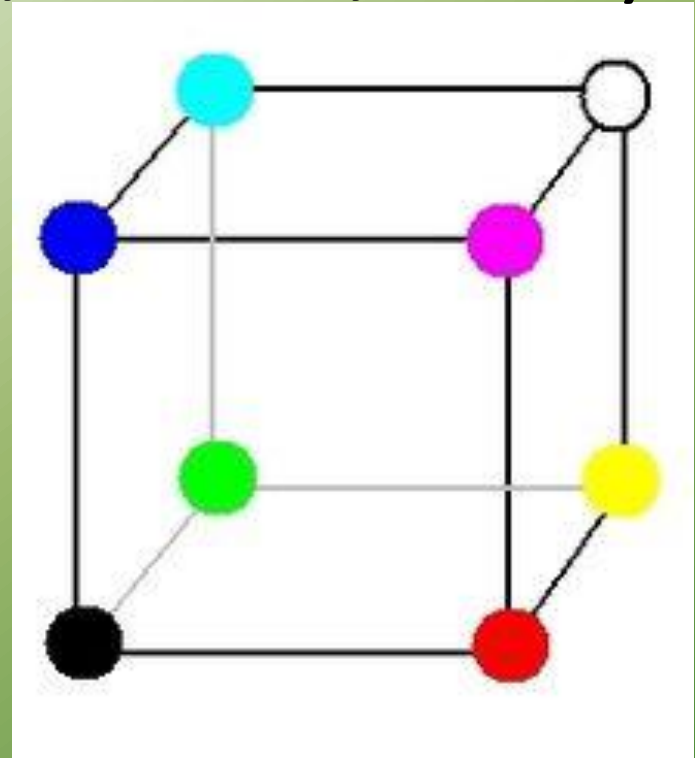
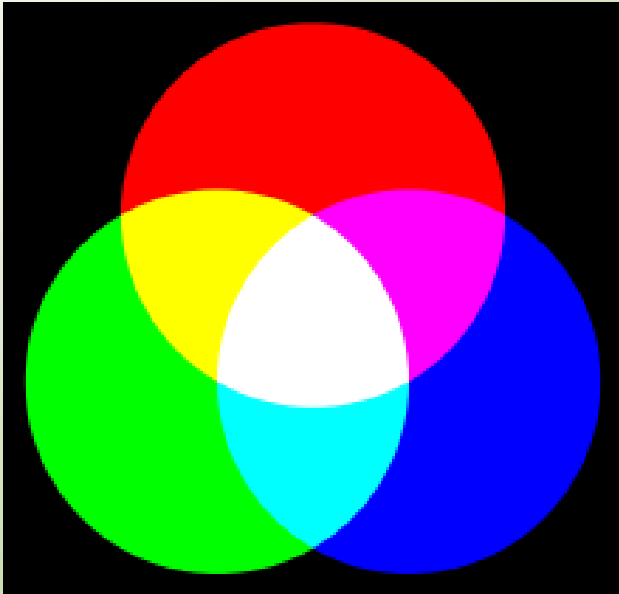


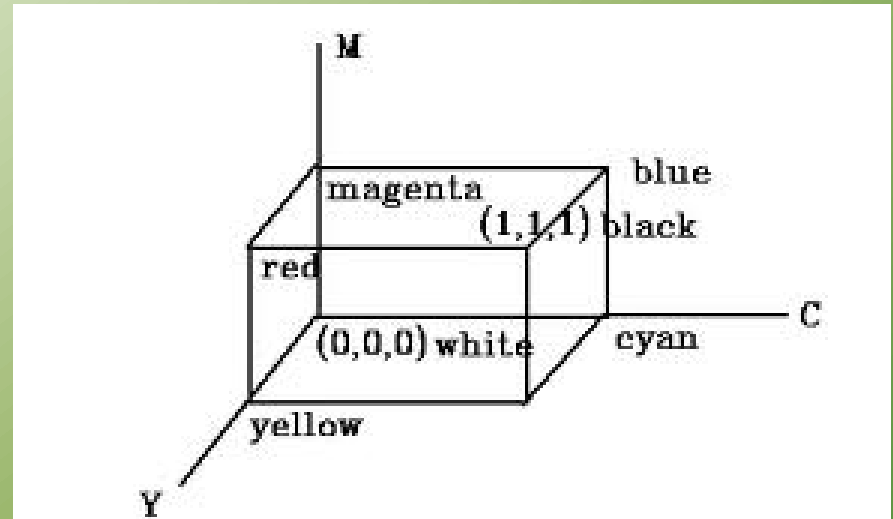
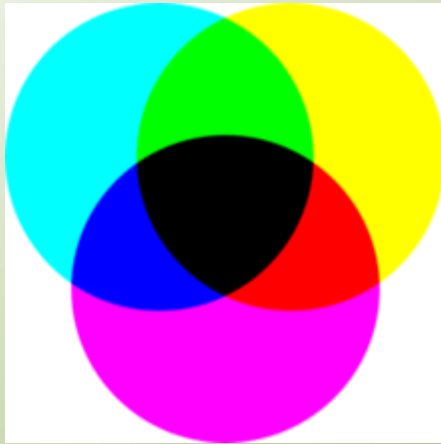
# Manejo de color

# Modelo RGB (Red, Green, Blue)



RGB es conocido como un espacio de color aditivo (colores primarios) porque cuando la luz de dos diferentes frecuencias viaja junta, desde el punto de vista del observador, estos colores son sumados para crear nuevos tipos de colores. Los colores rojo, verde y azul fueron escogidos porque cada uno corresponde aproximadamente con uno de los tres tipos de conos sensitivos al color en el ojo humano (65% sensibles al rojo, 33% sensibles al verde y 2% sensibles al azul). Con la combinación apropiada de rojo, verde y azul se pueden reproducir muchos de los colores que pueden percibir los humanos.

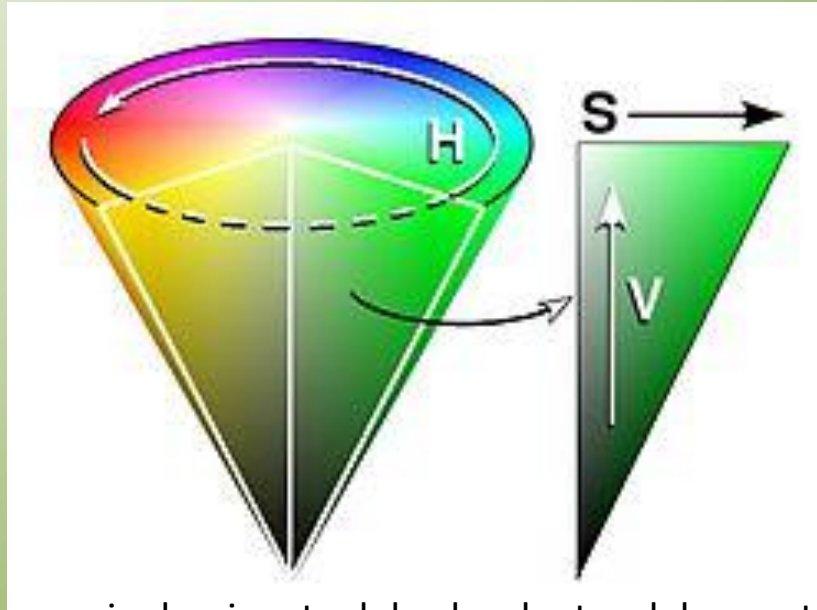
# Modelo CMY (Cian, Magenta, Yellow)



CMY trabaja mediante la absorción de la luz (colores secundarios).

Los colores que se ven son la parte de luz que no es absorbida. En CMY, magenta más amarillo producen rojo, magenta más cian producen azul, cian más amarillo generan verde y la combinación de cian, magenta y amarillo forman negro. El negro generado por la mezcla de colores primarios sustractivos no es tan denso como el color negro puro (uno que absorbe todo el espectro visible). Es por esto que al CMY original se ha añadido un canal clave (key), que normalmente es el canal negro (black), para formar el espacio CMYK o CMYB.

**Modelo HSV** (Hue, Saturation, Value – Matiz, Saturación, Valor) también llamado **HSB** (Hue, Saturation, Brightness – Matiz, Saturación, Brillo)



**Matiz (Hue):** Se refiere a la frecuencia dominante del color dentro del espectro visible. Es la percepción de un tipo de color, normalmente la que uno distingue en un arcoíris, es decir, es la sensación humana de acuerdo a la cual un área parece similar a otra o cuando existe un tipo de longitud de onda dominante. Incrementa su valor mientras nos movemos de forma antihoraria en el cono, con el rojo en el ángulo 0.

**Saturación (Saturation):** Se refiere a la cantidad del color o a la "pureza" de éste. Va de un color "claro" a un color más vivo (azul cielo – azul oscuro). También se puede considerar como la mezcla de un color con blanco o gris.

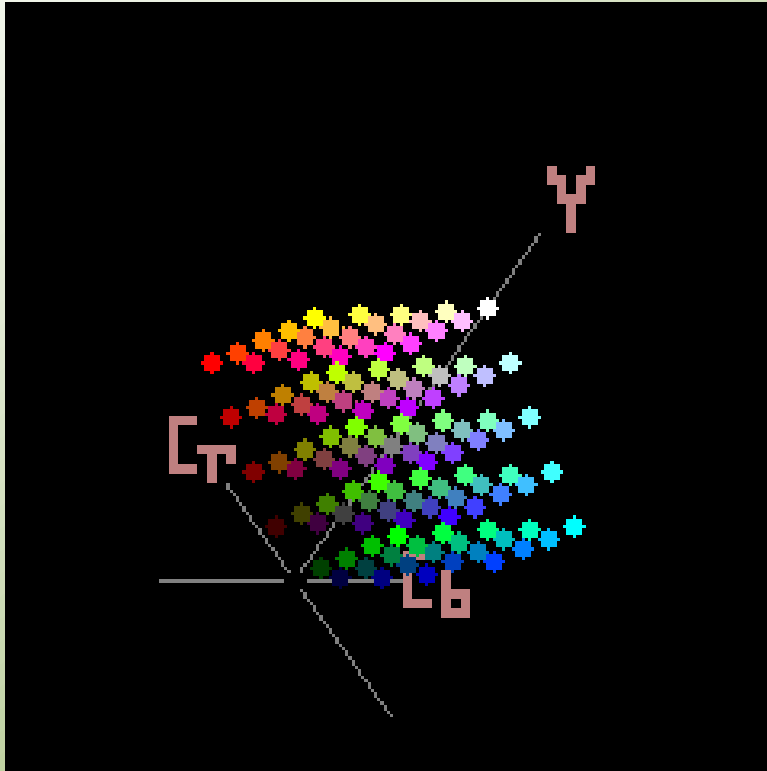
**Valor (Value):** Es la intensidad de luz de un color. Dicho de otra manera, es la cantidad de blanco o de negro que posee un color.

# Modelo HSL (del inglés Hue, Saturation, Lightness – Matiz, Saturación, Luminosidad)



El modelo HSL (del inglés Hue, Saturation, Lightness – Matiz, Saturación, Luminosidad), que es similar a HSV o HSI (del inglés Hue, Saturation, Intensity – Matiz, Saturación, Intensidad), define un modelo de color en términos de sus componentes constituyentes. El modelo HSL se representa gráficamente como un cono doble o un doble hexágono. Los dos vértices en el modelo HSL se corresponden con el blanco y el negro, el ángulo se corresponde con el matiz, la distancia al eje con la saturación y la distancia al eje blanco-negro se corresponde a la luminancia. Como los modelos HSI y el HSV, es una deformación no lineal del espacio de color RGB.

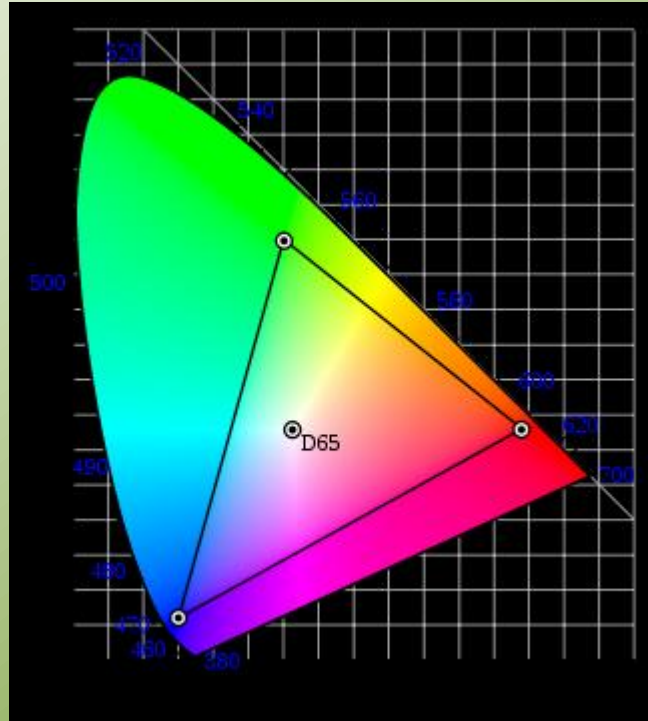
# YCbCr, Y'CbCr, o Y Pb/Cb, Pr/Cr



Y' es la componente luma, CB y CR son la componente cromática de diferencia azul y la diferencia roja. Y' distingue la luminisencia, significa que la intensidad de inluminacion no lineal es controlada por Y a partir de los primarios del RGB.



# Modelo sRGB



Es un espacio de color RGB creado en cooperación por Hewlett-Packard y Microsoft Corporation. Fue aprobado por el W3C, Exif, Intel, Pantone, Corel y otro muchos actores de la industria. Es también bien aceptado por el Software libre como el GIMP, y es utilizado en formatos gráficos propietarios y libres como el PNG.

# Modelo de color RYB



Rueda de color de Goethe, tomada de su libro Teoría de los colores de 1810.

El modelo de color RYB (Red, Yellow, Blue = rojo, amarillo, azul) es un modelo de síntesis sustractiva de color al igual que el modelo CMYK. Se basó en los estudios de Goethe en su libro Teoría de los colores de 1810, y adoptado desde aquellos tiempos por las escuelas de pintura y artes gráficas. Hoy en día, gracias a la evolución de la fotografía en color y la mezcla aditiva de color es fácil demostrar que dicho modelo es bastante impreciso (su corrección propiamente dicha es el modelo CMYK), aunque sigue estando presente en la teoría impartida en la educación artística, en especial en las bellas artes y el diseño gráfico.



# Transformaciones RGB - CMY

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

# Transformación RGB a HSV

$$H = \begin{cases} \text{no definido,} & \text{si } MAX = MIN \\ 60^\circ \times \frac{G-B}{MAX-MIN} + 0^\circ, & \text{si } MAX = R \\ & \text{y } G \geq B \\ 60^\circ \times \frac{G-B}{MAX-MIN} + 360^\circ, & \text{si } MAX = R \\ & \text{y } G < B \\ 60^\circ \times \frac{B-R}{MAX-MIN} + 120^\circ, & \text{si } MAX = G \\ 60^\circ \times \frac{R-G}{MAX-MIN} + 240^\circ, & \text{si } MAX = B \end{cases}$$

$$S = \begin{cases} 0, & \text{si } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{en otro caso} \end{cases}$$

$$V = MAX$$

Donde: MAX el valor máximo de los componentes (R, G, B), y MIN el valor mínimo de esos mismos valores

# Transformación HSV a RGB

$$H_i = \left[ \frac{H}{60} \right] \bmod 6, \quad f = \frac{H}{60} - H_i, \quad p = V(1-S), \quad q = V(1-fS), \quad t = V(1-(1-f)S)$$

$$\text{si } H_i = \left\{ \begin{array}{ll} 0, & \begin{array}{l} R = V \\ G = t \\ B = p \end{array} \\ 1, & \begin{array}{l} R = q \\ G = V \\ B = p \end{array} \\ 2, & \begin{array}{l} R = p \\ G = V \\ B = t \end{array} \\ 3, & \begin{array}{l} R = p \\ G = q \\ B = V \end{array} \\ 4, & \begin{array}{l} R = t \\ G = p \\ B = V \end{array} \\ 5, & \begin{array}{l} R = V \\ G = p \\ B = q \end{array} \end{array} \right.$$

# Conversión desde RGB a HSL

$$H = \begin{cases} 0, & \text{if } MAX = MIN \\ (60 \times \frac{G-B}{MAX-MIN} + 360) \bmod 360, & \text{if } MAX = R \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{if } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{if } MAX = B \end{cases}$$

$$L = \frac{1}{2}(MAX + MIN)$$

$$S = \begin{cases} 0, & \text{if } MAX = MIN \\ \frac{MAX-MIN}{MAX+MIN} = \frac{MAX-MIN}{2L}, & \text{if } L \leq \frac{1}{2} \\ \frac{MAX-MIN}{2-(MAX+MIN)} = \frac{MAX-MIN}{2-2L}, & \text{if } L > \frac{1}{2} \end{cases}$$

Los valores (R, G, B) deben ser expresados como números del 0 al 1. MAX equivale al máximo de los valores (R, G, B), y MIN equivale al mínimo de esos valores. La fórmula puede ser escrita como

# Detectando el color de piel (ejemplo)

```
for k=1:1
    nombre=['myDataBase\' num2str(k) '.jpg'];
    I=imread(nombre);
    figure(1)
    imshow(I)

    [x,y]=getpts.figure(1));
    I2 = imcrop(I,[x(1) y(1) (x(2)-x(1)) (y(2)-y(1))]);
    figure
    imshow(I2)
    pause
    [hue,s,v]=rgb2hsv(I2);
    figure
    imhist(hue)
    figure
    imhist(s)
    figure
    imhist(v)
    mean(mean(hue))

    I=double(I);
    [hue,s,v]=rgb2hsv(I);
    [w h]=size(I(:, :, 1));

    for i=1:w
        for j=1:h
            if (0<=hue(i,j) && hue(i,j)<=0.1 || 0.96<=hue(i,j) && hue(i,j)<=1) && 0.05<=s(i,j) && s(i,j)<=0.7 && 50<=v(i,j) && v(i,j)<=250
                segment(i,j)=1;
            else
                segment(i,j)=0;
            end
        end
    end
    % imshow(segment);
    im(:, :, 1)=I(:, :, 1).*segment;
    im(:, :, 2)=I(:, :, 2).*segment;
    im(:, :, 3)=I(:, :, 3).*segment;
    figure(2),imshow(uint8(im));
    pause
    clear segment
    clear im
end
```

# Detectando el color de piel (ejemplo)

```
I=imread('alemania.jpg');
I=double(I);
[hue,s,v]=rgb2hsv(I);
cb = 0.148 * I(:, :, 1) - 0.291 * I(:, :, 2) + 0.439 * I(:, :, 3) + 128;
cr = 0.439 * I(:, :, 1) - 0.368 * I(:, :, 2) - 0.071 * I(:, :, 3) + 128;
[w h]=size(I(:, :, 1));
for i=1:w
    for j=1:h
        if 140<=cr(i,j) && cr(i,j)<=165 && 140<=cb(i,j) && cb(i,j)<=195 && 0.01<=hue(i,j) &&
hue(i,j)<=0.1
            segment(i,j)=1;
        else
            segment(i,j)=0;
        end
    end
end
end
% imshow(segment);
im(:, :, 1)=I(:, :, 1).*segment;
im(:, :, 2)=I(:, :, 2).*segment;
im(:, :, 3)=I(:, :, 3).*segment;
figure,imshow(uint8(im));
```