



Recuperación de Información Multimedia

Descriptores Globales Color

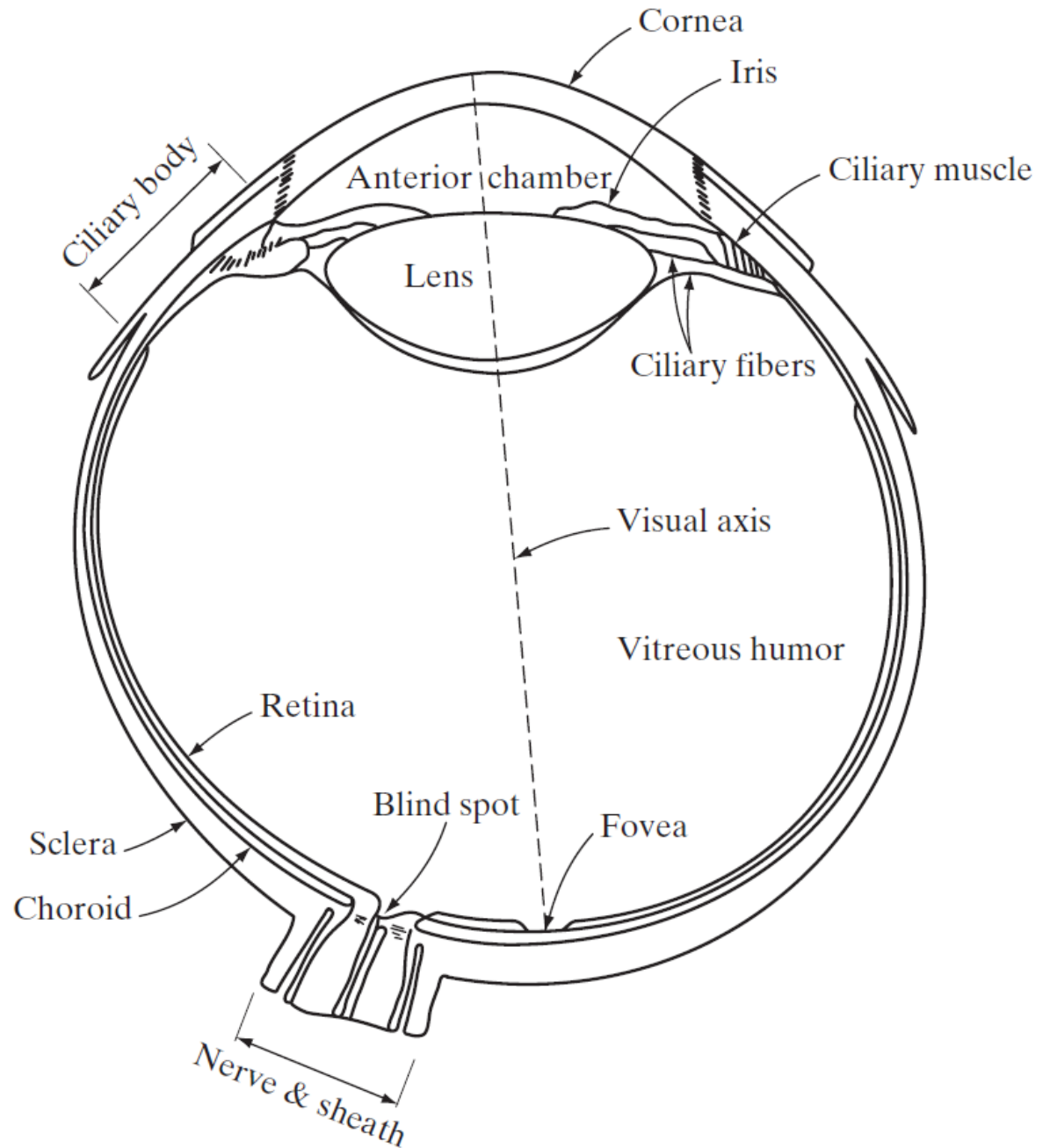
CC5213 – Recuperación de Información Multimedia

Departamento de Ciencias de la Computación

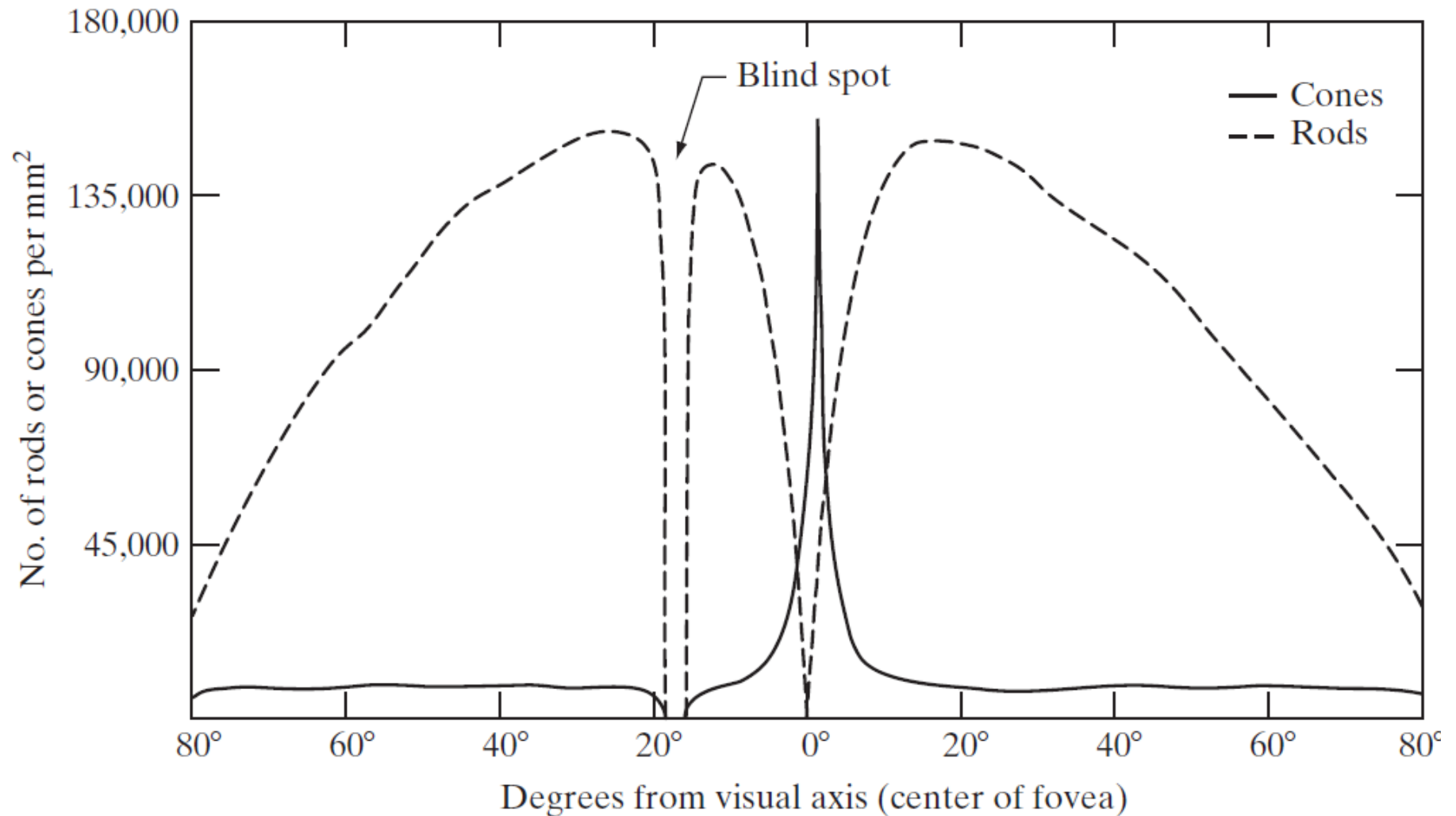
Universidad de Chile

Juan Manuel Barrios – <https://juan.cl/mir/> – 2019

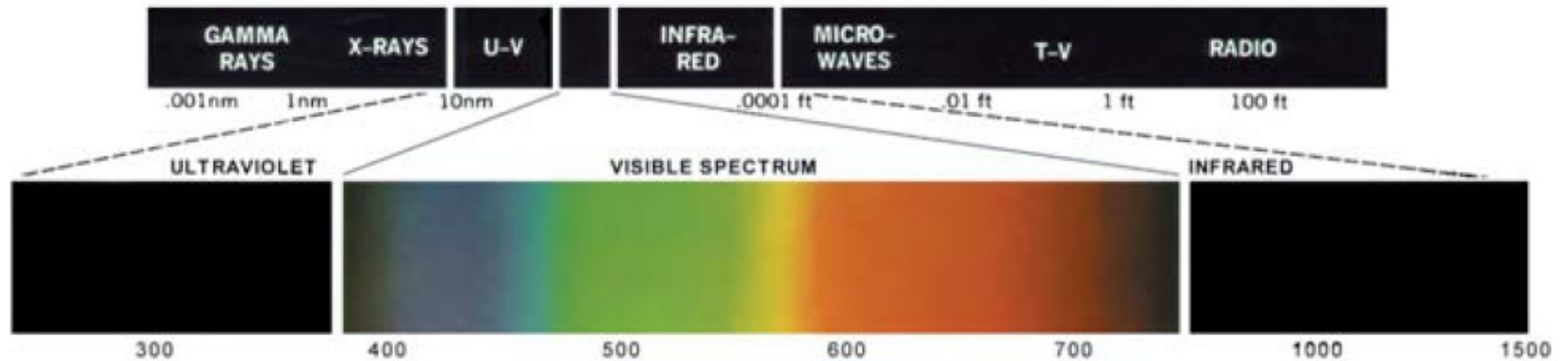
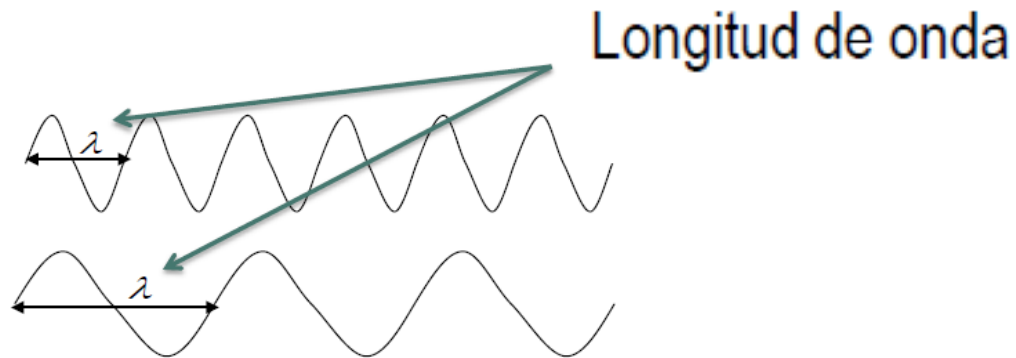
Sistema Visual



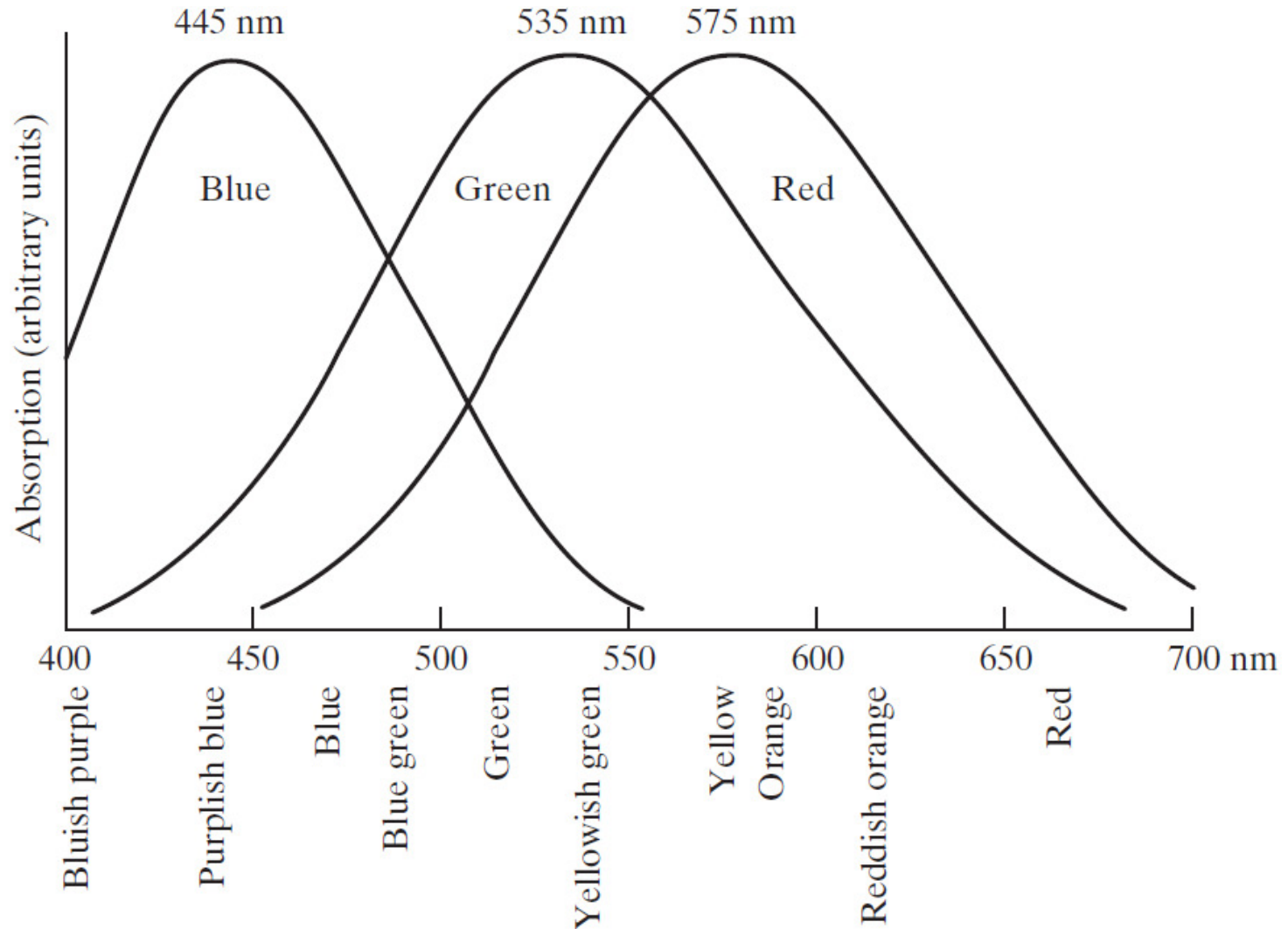
Bastones y Conos



Colores

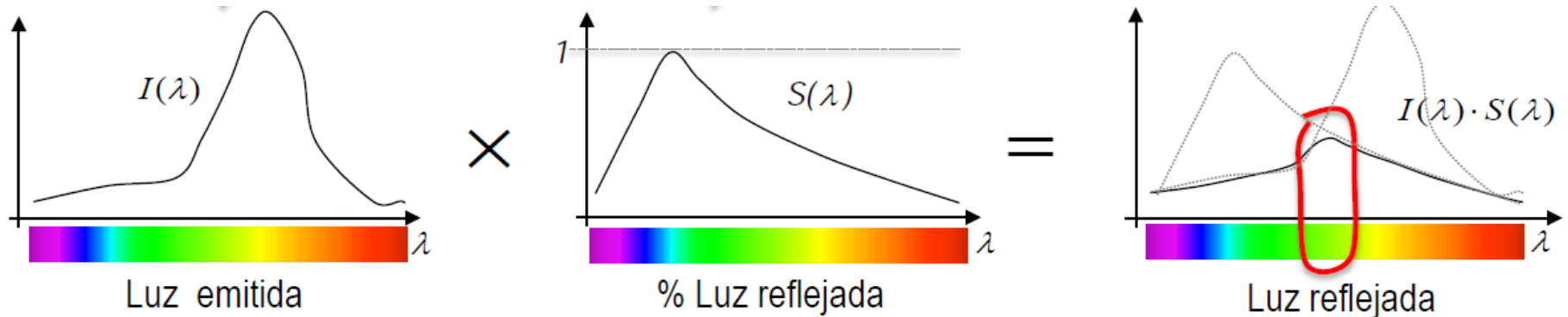
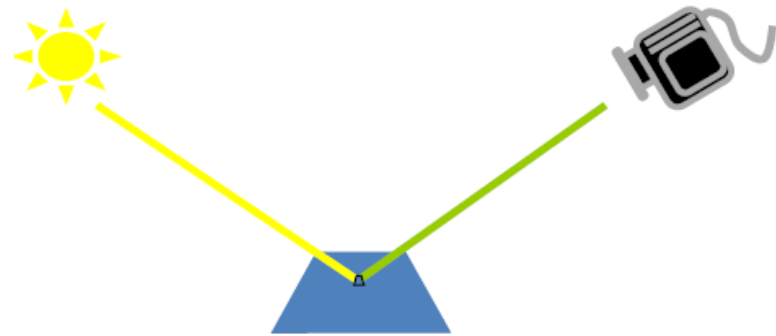


Tres tipos de conos



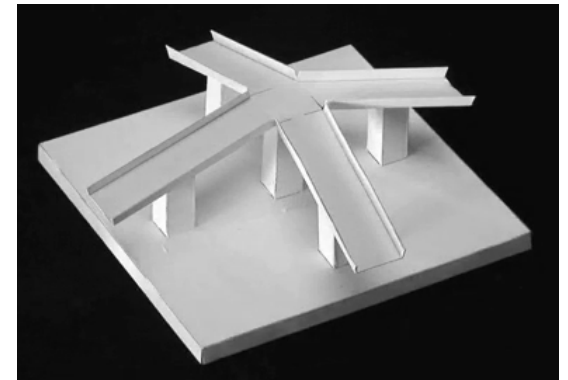
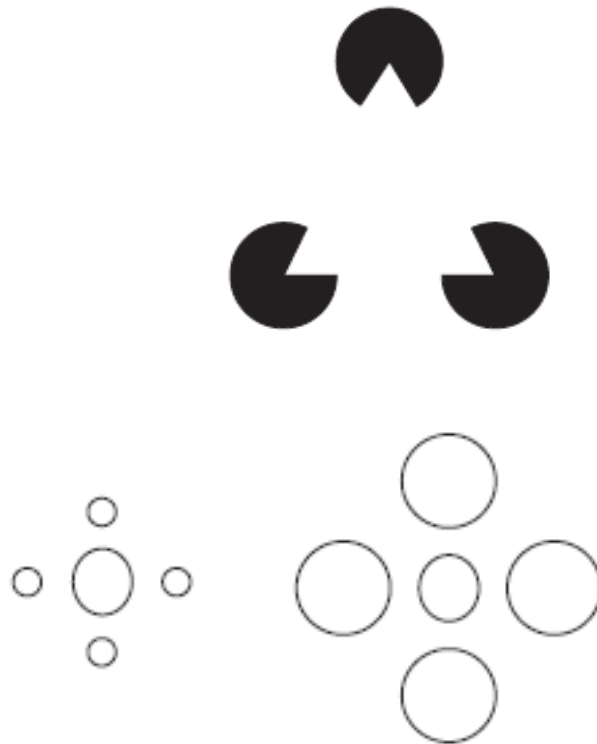
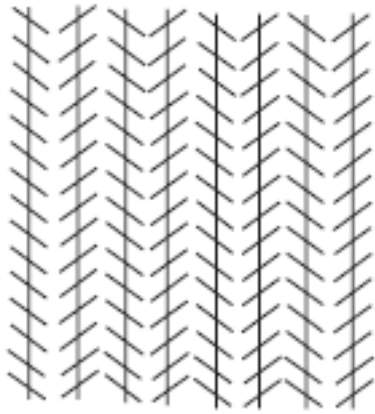
¿De qué color es un objeto?

- Color de la luz
- Material de la superficie
- Sensibilidad de la cámara



Sistema visual y percepción

- *“Las personas vemos con el cerebro”*



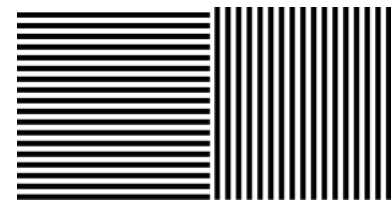
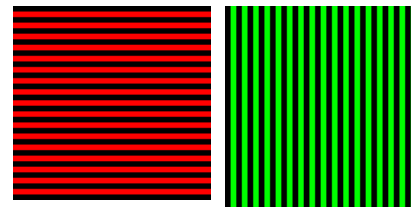
Sistema visual y percepción

- *“Las personas vemos con el cerebro”
(incluyendo los colores)*



Benham's disk

<https://youtu.be/hf3KTsRRPLs>

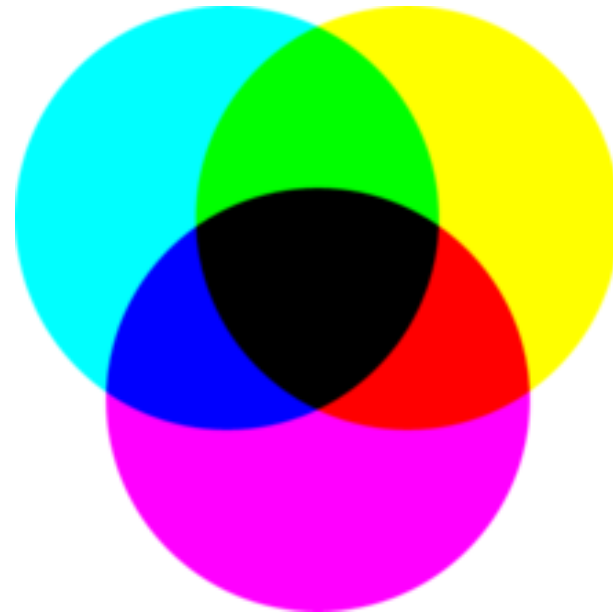
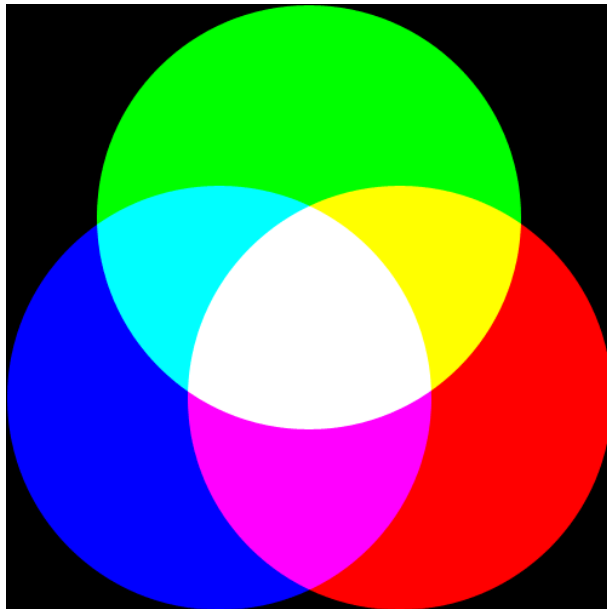


Efecto McCollough

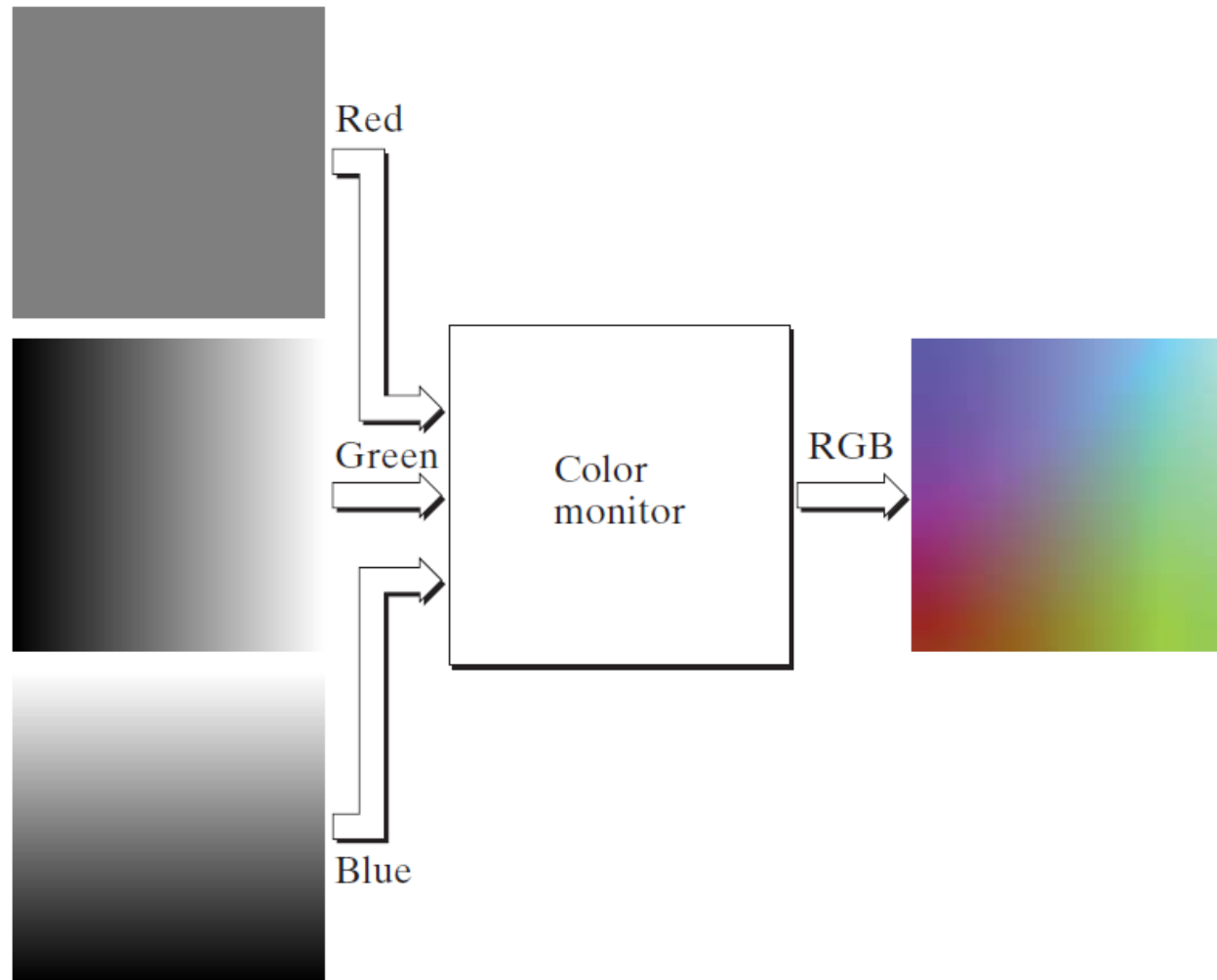


Colores primarios

- Modelo aditivo y sustractivo

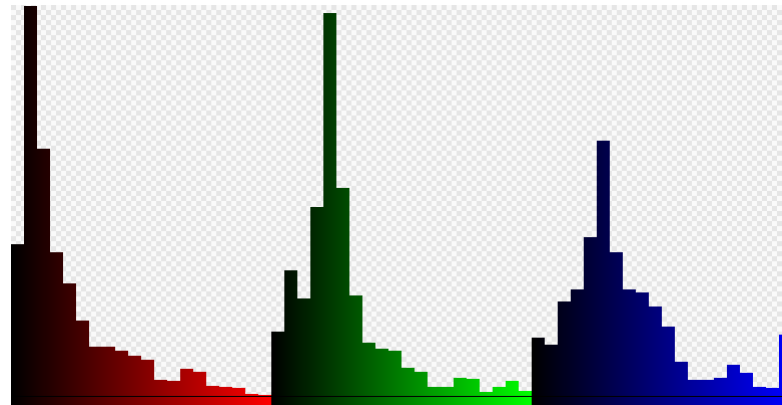


RGB como tres canales grises

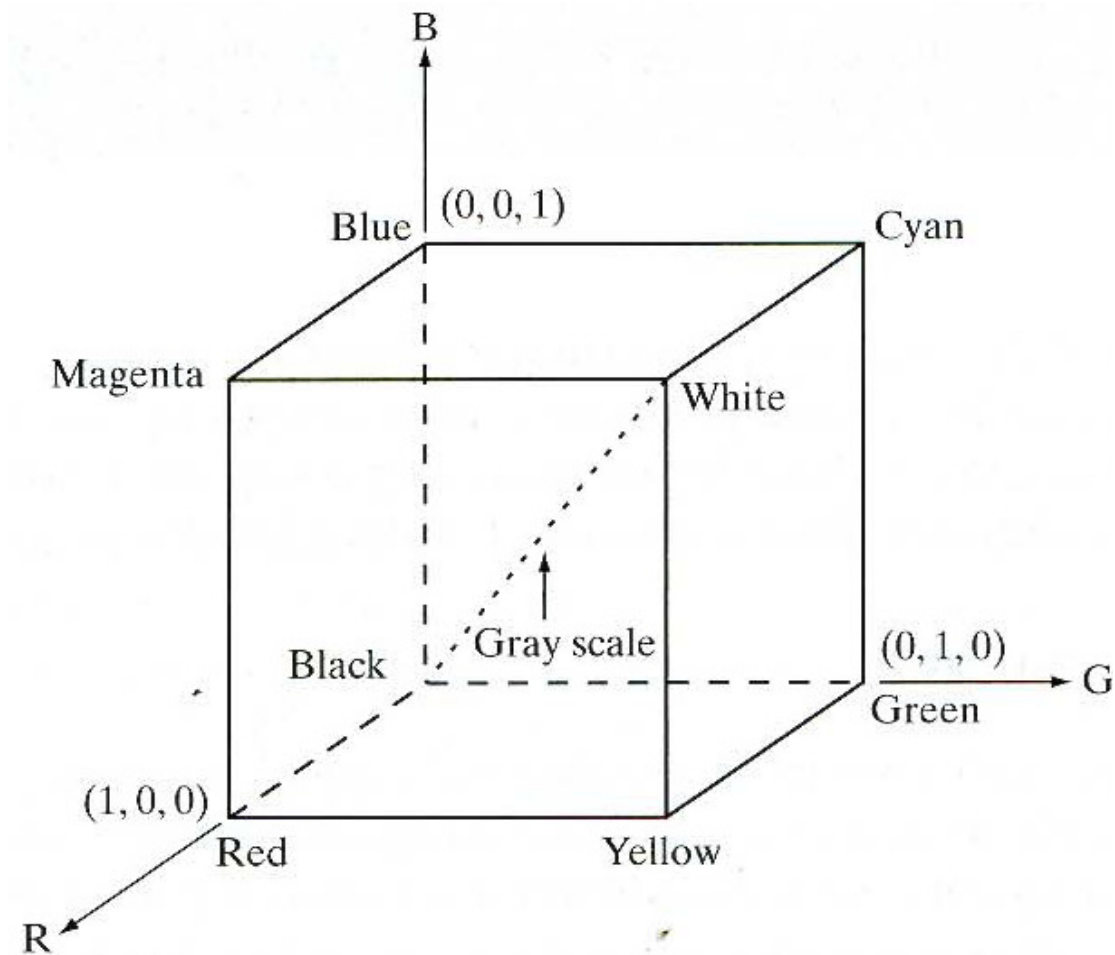


Histograma por Canal

- Calcular histograma independiente para cada canal R, G y B y concatenarlos
 - Útil para buscar duplicados
 - No es adecuado para búsqueda por similitud (no representa colores)



Cubo RGB





Normalizar luz blanca

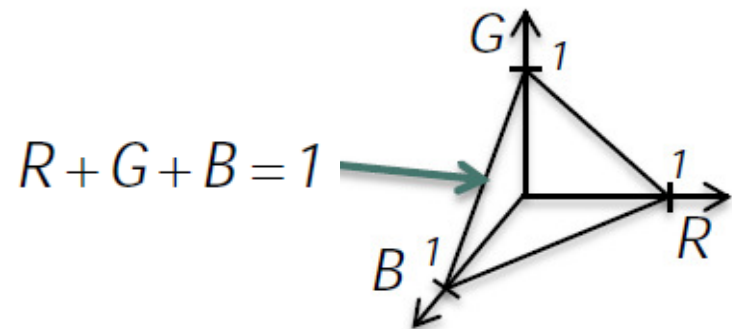
- White-patch. Asume que los valores máximos de color en los tres canales de la imagen es el color del blanco bajo la luz de la escena
 - Blanco = $(R_{\max}, G_{\max}, B_{\max})$
 - Escalar cada canal en forma independiente al rango 0-255
 - Para cada pixel:
 - $(r, g, b) \rightarrow (r * 255/R_{\max}, g * 255/G_{\max}, b * 255/B_{\max})$

Coordenadas Cromáticas

- Normalizar cada pixel por la suma de canales:

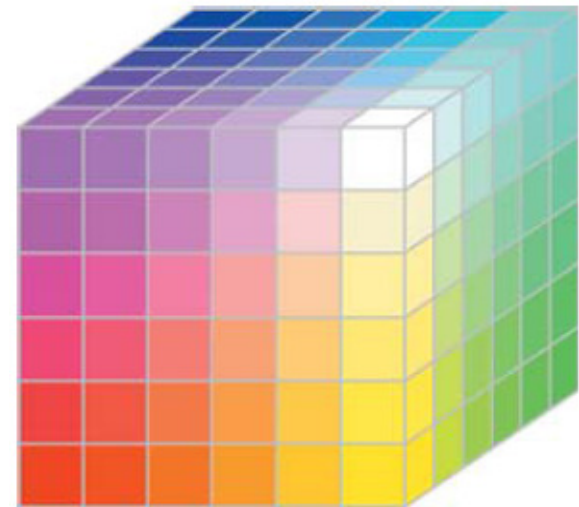
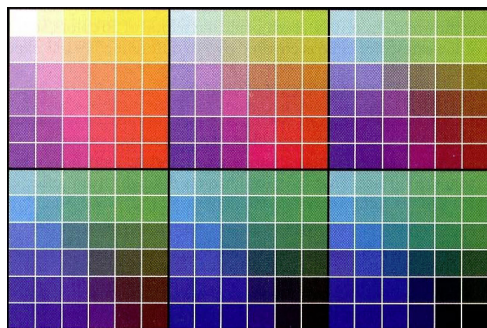
$$(R \ G \ B) \rightarrow \left(\frac{R}{R+G+B} \quad \frac{G}{R+G+B} \quad \frac{B}{R+G+B} \right)$$

- Notar que ante un cambio de intensidad (multiplicación por un factor) el valor no cambia
- Reducción del color a 2 dimensiones:



Histograma de Colores

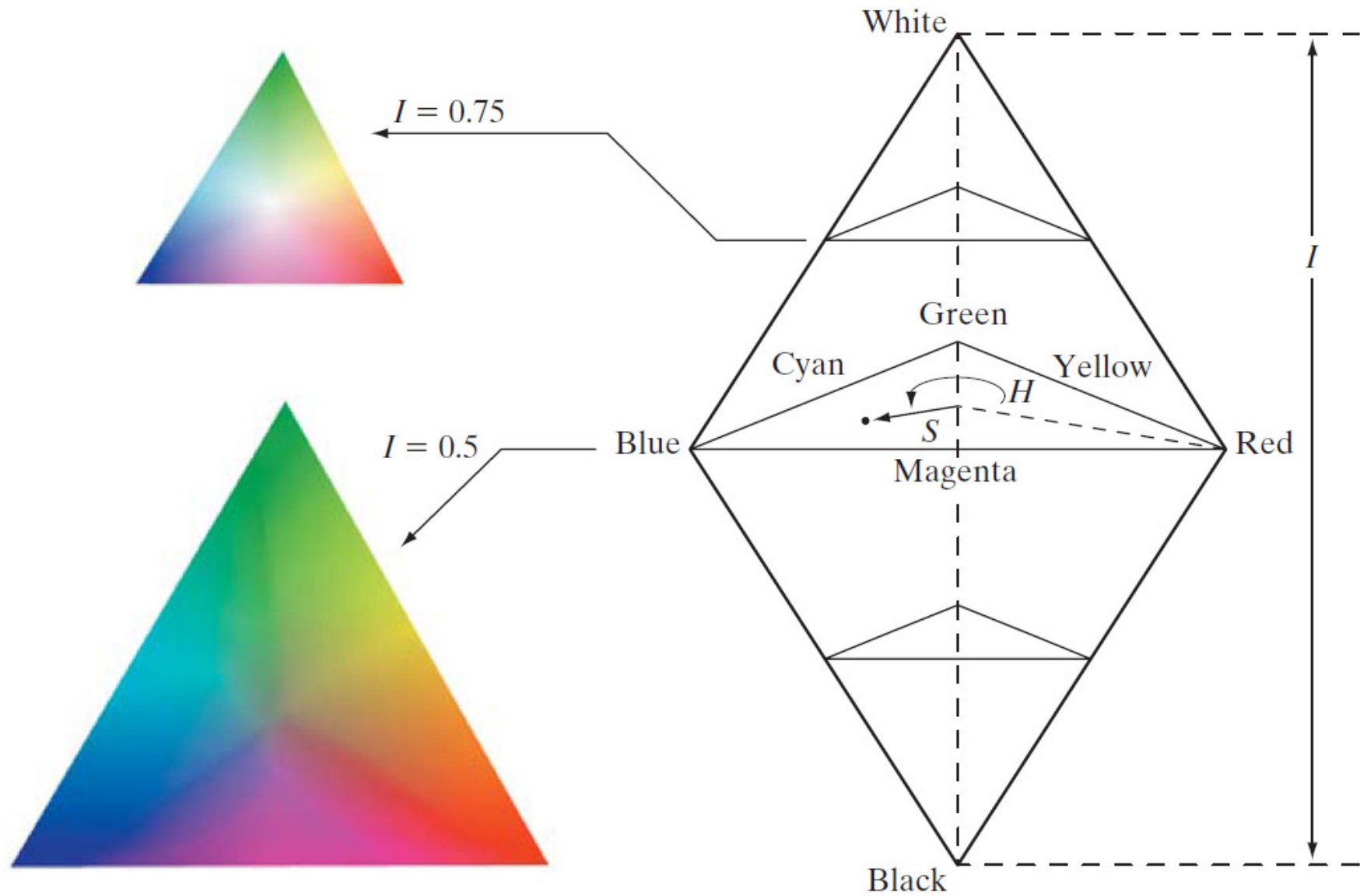
- Dividir cada dimensión R, G y B en tramos de un mismo tamaño
 - Por ejemplo, dividiendo cada canal en 6 tramos: [0,42] [43,84] [85,127] [128,170] [171,212] [213,255]
 - Total bins: $6^3=216$

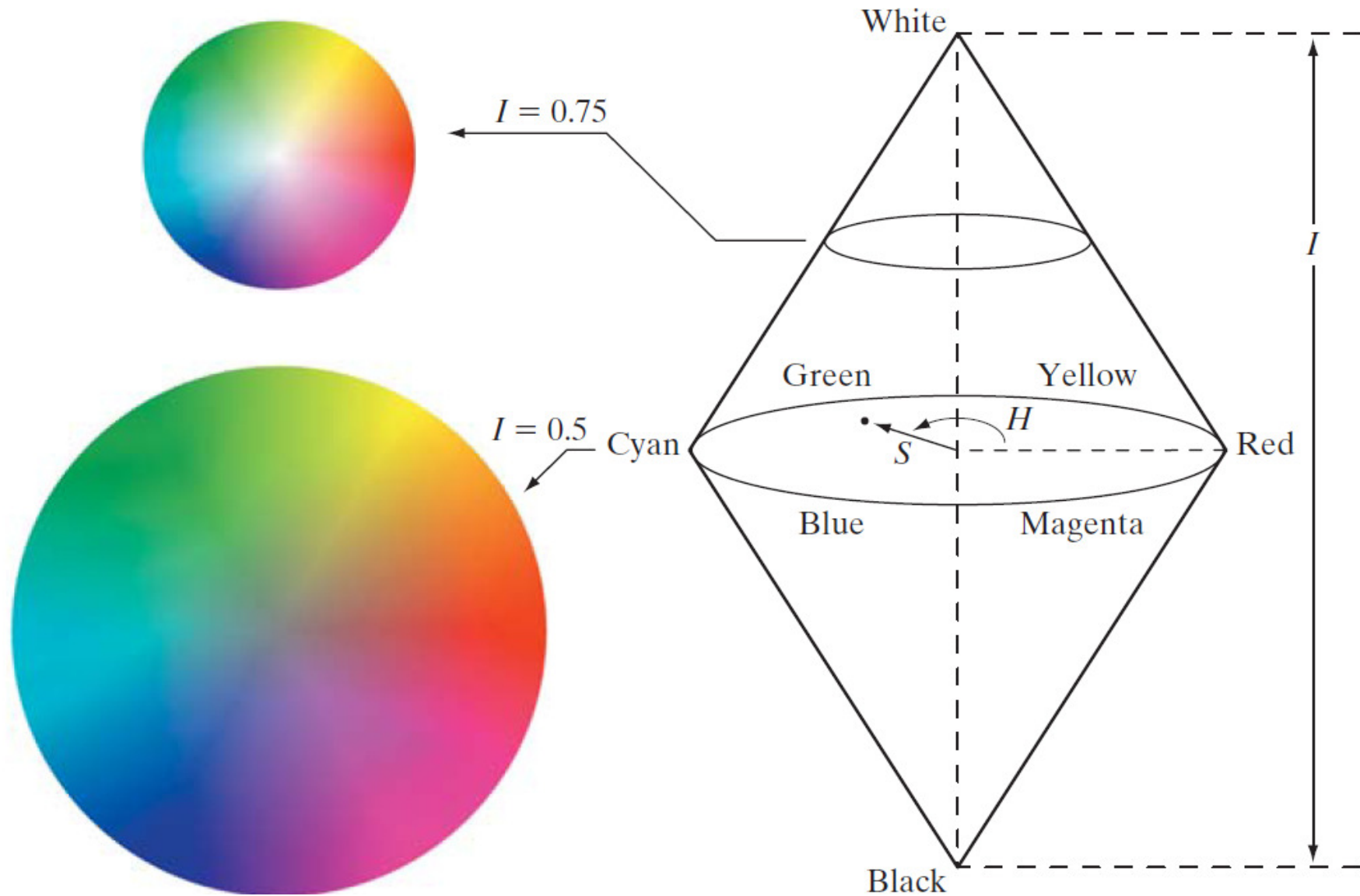




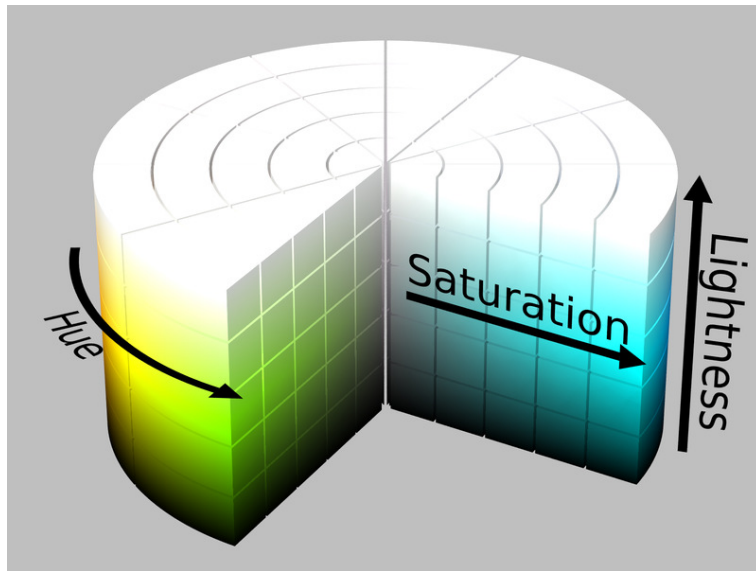
Histograma de Colores

- Al dividir en tramos iguales R, G y B se obtienen bins muy grandes:
 - Con 6 tramos, los colores (0,0,0) (42,0,0) (0,42,0) (42,42,42) son todos considerados iguales.
- Divisiones más finas producen muchos bins
 - Con 32 tramos, es decir [0,7] [8,15] ... se obtienen 32 mil bins
- Idea: cambiar el espacio de colores buscando asimetrías entre canales

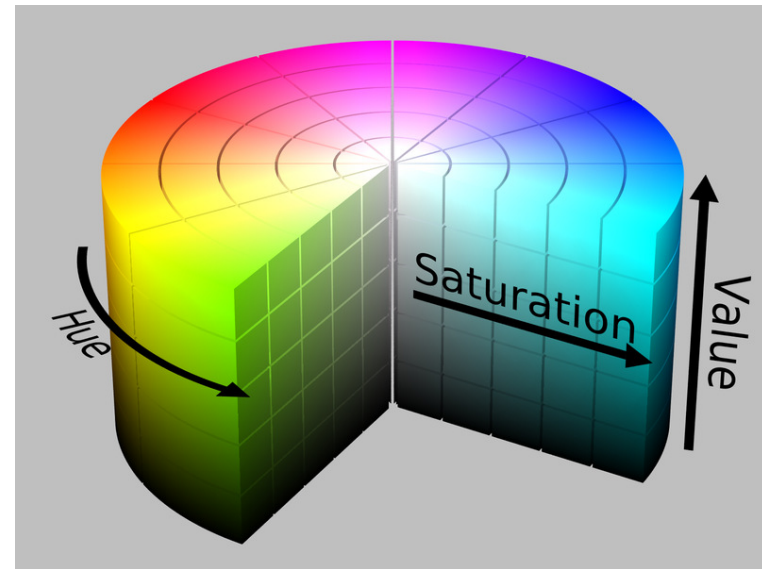




Espacios de Color “H__”



HSL=HSI=HLS



HSV=HSB

Espacios de Color “H__”

- RGB → HSV es una transformación geométrica:

```
Max = max(R, G, B); Min = min( R, G, B);  
Value = max(R, G, B);  
if( Max == 0 ) then  
    Saturation = 0; else  
    Saturation = (Max-Min)/Max;  
if( Max == Min ) Hue is undefined (achromatic color);  
otherwise:  
if( Max == R && G > B ) Hue = 60*(G-B)/(Max-Min)  
else if( Max == R && G < B ) Hue = 360 + 60*(G-B)/(Max-Min)  
else if( G == Max ) Hue = 60*(2.0 + (B-R)/(Max-Min))  
else Hue = 60*(4.0 + (R-G)/(Max-Min))
```

Conversión entre espacios de color en OpenCV:

https://docs.opencv.org/3.4.1/de/d25/imgproc_color_conversions.html

Espacios de Color “Y__”

- RGB → Y (Gris)

$$Y = (0.299)R + (0.587)G + (0.114)B$$

- RGB → YC_bC_r

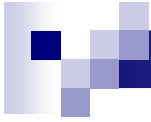
$$Cb = B - Y$$

$$Cr = R - Y$$

$$Cg = G - Y$$

$$\begin{aligned} Y &= 0.299*R + 0.587*G + 0.114*B \\ Cb &= -0.169*R - 0.331*G + 0.500*B \\ Cr &= 0.500*R - 0.419*G - 0.081*B \end{aligned}$$

- YUV, Y'UV, YC_bC_r , YC_rC_b , YP_bP_r , ...



Histogramas incluyendo “Hue”

- La coordenada **H** representa el color puro, mientras que las otras dos dimensiones se refieren a la “variante” del color
 - Hacer una división asimétrica de las dimensiones
 - Ej: Dividir **H** en 16 tramos y las otras dos en 4 tramos $\rightarrow 16 \times 4 \times 4 = 256$ bins



Descriptor: Scalable Color (SCD)

- Histograma HSV con división:

- 256 bins: $16 \times 4 \times 4$

- 128 bins: $8 \times 4 \times 4$

- 64 bins: $8 \times 2 \times 4$

- 32 bins: $8 \times 2 \times 2$

- 16 bins: $4 \times 2 \times 2$



Descriptor: Color Structure Histogram

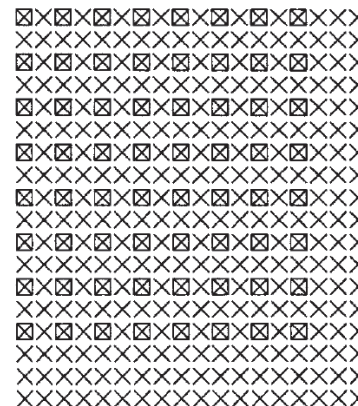
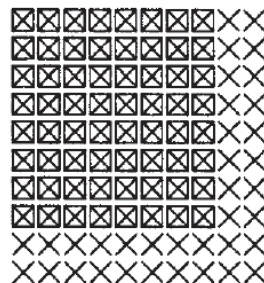
- Idea: Representar como está distribuido cada color en la imagen (si está agrupado en zonas o disperso por toda la imagen)
- Es un histograma del número de apariciones de cada color dentro de una máscara
 - Máscara de 8x8 que recorre la imagen (sliding window)
 - Para cada color se verifica si existe algún pixel dentro de la máscara con ese color



Descriptor: Color Structure Histogram

■ Máscara:

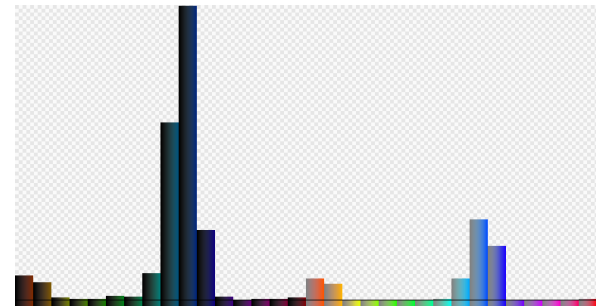
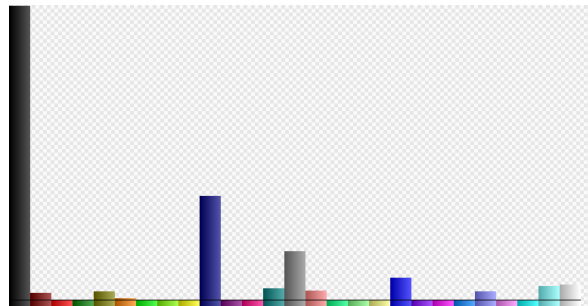
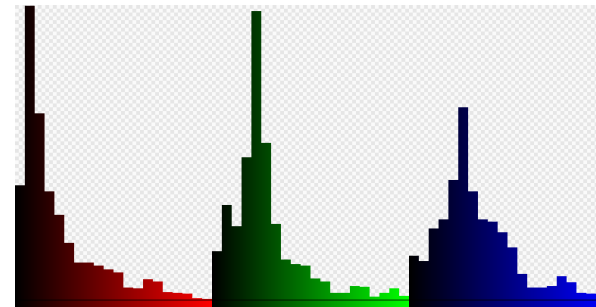
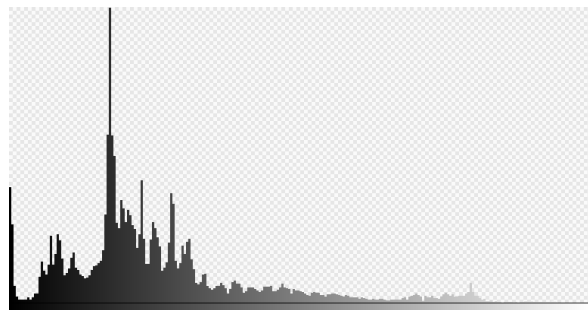
- Grilla de 8x8 con 64 posibles posiciones
- Para imágenes mayores a 256x256 se separan las celdas de la máscara (i.e., una máscara de 8x8 que abarca más espacio)





Histogramas de Color

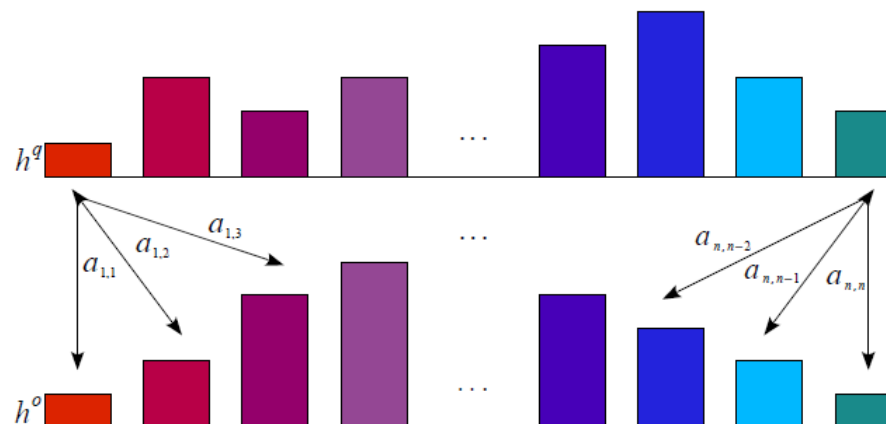
- Cuando dos histogramas de color usan una misma división del espacio de color, se pueden comparar con distancia L_1 o L_2
- Problema: la distancia euclidiana ignora similitud entre dimensiones
- Idea: Incluir en la función de distancia la información de similitud entre bins



Comparar vectores considerando similitud entre bins

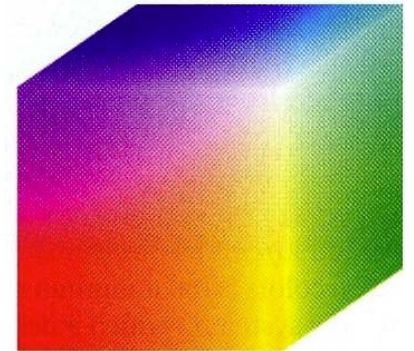
- Forma Cuadrática (Quadratic Form Distance)
 - Generalización de la distancia L_2
 - A es la **Matriz de Similitud**, donde a_{ij} es la similitud entre bin i y bin j
 - Costo de evaluación: $O(n^2)$

$$\text{QFD}(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \cdot A \cdot (\vec{x} - \vec{y})}$$



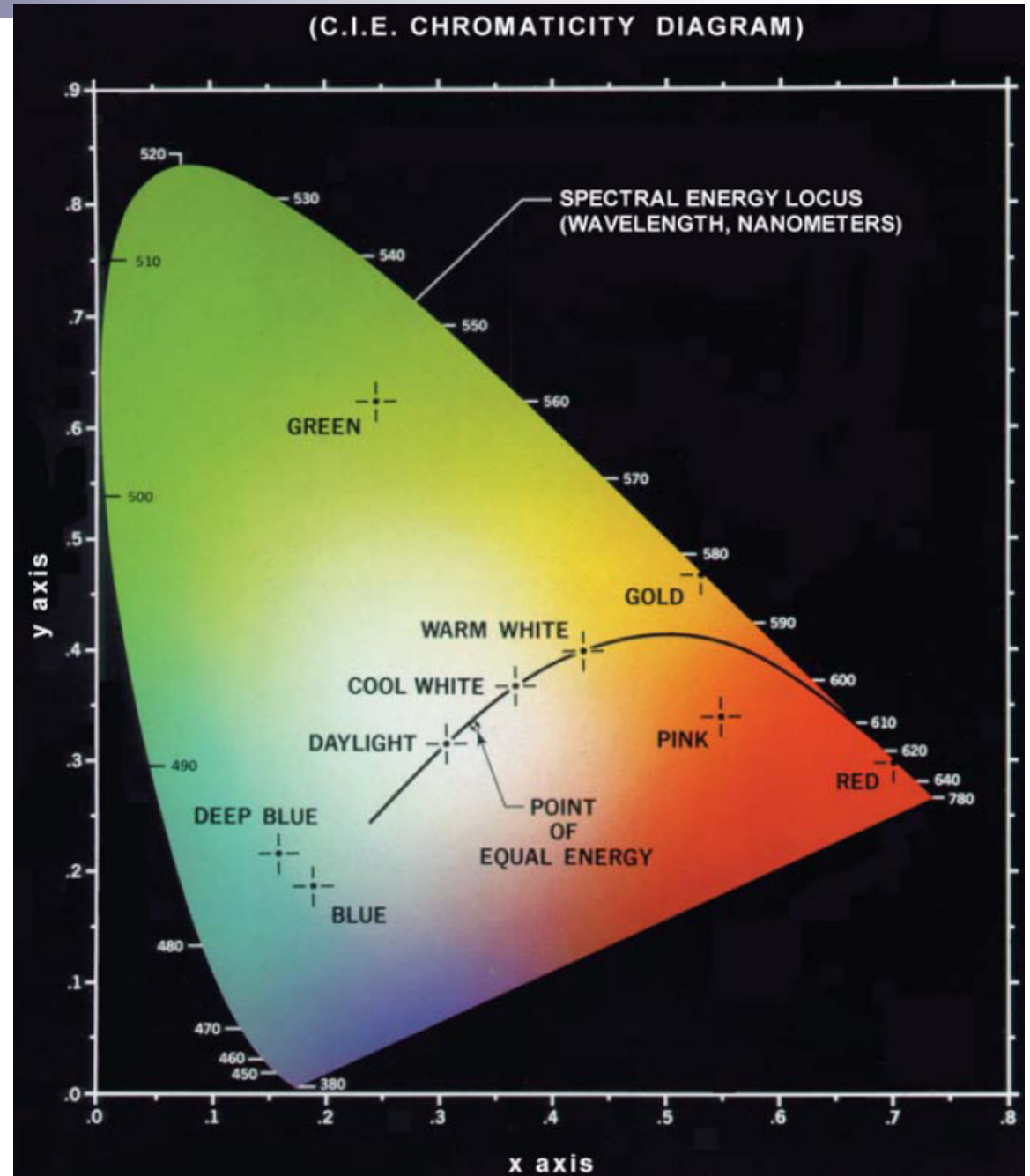
Similitud entre colores

- Problema: Cómo definir la Matriz de Similitud entre colores
- Idea: Comparar colores con distancia euclidiana en el cubo RGB
- Problema: En la percepción humana no son simétricos los canales RGB. Ej:
 - El amarillo debiera ser más parecido al blanco que al magenta, pero el amarillo y magenta están a misma distancia del blanco.
 - El rojo debiera ser más parecido al amarillo que al verde, pero rojo y verde están a misma distancia del amarillo.
- Utilizar distancias dentro del espacio de color CIELAB o CIELUV



CIE

- Commission Internationale de L'Eclairage
- Cromacidad:





Espacios de Color “CIE ____”

- Experimentalmente definen CIE RGB (~1920)
 - Basado en la longitud de onda de la luz
- 1931: CIE XYZ
 - Transformación lineal de CIE RGB
- 1976: CIELAB (L^*, a^*, b^*) y CIELUV (L^*, u^*, v^*)
 - Transformación de XYZ para que la distancia euclidiana entre colores sea similar a la perceptual
 - Adecuado para comparar colores
- 2002: CIECAM

Conversiones RGB a “CIE ____”

RGB \leftrightarrow CIE L*a*b*

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$X \leftarrow X/X_n, \text{ where } X_n = 0.950456$$

$$Z \leftarrow Z/Z_n, \text{ where } Z_n = 1.088754$$

$$L \leftarrow \begin{cases} 116 * Y^{1/3} - 16 & \text{for } Y > 0.008856 \\ 903.3 * Y & \text{for } Y \leq 0.008856 \end{cases}$$

$$a \leftarrow 500(f(X) - f(Y)) + \text{delta}$$

$$b \leftarrow 200(f(Y) - f(Z)) + \text{delta}$$

$$f(t) = \begin{cases} t^{1/3} & \text{for } t > 0.008856 \\ 7.787t + 16/116 & \text{for } t \leq 0.008856 \end{cases}$$

RGB \leftrightarrow CIE L*u*v*

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$L \leftarrow \begin{cases} 116 * Y^{1/3} - 16 & \text{for } Y > 0.008856 \\ 903.3Y & \text{for } Y \leq 0.008856 \end{cases}$$

$$u' \leftarrow 4 * X / (X + 15 * Y + 3Z)$$

$$v' \leftarrow 9 * Y / (X + 15 * Y + 3Z)$$

$$u \leftarrow 13 * L * (u' - u_n) \quad \text{where } u_n = 0.19793943$$

$$v \leftarrow 13 * L * (v' - v_n) \quad \text{where } v_n = 0.46831096$$

Conversión entre espacios de color en OpenCV:

https://docs.opencv.org/3.4.1/de/d25/imgproc_color_conversions.html



Número variable de bins

- Problema: Histogramas de color son muy largos, debido a que se necesita pre-definir una partición de colores
- Idea: Adaptive binning
 - Crear una partición del espacio de color variable, específica a los colores de cada imagen
 - Cada pixel corresponde a un punto (3d) en el espacio de color
 - Con clustering determinar N centroides (3d) por imagen
 - Calcular un histograma de N bins con las veces que aparece cada color correspondiente:
 - Hard Assignment: el color de cada pixel se “redondea” al centroide más cercano y se suma 1 a ese bin
 - Soft Assignment: cada color hace una votación ponderada a cada centroide según su distancia
- Problema: ¿Cómo comparar estos histogramas?



Histogramas con bins variables

■ Descriptor tipo **Signature**:

- Cada dimensión del descriptor corresponde a un vector s_i y el peso x_i de ese vector: $\mathbf{x}=\{(s_1, x_1), \dots, (s_n, x_n)\}$
- Usualmente s_i son vectores obtenidos por clustering
- Ej: Para histogramas de color, s_i son vectores 3d que forman una partición del espacio de colores

■ Distancia para comparar Signatures:

- **Earth Mover's Distance (EMD)** compara dos Signatures no necesariamente del mismo largo
- Calcula el costo mínimo para transformar un Signature en el otro
- Requiere resolver el problema de transporte (optimización)

Earth Mover's Distance (EMD)

- Dado dos vectores $\mathbf{x}=(x_1,\dots,x_n)$ $\mathbf{y}=(y_1,\dots,y_m)$ y una Matriz de Costos $\mathbf{C}_{n \times m}$ con los costos para mover una unidad entre bins, EMD corresponde al costo mínimo a pagar para convertir la distribución \mathbf{x} a la distribución \mathbf{y}
- EMD calcula la Matriz de Flujos $\mathbf{F}_{n \times m}$ que contiene los movimientos desde cada bin de \mathbf{x} hacia cada bin de \mathbf{y} que logran el mínimo costo total

$$\text{EMD}(\vec{x}, \vec{y}) = \min \left\{ \sum_{i=1}^n \sum_{j=1}^m c_{ij} f_{ij} \right\} \quad \text{sujeto a:} \quad \begin{aligned} f_{ij} &\geq 0 \\ \sum_{j=1}^m f_{ij} &= x_i \quad \forall i \in \{1, \dots, n\} \\ \sum_{i=1}^n f_{ij} &= y_j \quad \forall j \in \{1, \dots, m\} \end{aligned}$$

Función EMD en OpenCV:

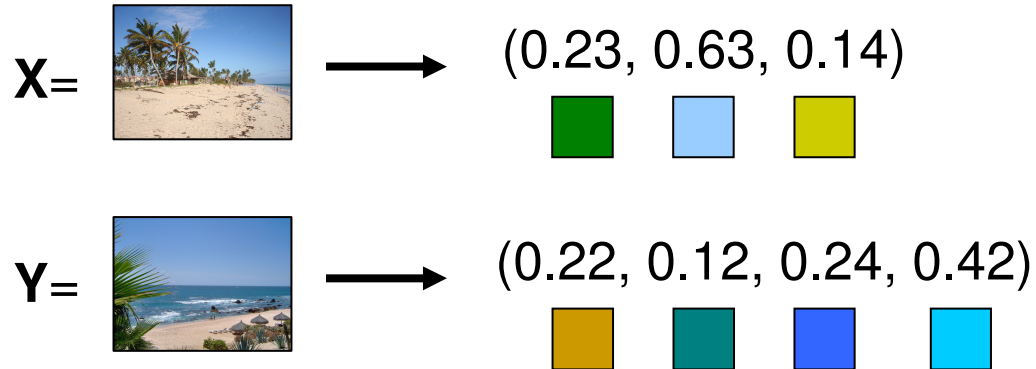
https://docs.opencv.org/3.4.1/d6/dc7/group_imgproc_hist.html#ga902b8e60cc7075c8947345489221e0e0



Earth Mover's Distance (EMD)








- La Matriz de Costos **C** representa el costo c_{ij} de mover una unidad desde el bin i de \mathbf{x} al bin j de \mathbf{y}
 - *Ground-distance*: Función que compara distancias entre bins
 - En descriptores tipo Signature, la ground distance corresponde a la distancia usada para comparar los vectores s_i de cada descriptor (ej: distancia entre colores)
- La Matriz de Flujos **F** representa una solución con las cantidades f_{ij} a mover desde el bin i de \mathbf{x} al bin j de \mathbf{y}
- Costo evaluación: en general exponencial, usualmente $O(n^3 \log n)$
- Propiedades métricas de EMD dependen de la ground-distance

EMD



1) Matriz de Costos:

C_{ij} : Lo que hay que “pagar” para mover una unidad del color i al color j (distancia en espacio CIE u otro)

				
	0.51	0.06	0.30	0.41
	0.37	0.43	0.19	0.06
	0.06	0.51	0.41	0.35

2) Matriz de Flujos:

F_{ij} : Plan de “movimientos” a seguir para convertir la distribución X en la distribución Y con el mínimo costo

	0.22	0.12	0.24	0.42
0.23	0.08	0.12	0.03	0
0.63	0	0	0.21	0.42
0.14	0.14	0	0	0

3) EMD:

Costo total “a pagar” para convertir X en Y :

$$\text{EMD}(X, Y) = \sum C_{ij} F_{ij} = 0.13$$

Notar que si se usa cualquier otra matriz de flujos el valor de $\text{EMD}(X, Y)$ puede aumentar.

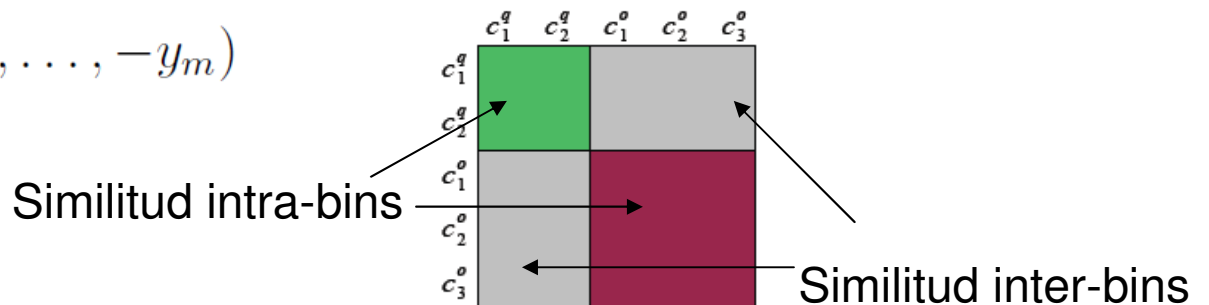
Alternativa a EMD

- Signature Quadratic Form Distance (SQFD)
 - Forma Cuadrática para comparar Signatures
 - Se concatenan los vectores de pesos cambiando el signo
 - La matriz de similitud **A** contiene la similitud intra-bins e inter-bins
 - Crear **A** requiere una *ground distance* convertida a similitud

$$\text{SQFD}(\vec{x}, \vec{y}) = \sqrt{(\vec{x} \mid -\vec{y})^T \cdot A \cdot (\vec{x} \mid -\vec{y})}$$

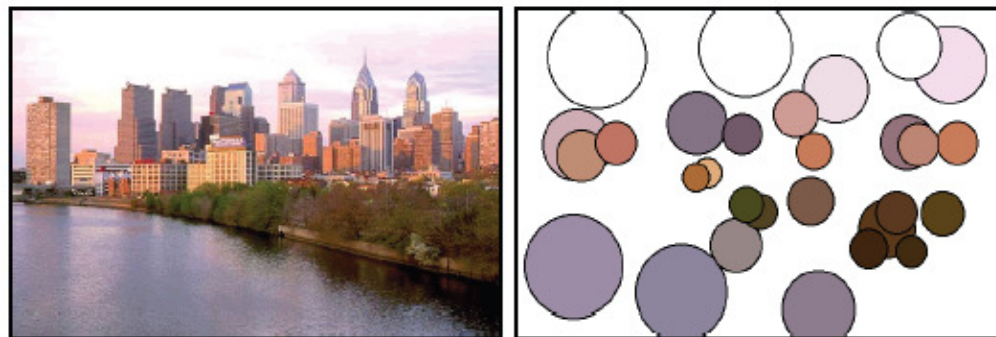
$$(\vec{x} \mid -\vec{y}) = (x_1, \dots, x_n, -y_1, \dots, -y_m)$$

$$A \in \mathbb{R}^{(n+m) \times (n+m)}$$



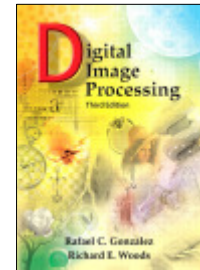
Histogramas incluyendo posición

- Idea: Al calcular el clustering de colores de cada imagen, además del color (r,g,b) incluir la posición (i,j) de cada pixel
 - Requiere una distancia para vectores 5-d que pondere distancia entre colores y distancia espacial
 - El clustering determina centroides que incluyen colores y ubicación
 - Descriptor de colores con ubicación espacial



Bibliografía

- **Digital Image Processing.** González et al. 2008
 - Cap 6 (Color)





Papers

- **Manjunath et al.** “Color and Texture Descriptors”. 2001.
- **Rubner et al.** “The Earth Mover’s Distance as a Metric for Image Retrieval”. 2000.
- **Beecks et al.** “Signature Quadratic Form Distance”. 2010.