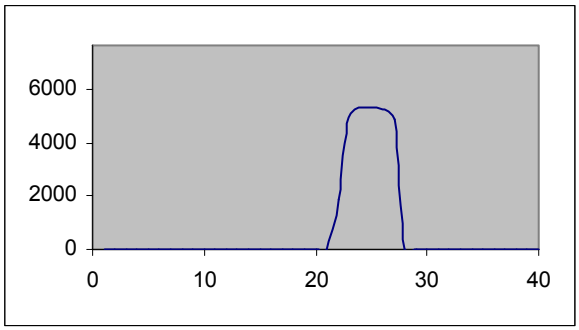

Figura 4.7 Histograma.

Debemos recordar que la imagen procesada para obtener este histograma es de 40 x 30 pixeles y es una imagen en blanco y negro, por lo cual en el histograma el eje "X" representa los cuarenta pixeles del largo de la imagen. En el eje "Y" se representa el nivel de saturación de pixeles blancos en las cuarenta columna de la imagen, dichas columnas tienen treinta posiciones que son los treinta pixeles del ancho de la imagen y cada posición contiene un valor 0 o 255, donde 0 representa la ausencia de color (negro) y 255 representa que en esa posición se encuentra una porción del obstáculo que en la imagen se encuentra

representado en color blanco, dado esto, si en toda una columna de la imagen se encuentran porciones del obstáculo (color blanco) el valor del eje "Y" será 7650 (30 x 255), si solamente la mitad de esa columna contiene porciones blancas, entonces el valor será 3825 (15 x 255)

La distancia óptima que debe haber entre el robot y el obstáculo para tomar una decisión acerca del movimiento que ejecutará el robot, se obtuvo de manera experimental, para esto, se colocó al robot a distancias diferentes, se hizo el procesamiento de la imagen mencionado en las secciones anteriores y se obtuvo su respectivo histograma para cada una de las distancias, estas distancias fueron medidas de forma manual; al comparar los histogramas y sus respectivas distancias, se llegó a la decisión que el valor del pico o de los picos del histograma, fuera de siete mil seiscientos cincuenta, que es el valor máximo que se puede tener en el histograma y se da cuando el robot se encuentra a una distancia aproximada de quince centímetros del obstáculo.

Si el o los picos que se encuentran en el histograma no son iguales a siete mil seiscientos cincuenta, entonces el movimiento que ejecutará el robot será el programado con el nombre adelante, esto hará avanzar el robot hacia adelante y se reducirá la distancia entre el robot y el obstáculo, al igual que la frecuencia de pixeles blancos del histograma también se incrementará, esto se hará hasta que le distancia se reduzca a la óptima que es de aproximadamente de quince centímetro como se mencionó anteriormente, y cuando pase esto, ya se podrá tomar la decisión del movimiento que tendrá que ejecutar el robot con el fin de evitar una colisión con el obstáculo.

Si se decide que se encuentra ante un obstáculo, se procede a verificar en que parte del campo de visión del robot se encuentra(n) dicho(s) obstáculo(s). Para esto se divide en tres el campo de visión (figura 4.7 y figura 4.8)

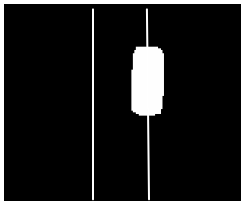

Figura 4.8 División del campo de visión.

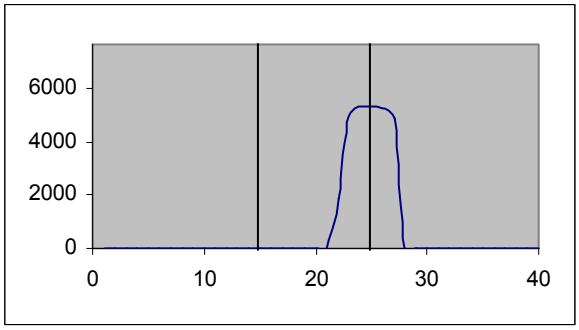

Figura 4.9 División en el histograma.

Las divisiones se hacen de la siguiente manera: se toman sobre el eje "X", la región uno está limitada a la izquierda por la columna uno y a la izquierda por la columna quince, la región central va desde la columna dieciséis hasta la columna veinticinco, la región derecha va desde la columna veintiséis hasta la columna cuarenta.

Con esta información se decide hacia donde se desplazará el robot para evitar el obstáculo, si el obstáculo se encuentra localizado en la tercera región (región de la derecha)

entonces se dará un giro hacia la izquierda, si se encuentra un obstáculo en la región uno (región de la izquierda) se dará un giro a la derecha, si se encuentran obstáculos en las regiones uno y tres simultáneamente, entonces se hará el movimiento de crisis, esto hará avanzar hacia atrás al robot y después dará un giro de ciento ochenta grados para evitar los obstáculos.

Si todos los valores que se obtienen del histograma son ceros, entonces esto quiere decir que nos encontramos frente a una pared y se procede a entrar en el movimiento de crisis, esto evitará que el robot colisione con la pared.

4.7 Conclusiones del capitulo.

En este capítulo se explicó la forma en que se implementó el comportamiento explorador de los vehículos de Braitenberg a un sistemas de visión, utilizando la cámara de video provista con el kit LEGO Vision Command , se explicó todo el tratamiento de la imagen original para decidir si es que nos encontramos frente a un obstáculo, si es que nos encontramos frente a una pared o si es que no hay un obstáculo en el rango de visión del robot. Además se mencionó la acción que se deberá ejecutar en caso de que se detecte un obstáculo dentro del rango de visión del robot.

Los resultados de adaptar el comportamiento explorador de los vehículos de Brainteberg al sistema de visión han sido en términos generales buenos, ya que el comportamiento que se esperaba del robot ha sido el que se esperaba. Se detectaron algunos problemas como por ejemplo, cuando el robot entra en estado de crisis y avanza hacia atrás,

lo hace de forma “ciega” y supone que no hay obstáculo atrás de el, de igual forma en el mismo estado de crisis cuando hace el giro de ciento ochenta grados si se encuentra un obstáculo al hacer ese movimiento inevitablemente chocará contra el. Por esta razón en un ambiente con obstáculos muy cercanos golpeará contra ellos, esto se podría resolver con el uso de otros sensores, como por ejemplo, sensores de proximidad, pero esto está fuera del alcance de esta tesis.