Visión por Computadora I

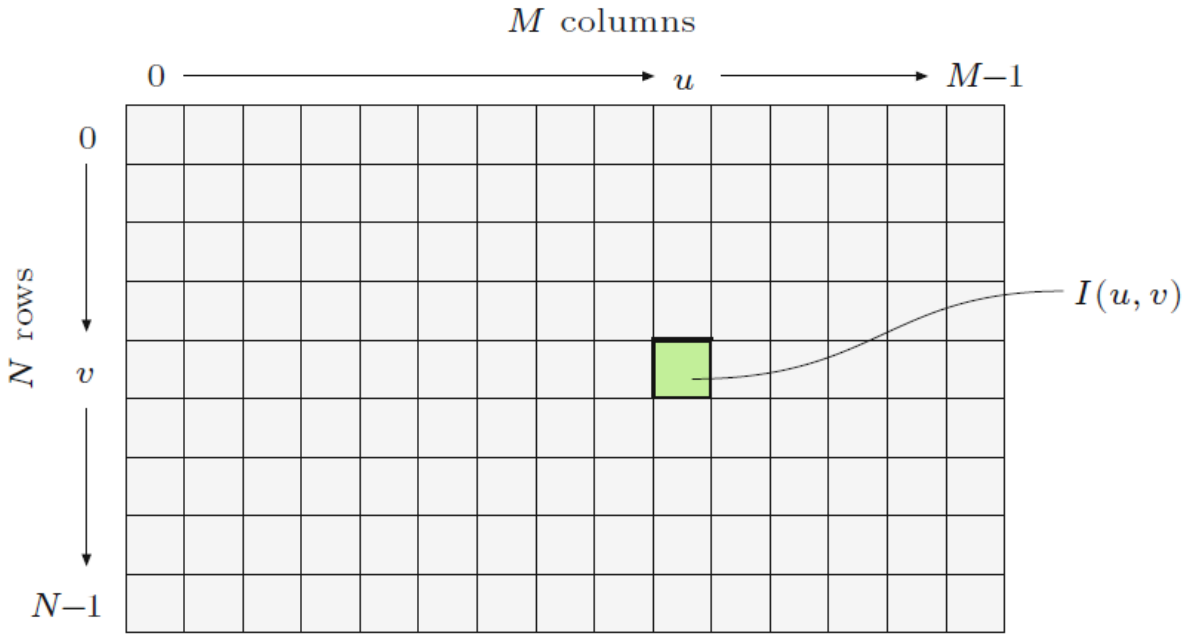
Ing. Maxim Dorogov

(mdorogov@fi.uba.ar)

Laboratorio de Sistemas Embebidos -FIUBA

FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

OPERADORES DE PÍXEL



Consideramos a las imágenes como funciones discretas 2D que mapean un dominio de coordenadas (u,v) a un conjunto discreto de valores P = I(u, v)

También podemos combinarlas:

- $g(\bar{x}) = h\big(f_0(\bar{x}), \dots, f_n(\bar{x})\big)$

Es decir, una imagen de salida g$(\bar{x})$ es función h$\big(f(\bar{x})\big)$ de una o más imágenes de entrada $f_i(\bar{x})$

  - x : Dominio N-dimensional
  - En imágenes discretas $x = (u, v)$

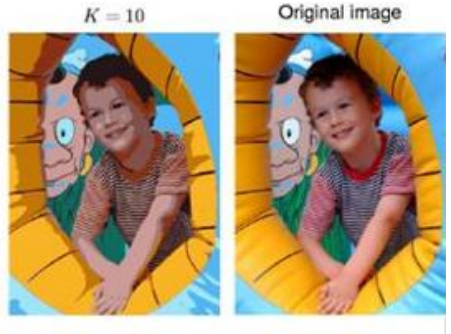
- $g(u, v) = h\big(f(u, v)\big)$

OPERADORES DE PÍXEL

Algunas definiciones

- **Contraste**: Diferencia entre el valor de intensidad máximo y mínimo en una imagen.
- **Rango dinámico**: Se caracteriza por los niveles de intensidad que tienen los diferentes pixeles en la imagen. Rango dinámico máximo en 8 bits: 256 valores.

- Podemos aumentar el contraste aplicando transformaciones sencillas a los valores de los pixeles presentes en la imagen.
- Para modificar el rango dinámico se utilizan técnicas mas complejas ya que es necesario generar valores adicionales de intensidad (por ejemplo mediante interpolación) que no formaban parte de la imagen original.
- Es deseable trabajar con imágenes de gran rango dinámico ya que se minimizan los efectos de degradación durante su procesamiento y las redundancias presentes en los niveles de intensidad se utilizan en la compresión.

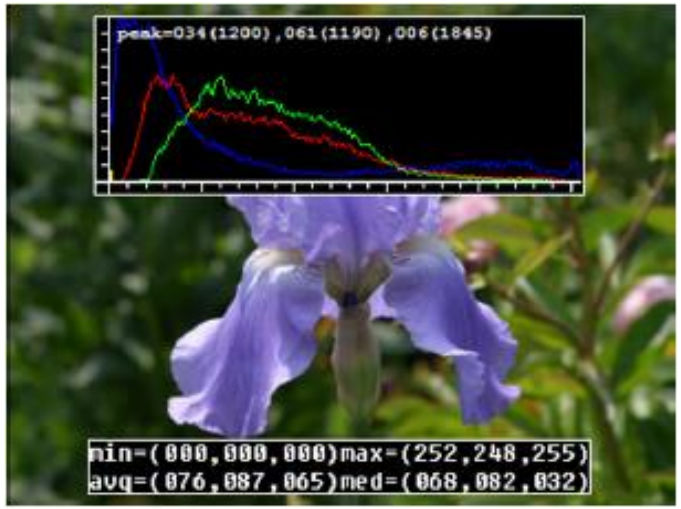


Derecha: Imagen original sin compresión

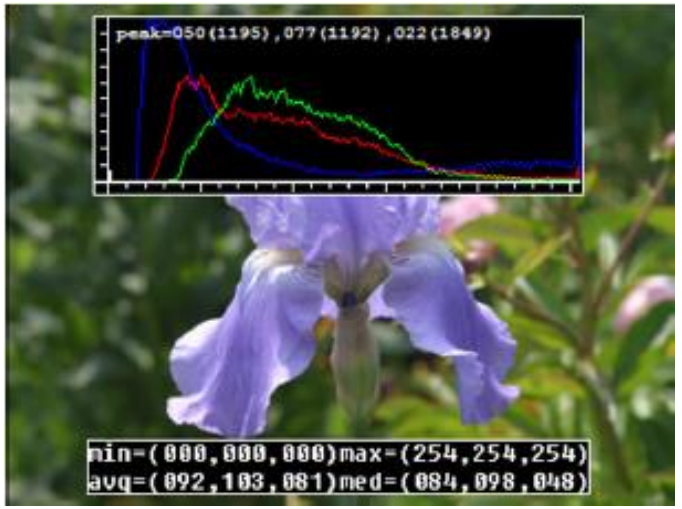
Izq: Imagen comprimida a 10 niveles de intensidad (muy bajo rango dinámico)

Bajo contraste

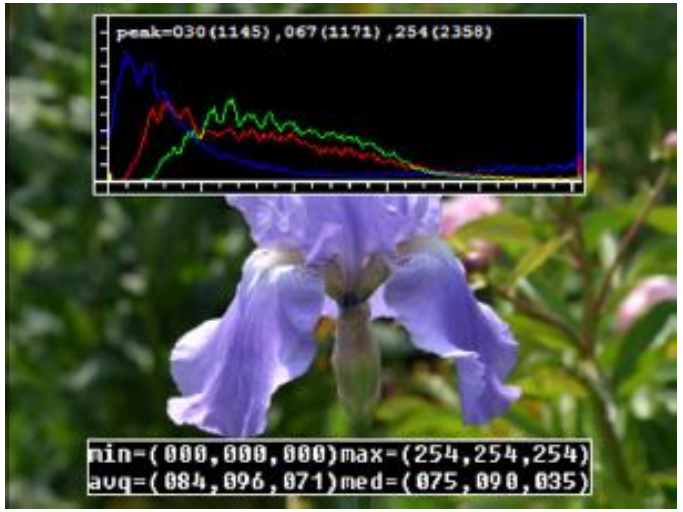
Alto contraste



Original



a=1, b=16



a=1.1, b=0

OPERADORES DE PÍXEL

- Un procesamiento común es:

$$g(x) = a.f(x) + b$$

- Donde
  - $a > 0$ es la ganancia y controla el contraste
  - b es el bias y controla el brillo
- También podría ser: $g(x) = a.f(x) + b(x)$

OPERADORES DE PÍXEL

- Los operadores de píxel lineales admiten superposición:

$$h(f_0 + f_1) = h(f_0) + h(f_1)$$

- Otra aplicación (transición entre dos imágenes):

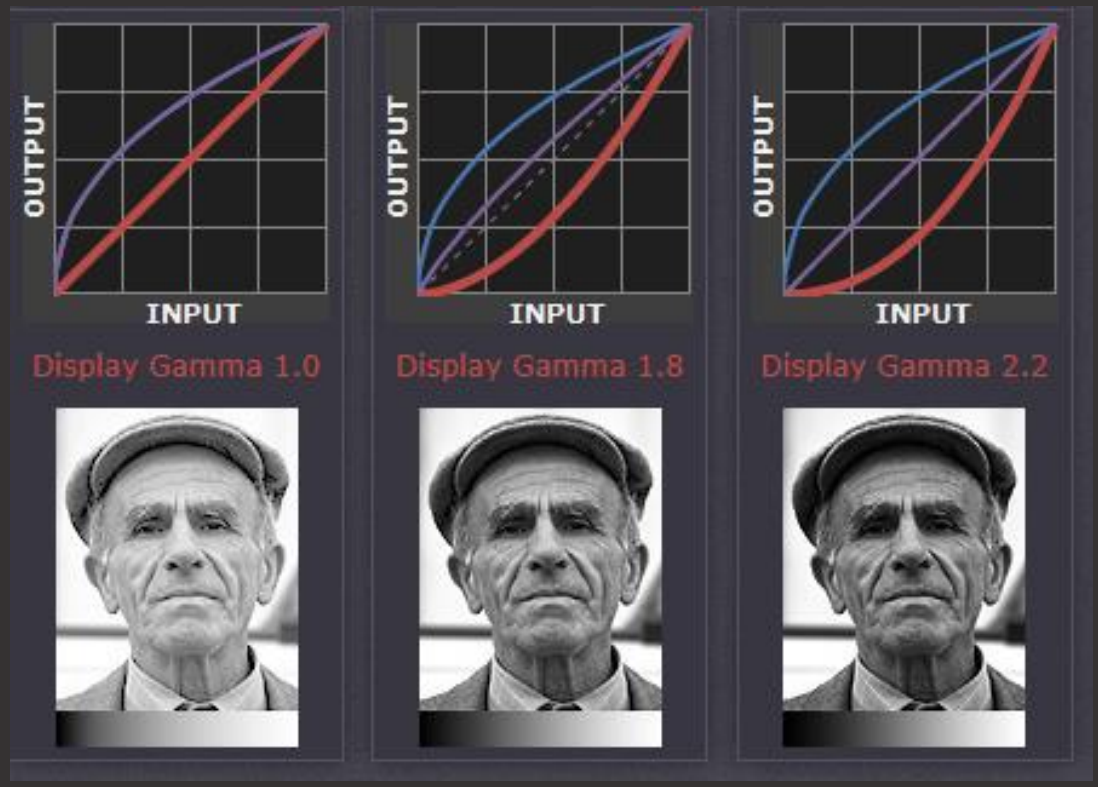
$$\boldsymbol{g(x) = (1 - \alpha).f_0(x) + \alpha.f_1(x)} \qquad 0 < \alpha < 1$$

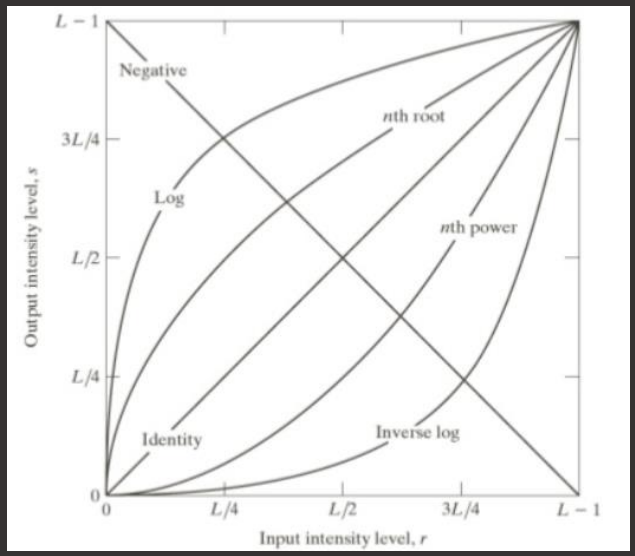
- Las transformaciones de píxel pueden ser también no lineales

$$\boldsymbol{g(x) = (f(x))^{1/\gamma}}$$

Esta trasformación es la conocida corrección de gamma (relación entre el valor numérico del píxel y su luminancia real, como la perciben los humanos). Ver este link

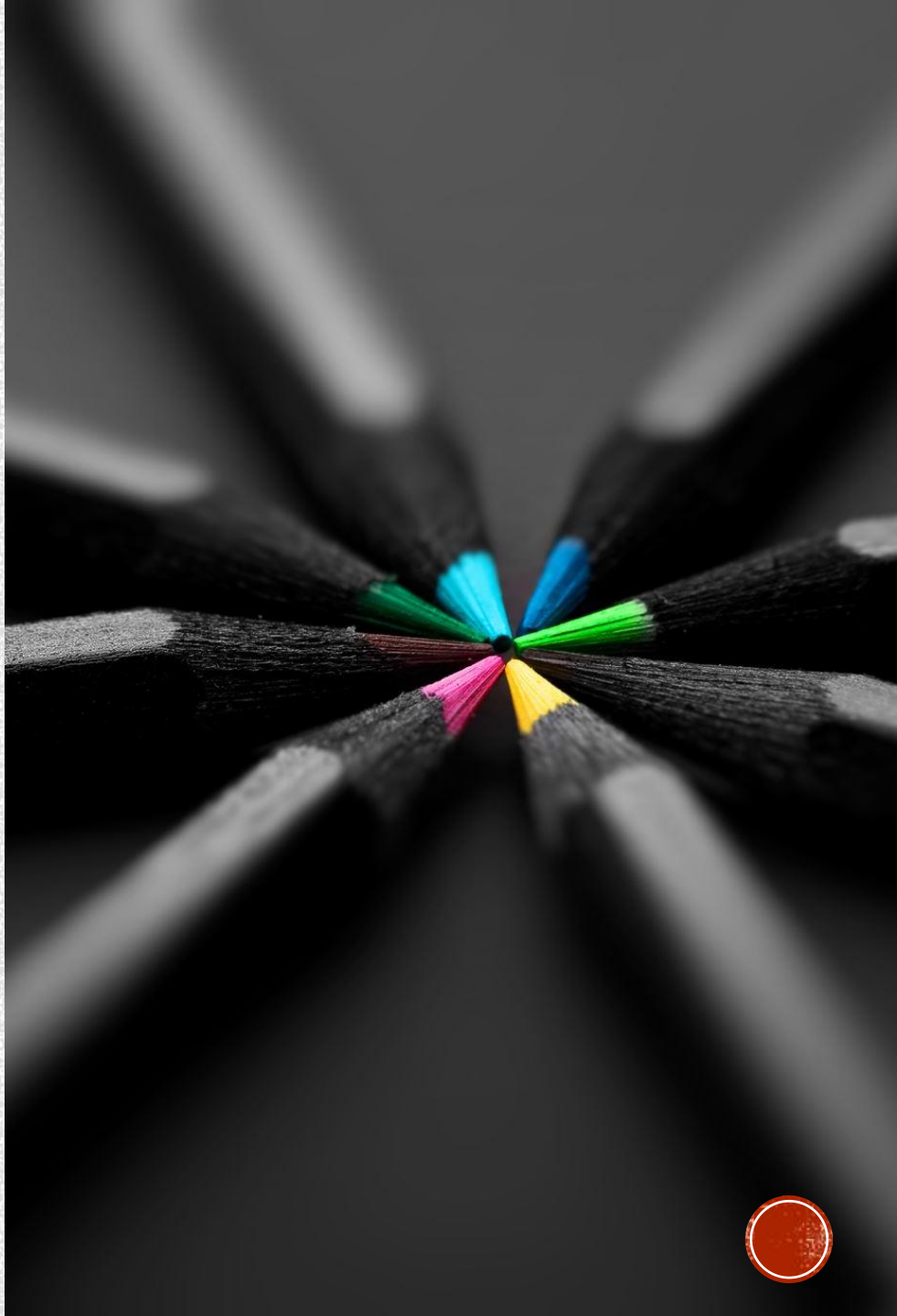
El valor de ɣ típico suele ser de 2.2





TRANSFORMACIÓN COLOR A GRISES

1. Promediado (método más común)
   - $I_{gris\,(i,j)} = \frac{R_{(i,j)}+G_{(i,j)}+B_{(i,j)}}{3}$
2. Brillo (Reducción de contraste)
   - $I_{gris\,(i,j)} = \frac{máx\left(R_{(i,j)};G_{(i,j)};B_{(i,j)}\right)-min\left(R_{(i,j)};G_{(i,j)};B_{(i,j)}\right)}{2}$
3. Luminosidad (Buena performance. Usado por GIMP)
   - $I_{gris\,(i,j)} = 0{,}21R_{(i,j)} + 0{,}72G_{(i,j)} + 0{,}07B_{(i,j)}$



COORDENADAS CROMÁTICAS

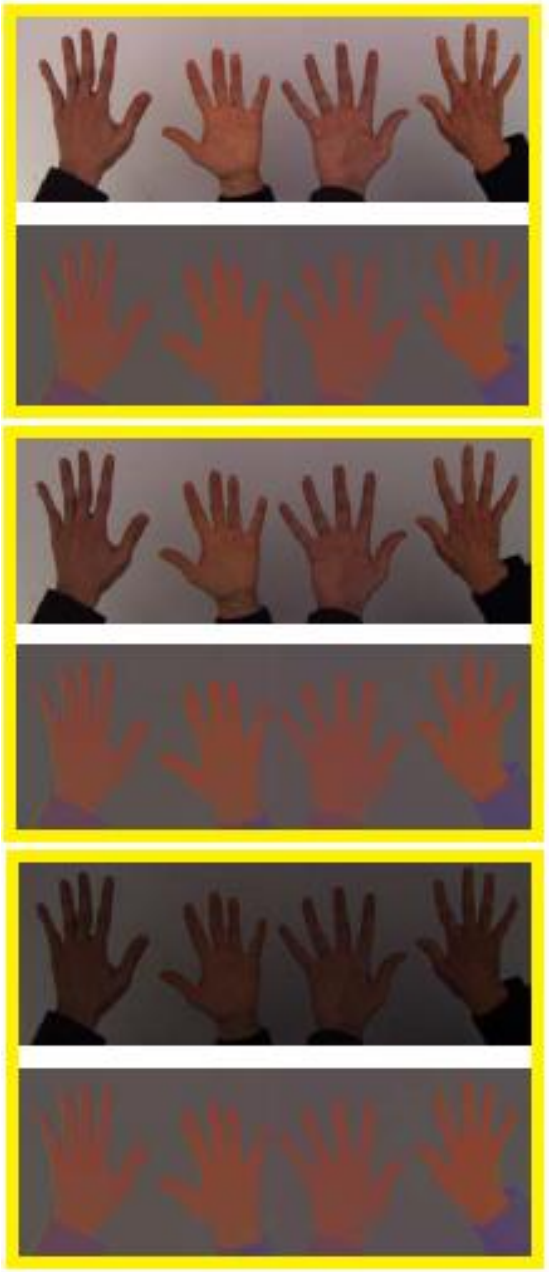
- Transformación de imágenes color

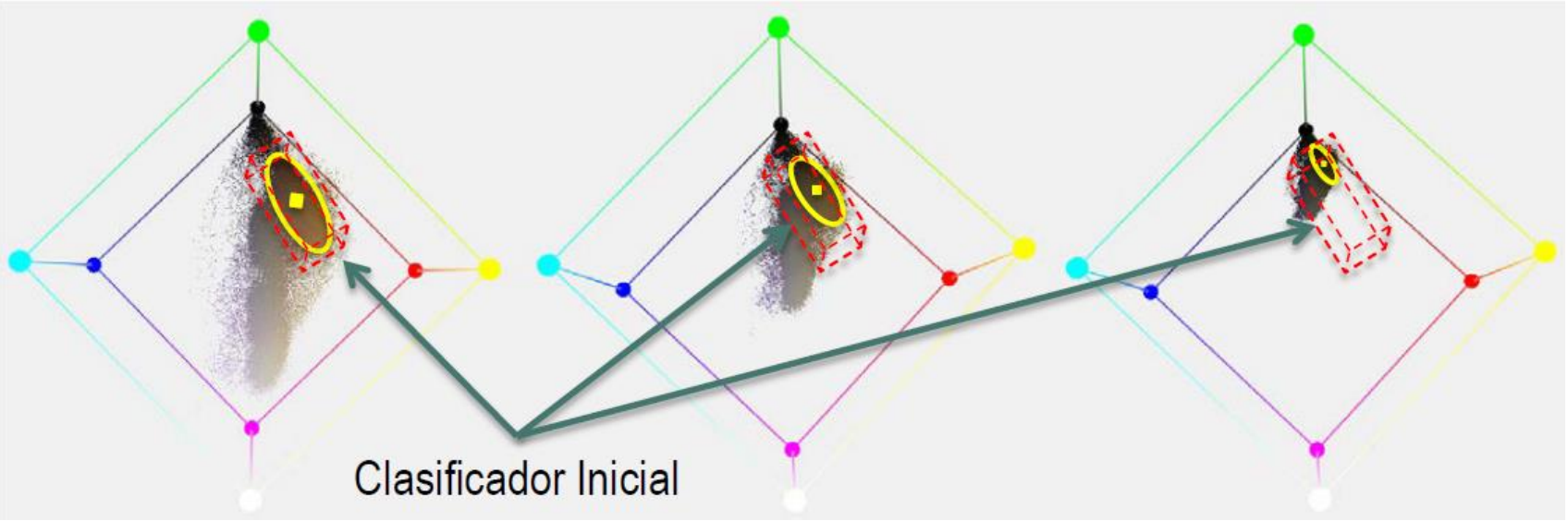
1. Sumar un bias a cada canal no solo modifica el brillo sino el matiz y la saturación (Solución: Modificar solo la luminancia en un espacio de color desacoplado por ej: YIQ o HSV y luego volver al espacio RGB)
2. Balance de color (por ejemplo, compensar por fuente de luz incandescente) se realiza multiplicando cada canal por un factor de escala diferente.

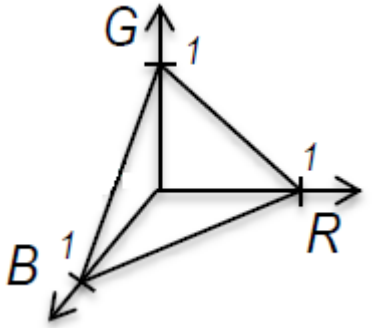
$$(R,G,B) \rightarrow \left(\frac{R}{R+G+B};\frac{G}{R+G+B};\frac{B}{R+G+B}\right)$$

Entonces si $s \in \mathcal{R}$ y s. $(R,G,B) \rightarrow \left(\frac{sR}{sR+sG+sB};\frac{sG}{sR+sG+sB};\frac{sB}{sR+sG+sB}\right)$

"**Descriptor** invariante a los cambios de contraste"



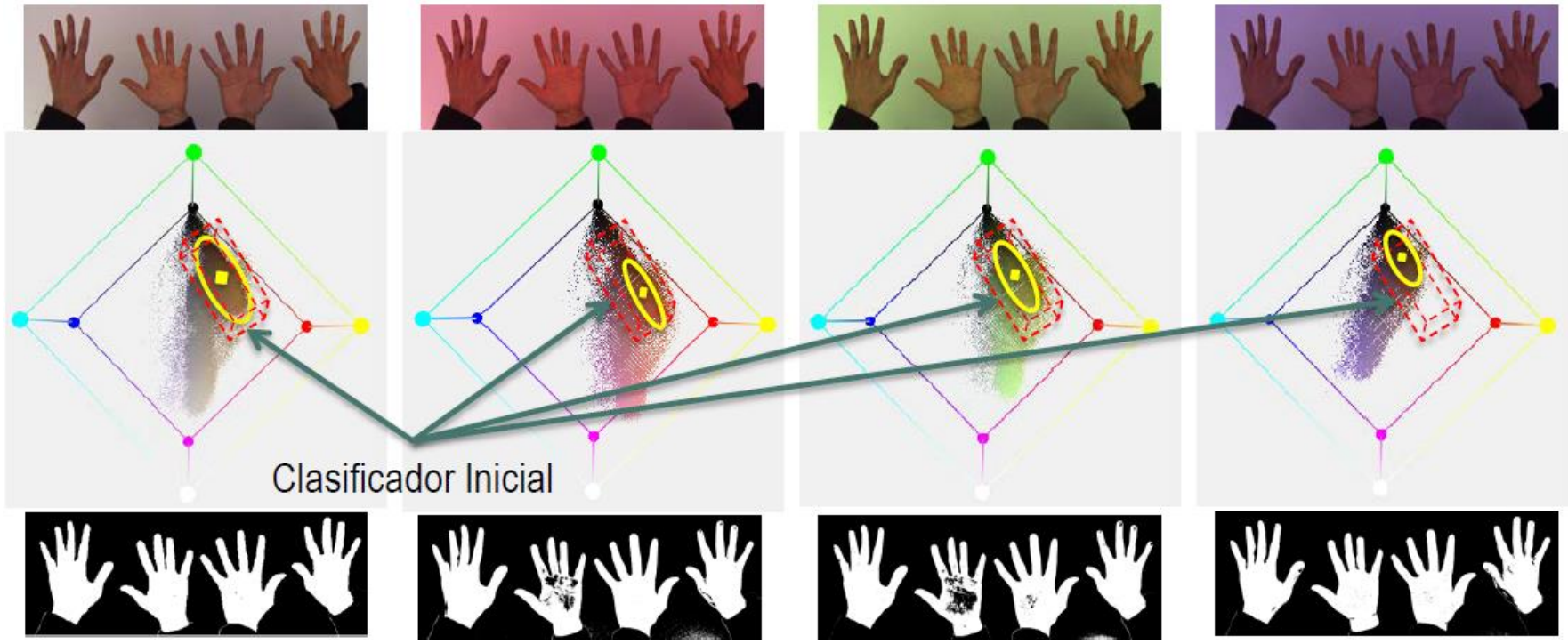




COORDENADAS CROMÁTICAS

- Este nuevo descriptor es de dimensión 2, es decir, la proyección sobre el plano R+G+B=1
- Ec. Plano:

$$\frac{X}{a}+\frac{Y}{b}+\frac{Z}{c}=1$$
$$a=b=c=(X+Y+Z)$$



ALGORITMO WHITE-PATCH

- Dependiendo el color de la fuente de iluminación tendremos variaciones en los colores reflejados.
- ¿Cómo hacemos para tratar de ser menos susceptibles esto?

ALGORITMO WHITE-PATCH

Original

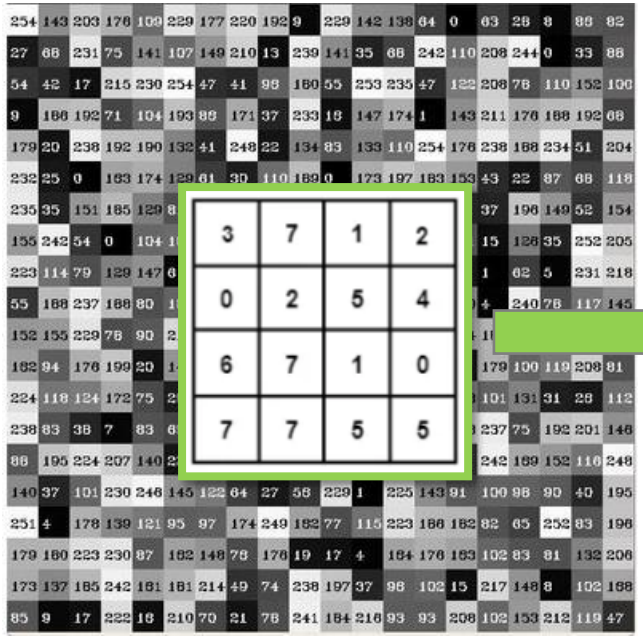
White Patch

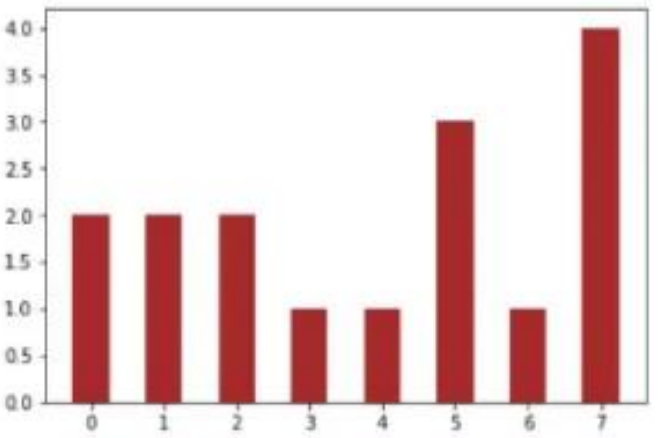
El color de la luz que ilumina la escena puede influir en la percepción que tenemos del objeto, para solventar esto podemos aplicar algoritmos de corrección con procesamiento digital:

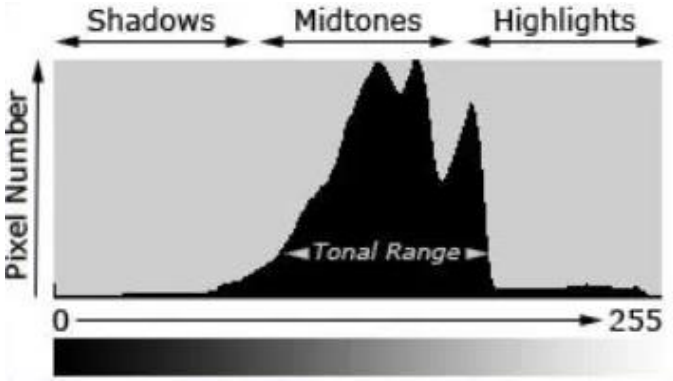
- Asumimos que los valores máximos de color en los tres canales es el color del blanco bajo la luz de la escena.
- $R_{max}(Img)$: $Valor\ máximo\ del\ canal\ R\ de\ la\ imagen$
- $G_{max}(Img)$: $Valor\ máximo\ del\ canal\ G\ de\ la\ imagen$
- $B_{max}(Img)$: $Valor\ máximo\ del\ canal\ B\ de\ la\ imagen$

Normalizamos entonces al color de la luz blanca puro (255, 255, 255)

- $(R, G, B) \rightarrow \left(\frac{255}{R_{max}(I)}R, \frac{255}{G_{max}(I)}G, \frac{255}{B_{max}(I)}B\right)$







HISTOGRAMAS

- Un histograma nos dice cómo están distribuidos los valores de intensidad en una imagen
- Permite ver posibles defectos
- Nos interesa modificar la imagen para llevar su histograma a un rango acorde para cada aplicación.

HISTOGRAMAS

La distribución de intensidades en una imagen sin procesar tiende a ser suave y "continua". Si la imagen fue modificada digitalmente algunas operaciones de pixeles, la alteración de contraste o la compresión se pueden reconocer fácilmente analizando el histograma:

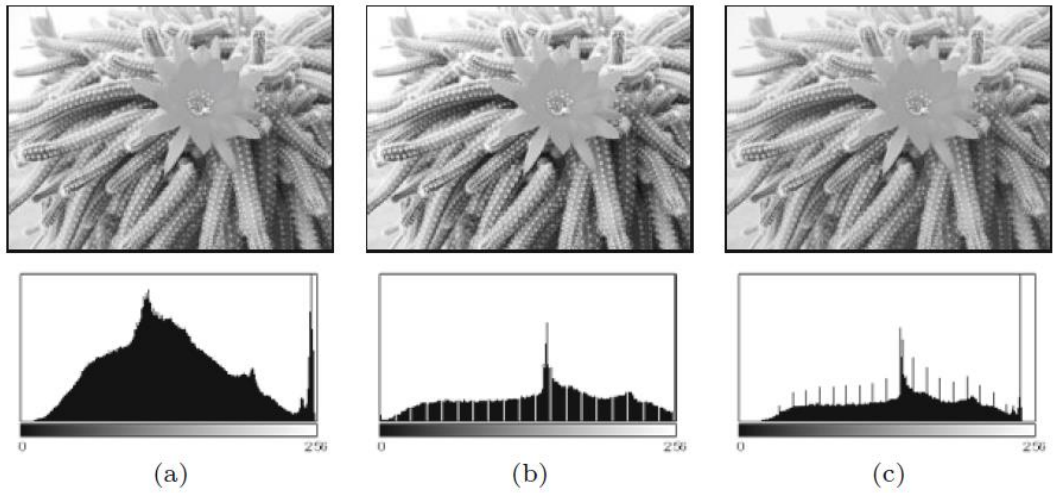
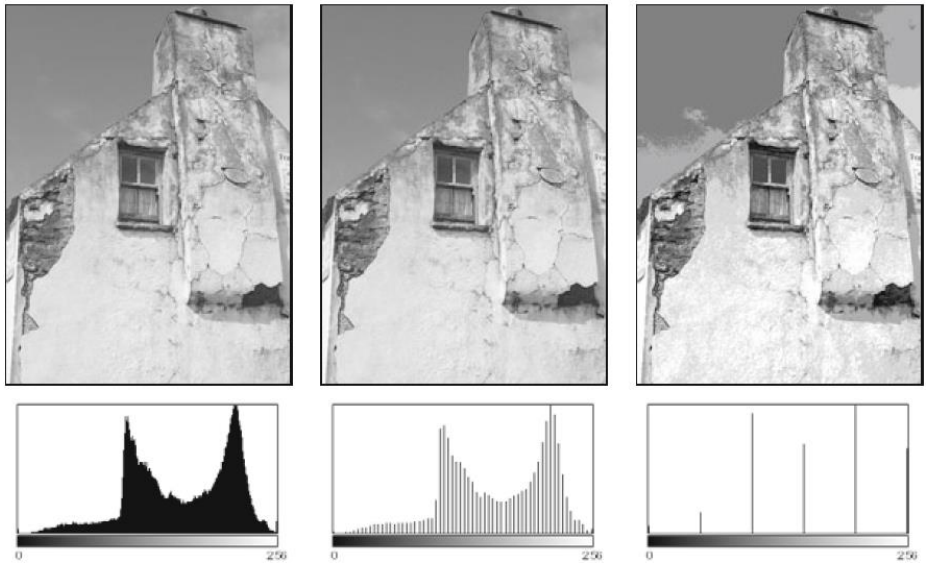


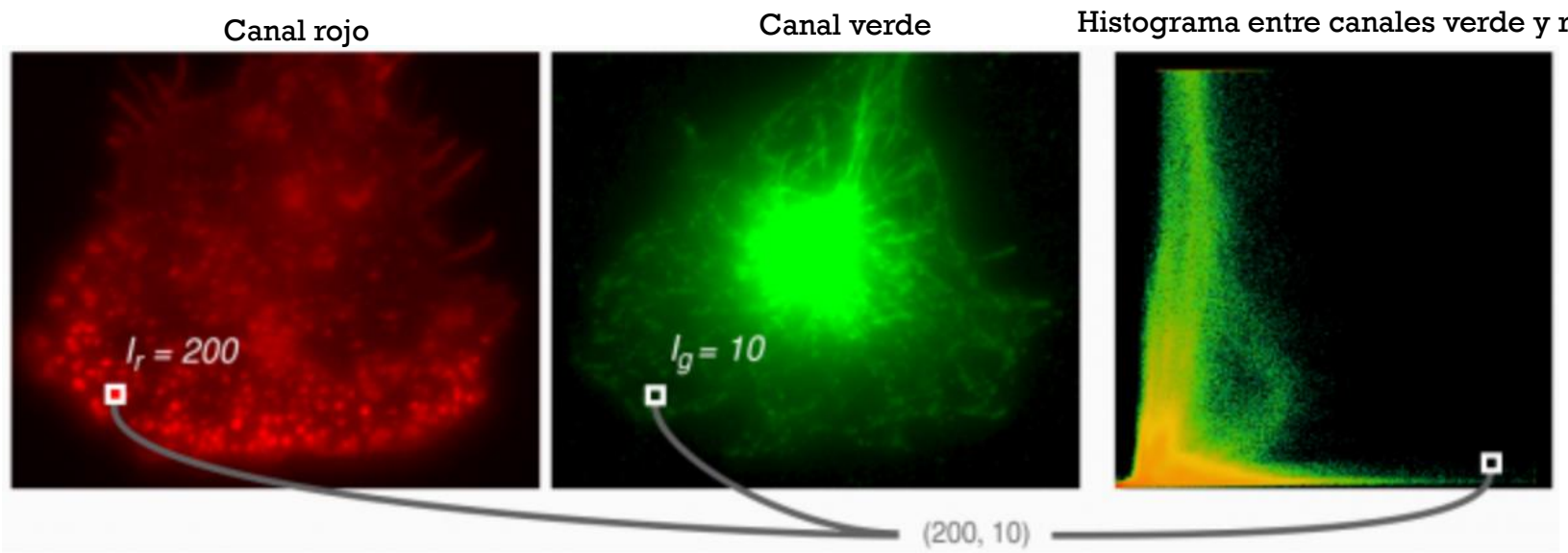
Imagen original (a), histograma luego de aplicar un ligero aumento de contraste (b) y luego de una ligera disminución de contraste (c).



Cambios en el histograma a medida que disminuye el rango dinámico: Cuantización de color.

HISTOGRAMAS

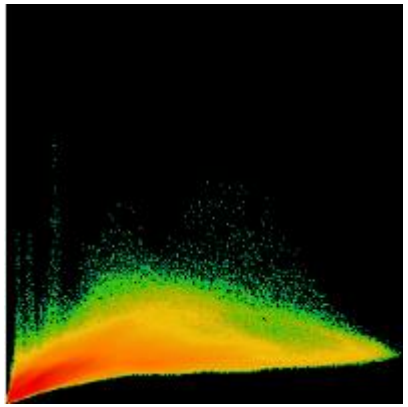
También podemos calcular histogramas N-Dimensionales, por ejemplo entre dos canales de una misma imagen.



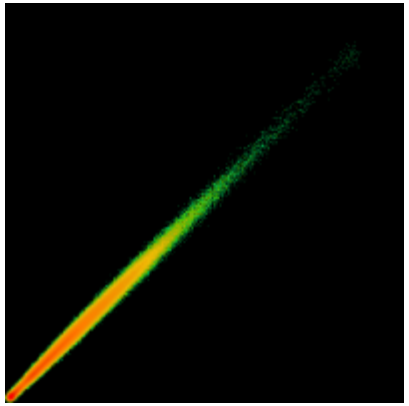
En estos casos se cuentan pares de ocurrencias: Si en la posición (30, 20) de la imagen la intensidad de los canales R y G, respectivamente, fue 200 y 10 acumulo 1 en la posición (200, 10) del histograma.

HISTOGRAMAS

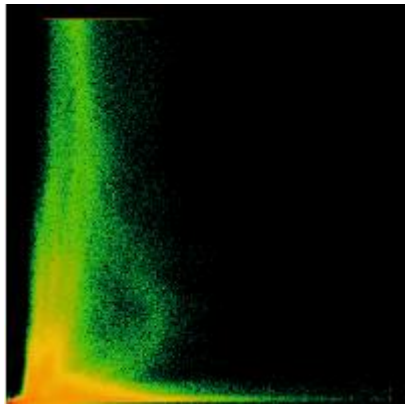
Muchas veces un histograma 2D puede aportar información importante que no es fácil de ver en la imagen original:



Podemos ver como el rango dinámico del canal representado en el eje X es mayor al rango del eje Y



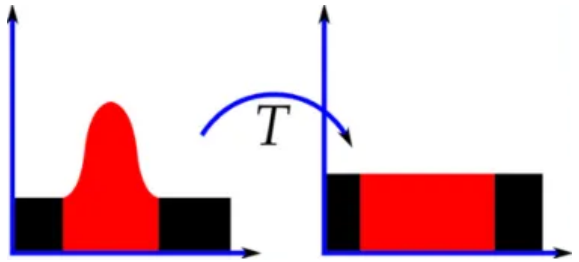
Alta correlación entre los valores de intensidad de ambos canales. Podríamos prescindir de canales que no aportan información.



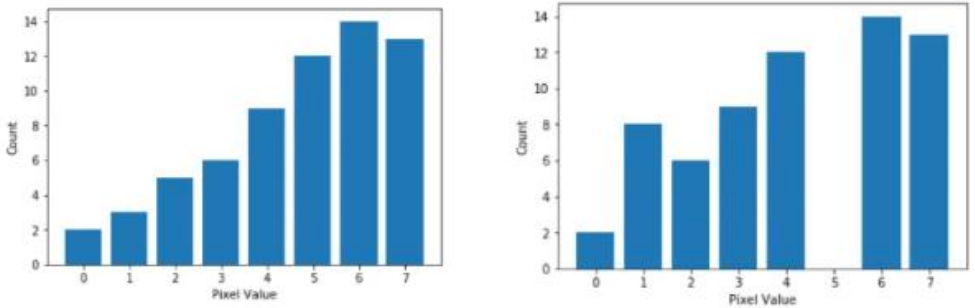
Se observa un efecto de saturación de color en el canal representado por el eje Y

ECUALIZACIÓN DE HISTOGRAMAS

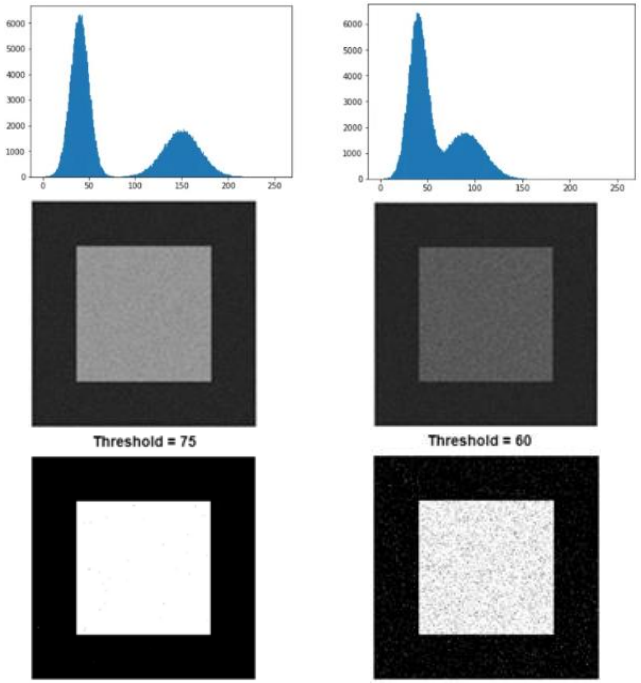
- La idea es "estirar" el histograma para "rellenar" el rango dinámico y a la vez mantener el histograma lo más uniforme posible.
  - Con esto logramos una imagen con mejor contraste y gran variedad de niveles de gris
- ¿Cómo lo hacemos?...utilizando la distribución acumulativa (integral del histograma)
  - $\boldsymbol{c_k = T(r_k) = (L-1)\sum_{i=0}^{k} p_r(r_i) = \frac{(L-1)}{N}\sum_{i=0}^{k} h(i)}$
  - N: *número de píxels en la imagen*
  - I: $[0,255]$ *(para una imagen de 8 bits)*
  - $p_r(r_i) = \frac{n_j}{N}$: *Probabilidad de ocurrencia del valor de intensidad* r_i *en la imagen*
  - $h(I)$: *Histograma original de la imagen*
- imagen_ecualizada = cv2.equalizeHist(imagen_original)

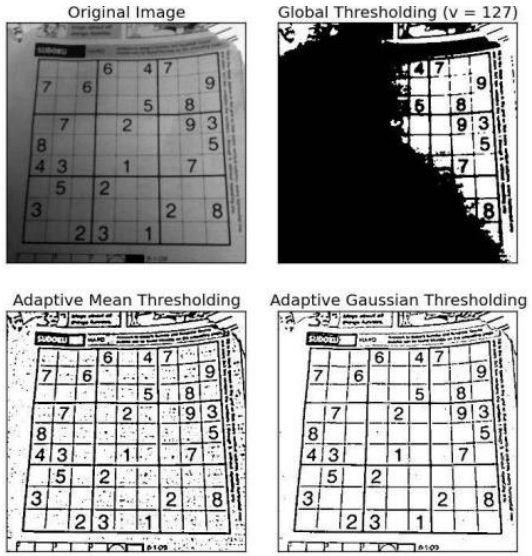


r_k	n_k	$p_r(r_k)$	s_k	Redondeo
0	2	0,03	0,21	0
1	3	0,05	0,56	1
2	5	0,08	1,12	1
3	6	0,09	1,75	2
4	9	0,14	2,73	3
5	12	0,19	4,06	4
6	14	0,22	5,60	6
7	13	0,20	7,00	7



Original → Ecualizada





BINARIZACIÓN

- **Métodos globales**
  1. **Umbral fijo**
     - $g(x) = \begin{cases} 1 & si\ f(x) \geq T \\ 0 & si\ f(x) < T \end{cases}$
  2. **Método Otsu**
     - Calcula el valor de umbral de forma que la dispersión dentro de cada segmento sea lo más pequeña posible, pero al mismo tiempo la dispersión sea lo más alta posible entre segmentos diferentes

- **Métodos locales**
  1. **Mediana**

     $$\boldsymbol{pixel} = (\boldsymbol{pixel} > \boldsymbol{mediana} - \boldsymbol{c})?\ \boldsymbol{objeto}: \boldsymbol{background}$$

     $c: por\ defecto\ vale\ 0$
  1. **Niblack**
     - Basado en la media y varianza local sobre una ventana de tamaño bxb alrededor del píxel

     $$\boldsymbol{pixel} = (\boldsymbol{pixel} > \boldsymbol{media} + \boldsymbol{k} * \boldsymbol{desvio\_std} - \boldsymbol{c})?\ \boldsymbol{objeto}: \boldsymbol{background}$$

     $k: por\ defecto\ 0.2\ para\ objetos\ claros, -0.2\ para\ objetos\ oscuros$
     $c: por\ defecto\ (algoritmo\ original) vale\ 0$
  2. **Sauvola**
     - Variación del método anterior

     $$\boldsymbol{pixel} = (\boldsymbol{pixel} > \boldsymbol{media} * (\boldsymbol{1} + \boldsymbol{k}\ * (\boldsymbol{desvio\_std}/\boldsymbol{r} - \boldsymbol{1})))?\ \boldsymbol{objeto}: \boldsymbol{background}$$

     $k: por\ defecto\ 0.5$
     $r: por\ defecto\ 128$
  3. **Bernsen**
     - El método original usa una ventana circular. Precisa un parámetro de contraste de entrada.

     $\boldsymbol{if}(\boldsymbol{contraste\ local} < \boldsymbol{contraste\ ingresado})$
     $\boldsymbol{pixel} = (\boldsymbol{gris\ medio} \geq \boldsymbol{128})?\ \boldsymbol{objeto}: \boldsymbol{background}$
     $\boldsymbol{else}$
     $\boldsymbol{pixel} = (\boldsymbol{pixel} \geq \boldsymbol{gris\ medio})?\ \boldsymbol{objeto}: \boldsymbol{background}$
  4. **Más..**

BINARIZACIÓN: OTSU

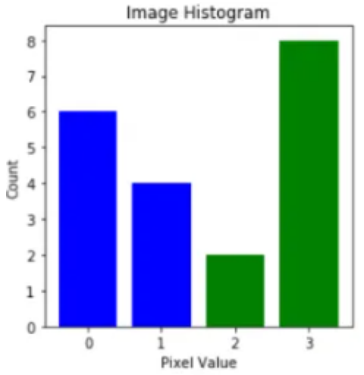
- Método de Otsu (creado por Nobuyuki Otsu)
  - Presupone que el histograma es bimodal y hay una relación de contraste razonable entre fondo y objetos.
  - Busca un valor de umbral que minimice la varianza ponderada dentro de la clase a la vez que maximice la varianza entre clases
  - Digamos que limitamos un histograma a un valor "t". Entonces tendremos dos regiones a izquierda y derecha de "t" con varianzas σ_0^2 y σ_1^2. Entonces la varianza ponderada vendrá dada por:

$$\boldsymbol{\sigma_w^2(t) = w_0(t)\sigma_0^2(t) + w_1(t)\sigma_1^2(t)}$$

$w_0(t)$ y $w_1(t)$ son los pesos dados a cada clase (proporción de píxeles totales en la clase respecto de los píxeles totales en la imagen).

El problema se reduce a encontrar el mínimo de $\boldsymbol{\sigma_w^2(t)}$. Esto es equivalente a encontrar el máximo de la varianza interclúster:

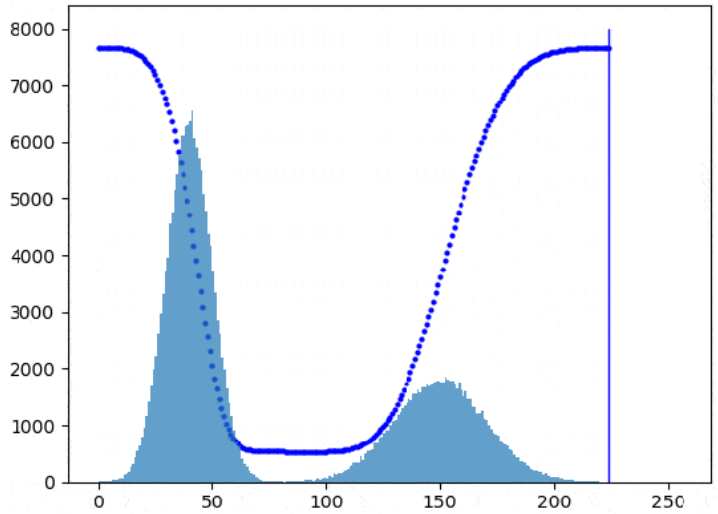
$$\boldsymbol{\sigma_B^2(t) = w_0(t)w_1(t)(\mu_0(t) - \mu_1(t))^2}$$



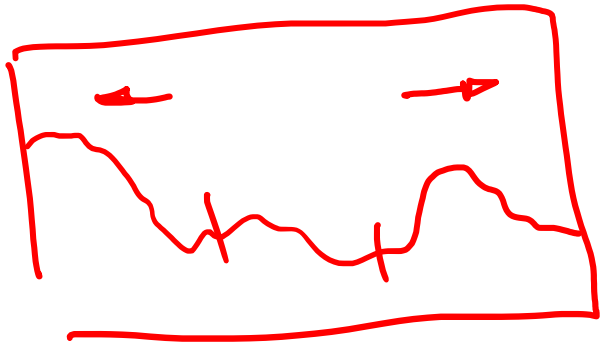
$$\boldsymbol{w_0 = \frac{pix\_región}{pix\_totales} = \frac{6+4}{20} = 0.5}$$
$$\boldsymbol{\mu_0 = \frac{Suma\ pesada\ int.}{pix\_región} = \frac{(0*6)+(1*4)}{10} = 0,4}$$
$$\boldsymbol{\sigma_0^2 = \frac{\left((0-0,4)^2*6\right)+\left((1-0,4)^2*4\right)}{10} = 0,24}$$

$$\boldsymbol{w_1 = \frac{2+8}{20} = 0.5}$$
$$\boldsymbol{\mu_1 = \frac{(2*2)+(3*8)}{10} = 2,8}$$
$$\boldsymbol{\sigma_1^2 = \frac{\left((2-2,8)^2*2\right)+\left((3-2,8)^2*8\right)}{10} = 0,16}$$

$$\boldsymbol{\sigma_w^2 = w_0\sigma_0^2 + w_1\sigma_1^2 = 0,5*0,24 + 0,5*0,16 = 0,2}$$



TP1

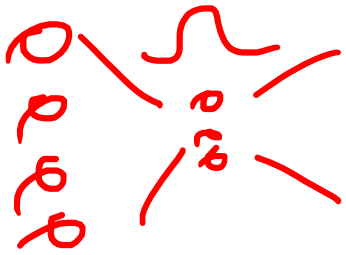


- Parte 1 (imágenes en `/white_patch` y `/coord_cromaticas`):

1. Implementar el algoritmo de pasaje a coordenadas cromáticas para librarnos de las variaciones de contraste.
2. Implementar el algoritmo White Patch para librarnos de las diferencias de color de iluminación.
3. Mostrar los resultados obtenidos y analizar las posibles fallas (si es que las hay) en el caso de White patch.

- Parte 2:

1. Para las imágenes `img1_tp.png y img2_tp.png` leerlas con OpenCV en escala de grisas y visualizarlas.
2. Elija el numero de bins que crea conveniente y grafique su histograma, compare los histogramas entre si. Explicar lo que se observa, si tuviera que entrenar un modelo de clasificación/detección de imágenes, considera que puede ser de utilidad tomar como 'features' a los histogramas?
3. Para la imagen `segmentacion.png` analice el histograma de los canales RGB. Segmente algunos de los elementos presentes en la imagen (agua, cielo, tierra) y muestre, aplicando mascaras, las regiones en imágenes separadas.



Resultado luego de aplicar corrección por coordenadas cromáticas