

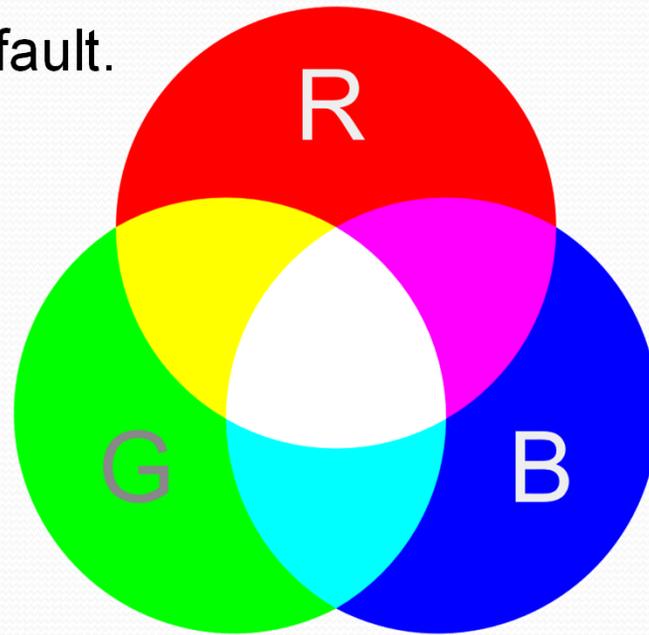
# Histogramas y balance de color

# Temario

- Espacios de color.
- Balance de color.
- Histogramas para imágenes en tonos de gris.
- Histogramas para imágenes a color.

# RGB

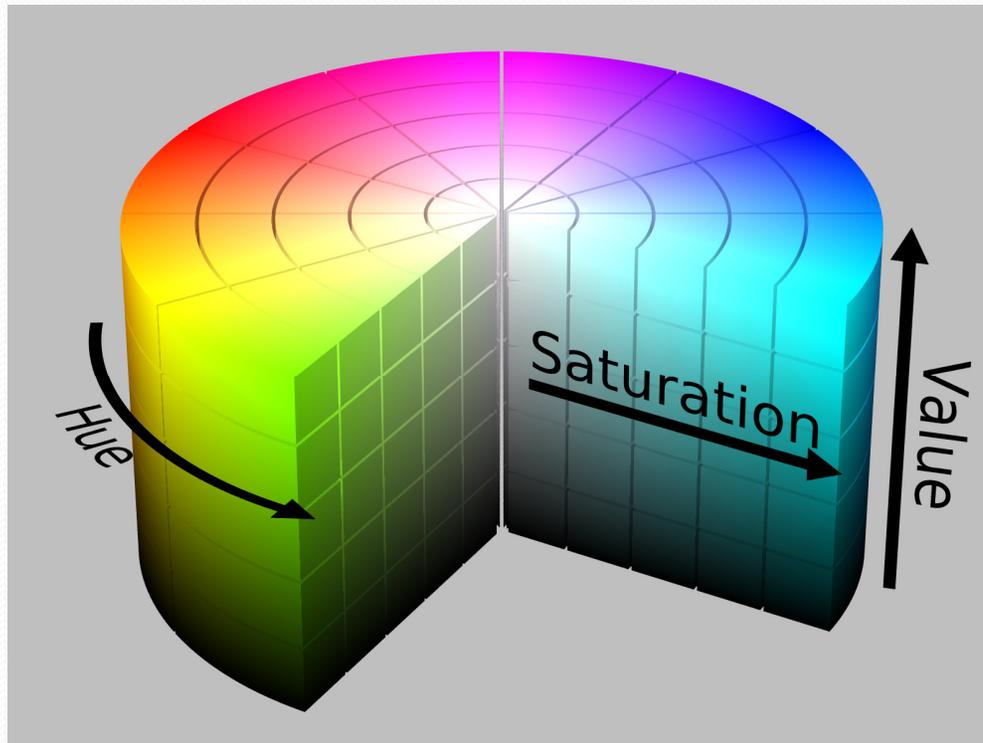
- Espacio de color por default.



Fuente: By Immanuelle - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=132469198>

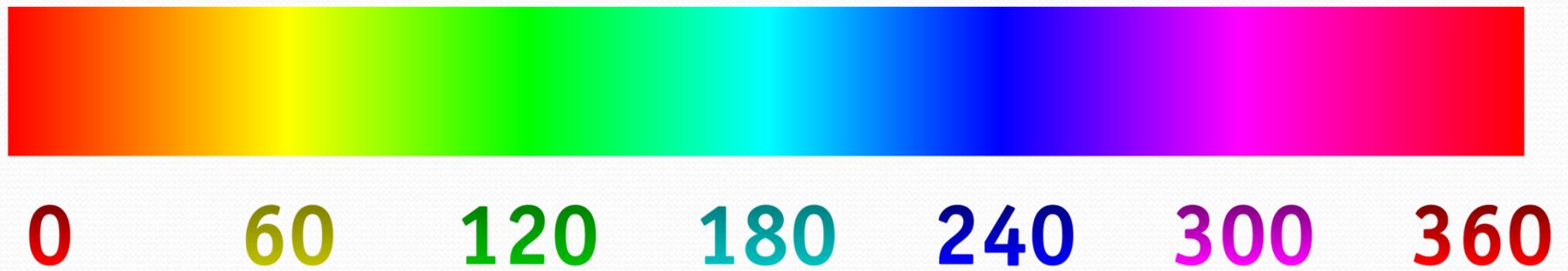
# HSV

- Hue, Saturation, Value.



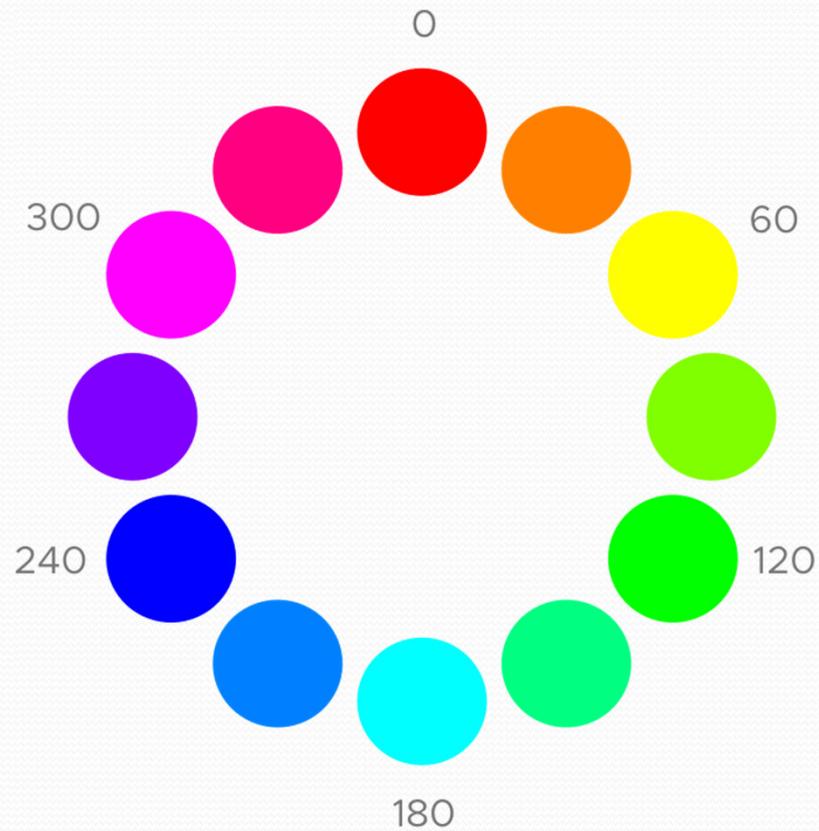
HSV\_color\_solid\_cylinder.png: SharkDderivative work: SharkD Talk - HSV\_color\_solid\_cylinder.png, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=9801673>

# Hue



By Kalan - Own work, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=1609109>

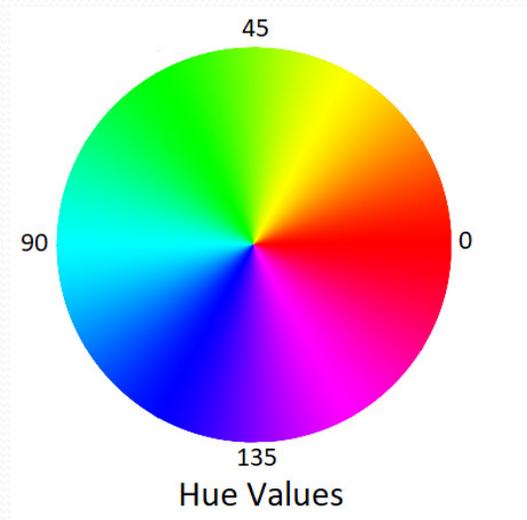
# Rueda de hues



<https://www.pinclipart.com/maxpin/bimmbJ/>

# HSV en OpenCV

- Nota: el rango del canal hue es entre 0 y 179.

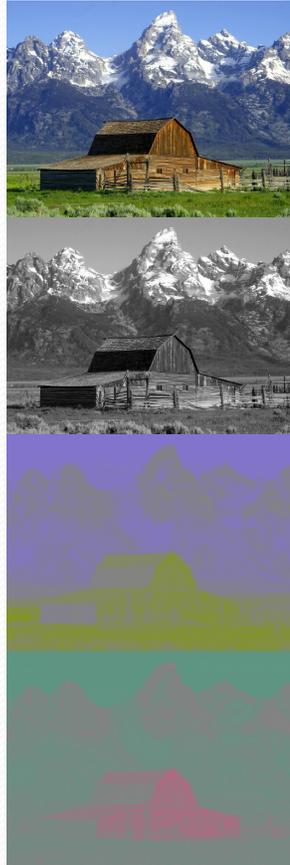


Fuente: <https://answers.opencv.org/question/184711/select-hsv-hue-from-30-to-30-in-python/>

# YCbCr

- Componentes:  $Y$  (luminancia o luma),  $Cb$  (diferencia de azul),  $Cr$  (diferencia de rojo).
- Se usa en televisión digital y para comprimir imágenes (p.e. JPEG).

# YCbCr

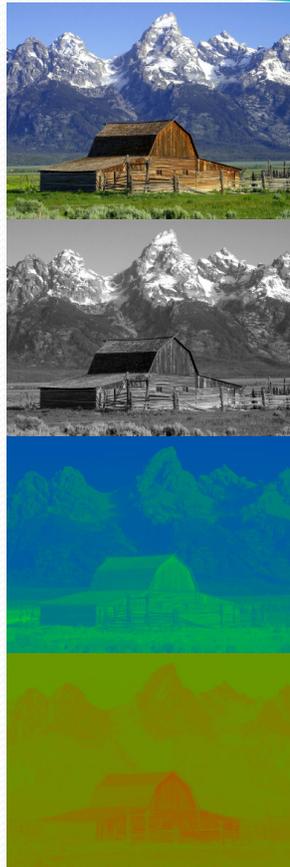


Fuente: By Mike1024 - Based on the (public domain) photo Image:Barns grand tetons.jpg. Code above and resulting output by Mike1024., Public Domain, <https://commons.wikimedia.org/w/index.php?curid=1493370>

# YUV

- Componentes:  $Y$  (luminancia o luma),  $U$  (proyección azul),  $V$  (proyección roja).
- Se usa en televisión analógica PAL.

# YUV

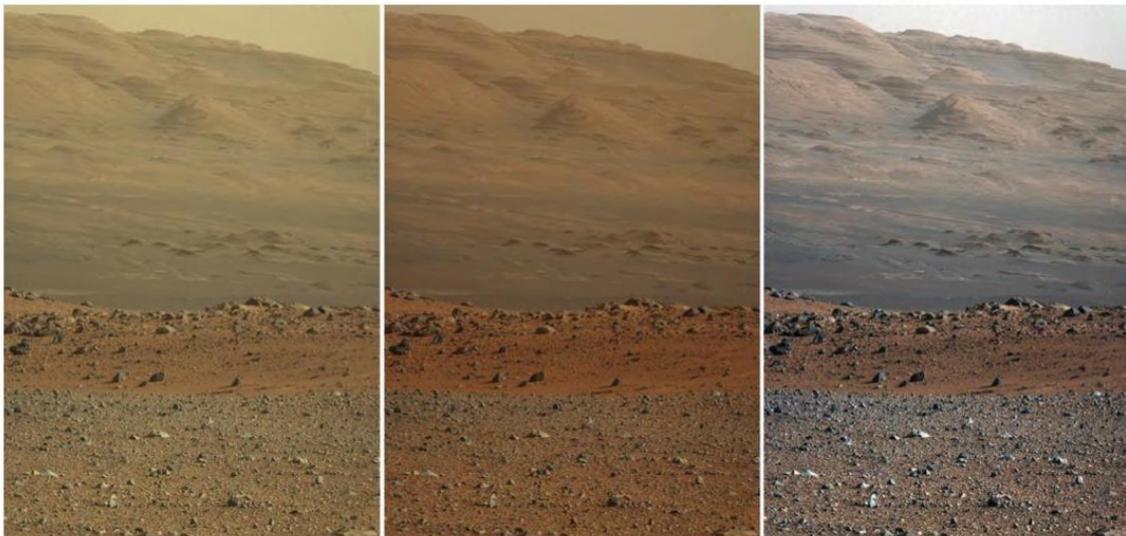


Fuente: By User:Brianski - Concept from en:Image:YUV\_components.jpg, original public domain image at en:Image:Barns\_grand\_tetons.jpg, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=2792866>

# Balance de color

- **Balance de color.** Ajuste global de las intensidades de los colores.
- El objetivo es representar correctamente colores específicos, particularmente colores neutros como el blanco o el gris.
- A veces se denomina **balance de grises**, **balance neutro** o **balance de blancos**.

# Ejemplo



**Unprocessed Color (JPL Web site)**  
(raw data from Mars, uncalibrated)

**"Natural" Color**  
(uses calibrated data)

**"White Balanced" Color**  
(Assumes something in the scene is white)

Fuente: By NASA/JPL-Caltech/MSSS - <http://photojournal.jpl.nasa.gov/catalog/PIA16800> (image link), Public Domain, <https://commons.wikimedia.org/w/index.php?curid=25188912>

# ColorChecker



Fuente: [https://en.wikipedia.org/wiki/File:Color\\_Checker.pdf](https://en.wikipedia.org/wiki/File:Color_Checker.pdf)

# Mejora de contraste



Fuente: By original Phillip Capper, modified by User:Konstable - modified Hawkes Bay NZ.jpg, CC BY 2.0,  
<https://commons.wikimedia.org/w/index.php?curid=855363>

# Ideas importantes

- Las imágenes típicas son grises en promedio; esto se puede utilizar para detectar distorsiones.
- Las diferencias más grandes son más visibles, por lo que utilizar todo el rango de intensidad mejora la visibilidad.
- A menudo es más fácil trabajar en un espacio de color que no sea RGB.

# Balance de color mediante ajuste lineal

- Idea: multiplicar los valores  $R$ ,  $G$  y  $B$  por distintas constantes.

- $$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \alpha_r & 0 & 0 \\ 0 & \alpha_g & 0 \\ 0 & 0 & \alpha_b \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

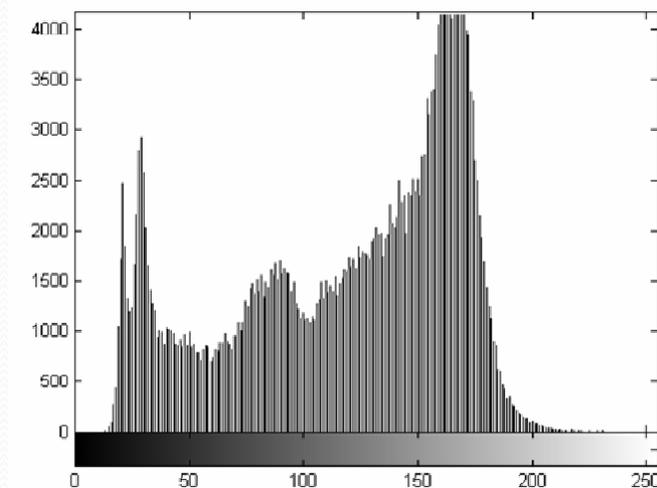
- ¿Cómo elegir las constantes?
  - Supuesto del “mundo gris”: el valor medio debería ser gris.
  - Balance de blancos: elegir una referencia como color blanco o gris.
  - Es mejor balancear el RGB de la cámara (lineal) que el RGB de la pantalla (no lineal).

# Ejemplo de balance de color

- Ver <https://jmanansala.medium.com/image-processing-with-python-color-correction-using-white-balancing-6c6c749886de>

# Histograma

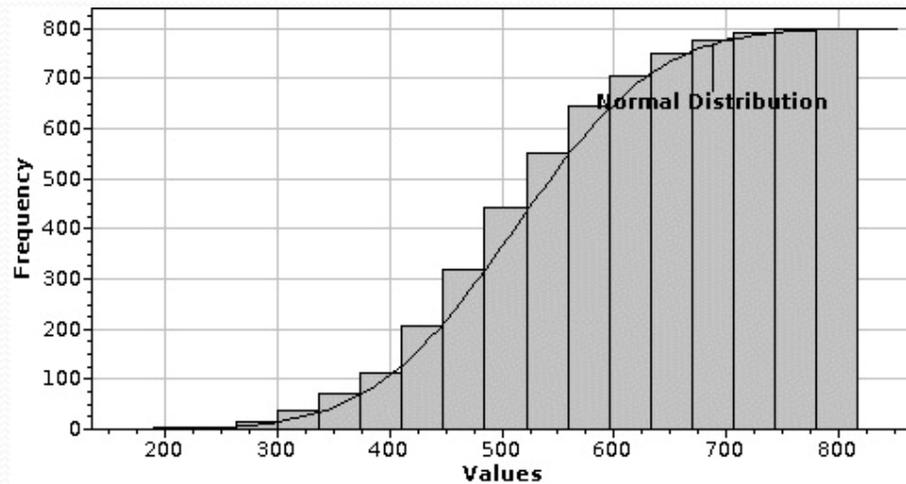
- **Histograma.** Conteo de cuántos pixeles tienen cada valor.
- Python: `cv2.calcHist()`



Fuente: [https://www.researchgate.net/figure/Histogram-Analysis-of-grayscale-image\\_fig2\\_228744430](https://www.researchgate.net/figure/Histogram-Analysis-of-grayscale-image_fig2_228744430)

# Histograma acumulativo

- **Histograma acumulativo.** Conteo del número de pixeles menores o igual a cada valor.



Fuente: <https://mipav.cit.nih.gov/pubwiki/images/1/19/CumulativeHistogramSample.jpg>

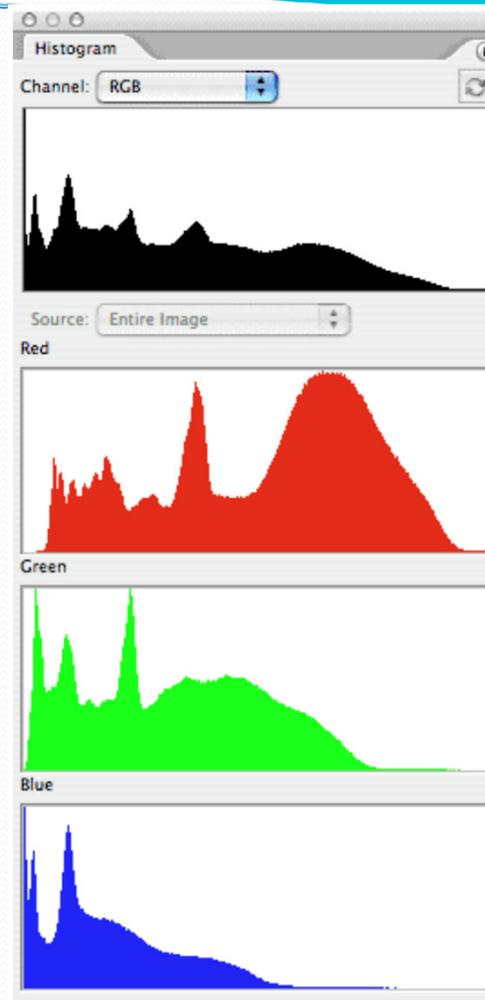
# Ejemplo

- $A = [1\ 1\ 2\ 3\ 3\ 3\ 5\ 6]$
- $H = [2\ 1\ 3\ 0\ 1\ 1]$
- $C = [2\ 3\ 6\ 6\ 7\ 8]$

# Histogramas de imágenes a color

- Dos tipos de histogramas para imágenes a color:
- Histograma de intensidad (luminancia).
- Histogramas individuales por cada canal de color.

# Ejemplo



Fuente: <https://kenrockwell.com/tech/yrgb.htm>

# Histogramas de intensidad

- El histograma de intensidad o luminancia,  $Y$ , es el histograma de la imagen a color en escala de grises.
- La imagen en escala de grises se obtiene calculando la luminancia.
- La luminancia se puede calcular mediante un promedio:
- $$Y = \frac{R+G+B}{3}$$
- Los humanos percibimos el rojo y el verde como más brillantes que el azul.

# Histogramas de intensidad

- En la práctica se usa un promedio ponderado:
- $Y = w_R \cdot R + w_G \cdot G + w_B \cdot B$
- Los siguientes pesos se desarrollaron para codificar señales analógicas de televisión en color:
- $w_R = 0.299, w_G = 0.587, w_B = 0.144$
- Los pesos recomendados en el standard ITU-BT.709 para la codificación de color digital son:
- $w_R = 0.2126, w_G = 0.7152, w_B = 0.0722$

# Histogramas

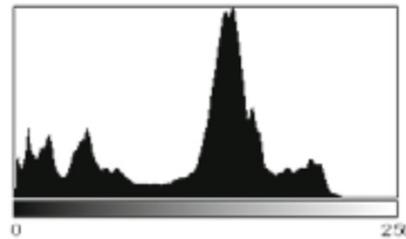
- ¿Para qué sirve el histograma de una imagen?
- Da información sobre el contraste y rango dinámico de una imagen.
- Ofrece información sobre la exposición de la imagen.
- El histograma se puede manipular para mejorar una imagen.

# Ejemplos de exposición

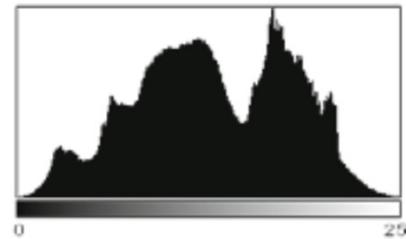
## 3 HISTOGRAMS AND IMAGE STATISTICS

**Fig. 3.6**

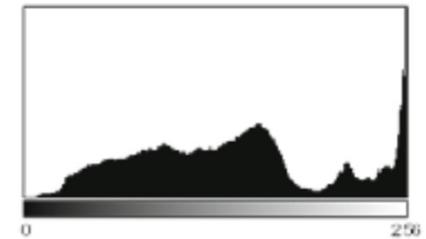
Exposure errors are readily apparent in histograms. Underexposed (a), properly exposed (b), and overexposed (c) photographs.



(a)



(b)

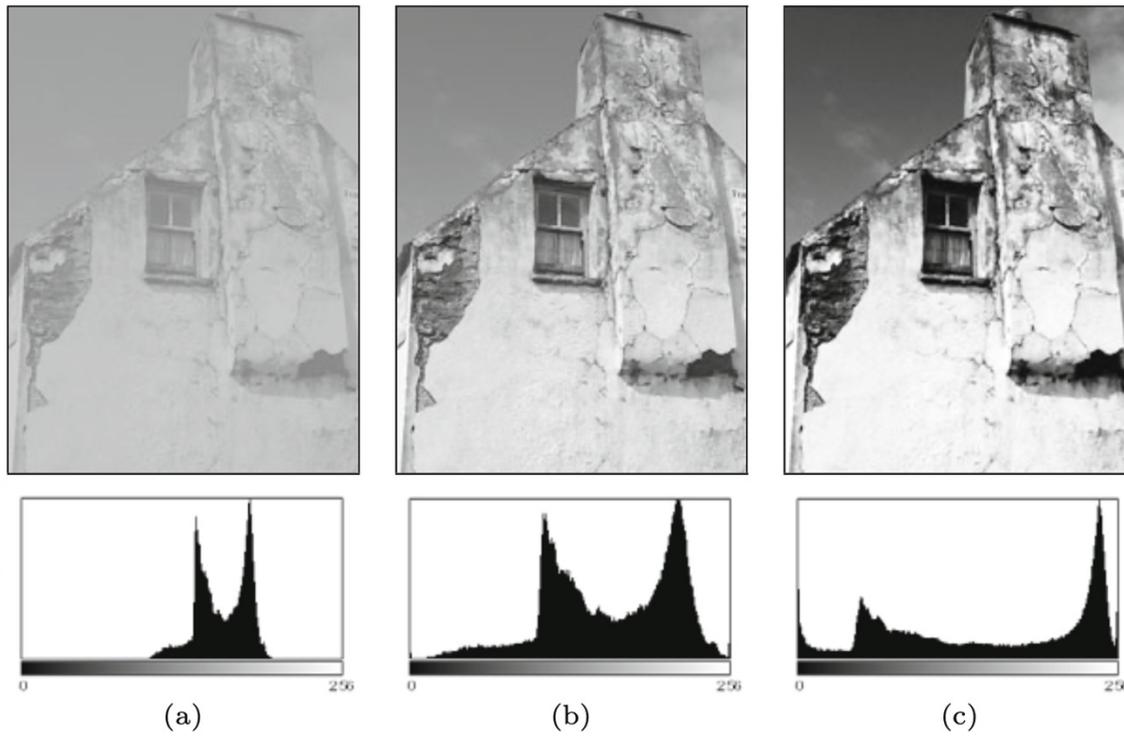


(c)

# Contraste

- El contraste es el rango de valores de intensidad utilizados por una imagen.
- Es la diferencia entre los valores máximo y mínimo de los píxeles de la imagen ( $a = a_{max} - a_{min}$ ).
- Una imagen de contraste completo (*full-contrast*) es la que hace un uso efectivo de todo el rango de valores de intensidad disponibles, es decir  $a_{min} = 0$  y  $a_{max} = 255$  (negro a blanco).
- Con esta definición, el contraste de la imagen se puede leer fácilmente directamente desde el histograma.

# Contraste



## 3.2 INTERPRETING HISTOGRAMS

**Fig. 3.7**

How changes in contrast affect the histogram: low contrast (a), normal contrast (b), high contrast (c).

# Rango dinámico

- **Rango dinámico.** La relación (ratio o razón) entre los valores más grandes y más pequeños de intensidad.
- Se mide como una relación, en decibeles o en stops.

# Rango dinámico

- Ejemplo: suponer que para una imagen dada de 8 bits  $a_{low} = 10$  y  $a_{high} = 250$ .
- El rango dinámico es:
- Como relación (ratio)  $DR = \frac{250}{10} = 25 = 25:1$ .
- En decibeles  $DR = 10 \times \log_{10} \left( \frac{250}{10} \right) = 13.98 \text{ dB}$ .
- En stops  $DR = \log_2 \left( \frac{250}{10} \right) = 4.64 \text{ stops}$ .

# Rango dinámico

- Fórmulas generales:
- Relación:  $\frac{a_{high}}{\max(1, a_{low})}$
- Decibeles:  $10 \times \log_{10} \left( \frac{a_{high}}{\max(1, a_{low})} \right)$  dB
- Stops:  $\log_2 \left( \frac{a_{high}}{\max(1, a_{low})} \right)$  stops
- En el denominador se usa  $\max(1, a_{low})$  porque es posible que  $a_{low}$  valga 0.
- Los decibeles se emplean en sonido y los stops en fotografía y en imágenes digitales.

# Rango dinámico alto

- High Dynamic Range (HDR).
- La mayoría de los dispositivos y imágenes digitales utilizan 8 bits por canal.
- Esto limita el rango dinámico del dispositivo a 256 niveles (255:1).
- El ojo humano puede adaptarse a condiciones de iluminación que varían en diez órdenes de magnitud ( $10^{10}: 1$ ).
- Cuando se toma fotografías de una escena del mundo real, las regiones brillantes pueden estar sobreexpuestas, mientras que las oscuras pueden estar subexpuestas.

# Rango dinámico alto



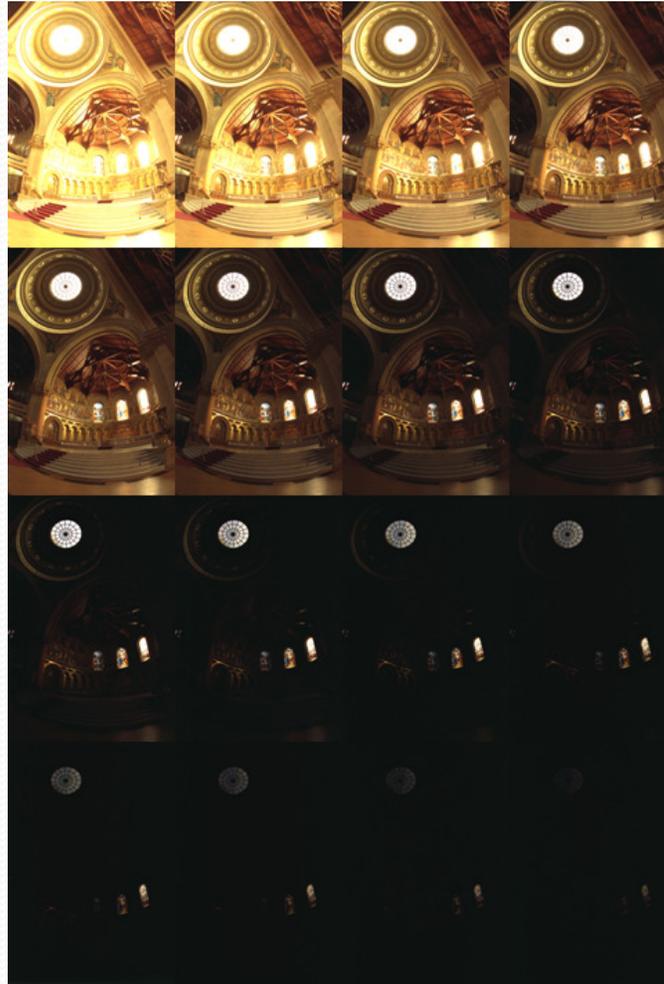
Fuente: <https://qph.cf2.quoracdn.net/main-qimg-e8ec2727cbd9cba0b290fbd0a1cfa411.webp>

# Rango dinámico alto

- Las imágenes HDR utilizan más de 8 bits por canal (normalmente valores flotantes de 32 bits), lo que permite un rango dinámico mucho más amplio.
- Una imagen HDR se obtiene capturando múltiples fotografías de la misma escena con diferentes exposiciones y luego combinarlos en una sola imagen.
- Para convertir una imagen HDR a una imagen LDR (rango dinámico 255:1) hay dos métodos: mapeo de tonos (tone mapping) y fusión de exposiciones (exposure fusion).

# Ejemplo

Secuencia de exposiciones



Fuente:

[https://docs.opencv.org/4.10.0/d3/db7/tutorial\\_hdr\\_imaging.html](https://docs.opencv.org/4.10.0/d3/db7/tutorial_hdr_imaging.html)

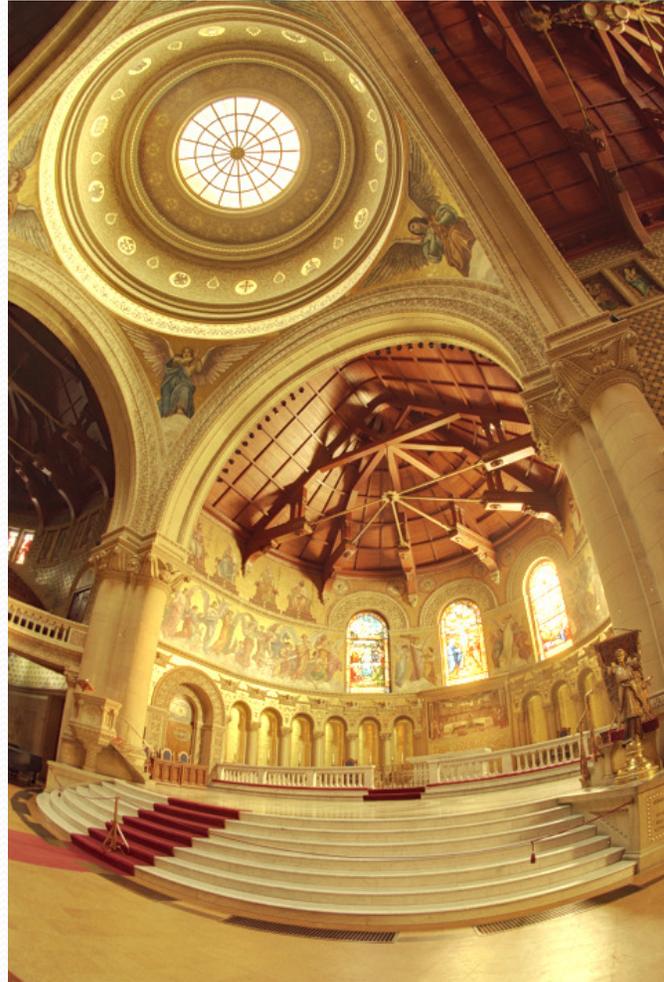
# Ejemplo

Mapeo de tonos



# Ejemplo

Fusión de exposiciones



# Ecualización del histograma

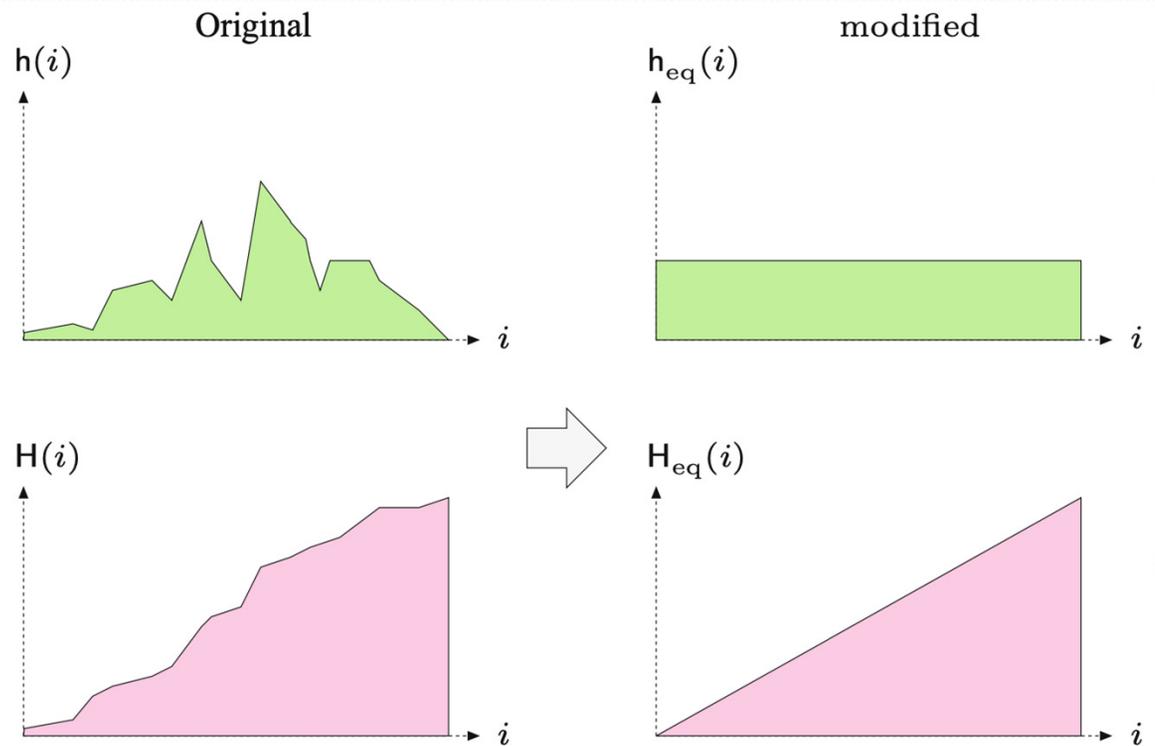
- El objetivo es encontrar y aplicar una operación de pixel para modificar una imagen de tal forma que el histograma de la imagen modificada se aproxime a una distribución *uniforme*.

# Ecuación del histograma

## 4 POINT OPERATIONS

**Fig. 4.8**

Histogram equalization. The idea is to find and apply a point operation to the image (with original histogram  $h$ ) such that the histogram  $h_{eq}$  of the modified image approximates a *uniform* distribution (top). The cumulative target histogram  $H_{eq}$  must thus be approximately wedge-shaped (bottom).



# Ecuación del histograma en OpenCV

**# File: histogram\_equalization.py**

```
import cv2 as cv
```

```
import numpy as np
```

```
image = cv.imread('Hawkes_Bay.jpg', cv.IMREAD_GRAYSCALE)
```

```
image_equ = cv.equalizeHist(image)
```

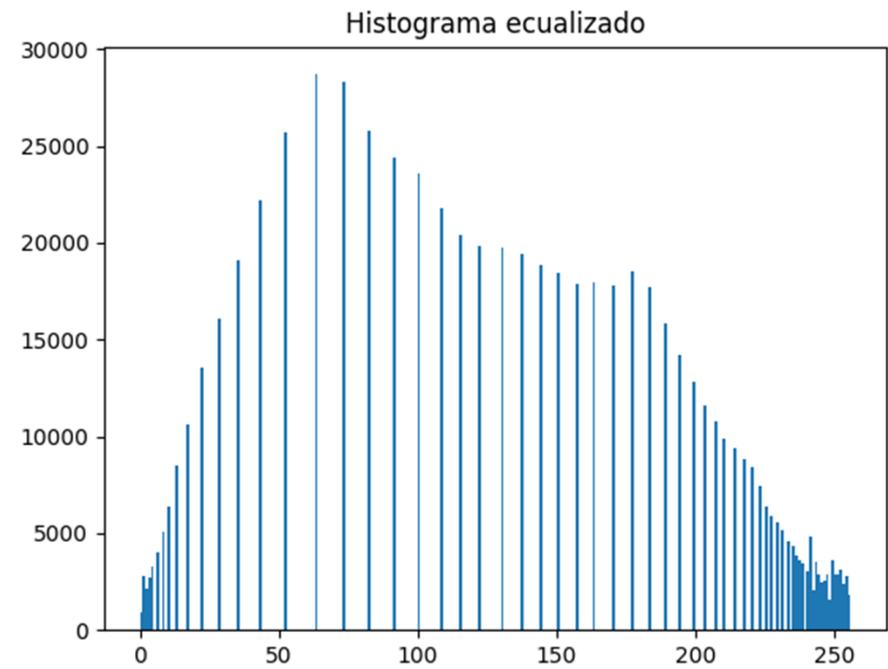
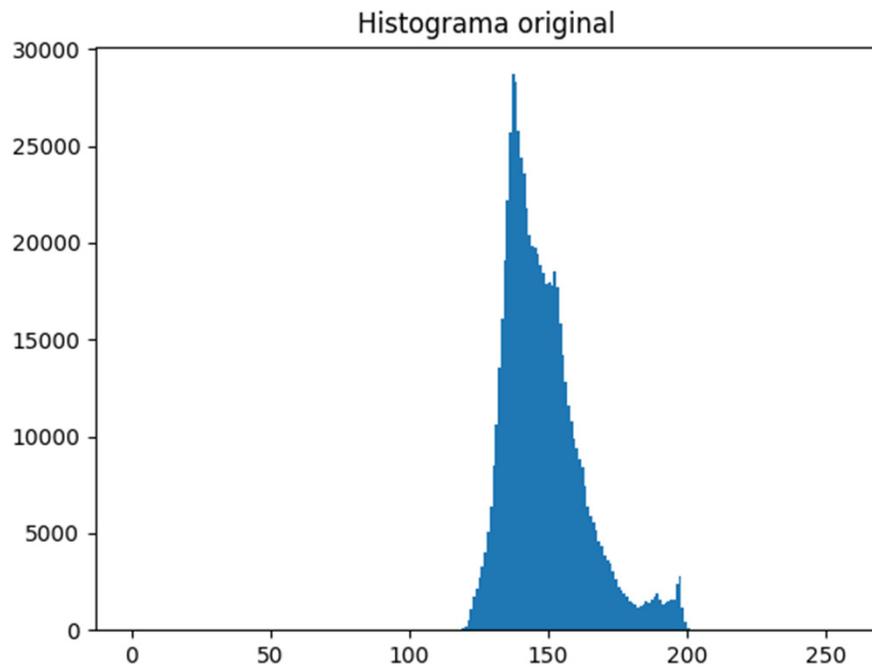
```
result = np.hstack((image, image_equ)) # stacking images side-by-side
```

```
cv.imwrite('Hawkes_Bay_Equ.png', result)
```

# Resultado



# Resultado



# CLAHE

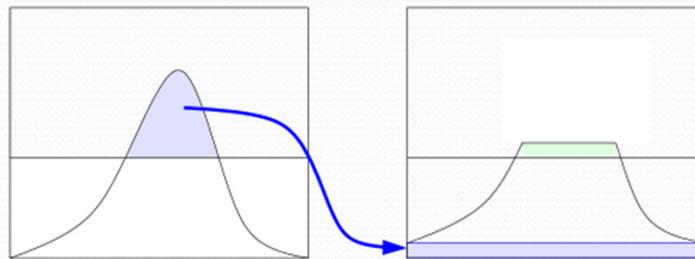
- Ecuación de histograma adaptativo limitado de contraste (Contrast Limited Adaptive Histogram Equalization).
- La ecuación del histograma considera el contraste global de la imagen.
- Esto puede no ser conveniente en algunas imágenes.

# Ecualización global



# CLAHE

- Divide la imagen en bloques (8 x 8 por default).
- A cada bloque se le aplica la ecualización del histograma.
- A cada bloque se le limita el contraste.
- Si algún bin de histograma se pasa del límite especificado de contraste (40 por default), los pixeles se cortan y se distribuyen uniformemente antes de aplicar la ecualización.



# CLAHE en OpenCV

- Paso 1: Crear un objeto CLAHE (los argumentos son opcionales).  
`clahe = cv.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))`
- Paso 2: Invocar el método `apply`.  
`image_clahe = clahe.apply(image)`

# Ejemplo

**# File: clahe.py**

```
image = cv.imread('tsukuba_gray.png', cv.IMREAD_GRAYSCALE)
clahe = cv.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
image_clahe = clahe.apply(image)
result = np.hstack((image, image_clahe)) # stacking images side-by-side
cv.imwrite('tsukuba_clahe.png', result)
```

# Resultado



# Ecualización del histograma en color

- Aplicar una ecualización a cada canal de una imagen RGB puede distorsionar el color.
- El concepto de ecualización del histograma solo es aplicable a los valores de intensidad de una imagen.
- La solución es convertir la imagen a un espacio en donde la intensidad esté separada de la información del color.
- Por ejemplo YUV o YCbCr.

# Ejemplo: ecualización BGR

**# File: histogram\_equalization\_color.py**

```
b, g, r = cv.split(image)
```

```
equ_b = cv.equalizeHist(b)
```

```
equ_g = cv.equalizeHist(g)
```

```
equ_r = cv.equalizeHist(r)
```

```
image_bgr = cv.merge((equ_b, equ_g, equ_r))
```

# Resultado



# Ejemplo: ecualización YUV

**# File: histogram\_equalization\_color.py**

**# convert the BGR image to YUV format**

```
image_yuv = cv.cvtColor(image, cv.COLOR_BGR2YUV)
```

**# equalize the histogram of the Y channel**

```
image_yuv[:, :, 0] = cv.equalizeHist(image_yuv[:, :, 0])
```

**# convert the YUV image back to BGR format**

```
output = cv.cvtColor(image_yuv, cv.COLOR_YUV2BGR)
```

# Resultado



# Aplicación: histogram backprojection

- **Histogram Backprojection.** Técnica utilizada para encontrar objetos de interés en una imagen.
- El objeto se busca en base a sus colores.
- Para entender esta técnica, se necesita revisar el concepto de histogramas 2D.

# Histogramas 2D

- **Histograma 2D.** Herramienta para encontrar la correlación entre los canales de una imagen.
- Ejemplo: un histograma 2D para los canales rojo y verde, cuenta cuántas veces aparece en la imagen el par  $(r, g)$ , donde  $r \in [0, 255]$  y  $g \in [0, 255]$ .

# Construcción de un histograma 2D

- Ejemplo tomado de <https://theailearner.com/2019/02/12/2d-histogram/>
- Suponer una imagen de 2 bits de 4 x 4 con los siguientes canales rojo y verde:

3	2	0	3
0	2	0	1
1	3	0	2
3	0	2	2

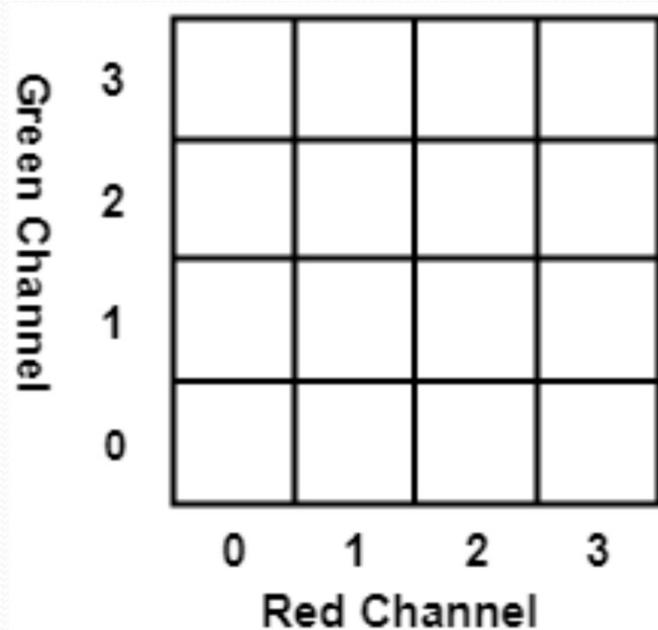
**Red Channel**

0	1	3	2
3	1	2	0
2	3	0	1
0	1	3	1

**Green Channel**

# Construcción de un histograma 2D

- El histograma 2D se guarda en una matriz de 4 x 4:



# Construcción de un histograma 2D

- Se recorre cada posición en los canales contando las parejas de intensidades:

Corresponding Intensity Pairs	Count
(0,0)	1
(0,1)	1
(0,2)	1
● ● ●	
(3,2)	1
(3,3)	1



Green Channel	3	2	0	1	1
	2	1	1	0	1
	1	1	0	4	0
	0	1	1	0	2
		0	1	2	3
		Red Channel			

Fill the 2D histogram with count

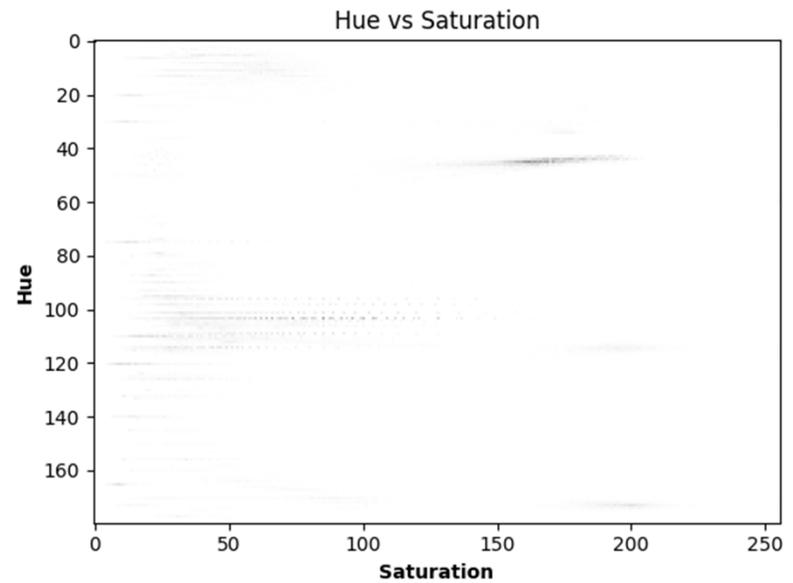
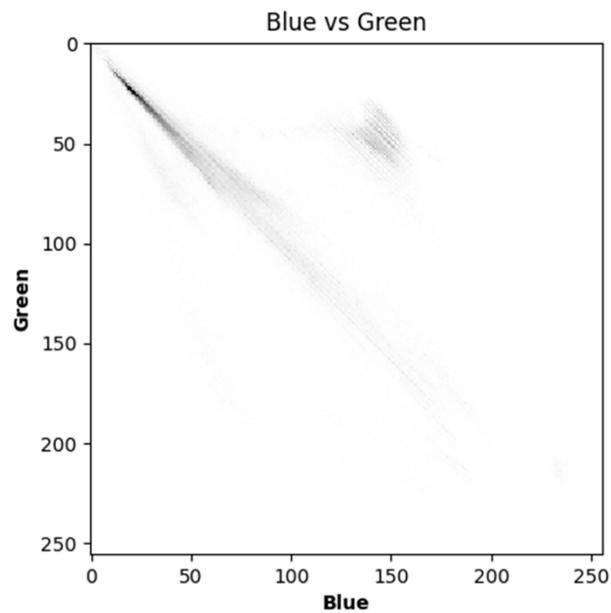
# Ejemplo

- Imagen original:



# Ejemplo

- Histogramas 2D:



# Histogramas 2D en OpenCV

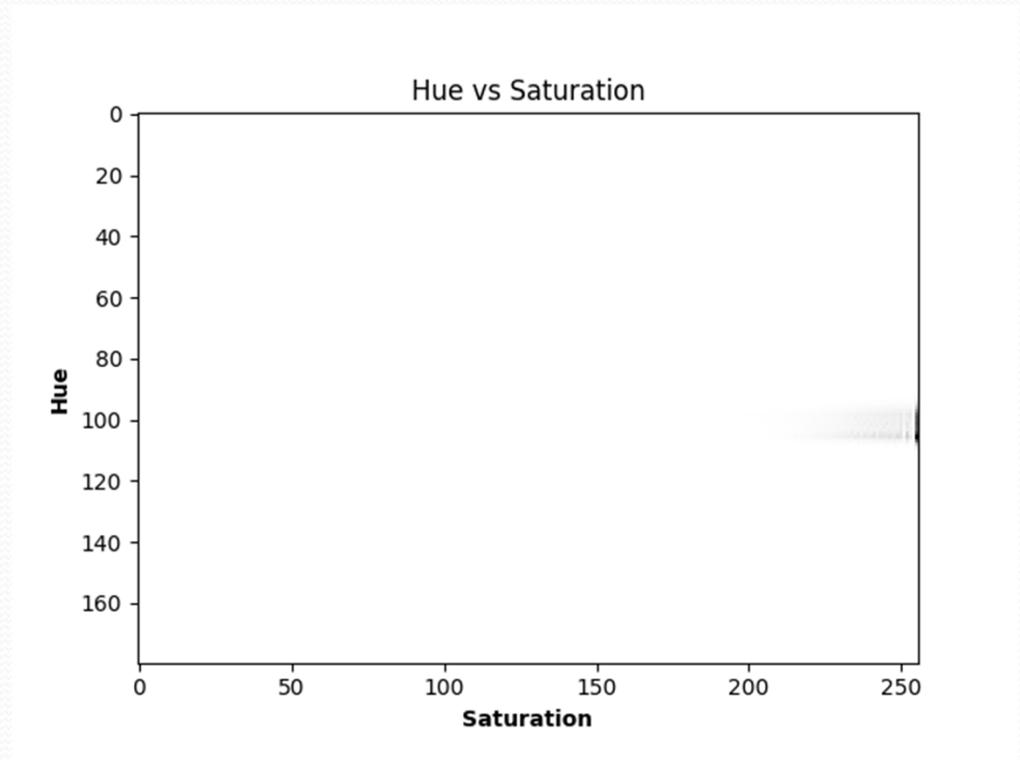
- Se usa la misma función `cv.calcHist()` que se utiliza para los histogramas convencionales.

- Ejemplos:

```
hist_bg = cv.calcHist([image_bgr], channels=[0, 1], mask=None,  
    histSize=[256, 256], ranges=[0, 256, 0, 256])
```

```
hist_hsv = cv.calcHist([image_hsv], channels=[0, 1], mask=None,  
    histSize=[180, 256], ranges=[0, 180, 0, 256])
```

# Ejemplo



# Histogram Backprojection

- Responde a la pregunta: “¿Dónde están los colores en la imagen que pertenecen al objeto que se está buscando?”
- El algoritmo calcula el histograma del objeto y luego lo usa para hacer una proyección de vuelta.
- Lo siguiente está basado en <https://theailearner.com/2019/04/18/histogram-backprojection/>

# Ejemplo



ROI



Target

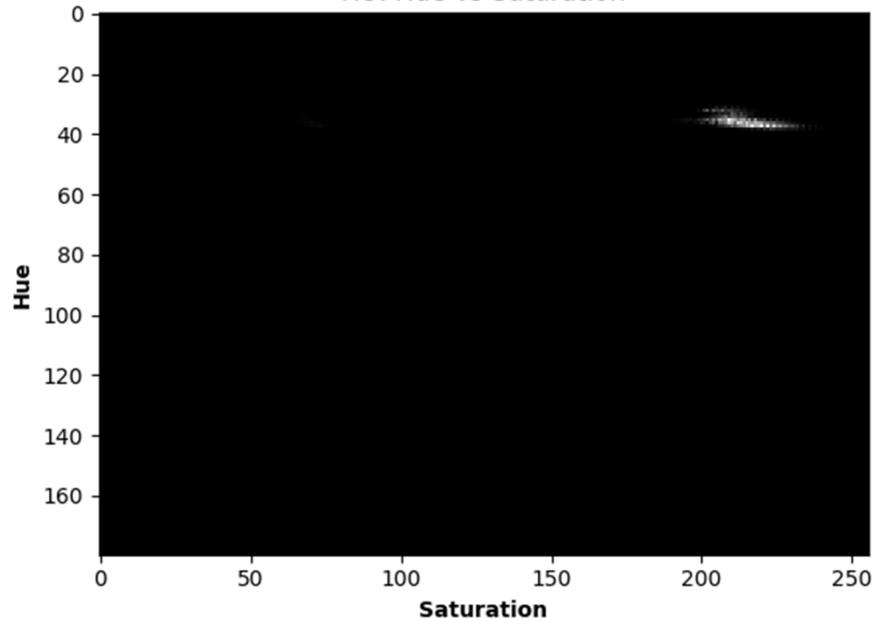


# Primeros pasos

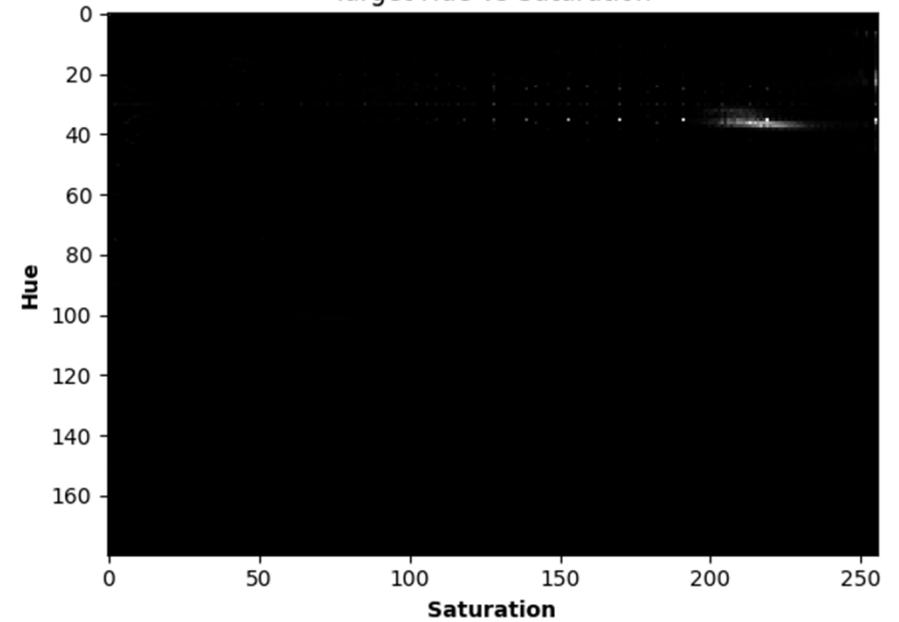
- Leer las imágenes.
- Obtener los histogramas 2D.

# Histogramas 2D

ROI Hue vs Saturation



Target Hue vs Saturation



# Explicación

- La imagen ROI tiene tonos verdes y su histograma está concentrado principalmente en los valores de H y S que representan el verde.
- Se puede decir lo mismo de la imagen destino.

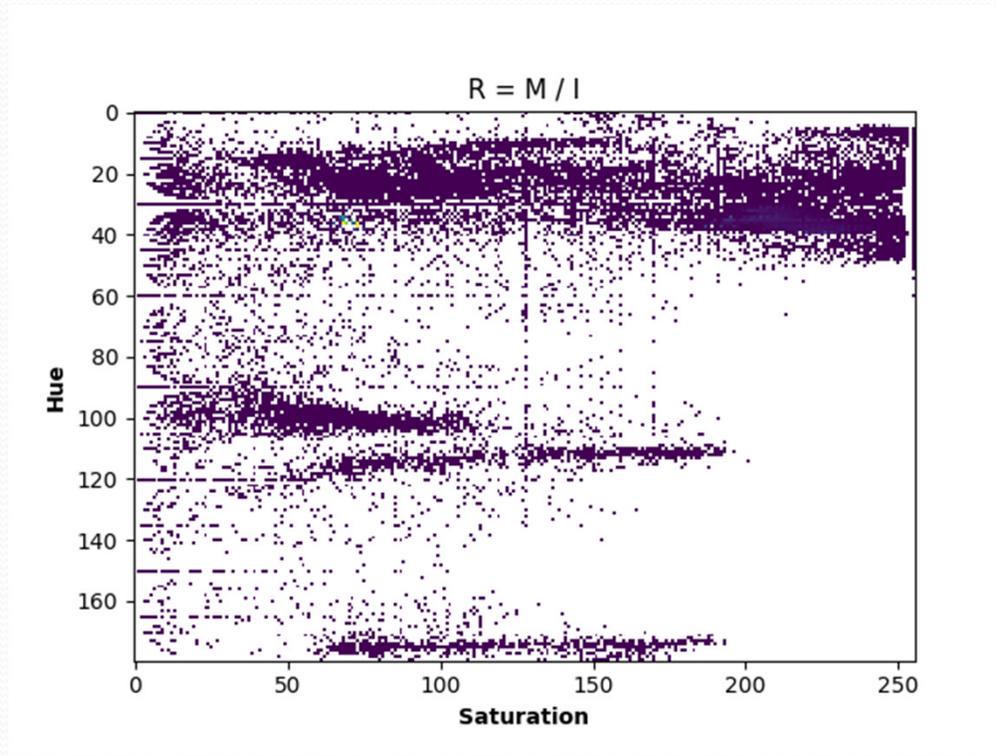
# Proyección de regreso

- Se busca extraer los colores de la imagen ROI en la imagen destino.
- Una solución es dividir  $M$  (el histograma 2D de la imagen ROI) entre  $I$  (el histograma 2D de la imagen destino).

$$R = M / I$$

- Los valores de  $R$  son:
- Mayores que 0 si  $M > 0$  e  $I > 0$
- 0 o nan en otro caso.

# Matriz $R$



# Explicación

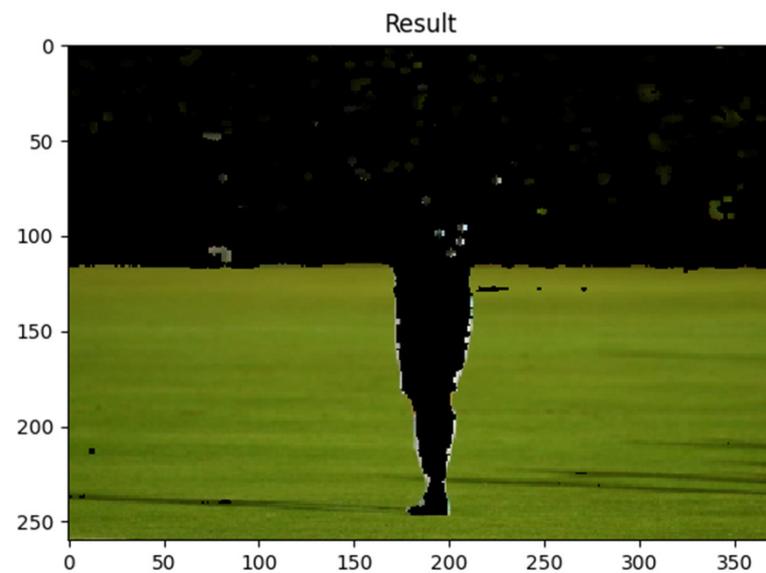
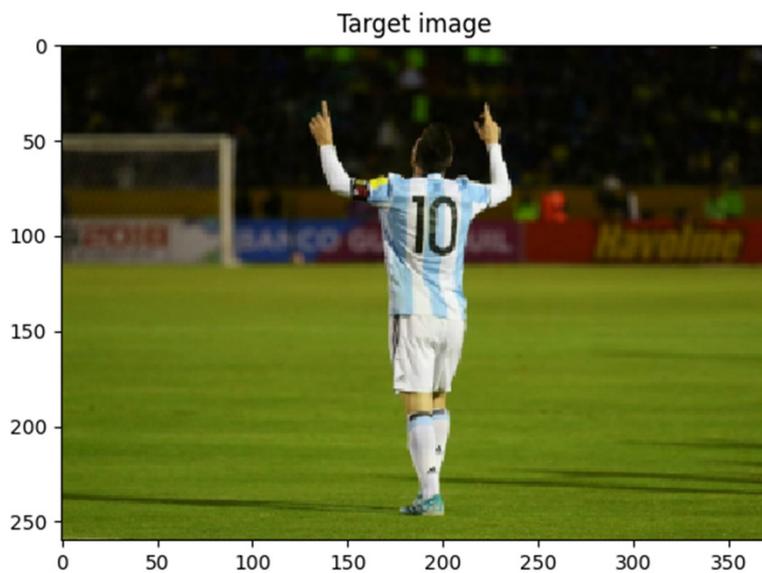
- El color cyan (alrededor de  $H = 30$  y  $190 \leq S \leq 200$ ) son valores mayores a 0.
- El color púrpura son valores iguales a 0.
- El color blanco son valores nan.

# Proyección de regreso

- Ahora  $R$  se usa como paleta y se proyecta de regreso.
- Es decir,  $B(x, y) = R[h(x, y), s(x, y)]$
- Donde  $h$  es el hue y  $s$  la saturación del pixel en  $(x, y)$  en la imagen destino.
- $B$  tiene valores distintos de 0 en la región de la imagen destino que corresponde a los colores de la imagen ROI.

# Resultado

- Ver los detalles en `histograms/histogram_backprojection.py`.



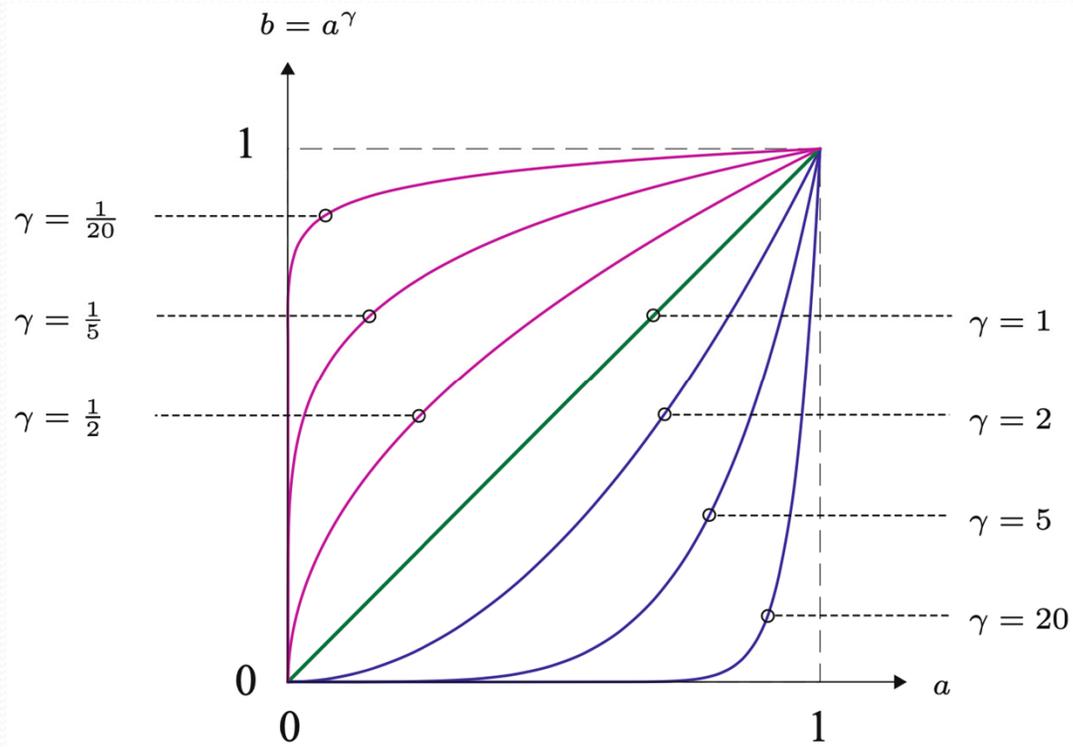
# Histogram backprojection in OpenCV

- Método `cv.calcBackProject()`.

# Corrección gamma

- **Corrección gamma.** Operación no lineal de la forma:
- $i_{out} = i_{in}^\gamma$
- Donde  $i_{out}$  es el valor actualizado del pixel,  $i_{in}$  es el valor original del pixel y  $\gamma$  es un valor real positivo.
- Si  $i_{in}$  se normaliza al intervalo  $[0, 1]$ , entonces  $i_{out}$  también se queda dentro de  $[0, 1]$ .

# Corrección gamma



**Fig. 4.18**

Gamma correction function  $f_\gamma(a) = a^\gamma$  for  $a \in [0, 1]$  and different gamma values.

# Corrección gamma

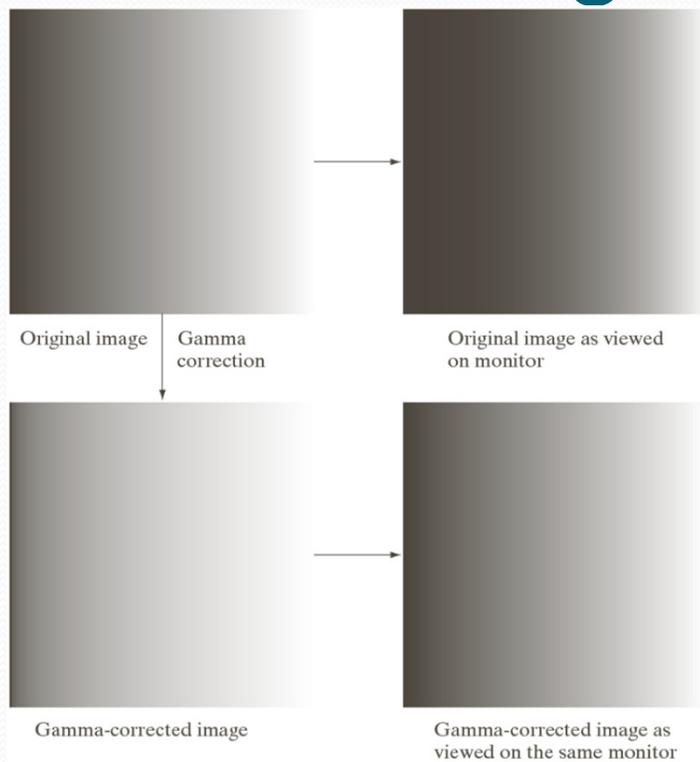


Fuente: [https://upload.wikimedia.org/wikipedia/commons/9/96/GammaCorrection\\_demo.jpg](https://upload.wikimedia.org/wikipedia/commons/9/96/GammaCorrection_demo.jpg)

# Corrección gamma

- Algunos dispositivos de captura, impresión y despliegado de imágenes responden a una ley de potencia.
- Por ejemplo, los dispositivos de rayos catódicos (CRT) tienen una intensidad-voltaje que responden a una ley de potencia con gamma entre 1.8 y 2.5.
- Esto implica que los monitores CRT producen imágenes mas oscuras de lo esperado.

# Corrección gamma



a	b
c	d

**FIGURE 3.7**  
(a) Intensity ramp image. (b) Image as viewed on a simulated monitor with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).

Corrección gamma de (a) a (c):

$$s = r^{1/2.5} = r^{0.4}$$

# Corrección gamma en OpenCV

- No hay una función en OpenCV para realizar corrección gamma.
- Una implementación esta en el archivo `gamma.py` en el código de la clase.

# Corrección gamma en OpenCV

Original

$\gamma = 0.5$

$\gamma = 2.8$



# Resumen

- Espacios de color básicos: RGB, HSV, YCbCr, YUV.
- Ajustes básicos de contraste: ecualización del histograma, corrección gamma.
- Imágenes en color: en muchas ocasiones conviene separar la luminancia y la crominancia para evitar cambios de color no deseados.
- Al mejorar el contraste en una imagen en color, por lo general es mejor operar en el canal de luminancia.