

Obtención de los canales RGB de una imagen.

Garfias López José De Jesús. Autor, *Student, IPN, (Instituto Politécnico Nacional, Calz. Ticomán 600, San José Ticomán, 07340 CDMX, garfiaslopez@gmail.com)*

Resumen— El modelo RGB, abreviación en inglés de los tres colores primarios red (rojo), green (verde) y blue (azul), es un modelo aditivo de color que toma como base la misma forma en la que los seres humanos percibimos los colores e imágenes de nuestro entorno, a través del sentido de visión dado por el órgano llamado ojo, el uso principal de este modelo se encuentra en los dispositivos electrónicos que tienen como objetivo la representación gráfica de imágenes o colores de cosas reales percibidas de la misma forma que el ojo humano. Los dispositivos mas comunes con este modelo son las camaras fotograficas, televisores, escaneres y monitores de computadoras.

I. INTRODUCCIÓN

El órgano conocido como ojo humano tiene un funcionamiento muy particular que nos ayudará a entender como funciona este modelo RGB, entonces primero necesitamos analizar y entender que es lo que provoca que nosotros como seres humanos podamos visualizar colores de nuestro entorno, despues extrapolaremos el mismo principio de funcionamiento hacia el modelo RGB dandole así la capacidad suficiente a los dispositivos electrónicos de la actualidad de poder sensar colores y representarlos de forma gráfica.

II. EL FUNCIONAMIENTO DEL OJO HUMANO

Si bien sabemos que el 50% de la información que recibimos de nuestro entorno, la obtenemos mediante el sentido de la vista, por lo que es muy importante para nosotros tener este sentido a la perfección si queremos llevar una vida mas facil, el ojo humano no es mas que un sistema optico formado por una lente y un dioptrio esférico que reciben respectivamente el nombre de cristalino y córnea.

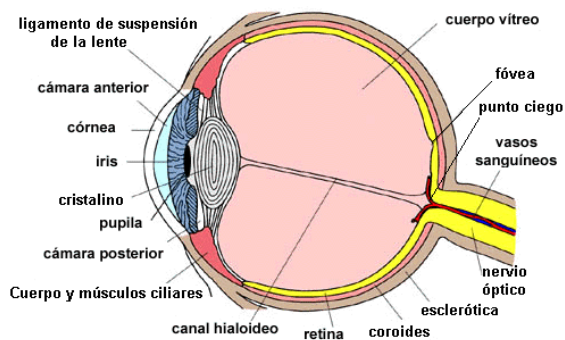


Fig. 1. Visualización de partes internas del ojo humano.

En la figura anterior se ven claramente las partes que forman el ojo. Tiene forma aproximadamente esférica y está rodeado por

una membrana llamada esclerótica que por la parte anterior se hace transparente para formar la córnea. Tras la córnea hay un diafragma, el iris, que posee una abertura, la pupila, por la que pasa la luz hacia el interior del ojo. El iris es el que define el color de nuestros ojos y el que controla automáticamente el diámetro de la pupila para regular la intensidad luminosa que recibe el ojo. El cristalino enfoca las imágenes sobre la envoltura interna del ojo, la retina. Esta envoltura contiene fibras nerviosas (prolongaciones del nervio óptico) que terminan en unas pequeñas estructuras denominadas conos y bastones muy sensibles a la luz.

Estos bastones y conos son células foto receptoras donde 120 millones son bastones y 6 a 7 millones son conos, estos últimos son los responsables de que el ojo humano pueda captar los colores, y se reparten en un 64% para conos que detectan el color rojo, 32% conos que detectan el color verde y un 2% para el color azul, estas células filtran estos colores y es transformada en información que es enviada al cerebro para interpretación quien se encarga de adicionarlos formando la extensa gama de colores que podemos apreciar, todos resultando de la base de mezclar esos 3 colores en diferentes porcentajes en un solo punto de la retina.

III. EL MODELO RGB

Un color expresado en modelo RGB sigue el mismo principio de funcionamiento de los bastones del ojo humano, en el cual especificamos en un rango desde cero hasta un valor máximo que tanto queremos adicionar de ese color primario a nuestro color que queremos expresar, formando una tripla de 3 valores (rojo, verde y azul) que describen un color en el espectro de la gama de colores que el ojo humano puede visualizar. Dependiendo la aplicación es como vamos a seleccionar en que formato vamos a expresar estos 3 valores, pueden ser aritméticos en un rango simplemente de 0 a 1, donde 0 es la ausencia de ese color y un 1 que es la máxima expresión de ese color, o también se puede usar porcentajes en un rango de 0% a 100%, y el mas común que es utilizado en los dispositivos electrónicos, el rango de 0 a 255, esto es porque un byte que es compuesto por 8 bits el máximo valor que puede expresar en notación decimal es 255.

IV. DESARROLLO

Para poder visualizar los componentes RGB de una imagen, se tiene procesar en la computadora, y para ellos nos apoyaremos del software "Matlab" el cual nos simplifica la forma en la que declaramos tareas para que la computadora las realice, así nos centramos mas en el proceso que en la codificación.

A continuación, explicare los procedimientos que se tienen que realizar para poder descomponer en sus componentes RGB cualquier imagen.

- 1.- Necesitamos declarar 3 matrices, una para cada componente correspondiente al modelo RGB.
- 2.- Leer nuestra imagen que deseamos descomponer e ingresarla en la estructura de datos de matriz.
- 3.- Mediante dos fors de iteración vamos recorriendo nuestra matriz por fila y columna, píxel por píxel.
- 4.- En cada píxel que iteramos solo escogemos el canal que nos interesa y lo guardamos en la matriz designada para ese componente anteriormente declarado.
- 4.- Imprimir las matrices de componentes.



Fig. 2. Imagen original.



Fig. 3. Imagen de la componente R.



Fig. 4. Imagen de la componente G.



Fig. 3. Imagen de la componente B.

V. CONCLUSIONES

Entendiendo la teoría de como funciona un fenomeno en la naturaleza lo podemos extrapolar para nuestro beneficio en aplicaciones de ingeniería, aprovechando el poder de una herramienta como matlab, el descomponer una imagen en sus componentes RGB parece trivial, pero nos ayuda a comprender y comprobar la teoría, para que podamos visualizarla de una mejor manera y comprenderla.

VI. ANEXOS

```
% GARFIAS LOPEZ JOSE DE JESUS
% RGB DESCOMPOSITION
clear all, close all, clc
A=imread('garfias2.JPG', 'jpg');
[m n l]= size(A);
R=zeros(m,n,l);
G=zeros(m,n,l);
B=zeros(m,n,l);
for x=1:l:n
    for y=1:1:m
        R(y,x,1) = A(y,x,1);
        G(y,x,2) = A(y,x,2);
        B(y,x,3) = A(y,x,3);
    end
end
figure;
image(uint8(R));
figure;
imshow(A(:,:,1));
imwrite(uint8(R),'practica3_output_r.JPG');
figure;
image(uint8(G));
figure;
imshow(A(:,:,2));
imwrite(uint8(G),'practica3_output_g.JPG');
figure;
image(uint8(B));
figure;
imshow(A(:,:,3));
imwrite(uint8(B),'practica3_output_b.JPG');
```