UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

CUANTIFICACION DEL COLOR DE IMAGENES PARA FRAME BUFFERS REDUCIDOS

T E S I S

QUE PARA OBTENER EL TITULO DE

A C T U A R I O

P R E S E N T A :

ANGELICA GUADALUPE SU RAMOS

ASESOR:
ANA LUISA SOLIS

MEXICO, D. F. 1992

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

CUANTIFICACION DEL COLOR DE IMAGENES

PARA FRAME BUFFERS REDUCIDOS

1 .- INTRODUCCION.

Crear imágenes realistas es a menudo uno de los principales objetivos de la generación de imágenes por computadora. Un elemento clave para alcanzar esta meta es describir y desplegar correctamente los colores de los objetos que forman la imagen.

La representación del color real de los objetos, requiere de una cuantificación de la escena, por lo que se han desarrollado modelos que describen su interacción (modelos de iluminación) para luego poder presentar los resultados en algún dispositivo de despliegue.

Por tanto la generación de imágenes por computadora requiere:

1) Describir las características de los objetos en la escena.

2) Desplegar los resultados.

Una de las características que se debe especificar es el color de los objetos, esta es una tarea difícil si no hay forma de que el usuario conozca los valores que requiere para producir dichos colores. Para ayudar al usuario en la selección de color, se utilizan los ***Editores de Color***.

El proceso para desplegar imágenes a color se inicia desde el momento que se tiene la información de la imagen original, la cual describe el color y la intensidad de cada punto, y termina cuando el observador percibe la imagen desplegada. Los pasos que se realizan para llegar de uno a otro, implican transformaciones de los datos de la imagen, para acoplarlos a las ***limitaciones*** del dispositivo de despliegue.

Los dispositivos que permiten desplegar imágenes a color son los ***"frame buffers"***, es decir, el área de memoria para la

descripción de la imagen. El número de colores que logran desplegar estos dispositivos de manera simultánea, por lo general, es menor al número de colores que contiene la imagen original.

Para poder *desplegar una imagen, cuyo número de colores sea mayor al que permite desplegar el frame buffer*, se tiene que encontrar el *conjunto de colores* que represente en forma óptima la gama de colores de esta imagen. Una vez encontrado este conjunto se tiene que *mapear* la gama de colores de la imagen original a dicho conjunto.

Los objetivos que se persiguen en el presente trabajo son:

1) *Desarrollar un editor de color que ayude en la selección de color.*
2) *Presentar e implementar un algoritmo para encontrar el conjunto de colores óptimo para desplegar la imagen.*

2 .- PERCEPCION DEL COLOR.

2.1 .- Introducción.

El sistema visual humano es el que asimila, detecta e interpreta la información que está a nuestro alrededor, y es el que nos permite, por medio del ojo, detectar los diferentes colores.

La generación y despliegue de imágenes dá una alternativa para la simulación del mundo que nos rodea, pero las técnicas disponibles para desplegar imágenes sólo permiten reproducir un subconjunto limitado de los estímulos que percibimos del medio ambiente real.

Para obtener imágenes que se asemejen a la realidad es necesario relacionar el proceso perceptual de la visión con la tecnología que se tiene para desplegar imágenes.

El color es descrito por Webster como "la sensación resultante de los estímulos que recibe la retina del ojo de ciertas longitudes de ondas de luz". Esto implica que el color tiene un componente perceptual humano (la sensación producida), y un componente físico (las longitudes de onda de luz).

Los estímulos son dependientes de diversas variables, las cuales producen variaciones en la percepción del color.

1.- La cantidad de energía electromagnética que entra en el ojo.

2.- La cantidad de energía presente en cada longitud de onda de los estímulos.

3.- Las variaciones en las relaciones de espacio que existen entre los elementos en el campo de visión. Para ilustrar esto último, veamos lo siguiente: si tenemos un cuadrado de color gris inmerso en otro mayor de color verde, tiende a ser rojizo.

4.- Finalmente, un efecto conocido como la adaptación es el que permite al observador detectar los mismos colores para

objetos que están en escenas iluminadas, ya sea por la luz del día o por la luz de una lámpara. A pesar de que estos tipos de luz tengan magnitudes y características espectrales completamente diferentes, nos permiten observar el mismo color.

Por lo mencionado anteriormente, podemos darnos cuenta de que el color de un objeto no depende solamente del objeto en sí, sino que hay infinidad de factores que influyen en la determinación del color final. Algunos de estos factores son: la fuente de la luz que lo ilumina, el color del área que rodea al objeto y el sistema visual humano. Además, debemos considerar que algunos objetos reflejan la luz, y otros la absorben.

Los aspectos teóricos necesarios para dar sustento a lo antes dicho son tres :

a) La colorimetría.

Ciencia que estudia el color, relacionando los aspectos físicos y perceptuales del ojo humano.

b) Modelos de color.

Modelos que permiten determinar, de manera precisa, la gama de colores disponibles y, en consecuencia, obtener imágenes realistas.

c) Especificación interactiva del color.

Editores de color que el usuario utiliza para especificar de forma interactiva los colores.

A continuación, cada uno de los apartados señalados serán tratados con mayor detalle.

2.2 .- Colorimetría.

La colorimetría es una ciencia perceptual; estudia y trata de cuantificar cómo el sistema visual humano percibe el color. Este estudio está basado en el espectro electromagnético y en la distribución de la energía electromagnética, donde ésta última representa la forma en que la energía de los estímulos que recibe el ojo se distribuye en función de las longitudes de onda.

El espectro electromagnético incluye rayos gamma, rayos-X, rayos ultravioleta, e infrarrayos entre otros. La energía electromagnética que incide en el ojo humano viaja a través de diferentes longitudes de onda. El rango del espectro al cual es sensible el ojo humano, es llamado rango visible y se encuentra aproximadamente entre los 400 y 800 nanómetros. El ojo recibe esta energía como luz, que es percibida por el ojo en forma de colores que van desde el violeta hasta el rojo, pasando por el índigo, el azul, el verde, y el amarillo. El color percibido es

el resultado de mezclar diferentes longitudes de onda.

Un observador, al tratar de reproducir un color, va creando distribuciones de energía espectral; sin embargo, cuando el observador siente que ya logró obtener el color, se puede percatar de que las distribuciones de energía espectral de ambos colores son completamente diferentes. Esto nos permite especular que el sistema visual humano no tiene un receptor para cada longitud de onda. Estudios que se han realizado sobre el ojo humano observan que son sólo tres los receptores del color, y estos son llamados conos. Seleccionando estos tres conos correctamente podemos producir la sensación del color deseado. Con estos estudios, se ha detectado también que, seleccionando tres luces de control con longitudes de onda corta, mediana y larga, se puede lograr igualar más colores. Esto se debe a que con una sola luz de control sólo produciríamos luz monocromática, pero es posible igualar luces no monocromáticas, si el observador no distingue la diferencia de matiz y saturación entre ambas

luces (colores). Esta forma de igualar los colores es conocida como fotometría. Dos luces de control permiten igualar más colores que utilizando una sola, y la introducción de una tercera luz de control incrementa en forma dramática el número de colores que pueden ser igualados. La única restricción que se tiene al elegir las luces de control, es que una de ellas no pueda ser igualada por las otras dos.

Es usual igualar un color con alguna combinación de las intensidades de las tres luces de control si las intensidades que se requieren pueden ser predecidas dando la curva espectral del color que se quiere igualar. La gráfica de intensidad, como una función de la longitud de onda, es la curva espectral del color a prueba. La sensación del color a prueba es la suma de las sensaciones producidas por la intensidad del color en cada longitud de onda del rango visible. Si esta misma suma es producida por otra combinación de las intensidades de las tres luces de control, obtendremos el mismo color.

La sensación del color producida por una mezcla de longitudes de onda puede ser igualada sumando las intensidades de las luces de control requeridas para igualar cada una de las longitudes de onda. Las curvas de igualación ("matching curves") para un conjunto de luces de control se obtienen al graficar las intensidades de las luces de control necesarias para igualar una longitud de onda, en función de esta. Predecir las intensidades de las tres luces de control para igualar un color a prueba es el resultado de multiplicar la curva espectral del color a prueba, por las curvas de igualación.

El diagrama de cromaticidad es utilizado para representar las tres luces de control. Cada eje ortogonal representa cada una de estas luces: el eje L (luz de control con longitud de onda corta, azul), M (luz de control con longitud de onda mediana, verde), y H (luz de control con longitud de onda larga, rojo).

El triángulo creado por las intersecciones de los 3 ejes y el plano describe todos los colores que pueden ser reproducidos

por las tres luces (primarios). La curva describe los colores puros. El interior de esta curva representa todos los colores visibles. La cromaticidad de un color es su posición dentro de la gráfica: describe matiz y saturación. Para ilustrar lo anterior se presenta la siguiente figura:

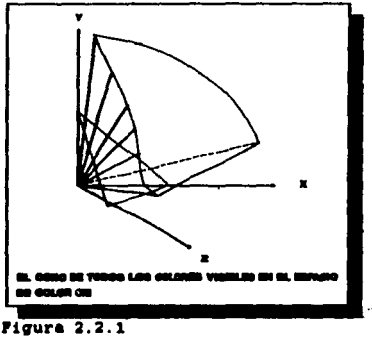


Figura 2.2.1

Lo mencionado anteriormente ha sido desarrollado y ha dadó resultados que pueden ser aplicados en la reproducción del color. De tal forma que si tenemos un monitor con colores primarios RGB y conocemos las curvas espectrales de cada uno de estos primarios, podemos determinar las intensidades LMH para

igualar cada uno de estos de la siguiente manera:

$$R = aL + bM + cH$$
$$G = dL + eM + fH$$
$$B = gL + hM + iH$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} L \\ M \\ H \end{bmatrix} = T \begin{bmatrix} L \\ M \\ H \end{bmatrix}$$

. Lo anterior es importante, pues significa que un conjunto de primarios puede ser puesto en términos de otro. Específicamente, si conocemos las cromaticidades de dos monitores (requeridas para pasarlos a un modelo estándar) podremos pasar del espacio de un monitor a otro.

2.3 .- Modelos de Color.

Un modelo de color es una especificación de un sistema de coordenadas de color en 3D y es un subconjunto visible del sistema de coordenadas, dentro del cuál se encierran todos los colores de una gama en particular. Por ejemplo, el modelo de color RGB es un cubo unitario, subconjunto de las coordenadas del sistema cartesiano.

El propósito de un modelo de color es permitir una especificación conveniente de los colores dentro de alguna gama de color. Un modelo de color puede ser utilizado para especificar todos los colores visibles.

Los tres modelos de color orientados hacia dispositivos de despliegue son el RGB (usado para los monitores CRT o raster) el YIQ (utilizado en las televisiones de color) y el CMY (usado para dispositivos de impresión). Desafortunadamente, ninguno de

estos modelos son fáciles de usar, porque no se relacionan directamente con las nociones intuitivas del color que tiene el observador, las cuales son matiz, saturación y brillo. Por tal razón se han desarrollado otros modelos, basados en estas nociones intuitivas, que tienen como meta su facilidad de uso, tales como el HSV (llamado algunas veces HSB), HLS, y HVC.

Cada modelo puede ser pasado a otro modelo, por medio de alguna transformación. Para el modelo RGB, se utiliza la conversión hacia el espacio CIEXYZ. Esto es de gran importancia, pues el espacio CIEXYZ es el modelo estándar a nivel mundial. A los demás modelos se les aplica una transformación para pasarlos al modelo RGB.

2.3.1 .- Modelo CIEXYZ.

En 1931, la Commision Internationale de l'Eclariage (CIE) definió a X, Y, y Z como los estándares primarios. Las tres funciones utilizadas para igualar un color se muestran a continuación. Estas se determinan de manera experimental.

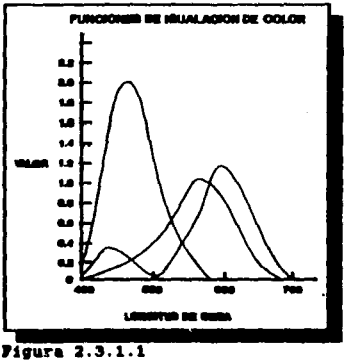


Figura 2.3.1.1

Los valores que toman los colores primarios XYZ para igualar un color son solamente positivos. La Y primaria es la función que iguala la sensación de luminosidad.

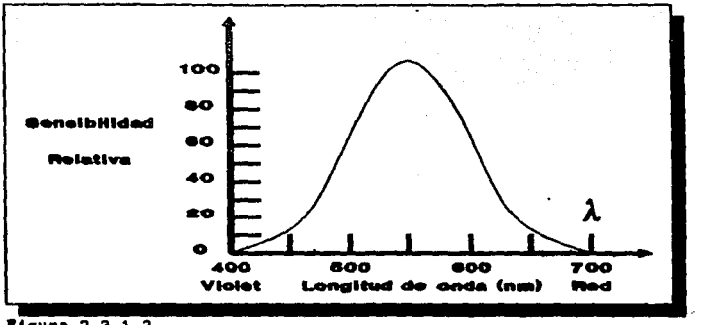


Figura 2.3.1.2

Los valores x_λ, y_λ y z_λ no son las distribuciones espectrales de X, Y, y Z, sino que son funciones auxiliares usadas para calcular cuanto de X, Y, y Z se debe mezclar para generar una sensación de color.

Las tres funciones CIE usadas para igualar un color son combinaciones lineales de las funciones auxiliares que se utilizan para generar una sensación de color por medio del rojo, verde, y azul. Esto significa que la definición de un color por medio de los colores primarios rojo, verde, y azul puede ser

convertido a coordenadas CIEXYZ a través de una transformación lineal, y viceversa.

Las cantidades de X, Y, y Z necesarias para igualar un color con una distribución de energía espectral P(λ) son :

$$X_1 = k\int P(\lambda)\, x_\lambda d\lambda \quad Y_1 = k\int P(\lambda)\, y_\lambda d\lambda \quad Z_1 = k\int P(\lambda)\, z_\lambda d\lambda$$

La siguiente figura muestra el espacio XYZ que contiene todos los colores visibles.

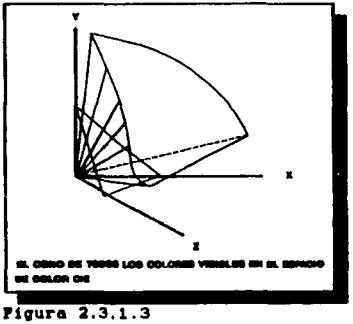


Figura 2.3.1.3

Dado (X_1, Y_1, Z_1) las cantidades necesarias para igualar un

color C. Entonces $C = X_1 X + Y_1 Y + Z_1 Z$. Definimos los valores cromáticos normalizando con $X_1+Y_1+Z_1$. lo cual puede ser pensado como la cantidad total de la energía de la luz.

$$x = \frac{X_1}{X_1 + Y_1 + Z_1}, \qquad y = \frac{Y_1}{X_1 + Y_1 + Z_1}, \qquad z = \frac{Z_1}{X_1 + Y_1 + Z_1}$$

Hay que notar que $x + y + z = 1$, y que x, y, y z están en el plano $X_1 + Y_1 + Z_1 = 1$. Si especificamos x, y y entonces $z = 1 - x - y$. Sin embargo no podemos obtener X_1, Y_1, y Z_1 a partir de x, y y. Para obtenerlos necesitamos conocer la luminosidad, es decir Y. Así que dado (x, y, Y_1) la transformación correspondiente (X_1, Y_1, Z_1) es :

$$X_1 = \frac{x}{y} Y_1, \quad Y_1 = Y_1, \quad Z_1 = \frac{1 - x - y}{y} Y_1$$

Estos valores dependen de la longitud de onda dominante y de la saturación, pero no de la energía luminosa. Para obtener el diagrama de cromaticidad CIE, hacemos la proyección del plano $X_1 + Y_1 + Z_1 = 1$ sobre el plano X_1Y_1, quedando la figura siguiente

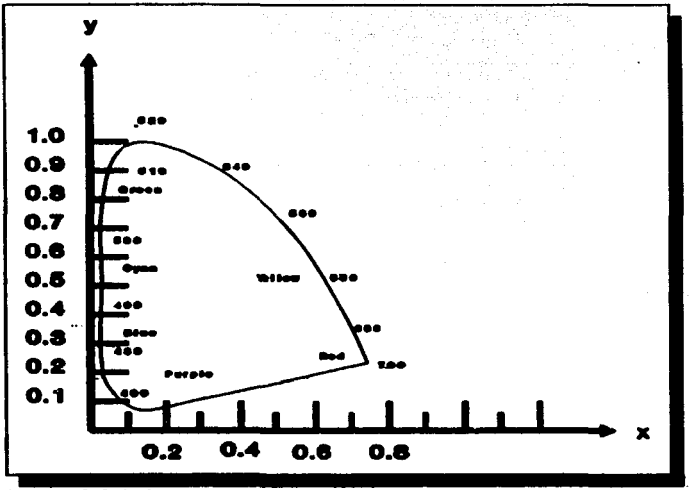


Figura 2.3.1.4

Otro concepto importante que nos permite manejar el sistema CIE, son los *colores complementarios*. Estos son los colores que pueden ser mezclados para obtener luz blanca. Hay colores que no tienen longitud de onda dominante y son llamados no espectrales (*colores complementarios*). Estos colores deben ser

expresados utilizando una longitud de onda complementaria como es el caso de los colores púrpuras y magentas. (que están ubicados en la parte baja del diagrama de cromaticidad (CIE)).

El diagrama de cromaticidad CIE también se usa para definir gamas o rangos de colores, y mostrar cuál es el efecto de juntar los colores. Así que si tenemos dos colores I, J, pueden ser sumados de tal forma que obtenemos una línea de colores, y si agregamos otro color K, obtendremos un triángulo.

Otra utilidad que tiene el diagrama de cromaticidad es que puede utilizarse para comparar las gamas disponibles de varios dispositivos de despliegue a color y dispositivos de impresión.

2.3.2 .- Modelo de Color RGB.

El modelo de color Red, Green, y Blue (RGB) es usado en monitores raster y CRT y emplea un sistema de coordenadas cartesianas. El subconjunto de interés es el cubo unitario mostrado en la figura 2.3.2.1. La diagonal que atraviesa el cubo representa los niveles de grises, que van desde el negro con coordenadas (0, 0, 0) al blanco con coordenadas (1, 1, 1). En esta diagonal las cantidades correspondientes a cada color primario son iguales. El cubo unitario representa la gama de colores disponibles que se tienen en el monitor.

Hay que considerar los siguientes tres puntos en el modelo RGB :

(1) El cubo RGB difiere de monitor a monitor dependiendo de las características físicas de los fósforos. En otras palabras, un punto definido con los valores (R, G, B), producirá diferentes colores de monitor a monitor.

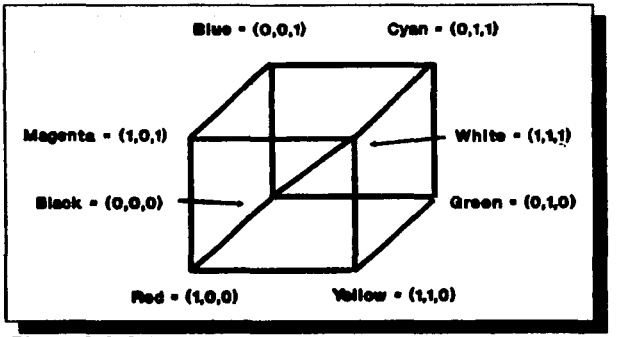


Figura 2.3.2.1

(2) El cubo RGB representa un subconjunto de todos los colores percibidos por el sistema visual humano.

(3) El cubo no es un espacio de color en el cual se tenga una percepción uniforme. Esto se refiere al hecho de que si tomamos intervalos uniformes en alguna línea recta que esté dentro del espacio, los colores producidos no son uniformemente diferentes para el sistema visual humano.

En los tres puntos citados anteriormente se mencionó el hecho de que el cubo RGB varía de monitor a monitor, de tal forma que si queremos convertir un color especifico de un monitor a otro monitor debemos pasar del espacio RGB de cada monitor al espacio CIEXYZ, por medio de transformaciones. La forma que debe tener cada transformación es la siguiente :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{..... ECUACION 1}$$

Los valores Xr, Xg, y Xb son los pesos aplicados a los colores del monitor RGB para encontrar X, y así sucesivamente para Y y Z.

Definiendo a M como la matriz de 3x3 utilizada en la transformación anterior, la ecuación (1) queda como:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{.....ECUACION 8}$$

Definimos M1 y M2 como las matrices correspondientes a dos monitores para convertir a la gama CIEXYZ. $M2^{-1}$ M1 convierte del monitor 1 al monitor 2.

Las coordenadas cromáticas para los fósforos del RGB podemos obtenerlas por medio de las especificaciones de su diseño. Denotando las coordenadas por (x_r, y_r) para el rojo, (x_g, y_g) para el verde, y (x_b, y_b) para el azul, y definiendo $C = X_r + Y_r + Z_r$, podemos escribir las siguientes relaciones para el color primario rojo :

$$x_r = \frac{X_r}{X_r + Y_r + Z_r} = \frac{X_r}{C_r}, \quad X_r = x_r C_r$$

$$y_r = \frac{Y_r}{X_r + Y_r + Z_r} = \frac{Y_r}{C_r}, \quad X_r = y_r C_r$$

$$z_r = (1 - x_r - y_r) = \frac{Z_r}{X_r + Y_r + Z_r} = \frac{Z_r}{C_r}, \quad X_r = z_r C_r$$

..... ECUACION 3

Haciendo algo similar para el verde y azul la ec. (1) adquiere la siguiente forma :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_r C_r & x_g C_g & x_b C_b \\ y_r C_r & y_g C_g & y_b C_b \\ (1 - x_r - y_r) C_r & (1 - x_g - y_g) C_g & (1 - x_b - y_g) C_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

..... ECUACION 4

Los valores desconocidos C_r, C_g y C_b pueden ser encontrados de dos formas. Primero, las luminosidades de máximo brillo Y_r, Y_g y Y_b, para el rojo, verde y azul respectivamente, pueden ser medidas por un fotómetro. Estos valores se combinan

con los valores conocidos y_r , y_g y y_b para producir :

$$C_r = Y_r / y_r, \quad C_g = Y_g / y_g, \quad C_r = Y_b / y_b$$

Sustituyendo estos valores en la ec. (1) nos queda :

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x_r \frac{Y_r}{y_r} & x_g \frac{Y_g}{y_g} & x_b \frac{Y_b}{y_b} \\ y_r & y_g & y_b \\ (1-x_r-y_r)\frac{Y_r}{y_r} & (1-x_g-y_g)\frac{Y_g}{y_g} & (1-x_b-y_g)\frac{Y_b}{y_b} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

..... ECUACION 6

Otra forma en la que podemos encontrar los valores para estas variables, es si conocemos X_w , Y_w y Z_w para el color blanco producidos cuando R = G = B = 1. Así la ec. (3) se transforma en lo siguiente :

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ (1-x_r-y_r) & (1-x_g-y_g) & (1-x_b-y_g) \end{bmatrix} \begin{bmatrix} C_r \\ C_g \\ C_b \end{bmatrix}$$

..... ECUACION 6

resolviendo este sistema de ecuaciones encontramos C_r , C_g y C_b .

2.3.3 .- Modelo de color CMY.

El cyan, magenta, y amarillo son los complementos del rojo, verde, y azul, respectivamente. Cuando son usados como filtros para restar color de la luz blanca, son llamados primarios subtractivos. El subconjunto de las coordenadas del sistema cartesiano para el CMY es el mismo que el del RGB, solo que el blanco es el origen en vez del negro. Los colores son especificados sustrayéndolos de la luz blanca, más que agregándoles oscuridad.

Esta especificación de los colores podemos verla de la siguiente manera: si tenemos una pieza de papel blanca y depositamos tinta amarilla, lo que estamos haciendo es quitarle el color azul, dejando los componentes verde y rojo de la luz reflejada. Similarmente el color cyan (azul y verde) lo obtenemos quitando el rojo del espectro blanco, y si quitamos el verde en

vez del rojo del espectro blanco obtenemos el color magenta (rojo y azul)

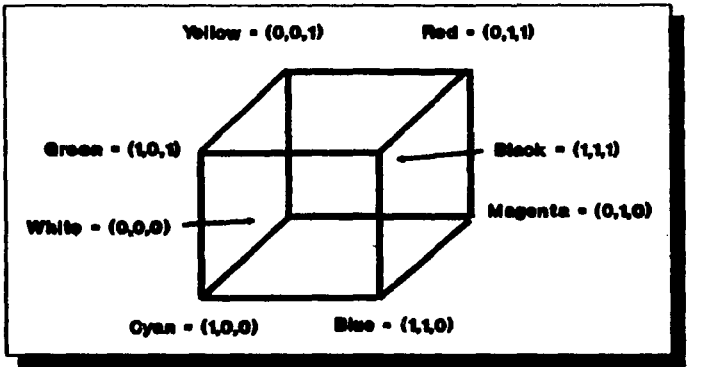


Figura 2.3.3.1

El modelo CMY es importante en dispositivos de impresión tales como impresoras electrostáticas, los cuales permiten mezclar colores.

La relación entre el modelo CMY y el modelo RGB está dada por:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{.....[illegible].}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \quad \text{.....[illegible].}$$

Los puntos que fueron considerados para el modelo RGB también son aplicados al modelo CMY. Las dos desventajas generales de esos modelos de hardware son:

(1) Es difícil trabajar en tales espacios.

(2) Un color C es identificado solamente por el valor de los parámetros (C,M,Y) o bien el valor de los parámetros (R,G,B). Los mismos valores del RGB pueden producir un color diferente en otro sistema.

Otro modelo de color, es el CMYK, el cual usa el negro (abreviado como K) como cuarto color. CMYK es usado en impresoras con cuatro colores y algunos otros dispositivos de impresión

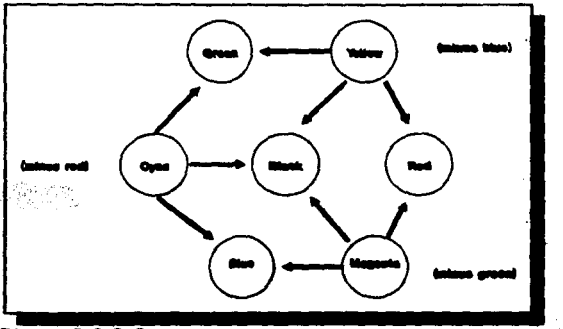


Figura 2.3.3.2

A partir de la especificación del CMY, podemos producir el modelo CMYK, de la siguiente manera:

$$K_1 = \min(C, M, Y)$$

$$C_1 = C - K_1$$

$$M_1 = M - K_1$$

$$Y_1 = Y - K_1$$

donde C_1, M_1, Y_1, K_1 son las coordenadas en el modelo CMYK.

2.3.4 .- Modelo YIQ.

El sistema YIQ es un sistema de color primario adoptado por la NTSC ("National Television Standards Committe") para la transmisión de las televisiones de colores. El sólido de colores formado por el sistema YIQ es una transformación lineal del cubo RGB. Su propósito es explotar ciertas características del sistema visual humano para maximizar la longitud de la banda fija y proveer compatibilidad con la televisión de blanco y negro.

La componente Y es conocida como luminosidad, y es el componente usado para recibir el negro y el blanco; I y Q representan la cromaticidad y son formadas de la diferencia del rojo con la componente Y *(R - Y)* y la diferencia del azul con la componente Y *(B - Y)* respectivamente.

La matriz que forma la transformación YIQ es expresada como:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ 0.60 & -0.28 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

y la inversa

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.00 & 0.96 & 0.62 \\ 1.00 & -0.27 & -0.65 \\ 1.00 & -1.10 & 1.70 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

La característica principal del sistema NTSC es que la señales de crómaticidad son transmitidas en una longitud de banda muy reducida comparada con la señal de luminosidad. Esta estrategia está basada en ciertos aspectos del sistema visual humano; principalmente en que el ojo es más sensitivo a los cambios en luminosidad que a los cambios en matiz o saturación; esto indica que nuestra habilidad para discriminar los colores es más débil que la habilidad que tenemos para discriminar la información monocromática. Lo anterior implica que la longitud de banda requerida para I y Q, es menor que la se requiere para Y.

Otro aspecto a considerar es el siguiente: objetos muy pequeños producen sensaciones limitadas de color, las cuales pueden ser especificadas con una dimensión de color.

2.3.5 .- Modelo HSV.

Este modelo fue propuesto por Smith (1978), es un modelo que permite al usuario seleccionar los colores. El modelo se basa en la forma en que un artista mezcla sus colores.

Smith hace referencia a la dificultad de producir un color por medio del RGB en el siguiente comentario:

Trata de variar el RGB mentalmente para obtener un rosa o un café. Esto es muy difícil.

...el siguiente modelo (HSV) imita la forma como el artista mezcla sus pinturas en su paleta: escoge un color puro, ó pigmenta e ilumina este color, agregándole blanco, obteniendo una tinta, o bien lo oscurece agregándole negro, logrando una sombra; en términos generales obtiene un tono de color agregando una mezcla de blanco y negro.

Los términos de matiz, saturación, y valor son descritos a continuación:

Matiz "Hue" : Cualidad por la cuál distinguimos una familia de colores de otra, por ejemplo el rojo del amarillo, o el verde del azul.

Saturación "Saturation" : Cualidad que nos permite distinguir un color fuerte de un color débil; intensidad del color. Como ejemplo podemos citar al rojo como color saturado y al rosa como un color desaturado.

Valor "Value" : Cualidad que hace distinción entre un color iluminado y un color oscuro. Este término se relaciona con el brillo del color.

El sistema es un sistema de coordenadas cilíndricas, y el subconjunto dentro del cual el modelo está definido es un hexcono, o una pirámide de seis lados, como se muestra en la siguiente figura:

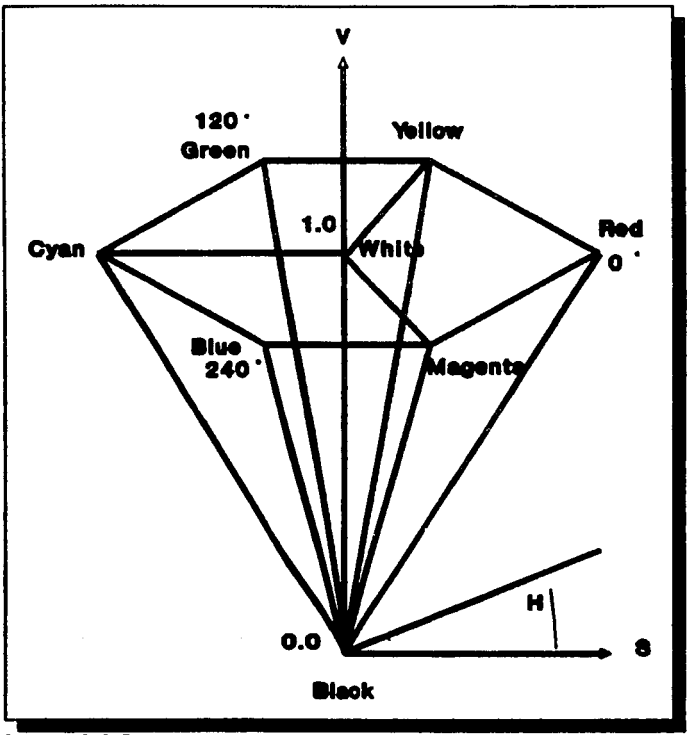

V
120 °
Green
Yellow
1.0
Cyan
White
Red
0 °
Blue
240 °
Magenta
H
0.0
S
Black

Figure 2.3.5.1

El tope del hexcono corresponde a V = 1, y contiene los colores relativamente brillosos. Sin embargo, el brillo percibido no es igual para todos.

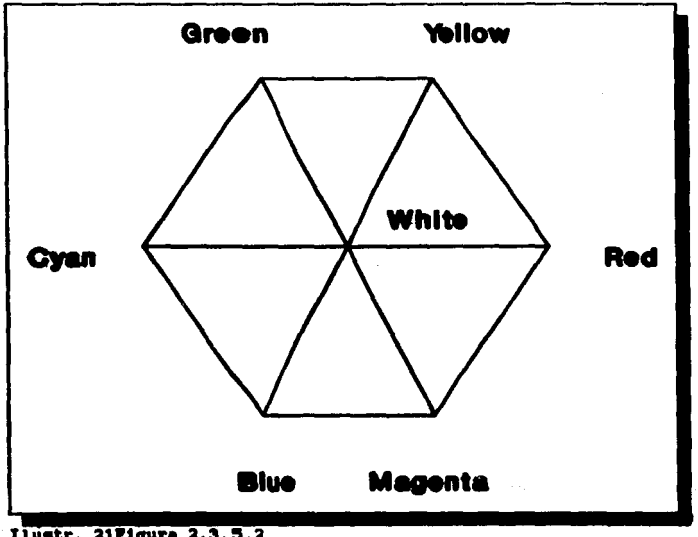
El matiz (H), es medido por el ángulo que se forma alrededor del eje vertical y está en el rango de 0 - 360. El valor de S (Saturación) varía de 0 a 1; en el eje central del hexcono el valor es cero, y en los lados triangulares del hexcono alcanza el valor de 1.

El hexcono tiene una unidad de altura en V, con el pico en el origen. El punto que está en el pico es el color negro y se tiene V = 0. En este punto los valores de H y S son irrelevantes. El punto S = 0 y V = 1 es el color blanco. Los valores intermedios de V para S = 0, son los grises.

En este modelo variar H corresponde a seleccionar un color. Decrementar S (desaturar el color), corresponde a agregar

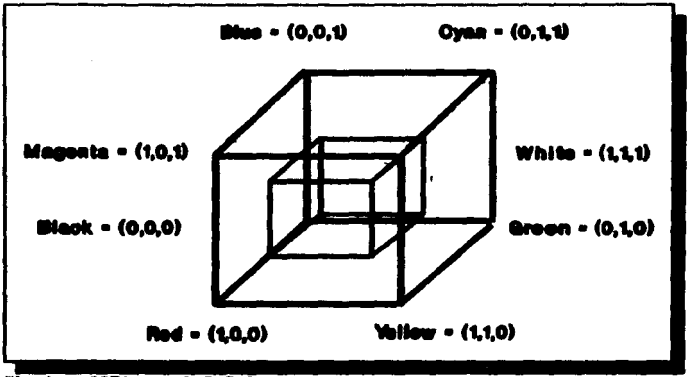
blanco. Decrementar V, corresponde a agregar negro. Si tenemos S = 1 y decrementamos V; conseguimos sombras. Decrementar S y V es lo que nos permite crear tonos. Cuando S = 0, el valor de H es irrelevante, en caso contrario, el valor de H sí importa.

El tope del hexcono puede ser visto como la proyección que se obtiene al ver el cubo RGB a través de la diagonal que va del blanco al negro, como se muestra a continuación:



Ilustr. 21Figura 2.3.5.2

El cubo RGB tiene subcubos; al hacer una proyección en cada uno de estos subcubos a través de su diagonal principal, también obtenemos un hexágono como con el cubo principal (véase figura anterior), pero de menor tamaño.



Ilustr. 22Figura 2.3.5.3

La siguiente relación se establece entre los seis vértices del RGB y los seis puntos del HSV.

RGB	color	HSV
(100)	red	(0, 1,1)
(110)	yellow	(60, 1,1)
(010)	green	(120,1,1)
(011)	cyan	(180,1,1)
(001)	blue	(240,1,1)
(101)	magenta	(300,1,1)

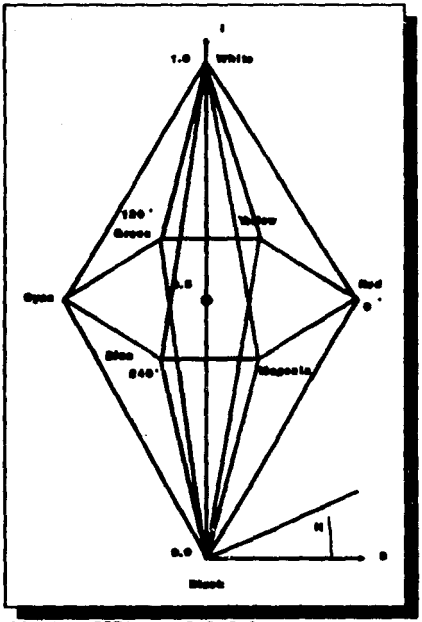
Los siguientes algoritmos definen de forma precisa la conversión del modelo RGB al modelo HSV y viceversa.

```
Procedure RGB_a_HSV(r,g,b : real; Var h,s,v : real);
{Entrada : r,g,b en el rango [0,1]
 Salida   : h en el rango [0.360], s y v en el rango
            [0,1] excepto si s=0, entonces h=INDEFINIDO,
            en este caso se le asigna el valor de una
            constante definida con un valor fuera del
            intervalo [0,360].}
begin
   max := Maximum(r,g,b);
   min := Minimum(r,g,b);
   v := max;       {Este es el valor de v (value)}
   {Siguiente paso, calcular la saturación}
   if max <> 0 then
      s := (max - min)/max
   else
      s  := 0;  {La saturación es cero si el rojo, el
                 verde, y el azul son 0}
   if s = 0 then
      h := INDEFINIDO;
   else
   begin
      delta := max - min;
      if r = max then
         h := (g - b)/delta; {El color resultante está
                              entre amarillo y magenta}
      else if g = max then
         h:=2 + (b - r)/delta; {El color resultante está
                                entre cyan y amarillo}
      else if b = max then
         h:=4 + (r-g)/delta; {El color resultante está
                              entre magenta y cyan}
      h := h * 60; {Convertir el matiz a grados}
      if h < 0 then
         h := h + 360;
   end;
end; {RGB_a_HSV}
```

```
Procedure HSV_a_RGB(Var r,g,b : real; h,s,v : real);
{Entrada: h en el rango [0,360] o valor INDEFINIDO, s y v
          en el rango [0,1].
 Salida : r,g,b en el rango [0,1].}
begin
   if s = 0 then {El color está en la linea central
                  negro-y-blanco}
      if  h =  INDEFINIDO then  {Color  no cromático:  no
                                 tiene matiz}
      begin              {Caso no cromático}
         r := v;
         g := v;
         b := v;
      end
      else Error
   else      {color cromático: s ≠ 0, asi que hay matiz}
   begin     {Caso cromático}
      if h = 360 then
         h := 0;
      h := h/60;
      i := Floor(h); {Floor regresa el entero más grande
                      <= h}
      f := h - i; {f es la parte fraccionaria de h}
      p := v * (1 - s);
      q := v * (1 - (s * f));
      t := v * (1 - (s * (1 - f)));
      case t of
         0 : (r,g,b) := (v,t,p);
         1 : (r,g,b) := (q,v,p);
         2 : (r,g,b) := (p,v,t);
         3 : (r,g,b) := (p,q,v);
         4 : (r,g,b) := (t,p,v);
         5 : (r,g,b) := (v,p,q);
      end; { case }
   end; {Caso cromático}
end { HSV_a_RGB }
```

2.3.6 .- Modelo HLS.

El modelo HLS (Hue, Lightnes, Saturation) está muy relacionado con el modelo HSV. Está basado en el sistema de color de Ostwald (Osttwald, 1931) y es usado por Tektronix. Puede ser considerado como una deformación del modelo HSV, solo que el blanco es jalado de tal forma que se forme otro hexcono.



Ilustr. 23Figura 2.3.6.1

Como en el modelo HSV, H se mide en grados y tiene un rango de 0 a 360; S es medido desde el eje vertical del doble hexcono, hasta los lados triangulares del mismo, y su valor va

desde 0 a 1. La luminosidad es cero para el negro (pico del hexcono inferior) y uno para el blanco (pico del hexcono superior).

El modelo HLS, tal como el modelo HSV, es fácil de usar. Todos los grises tienen S = 0, pero los matices más saturados están en S = 1, L = 0.5. Análogamente al modelo HSV, los colores del plano L = 0.5 no tienen el mismo brillo, así que dos colores diferentes con igual percepción de brillo, tendrán valores diferentes de L.

Finalmente, se debe establecer que ambos modelos el HSV y HLS son solo convenientes para especificar el color o un cambio de color en monitores cuyo espacio de trabajo sea el RGB. Los modelos no están completamente de acuerdo con los aspectos psicofísicos de los espacios de color. En particular los modelos implican que la máxima saturación ocurre en el mismo punto para todos los colores. Esto no es consistente con el hecho de que hay

ciertos colores con luminosidad alta (amarillo) y otros colores que tienen máxima saturación en niveles de luminosidad baja (azules).

2.4 .- Especificación Interactiva del Color.

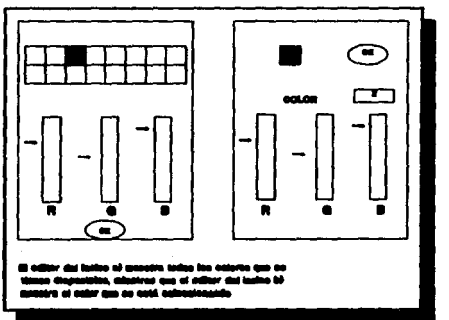
Es importante que al usuario se le permita seleccionar colores. Para lo cual se han desarrollado programas de aplicación, Editores de Color, que permiten especificar de manera interactiva los colores.

Si el conjunto de colores con el que cuenta el dispositivo de despliegue en el cual está trabajando el usuario es pequeño, es apropiado presentar un menú con todos los colores disponibles. Si, por el contrario, el conjunto es amplio, se tienen las siguientes opciones:

- Especificar las coordenadas del color en el espacio de color, así por ejemplo si se utilizará como espacio el modelo RGB se tendría que especificar los valores para el rojo (R), el verde (G), y el azul (B).

- Interactuar con la representación de un espacio de color, que permita realizar la especificación de color en forma intuitiva.

La especificación del color por medio de las coordenadas puede ser hecha por medio de marcadores, como se muestra en la siguiente figura, utilizando el modelo de color con el que cuenta el dispositivo de despliegue (RGB).

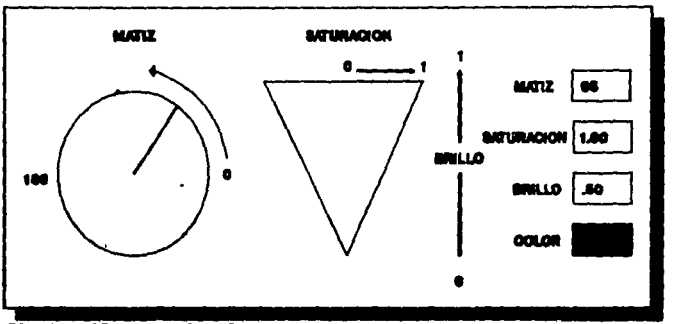


Ilustr. 24Figura 2.4.1

Esta especificación es apropiada si el usuario sabe como afecta cada coordenada del modelo de color a los colores.

Probablemente el mejor método de especificación interactiva, es aquel donde el usuario interactua con un espacio de color que le permita hacer una seleccción de color más

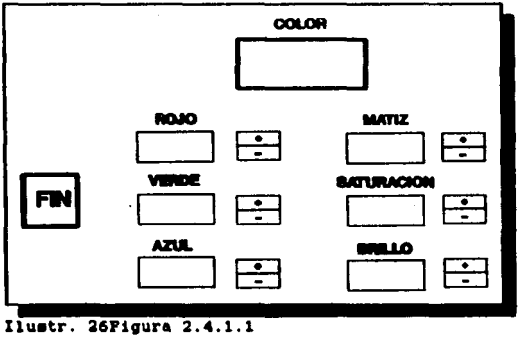
intuitiva. En este caso es común utilizar el modelo de color HSV. Un editor en el que se utilice este modelo se muestra a continuación.



Ilustr. 25Figura 2.4.2

2.4.1 .- Implantación de un editor de color.

El editor de color, desarrollado en este trabajo, permite que el usuario especifique los colores por medio de dos modelos de color HSV y RGB. Al especificar los colores por alguno de estos modelos, se da a conocer al usuario cuáles son los valores correspondientes en el otro, y se muestra en pantalla el color que se especificó.



Ilustr. 26Figura 2.4.1.1

Es importante haber implantado el modelo HSV junto con el modelo RGB, pues de esta manera se permite que el usuario especifique los colores de una manera intuitiva (por medio del matiz, la saturación y el brillo) y que conozca los valores que se requieren en el dispositivo de despliegue para generar dichos colores (es decir se conozcan los valores RGB).

Por tanto, el editor de color es una herramienta útil para el usuario en el proceso de generar imágenes, pues ayuda a describir las características de los objetos que estarán en la imagen; es decir, el usuario puede conocer los valores RGB que se requieren para producir los colores de dichos objetos.

3 .- DESPLIEGUE DE LA IMAGEN : CUANTIFICACION DEL COLOR.

3.1 .- Cuantificación.

La cuantificación se define como el proceso por medio del cual se dividen los rangos de las variables de entrada en intervalos y se les asigna valores que están en el dominio de las variables.

Hay dos clases de métodos de cuantificación: los uniformes y los aproximados. En la cuantificación uniforme los rangos de las variables de entrada son divididos en intervalos de igual longitud. En este caso, si se tiene que desplegar una imagen en un frame buffer de 8 bits por registro de color, se dejan 3 bits para el rojo, 3 bits para el verde y 2 bits para el azul. Por lo tanto los 256 colores desplegables serán la combinación de los 8 rojos, 8 verdes, y 4 azules. La cuantificación aproximada basa la división de los rangos de las variables de entrada en la distribución estadística de las

mismas. Para comparar la calidad de los métodos de cuantificación aproximada, frecuentemente se introduce una medida de distorsión o error métrico. Con esta medida de error se puede buscar el método que produzca una cuantificación "óptima". La variable que será utilizada es la imagen.

La *notación* que se utiliza en la cuantificación es la siguiente:

Dado x un punto $\in D^n$ (en 3-D para el propósito que se tiene), un cuantificador consiste de:

a) un conjunto de K puntos representativos en D^n:

$$Y = (Y_i,\ i = 1,2,\ldots,k),$$

b) una partición del espacio inicial en regiones:

$$R = (r_i,\ i = 1,2,\ldots,k),\quad r_i \subset D^n \text{ y } r_i \cap r_j = \emptyset$$

c) un mapeo de los puntos de entrada a los índices representativos:

$$p(x) = i \text{ si } x \in r_i .$$

d) la función de cuantificación, la cual mapea los puntos de entrada a los puntos de salida:

$q(x) = Y(p(x))$.

En la cuantificación del color de la imagen, Y es el conjunto de colores que mejor representa la gama de colores de la imagen original, k es el número de colores que nos permite desplegar el frame buffer, y p es un mapeo de los colores de la imagen original (los cuales tienen tres componentes), a un solo valor del pixel en la imagen cuantificada.

En la imagen original se denotan los colores de los pixels como $c_{i,j}$, donde i es el renglón que tiene en la imagen original y j es la columna, donde $0 \leq i \leq NI$ y $0 \leq j \leq NJ$. NI es la resolución vertical del dispositivo de despliegue en el cual fue generada la imagen y NJ es la resolución horizontal. Denotamos el valor del pixel del renglón i y columna j en la imagen final por $f_{i,j}$. Hay que notar que c es una matriz de

vectores, mientras que f es una matriz de escalares. Se supondrá que la imagen original y la imagen final tienen la misma resolución.

3.2 .- Cuantificación del Color de la Imagen.

La Cuantificación del Color de la Imagen se define como el proceso por medio del cual se selecciona el conjunto de colores que mejor representa a la gama de colores de la imagen inicial, y mapea el espacio de colores de la imagen original al conjunto de colores representativos.

La Cuantificación del Color de la Imagen se divide en cuatro fases:

1.- Tomar muestras de la imagen original, para obtener algunas estadísticas acerca del color.

2.- Seleccionar el conjunto de colores que represente de manera óptima el conjunto de colores de la imagen original.

3.- Mapear cada uno de los colores de la imagen original al color más cercano del conjunto de colores representativo

4.- Cuantificar y redibujar la imagen original.

La cuantificación uniforme del color de la imagen, nos da como resultado imágenes en las cuales se nota un cambio brusco de color, debido a que muchos de los colores del conjunto representativo no son utilizados en la imagen final. En cambio, si elegimos un conjunto de colores representativos, que se adapte a la gama de colores de la imagen original, aseguramos que utilizaremos cada uno de los colores de este conjunto, y así la imagen que obtendremos se acercará más a la original.
Este último concepto es el que se utiliza en la cuantificación aproximada del color de la imagen.

Cuando una imagen es cuantificada, cada uno de los colores definidos por los tres componentes de la imagen original debe ser transformado a un solo valor.

$$f_{i,j} = p(c_{i,j}) \text{ para } 0 \leq i \leq NI,\ 0 \leq j \leq NJ$$

Para desplegar la imagen cuantificada, debemos mapear el valor del pixel cuantificado, al color que le corresponde dentro

del conjunto de colores que se seleccionó, de la siguiente forma:

$$Yp(c_{i,j}) = q(c_{i,j})$$

Medida del error de un Cuantificador.

Para medir la diferencia entre la imagen original y la imagen cuantificada (el error total de cuantificación), se utiliza la siguiente fórmula:

$$D = \sum_{i,j} d(c_{i,j}, q(c_{i,j}))$$

donde d(x,y) es un función de distorsión, la que mide la "diferencia" entre los colores de la imagen original y la imagen final. Para medir esta diferencia se usará la distancia cuadrada en el espacio RGB:

$$d(x,y) = (x_r - y_r)^2 + (x_g - y_g)^2 + (x_b - y_b)^2$$

donde $x = (x_r, x_g, x_b)$ y

$y = (y_r, y_g, y_b)$.

3.3 .- Algoritmos para la cuantificación del color de la imagen.

Para cada una de las fases de cuantificación del color de la imagen se presentan varios métodos, los cuales son descritos a continuación:

FASE 1: Muestreo de la imagen original.

La información estadística que se necesita para seleccionar el conjunto de colores que represente de manera óptima la gama de colores de la imagen original, es un histograma de frecuencia de los colores de la imagen original. Con 24 bits por pixel se pueden generar alrededor de 16 millones de colores. Para almacenar todos estos colores en el histograma se requeriría una cantidad enorme de memoria; para conservarla, se utilizarán solo 15 bits por pixel, es decir 5 bits para cada componente del color (rojo, verde y azul).

Utilizando los 15 bits por pixel se obtendrá una tabla

con 2^{15} entradas.

Obviamente al tomar sólo 15 bits por pixel estamos reduciendo el número de colores de la imagen.

FASE 2: Seleccionar el conjunto de colores representativo.

En esta fase se expondrán dos algoritmos para la elección del conjunto representativo, y un proceso que permita mejorar la cuantificación.

Algoritmo Popular

Este algoritmo fue desarrollado por dos grupos independientemente en 1978: Tom Boyle y Andy Lippman en el grupo de arquitectura del MIT, y Ephraim Cohen en el Instituto Tecnológico de Nueva York.

La suposición en la que se basa este método es la

siguiente: el conjunto representativo puede ser construido encontrando las regiones más densas de color a partir de la distribución que tengan los colores en la imagen original.

Las regiones más densas se obtienen al elegir los k colores del histograma que tengan las frecuencias de ocurrencia más altas.

Este algoritmo funciona bien para muchas imágenes, pero no en aquellas que tienen una gama de colores muy amplia.

El tiempo aproximado que toma este algoritmo es de (Nk) donde N es el número de colores en el histograma.

Algoritmo del corte medio.

Este algoritmo fue propuesto por Paul Heckbert. Se basa en la suposición de que el conjunto representativo de colores se puede obtener dividiendo el cubo de colores RGB, en k regiones

que tengan igual número de colores, donde k es el número de colores que permite desplegar el monitor.

El cubo de colores RGB con el cual inicia este algoritmo, incluye a los NI x NJ pixels de la imagen original. El número de colores que estén dentro de este cubo depende de la resolución que se utilice.

Paso iterativo: División de un cubo de colores.

Lo primero que debe hacerse es encontrar los valores máximos y mínimos para cada componente del color (rojo, verde y azul), obteniéndose así el cubo mínimo de colores que contenga a los colores de la imagen original. Después se utiliza "una partición adaptativa" para decidir la forma en que será divido el cubo. Los colores del cubo son ordenados de acuerdo al componente de color que tiene la diferencia mayor entre su valor máximo y mínimo, posteriormente se divide el cubo original en el punto medio, esto es el punto donde se obtendrán dos regiones con

aproximadamente igual número de puntos, obteniendo dos cubos de colores en los cuales hay aproximadamente el mismo número de colores.

El paso mencionado anteriormente debe realizarse de forma recursiva hasta que se hayan generado k cubos de color. Si alguno de los cubos que va a ser dividido tiene sólo un punto, se buscará entre los cubos restantes el mayor, y este será dividido.

Después de haber generado k regiones de color, se le asigna un color representativo a cada una de estas regiones. El color representativo de cada región se obtiene promediando los colores contenidos en ella. Esta lista de colores que se obtiene es el conjunto Y.

El tiempo que toma este algoritmo en dividir cada región de colores, es proporcional al número de colores contenidos en ella. Generalmente, el tiempo que toma encontrar el conjunto

representativo es O(Nlogk), donde N es el número de colores que tiene la imagen original.

Se han realizado algunas pruebas de comparación sobre el algoritmo popular y el algoritmo del corte medio, y se ha visto que el segundo produce mejores resultados que el primero. En algunos casos la diferencia entre uno y otro es muy grande.

Se podrían utilizar otros criterios para la división de los cubos. Uno de ellos es dividir el cubo de colores en la coordenada que tiene mayor varianza.

Un Algoritmo del Punto Fijo para mejorar Un Cuantificador.

Gray, Kieffer, y Linde han descrito un algoritmo para encontrar un cuantificador localmente óptimo. Un cuantificador es

llamado localmente óptimo si la distorsión total D (error del cuantificador) no se puede decrementar a pesar de que se realicen pequeñas perturbaciones al conjunto Y (el conjunto representativo de colores).

Dado un conjunto representativo Y, la partición óptima R'(Y) es:

$$r'_k = \{ x : d(x,Y_k) \leq d(x,Y_j),\ j \neq k \}$$

que representa el lugar geométrico de los colores cuyo vecino más cercano es Y_k. Dada una partición R, el conjunto óptimo de colores representativos Y'(R) es el conjunto de las Y'_k tal que Y'_k es el centroide de todos los puntos $c_{i,j}$ que están en r_k. Utilizando R'(Y) y Y'(R) se define una función de mapeo T que perturba a Y, pero D nunca incrementa:

$$T(Y) = Y'(R'(Y)).$$

Con estas ecuaciones, se encuentra para cada color representativo Y_k, el lugar geométrico de todos los colores de

entrada $c_{i,j}$. cuyo vecino más cercano es Y_k.

El algoritmo del punto fijo puede ser usado para mejorar cuantificadores generados por el algoritmo del corte medio o el algoritmo popular.

FASE 3: Mapear los colores de la imagen original al conjunto de colores representativos.

Dada una distribución c y un conjunto representativo de colores Y, D se minimiza cuando q mapea un punto a un color representativo cercano a él:

$$p(x) = i \quad \left\{ \begin{array}{l} \\ \\ q(x) = Y_i \end{array} \right. \quad \text{Si } d(x,Y_i) \leq d(x, Y_j) \; j \neq i$$

La operación p(x) algunas veces es llamada "en búsqueda del vecino más cercano". Puede decirse que es un proceso "inverso al mapeo", pues mapea los colores en un valor de pixel.

Evaluando esta función para cada color de la imagen original, y salvando esta información en una tabla, podemos pasar rápidamente a la cuarta fase. Una alternativa es evaluar p una vez por pixel. Esta forma sería rápida si el número de pixels en la imagen es pequeño.

A continuación se presentan algunos métodos para calcular la función p:

Búsqueda Exhaustiva.

Para calcular p(x) se calculan las distancias a todos los colores representativos, y se toma la que sea mínima. Desafortunadamente, este método es lento. Se malgasta mucho tiempo al calcular distancias a puntos que probablemente no sean el vecino más cercano.

Búsqueda Ordenada Localmente.

Se crea una base de datos que consiste de una red de celdas cúbicas, las cuales contienen una lista ordenada de los colores representativos. Cada lista de la celda debe incluir todos los colores representativos quienes son los vecinos más cercanos de algún punto en la celda.

Cada entrada de la lista contiene dos variables: un número representativo (rep_no), que se refiere a un número asociado a los colores representativos, y su distancia (**dist**) al punto más cercano en la celda. A la variable **dist**, para los colores representativos que están en la celda, se le asocia el valor de cero. Para crear la lista, calculamos las distancias de los puntos que estén más cercanos a algún color representativo, los guardamos en la lista, y después ordenamos ésta tomando como llave la distancia.

Una forma de restringir la longitud de las listas es eliminar los colores representativos, que probablemente no serán los vecinos **más** cercanos de ningún punto de la celda.

Para calcular la función p(x) con esta base de datos, primero se encuentra la celda en donde está x, y después se ejecuta el siguiente procedimiento:

```
min = infinity;
```

```
i = 0;
while (min > entry[i].dist)
begin
   distance = d(x,y(entry[i].rep_no));
   if (distance < min) then
   begin
      nearest = i;
      min = distance;
   end
   i = i + 1;
end;
return(nearest);
```

La memoria y el tiempo que toma este algoritmo dependen del tamaño y número de celdas. Si se usa una red con N^3 celdas y la lista tiene aproximadamente L entradas, la memoria que se requiere para la base de datos es de $O(N^3 L)$. Las distancias que se calculan desde cada color representativo a una celda de colores toma un tiempo de O(k), y ordenar la lista toma O(LlogL); por lo tanto, el tiempo total que toma el algoritmo es de $O(N^3 k+N^3 LlogL)$.

FASE 4: Cuantificando y desplegando la imagen.

Para cuantificar, cada pixel de la imagen original $c_{i,j}$ se mapea a su valor de pixel, o índice dentro de la tabla de colores, por medio de las tablas de mapeo generadas en la fase 3. Los k colores del conjunto representativo se escriben en cada uno de los índices de la tabla de colores. De esta forma se redibujará la imagen con solamente k colores.

Dependiendo de la imagen, los errores de cuantificación serán obvios o invisibles. Las imágenes con frecuencias altas, (brillosas; tales como el cabello o el pasto) presentan errores de cuantificación menores que las imágenes en las que se tienen áreas grandes, sombreadas de forma plana, (es decir, no brillosas, tales como caras).

Cuando se cuantifica con muy pocos colores, o el conjunto de colores representativo elegido es muy pobre, los contornos en la imagen no son notorios. Las imágenes en las cuales se nota un

cambio brusco de color, donde los contornos son notorios, pueden ser mejoradas con la técnica de "dithering".

Dithering.

La estrategia básica de "dithering" es cambiar resolución de intensidad por resolución espacial. Promediando las diferentes intensidades de los pixels vecinos se pueden obtener colores que no están dentro del conjunto representativo. Si la resolución del frame buffer es lo suficientemente alta, el ojo se encargará de realizar la combinación de los colores. Esto es de gran ventaja, pues es posible reproducir imágenes con una amplia gama de colores con solamente 4 colores base.

Un camino utilizado por la técnica "dithering" es modularizar la imagen original con señales de frecuencia alta antes de cuantificar.

La técnica de "dithering" fue propuesta por Floyd y Stenberg. Su algoritmo trata de compensar el error de cuantificación introducido a un pixel, propangando este error a los pixels vecinos. Si la propagación es directa, solamente toma los pixels que están debajo o a la derecha del "pixel actual", y se pueden realizar la cuantificación y la propagación en un solo paso de arriba hacia abajo ("top-to-botton") sobre la imagen.

Un programa que realice la cuantificación y propagación utilizando el algoritmo de Floyd y Stenberg se vería de la siguiente manera:

```
For i=0 to NI-1 do
   For j=0 to NJ-1 do
      x = c(i,j);          ( se lee el color )
      k = p(x);            ( se encuentra el vecino más
                             cercano )
      f(i,j) = k;          ( dibujar la imagen cuantificada )
      e = x - Y(k);        ( error de cuantificación )
      c(i,j+1)   = c(i,j+1)   + e*3/8;
      c(i+1,j)   = c(i+1,j)   + e*3/8;
      c(i+1,j+1) = c(i+1,j+1) + e/4;
   End
```

En lo anterior x y e son vectores; i, j, y k son escalares.

Podemos garantizar que el conjunto de colores elegido no se afectará al utilizar la técnica de "dithering"; es decir, podremos obtener todos los colores de la imagen original con alguna combinación lineal del conjunto, si los colores de entrada (de la imagen original) caen en el conjunto convexo de los colores representativos.

4 .- IMPLANTACION DE UN ALGORITMO PARA LA CUANTIFICACION DEL COLOR DE IMAGENES.

4.1 .- Introducción.

Actualmente las imágenes digitales utilizan 24 bits por pixel, en equipos tales como las estaciones de trabajo. Esto es 16 millones de colores. El problema es que muchos de los monitores a color solo pueden mostrar 256 colores a la vez. De esta manera puede verse que es necesario capturar imágenes desplegadas con 24 bits y hacer que se vean muy bien en las pantallas de 8-bits.

De alguna manera habria que seleccionar los 256 colores que mejor representen a la Imagen Original, y habría que mapear los miles de colores dentro de los 256 colores. Esto es un problema que se ha vuelto común dentro del software para el manejo de imagenes.

El algoritmo implantado demuestra cómo se puede hacer una selección de color en monitores con una paleta limitada como son las tarjetas EGA y VGA.

4.2 .- Cuantificación del color de imágenes.

La cuantificación del color de imágenes se llevó a cabo en 4 fases :

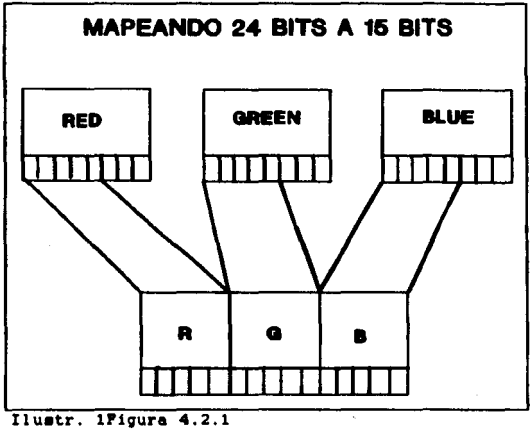
FASE 1: Muestrear la Imagen Original.

A partir de la imagen original, formada por un archivo del tamaño de la resolución de la imagen y un RGB para cada pixel, se crea un histograma de frecuencias de los colores, lo cual permitirá conocer el número de veces que ocurre cada color dentro de la imagen.

Con 24 bits por pixel, se pueden generar 16 millones de colores. Para almacenar la frecuencia de uno de ellos se necesita aproximadamente una palabra de memoria, así que para almacenar todos ellos se necesitaría un total de 32 MB de memoria. Para conservar memoria se utilizarán solamente 15 bits por pixel, esto es 5 bits para el componente de color rojo, 5 bits para el

componente de color verde, y 5 bits para el componente de color azul.

En este trabajo el histograma de frecuencias de colores será almacenado en una lista, la cual tendrá en cada entrada el número de color (formado por los 5 bits de cada componente de color (RGB), como se muestra en la siguiente figura); y un contador del mismo.



Ilustr. 1Figura 4.2.1

FASE 2: Seleccionar el conjunto de colores que represente de manera óptima el conjunto de colores de la imagen original.

Una vez creado el histograma de frecuencias de colores, se selecciona el conjunto óptimo de colores a partir de este.

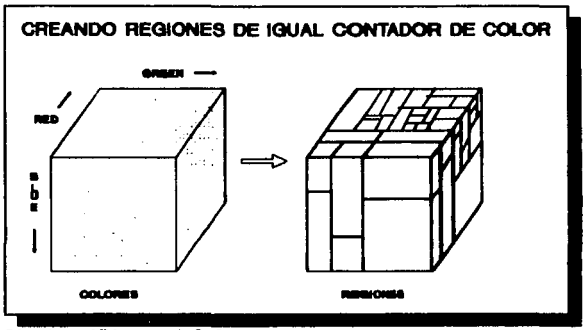
Todos los colores posibles, se pueden representar por medio de un cubo, si se piensa en ellos como una matriz tridimensional donde los 3 ejes son los componentes de color. Para seleccionar el conjunto de colores óptimo, se divide este cubo de colores en 256 regiones.

Si se tomaran regiones de igual volumen, no se consideraría la distribución actual de los colores en la imagen. En otras palabras, se le dá el mismo tratamiento a una imagen en la cual se muestra el mar que a la imagen de un ocaso. En cambio, si la división del cubo se realiza utilizando uno de los

algoritmos propuestos: 1) algoritmo popular y 2) algoritmo del corte medio, esta distribución sí es considerada.

En este trabajo se implantó el algoritmo del corte medio, porque su sensibilidad a la distribución de los colores en la imagen es mejor que la del algoritmo popular. Esto se debe a que el primero de ellos considera a todos los colores en la imagen, para seleccionar el conjunto representativo, mientras que el segundo de ellos solo tomo en cuenta los colores con frecuencias más altas en el histograma.

Tal como el algoritmo seleccionado lo propone el cubo se dividió en regiones con igual número de colores. Tomar regiones con igual número de colores, implica que las regiones deben ser más pequeñas y más numerosas donde las frecuencias del histograma sean más altas.



Ilustr. 1Figura 4.2.2

Para la implantación del algoritmo se utiliza un arreglo que contiene los colores que están en la lista, es decir los colores de la imagen.

La estructura que se utilizará para almacenar la información correspondiente a cada región es la siguiente:

first y last : Los cuales son índices dentro del arreglo que definen los colores que pertenecen a esa región.

histSum Indica el numero total de colores

que están en la región

MinRed,MaxRed

MinGreen, MaxGreen

MinBlue, MaxBlue: Definen los valores máximos y mínimos para cada componente.

widestcolor : Representa la componente del color con mayor longitud, es decir, la componente cuya diferencia entre su valor máximo y mínimo es mayor.

mediancolor : Este campo representa al componente cuya diferencia es inmediata inferior a la de widestcolor.

Red, Green, Blue : Se utilizan para guardar los componentes del color promedio asignado a la región.

Indice . Indice dentro de la tabla de colores que le corresponde a todos los colores de la región

El Algoritmo.

El conjunto de pasos que a continuación se presenta se aplicará a cada región, hasta que el número de regiones sea de 256 :

a) Se revisa el arreglo de colores en el rango delimitado por índices de inicio y fin (first y last), con la finalidad de encontrar los valores máximos y mínimos de cada componente. A partir de esta información se obtiene el widestcolor, y el mediancolor.

DifRed = RedMax - RedMin

DifGreen = GreenMax - GreenMin

DifBlue = BlueMax - BlueMIn

widestcolor = max(DifRed, DifGreen, DifBlue)

Al mismo tiempo se calcula el contador de color de la región (histSum), sumando las frecuencias de los colores que pertenecen a la región.

$$histSum = \sum_{i=first}^{last} frecuencia(color[i])$$

b) Se ordena el arreglo de colores tomando como llave primaria a widestcolor y a mediancolor como llave secundaria.

c) Encontrar el punto medio: Se recorre la lista de colores otra vez, desde first, sumando los contadores de color en una variable auxiliar. Cuando esta variable sea mayor o igual que la mitad del contador de la región (histSum), se habrá encontrado el punto medio. Este punto se debe de ajustar de tal forma que el color que se encuentra en el punto medio, y el color que está una posición más adelante en el arreglo de colores, tengan diferente valor en la componente de color correspondiente al widestcolor.

d) Una vez encontrado el punto medio se divide la región actual, en dos regiones, la primera de ellas tendrá un rango delimitado por first y punto medio, la segunda

iniciará una localidad después del punto medio dentro del arreglo y terminará en last. Para cada una de estas regiones calculamos el widestcolor, mediancolor, valores mínimos y máximos de cada componente, y el contador de color.

Para construir las 256 regiones se crea una lista de regiones que estará ordenada por el contador de color de la región (número de colores que están en la región) en forma descendiente; la región a la que se le aplicarán los pasos descritos anteriormente será la que está en la cabeza de la lista. Si en alguna ocasión la región que se toma tiene un solo punto, un color con un contador alto, se guarda esta región en una lista auxiliar, y se toma la siguiente región que está en la lista de regiones. Finalmente se copia la lista auxiliar en la lista de regiones.

Una vez que se han creado las 256 regiones se asigna el color correspondiente a cada una de ellas, para lo cual se suman los colores encerrados en la región, multiplicando cada uno de ellos por su frecuencia en el histograma, y dividiendo este resultado entre el contador de la región (histSum).

$$\text{color promedio} = \left(\sum_{i=first}^{last} \text{colorList}[i] * \text{frecuencia}(\text{colorList}[i])\right) / \text{histSum}$$

FASE 3: Mapear los colores de la imagen original al conjunto de colores representativo

Después de haber seleccionado el conjunto de colores representativo, se mapea la gama de colores de la imagen original a este conjunto.

En este caso se utiliza el algoritmo de **Búsqueda Ordenada Localmente**. En este método se plantea tener una lista en la cual cada celda tiene un color representativo y sus colores vecinos más cercanos; pero esto es precisamente lo que se tiene en la lista de regiones, pues con los valores máximos y mínimos se delimita cuáles son los colores más cercanos al color representativo de esa región.

Como se mencionó anteriormente la imagen original contiene para cada pixel un valor RGB; lo que se pretende obtener es un archivo que contenga para cada pixel el índice que le

corresponde dentro de la tabla de colores, así como los colores que se deben cargar en la tabla de colores (conjunto de colores representativo). Para lo cual se realiza lo siguiente:

A cada región se le asigna un índice, que es el que tiene el color representativo de esa región dentro de la tabla de colores (look up table).

Para cada color de la imagen se revisa la lista de regiones, encontrando de esta forma la región a la cual pertenece. Esta búsqueda se realiza a través de los valores máximos y mínimos de los componentes de color que tiene cada región. Una vez encontrada la región a la que pertenece ese color, se le asigna el índice que tiene.

FASE 4: Cuantificando y Desplegando la Imagen

Finalmente se despliega la imagen y al cargar en la tabla de colores (look up table) el conjunto de colores óptimo, la imagen se vuelve a desplegar con estos colores

5 .- RESULTADOS Y CONCLUSIONES.

En el presente trabajo se desarrolló:

1) Editor de Color.

2) Implementación de un algoritmo para la cuantificación del Color de Imágenes.

Con lo cual se consiguió lo siguiente:

El Editor de Color permite que el usuario seleccione de una manera intuitiva los colores, por medio del modelo HSV. A la vez el usuario puede conocer la especificación de ese color en el modelo inherente al dispositivo (RGB). Por lo tanto el Editor de Color es una herramienta útil, pues ayuda al usuario en la tarea de selección de color al momento de generar imágenes.

Con la Implantación de un algoritmo para la Cuantificación del Color de Imágenes se logra encontrar, con base a la distribución actual de los colores en la imagen original, el conjunto que mejor representa la gama de colores de esta Imagen.

Con este algoritmo también se consigue mapear los colores de la Imagen Original al conjunto seleccionado.

Es decir, con este método se realiza un recorte sobre el número de colores de la Imagen Original, de tal forma que la imagen se acopla a las limitaciones que presentan los dispositivos de despliegue utilizados, permitiendo que la imagen sea desplegada en ellos.

Por lo tanto, el algoritmo implantado en este trabajo cumple con el objetivo planteado al inicio del mismo:

Presentar e implementar un algoritmo que encuentre el conjunto de colores óptimo para desplegar la imagen.

La calidad en la que se desplegaron las imágenes a las cuales se les aplicó dicho algoritmo fue alta, percibiéndose los siguientes problemas :

- Se notan algunos cambios bruscos de color, es decir, se ven franjas de color en lugar de una degradación continua del mismo.
- Se presenta el efecto de escalera, esto es, las líneas no tienen una trayectoria recta, sino una trayectoria con picos como los de una escalera.

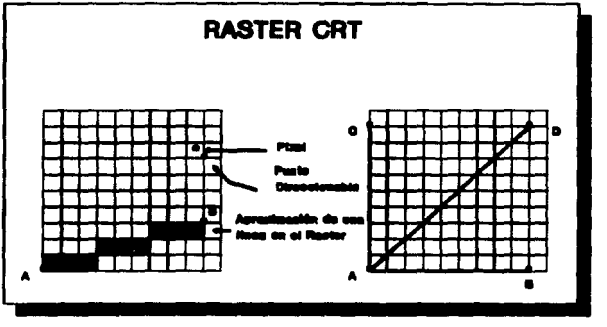
Para dar solución al primero de ellos, se debe desarrollar la técnica de **dithering**. Mientras que para dar solución a el segundo, se propone la técnica de **anti-aliasing**.

De lo anterior se puede ver, que este trabajo no termina aquí, sino que da pauta para que se inicien nuevas investigaciones que resuelvan los problemas antes mencionados.

6 .- APENDICE.

DISPOSITIVO RASTER

Un dispositivo raster CRT puede ser considerado como una matriz de celdas, a cada una de las cuales se les asigna intensidades; esto permite generar líneas de manera aproximada (discreta). No es posible, excepto en caso especiales, dibujar directamente una línea recta de un pixel a otro pixel. La línea solamente puede ser aproximada por una serie de puntos (pixels) que siguen una trayectoria.



FRAME BUFFER.

Un frame buffer es una parte de la memoria de la computadora, el cual está compuesto de bit-planes, donde un bit-plane es la mínima memoria para almacenar una imagen con dos niveles de intensidad.

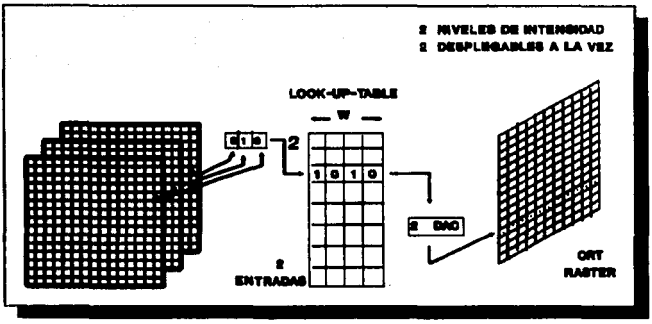
Como el frame buffer es un dispositivo digital, y el raster CRT es un dispositivo análogo que requiere voltaje eléctrico, se debe hacer una conversión de la representación digital a una señal análoga cuando se lee la información del frame buffer y se despliega en el dispositivo CRT. Esta conversión es realizada por medio de un convertidor digital-a-análogo. (digital-to-analog converter, DAC).

Si se agregan bit-planes al frame buffer se pueden obtener más tonalidades de color.

La intensidad de cada pixel en el CRT es controlada por la correspondiente localización del pixel en cada uno de los N bit planes. El valor binario (0 o 1) de cada uno de los bit planes es cargado dentro de las posiciones correspondientes en un registro. El número binario que resulta en este registro es interpretado como el nivel de intensidad entre 0 y $2^N - 1$. Un total de 2^N intensidades pueden ser producidas.

El Look-Up-Table debe contener 2^N entradas. Cada entrada en el Look-Up-Table puede contener W bits. W puede ser mayor que N. Cuando esto ocurre, se tiene 2^W intensidades disponibles, pero solamente 2^N diferentes intensidades están disponibles al mismo tiempo.

Como son tres los colores primarios, un frame buffer puede ser implantado con 3 bit-planes, uno para cada color primario. Estos 3 colores son combinados en el CRT para producir los 8 colores siguientes:



	Red	Green	Blue
Black	0	0	0
Red	1	0	0
Green	0	1	0
Blue	0	0	1
Yellow	1	1	0
Cyan	0	1	1
Magenta	1	0	1
White	1	1	1

Se pueden utilizar bit-planes adicionales para cada uno de los colores. Por ejemplo si se usan 8 bit-planes por color, es decir un frame buffer con 24 bit-planes. Cada grupo de bit-planes genera 256 intensidades (2^8) intensidades de rojo, verde ó azul.

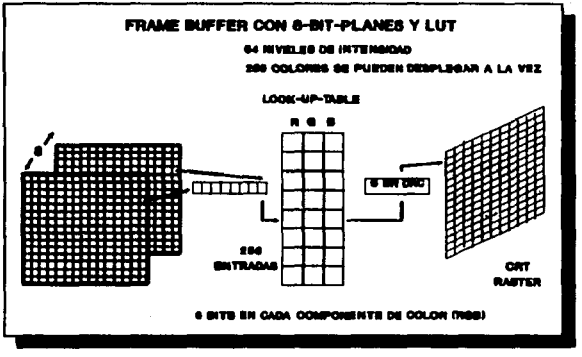
Estas intensidades pueden ser combinadas, obteniendo 16.777.216 [$(2^8)^3 = 2^{24}$] posibles colores.

Para N bit-planes por color con Look-Up-Tables con ancho W, $(2^3)^W$ colores pueden ser desplegados a la vez mientras que se tienen disponibles $(2^3)^N$ colores.

Los dispositivos que se utilizaron para el trabajo son los siguientes:

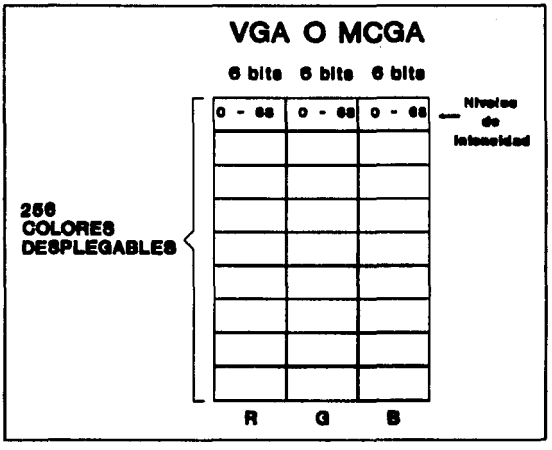
MONITOR VGA y MCGA.

Estos Frame Buffers cuentan con 8 bit-planes, con 6 bits por cada componente de color (RGB). Lo cual implica que tenemos disponibles niveles de intensidad de 0 a $2^6-1 = 63$.



El número de colores disponibles en estos Frame Buffer es $(2^6)^3 = 262,144$.

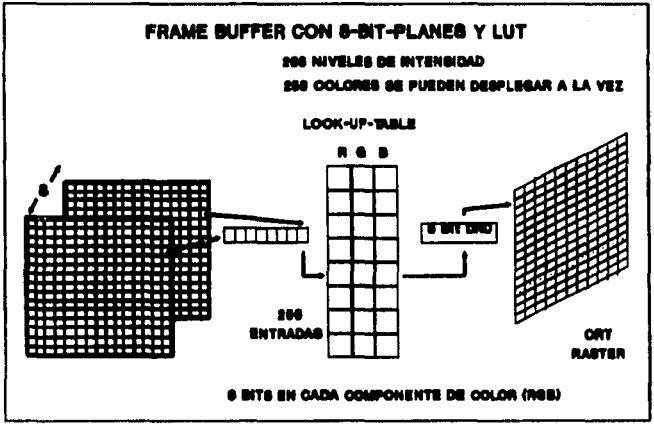
El número de colores que el dispositivo permite mostrar a la vez es de 256.



Dispositivo en el cual se generaron algunas imágenes:

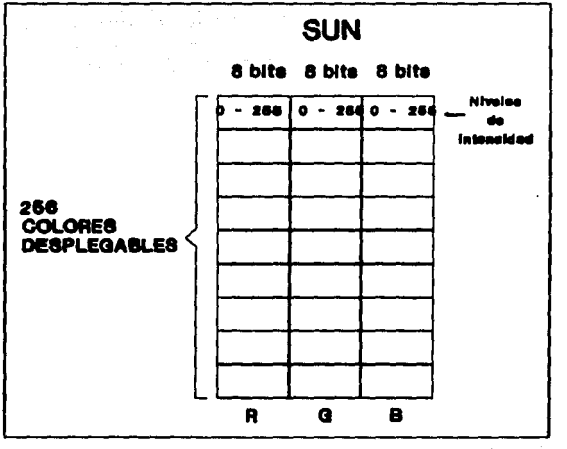
SUN:

Este Frame Buffer tiene 8 bit-plane , con 8 bits por componente de color RGB. Por lo tanto se tienen niveles de intensidad de 0 a $2^8 - 1 = 255$.



El número de colores disponibles en este Frame Buffer es de $(2^{24}) = 16.777.216$.

El número de colores que se pueden desplegar a la vez es de 256.



7 .- BIBLIOGRAFIA.

A Consumer's and Developer's Guide to Image Synthesis.
SIGFRAPH'88 - Atlanta, Georgia.
Agosto 1988.
Curso # 10.

Brown Judith, Cunningham Steve
Programming The User Interface: Principles and Examples.
John Wiley & Sons.

Foley, Van Dam, Feiner, Hughes.
Computer Graphics: Principles and practice.
Segunda Edición, Addison Wesley.

Heckbert, Paul.
Color Image Quantization for Frame Buffer Display.
Computer Graphics 16(3).

Hall, Roll.
Ilumination and Color Generated Imagery.

Pomerantz, Dave.
A Few Good Colors..
Computer Language August 1990.

Rogers, David.
Procedural Elements for Computer Graphics.
Mc. Graw-Hill Book Company.

Watt. Alan.
Fundamentals of Three-Dimensional Computer Graphics.
Addison Wesley.