

# **Estudio e Implementación de un Algoritmo para el Seguimiento de Color en Video y la Respectiva Graficación de la Ruta del Objeto, Usando la Herramienta de Adquisición y Procesamiento de Imágenes de Matlab**

Baque Yoza José<sup>1</sup>, Chávez Burbano Patricia<sup>2</sup>  
Facultad de Ingeniería Eléctrica y Computación (FIEC)  
Escuela Superior Politécnica de Litoral (ESPOL)  
Campus Gustavo Galindo, Km 30.5 Vía Perimetral  
Apartado 09-01-5863. Guayaquil, Ecuador  
jbaque@fiec.espol.edu.ec<sup>1</sup>, pchavez@espol.edu.ec<sup>2</sup>

## **Resumen**

*El presente artículo presenta la descripción de un programa para detectar un determinado color en una secuencia de video. Este programa fue desarrollado usando el software Matlab 7.7 que nos permitió manejar video proveniente de una cámara web o de un archivo en formato AVI almacenado en la memoria. El objetivo de éste programa es poder hacer el seguimiento de un determinado objeto que se caracterice por su color, teniendo la opción de visualizar la trayectoria del objeto en movimiento y la posición de las coordenadas (x,y) del centro del objeto a seguir. El seguimiento de un objeto de color característico se puede aplicar en el análisis de organismos celulares que tienen un color determinado dada su propia naturaleza, al cual se les puede dar seguimiento y analizar su comportamiento de movimiento para una predicción futura de su trayectoria final y tomar decisiones favorables a los pacientes en caso de que éstos organismos sean malignos. El seguimiento de estrellas por parte de los astrólogos, como el seguimiento de robots en la robótica son ramas en las que se han aplicado el seguimiento de objeto de color característico.*

**Palabras Claves:** Matlab, localización, video, color.

## **Abstract**

*This paper presents a program's description for detecting a specific color in a video sequence. This program was developed using the software Matlab 7.7 which allows us to manage video from a Webcams or an AVI format file in the memory. The goal of the program is to follow an object with an specific color with the option of plotting its trajectory and obtaining its center position (x,y). The color pursuit can be applied in the analysis of cellular organisms which have specific color from their nature. The pursuit of a cellular organism can help in the prediction of its behavior and the correct decisions can be taken. The pursuit of stars is vital for the astronomer.*

**Key Words:** Matlab, location, video, color

## Introducción

Si estuviéramos interesados en analizar a través de un video el *comportamiento del movimiento* de un objeto de color característico, como por ejemplo un satélite natural, satélite artificial o inclusive un avión, se podría desarrollar un programa *seguidor de movimiento por color*.

Desarrollaremos un programa en MATLAB que adquiera imágenes por medio de una cámara web y busque en tiempo real un color determinado por el usuario, a través de un vector de valores RGB. Adicionalmente, el programa graficará su trayectoria y posición.

El proyecto se lo realizará a nivel de software basado en el estudio del procesamiento de adquisición de imágenes considerando la resolución, número de bandas, tramas por segundo entre otros parámetros.

Se realizará un estudio acerca de la forma como Matlab trabaja con la imágenes a través de la Matriz RGB, planos de la imagen, extracción de planos entre otras variables.

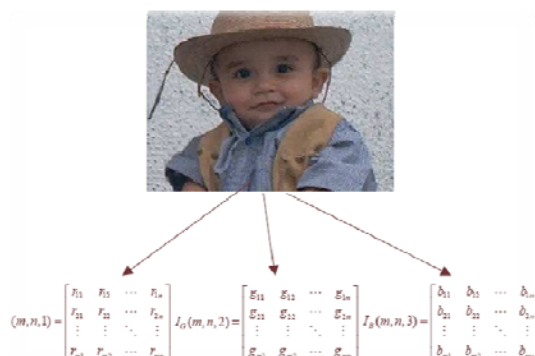
La búsqueda de color dentro de una imagen, obtenida de una secuencia de video se centrará en el uso de un *umbral* de comparación entre la matriz de la imagen de entrada y el vector de valores RGB a buscar, vector que se adquirirá cuando el *usuario* haga click derecho sobre el color a seguir.

Para darle una mejor visualización al seguimiento del objeto, se generará una gráfica de la ruta que este objeto sigue.

## 1. Propiedades de la imagen

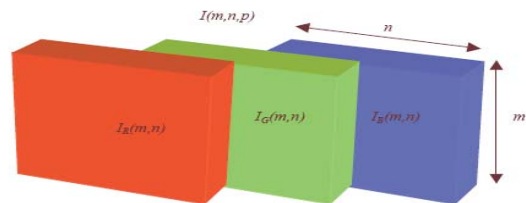
La resolución de una imagen es el número de filas y columnas que forman los píxeles. Una imagen que tiene  $m$  filas y  $n$  columnas tiene una resolución  $m \times n$ .

Ésta imagen tiene  $n$  píxeles en su eje horizontal y  $m$  píxeles en su eje vertical.



**Figura 1.1** Representación matricial (3 matrices) de una imagen a color en Matlab

El número de planos es el número de arreglos de píxeles que componen la imagen. Una imagen con escala de grises (gray-level) está compuesta por un sólo plano, mientras una imagen de color verdadero (true-color) está formada por 3 planos que son: Rojo(R), Verde (G) y Azul (B).



**Figura 1.2** Planos rojo, verde y azul que componen una imagen a color en Matlab

## 2. Adquisición de imágenes

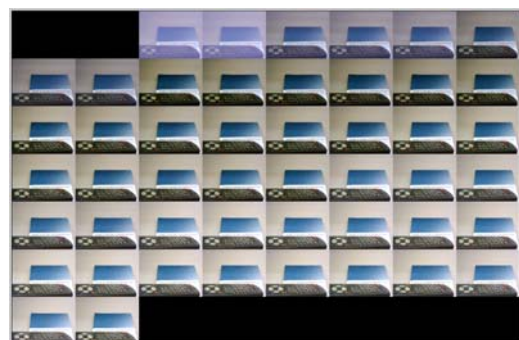
La adquisición de imágenes para el procesamiento provienen de dos fuentes: *cámara web* o un archivo de video de formato *AVI* ya almacenado en el computador.

Para la adquisición de las imágenes con la cámara web se usó la herramienta de adquisición de imágenes, utilizando funciones tales como *imaqhwinfo* para determinar el hardware conectado y establecer sus propiedades.

### 2.1. Parámetros para adquisición de imagen

La configuración de la cámara web se realiza con la función *videoinput*, configurada de tal manera que la adquisición sea de 50 tramas por disparo (trigger), el cual es activado por software. La adquisición empieza apenas inicia el video y tiene un intervalo de adquisición de trama de 1 (valor por defecto) para que la presentación del video sea considerada en tiempo real.

Se establecen 50 tramas por disparo debido a que al iniciar la cámara web las dos primeras imágenes son de valor 0 (pantalla negra) y las siguientes van poco a poco aclarando su nitidez. Por lo tanto, se dejan pasar las primeras 50 tramas para estabilizar la imagen. La siguiente figura ilustra este fenómeno.



**Figura 2.1** Adquisición de imágenes con cámara web

Una vez establecida las propiedades de adquisición de la cámara web, se adquiere la imagen con la función *getdata*, la cual retorna por defecto una imagen en formato *uint8* (entero sin signo de 8 bits).

La función también permite establecer el formato de la imagen, por ejemplo *double* (número racional). Sin embargo, se conserva el formato *uint8* para simplificar la presentación de la imagen con la función *imshow*.

Para la adquisición de la imagen de un video con formato AVI, se usa la función *aviread*, la cual retorna una estructura cuyo campo *cdata* es el que contiene las tramas del video.

### 3. Identificación del color

#### 3.1 Ecuación “Diferencia” y Umbral

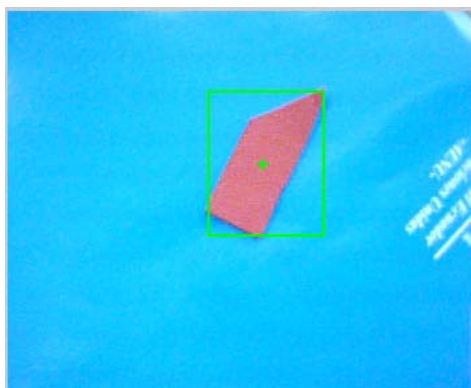
La identificación del color se realiza mediante una comparación (resta) entre los planos RGB de la imagen de entrada con el valor RGB ingresado por el usuario.

$$Diferencia = [(R_{img} - R_{us}) < umbral] \wedge [(G_{img} - G_{us}) < umbral] \wedge [(B_{img} - B_{us}) < umbral]$$

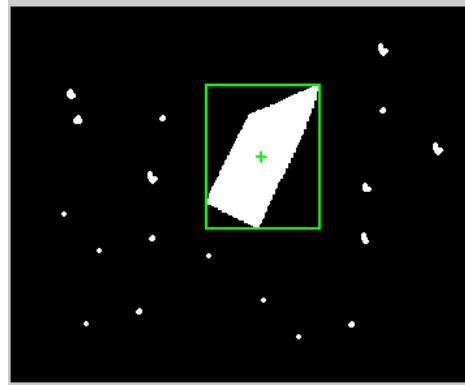
El *umbral* se determina de forma *experimental*, llegando a la conclusión que entre más semejanza tenga el color a buscar con el fondo, mayor debe ser calibrado este umbral.

Esta ecuación retorna una matriz *binaria* del mismo tamaño que la imagen de entrada donde *1* representa píxel *blanco* (color a seguir) y *0* píxel *negro*. Para *precisar* el color a buscar, se utilizan dos operaciones *morfológicas* para eliminar pequeñas áreas dentro de la imagen resultante.

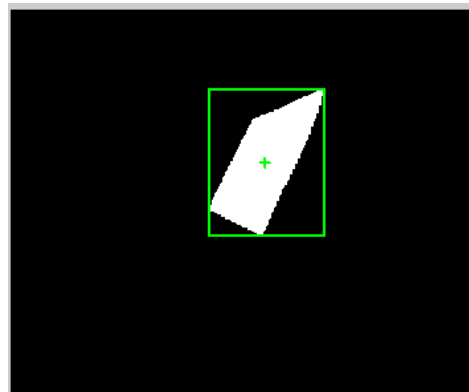
Se usó dilación seguida por erosión usando la función *bwmorph* para luego realizar una erosión seguida por la dilación. Los resultados de estas operaciones se muestran en las siguientes figuras:



**Figura 3.1** Imagen con el objeto a buscar (R<sub>us</sub>=255, G<sub>us</sub>=138, B<sub>us</sub>=139)



**Figura 3.2** Imagen resultado de la Diferencia, 1=píxel blanco (color a seguir), 0=píxel negro



**Figura 3.3** Imagen luego de realizar las operaciones morfológicas

### 4. Encuadre del objeto y trazado de ruta

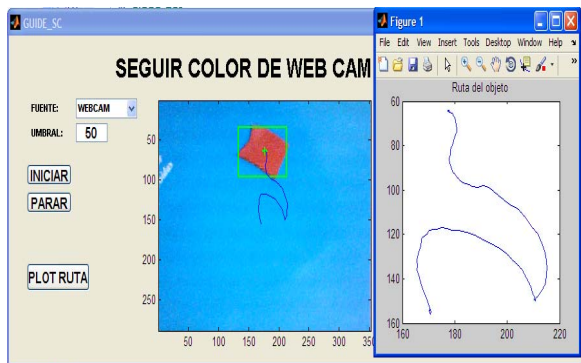
Luego de haber obtenido la matriz de Diferencia y de haber realizado las operaciones morfológicas, el siguiente paso es encontrar el área que representa al color buscado en la imagen. Para este fin se usa la función *regionprops*, la cual tiene como argumento de entrada una matriz de identificadores que se la obtenemos de la función *bwlabel*.

La función *regionprops*, en su forma básica, retorna tres parámetros: área, coordenada del centro y coordenada de un rectángulo. Todas estas son propiedades de los objetos (píxeles blancos) de la figura.

El rectángulo del objeto nos sirve para visualizar su movimiento, mientras que la coordenada del centro para graficar y almacenar su trayectoria. El área del objeto la usamos para eliminar los objetos menores a 200 píxeles.

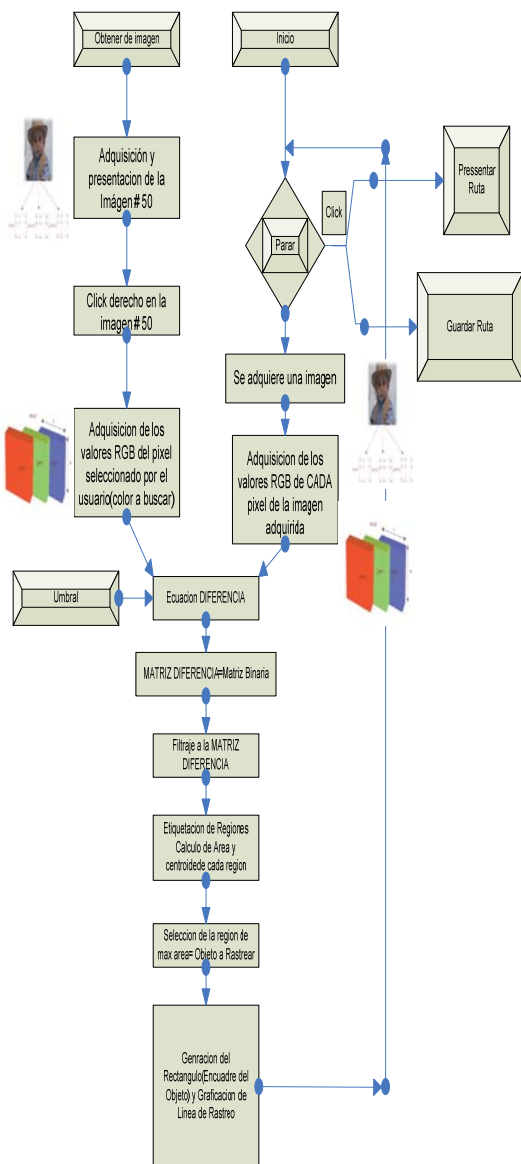
La trayectoria se almacena en una matriz nula, concatenando cada valor de posición. La interfaz gráfica (GUI) del programa tiene una opción para almacenar la ruta recorrida usando la función *save* en un archivo *.mat*.

La siguiente figura muestra la GUI y la ruta del objeto:



**Figura 4.1** GUI de seguimiento de color y trazado de la ruta

#### 4.1 Diagrama de Flujo



**Figura 4.2** Diagrama de flujo

### 5. Conclusiones y Recomendaciones

El umbral no debe ser muy alto, se recomienda entre 30 y 60, menor a 30 es probable que el programa no detecte el objeto con el color buscado y si es más de 60 es probable que se pierda la identidad del objeto debido a que hay un mayor nivel de tolerancia en los tonos del color a buscar y pueda que se rastree a un objeto distinto.

Entre más cercana estén los valores RGB del fondo (digamos la pared) a los valores RGB del objeto, menor debe ser el valor del umbral, esto se debe al parecido ya que la diferencia de tonos será menor.

Si existen 2 o más objetos del mismo color, el programa sólo tomará en consideración para el rastreo al de máxima área.

El programa tiene un cierto grado de iteratividad debido a que el umbral se lo puede cambiar hasta encontrar el valor del umbral que optimice el rastreo.

La Iluminación es un factor muy importante para evitar ruido o mala interpretación del color a buscar. Además si se usa una cámara web de mejor captura el programa trabajaría mejor ya que la nitidez de la imagen es también importante. La cámara web que se utilizó, fué de marca Genius, modelo Eye 110 con resolución 352x288 pixeles. Una alternativa a futuro se podría considerar la minicámara digital filma con sonido Gtia Navidad Sp, guarda en formato AVI con una resolución de 640x480 pixeles.

Las operaciones morfológicas abrir (open) y cerrar (close) reducen los espacios “blancos” que crea el ruido provocado por perturbaciones de la luz o por un umbral inadecuado.

Si se generan muchas trayectorias de forma casi aleatoria sólo cambie el valor del umbral a un valor inferior, éste le dará efectividad en la graficación de la trayectoria del color escogido por el usuario.

### 6. Agradecimientos

A mi madre, mis hermanos y a la Msc. Patricia Chávez profesora de la Materia de Graduación quién estuvo atenta y presta para apoyarnos en todo momento y de todas las formas posibles para culminar este proyecto y con el proceso de graduación.

### 5. Referencias

- [1] Cuevas Jimenez, Erik Valdemar; Zaldivar Navarro, Daniel, “Visión por computador utilizando Matlab y la herramienta de procesamiento digital de imágenes”
- [2] Gonzáles, Wood, Eddins, “Digital image process”
- [3] Stephen Westland, Caterina Ripamonti, “Computational Colour Science Using MATLAB”
- [4] Webinar de procesamiento de imágenes de mathworks
- [5] <http://www.imageprocessingplace.com/>

[6] Ing.DiegoBarragán,toriales,<http://www.matpic.com>

[7]

<http://www.proton.ucting.udg.mx/tutorial/vision/cursovision.pdf>

[8]

[http://www.matpic.com/VIDEOS\\_TUTORIALES.html](http://www.matpic.com/VIDEOS_TUTORIALES.html)

[9]

<http://www.youtube.com/user/diegokillemall>