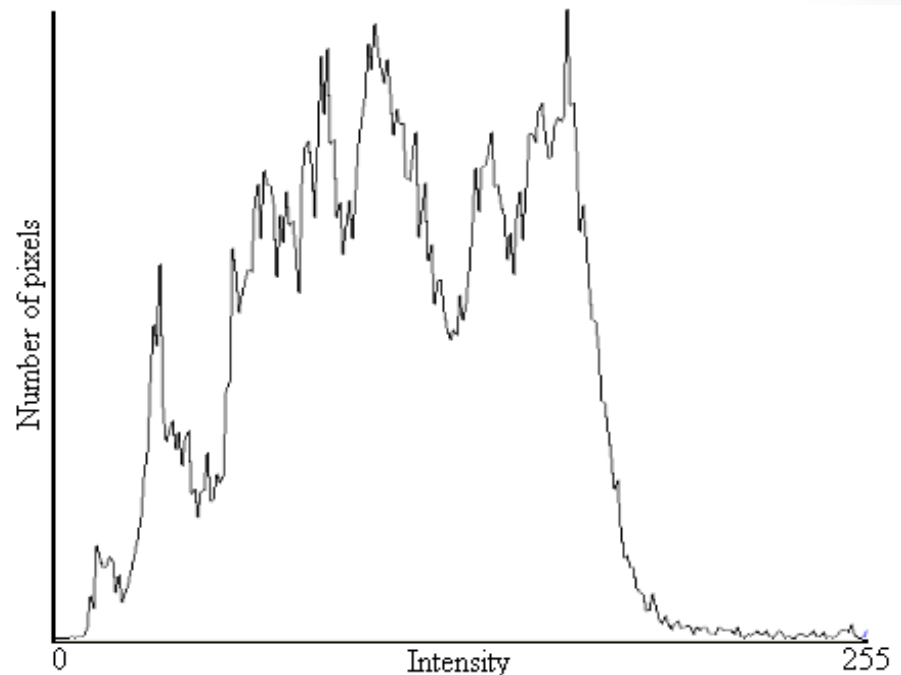
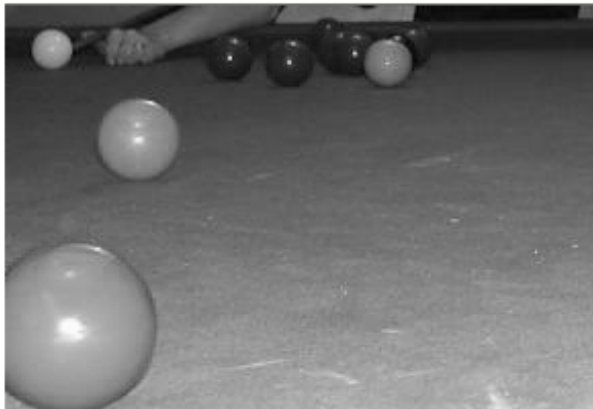


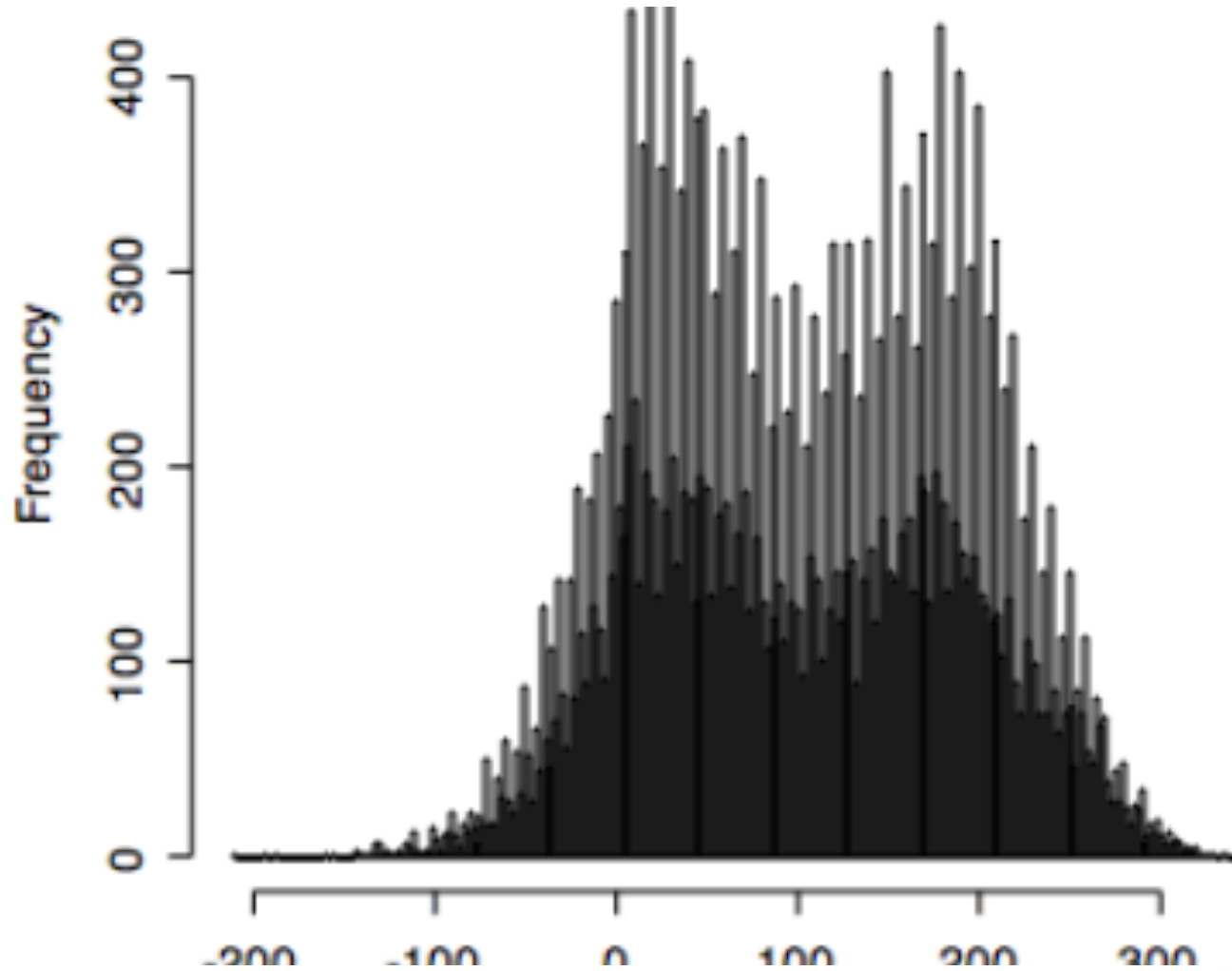
# Histogramas e Matching

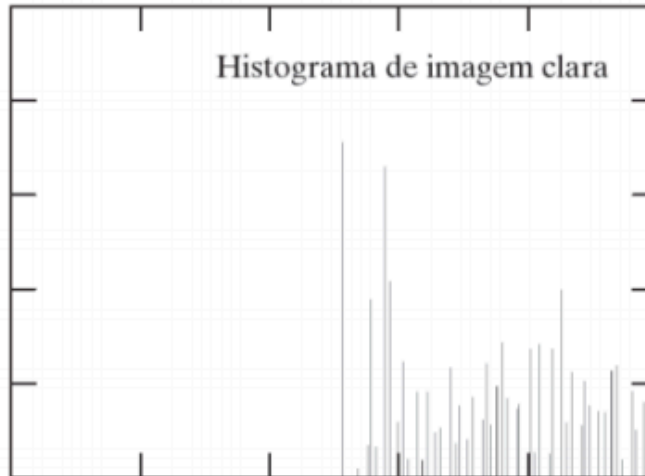
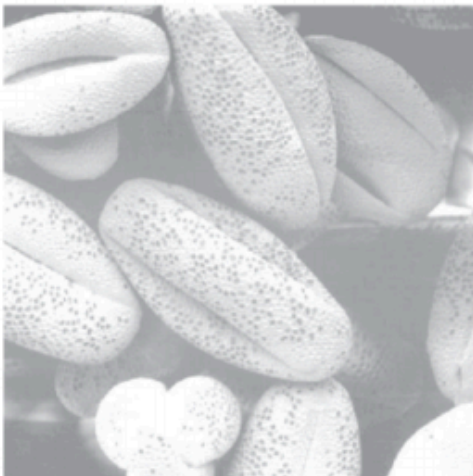
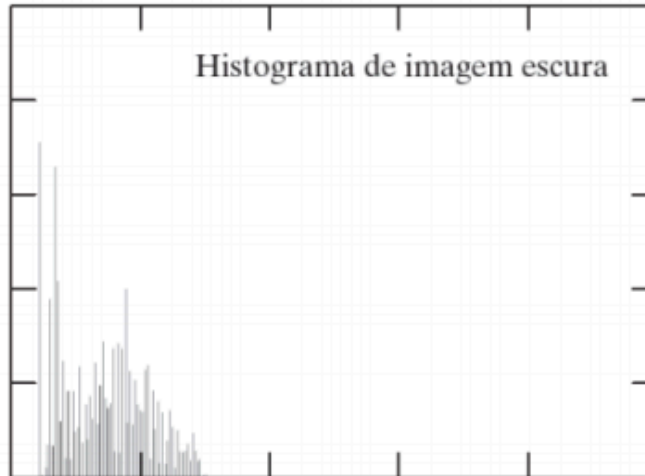
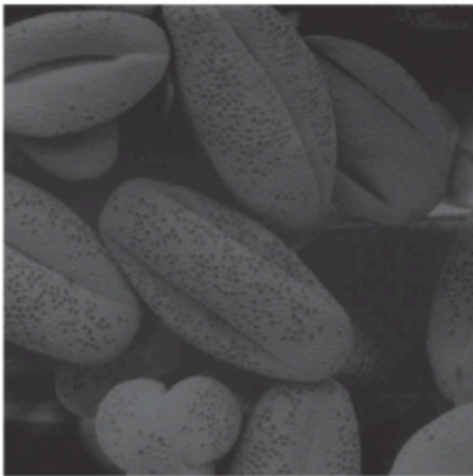
# Definição Histograma 1D

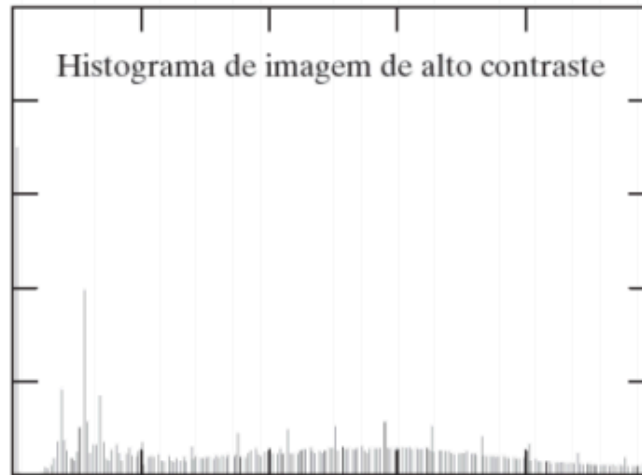
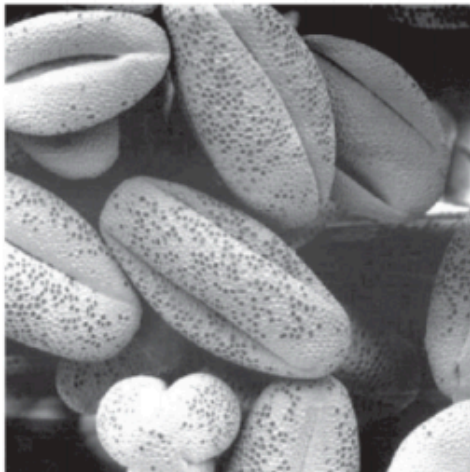
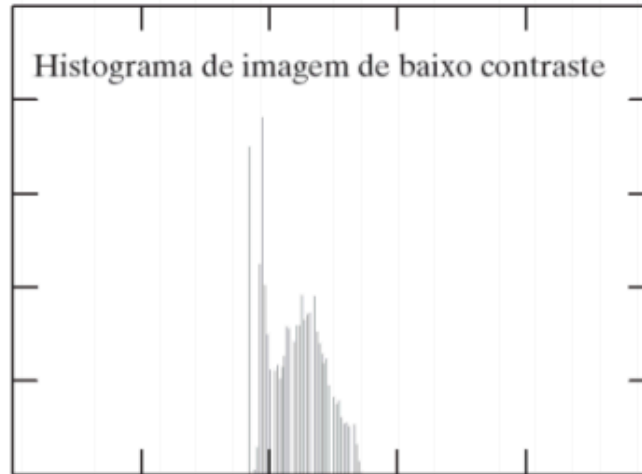
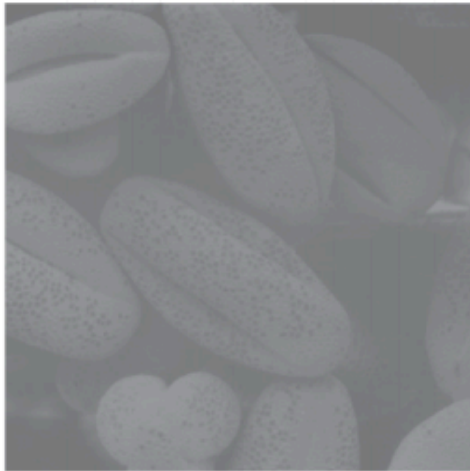
- O histograma de uma imagem em tons de cinza é uma **função  $H(k)$**  que produz o número de ocorrências de cada nível de cinza de uma imagem (função de distribuição de probabilidades)



# Exemplos de Histograma







# Qual a informação o Histograma fornece?

- Informação global da distribuição de tonalidades
  - Probabilidades
- Independente de localização e orientação
- Pode ser usada em problemas de classificação?
  - cuidado, imagens diferentes podem possuir o mesmo histograma

# Processo Histograma 1D

Para todo valor de  $Z$ : inicialize  $H(z) = 0$

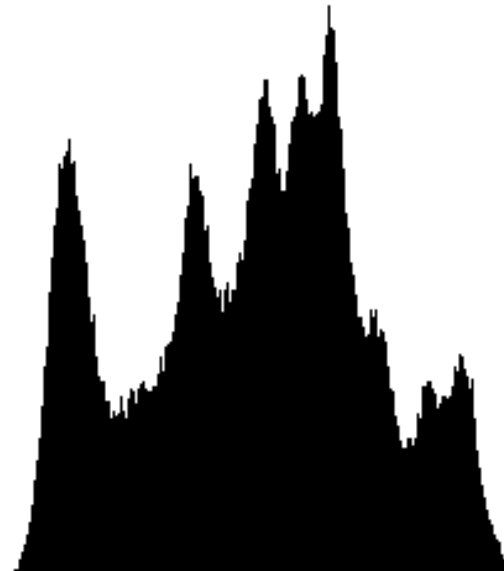
onde  $z$  varia de 0 a  $L-1$ , sendo  $L$  a quantidade de tons diferentes possíveis para a imagem

**$z$  também é chamado de bin**

Para cada pixel da imagem

- $H(\text{pixel}(\text{img}(l,j))) ++;$

# Usando opencv para obter o histograma



```
Mat gray_image = imread("lenna.png", CV_8U);  
MatND histogram;  
int number_bins = 256;  
calHist( &gray_image, 1, 0, Mat(), histogram, 1, &number_bins,  
NULL );
```



# Para realmente obter Probabilidades

- O histograma normalizado é uma função  $[0,1]$  onde se basicamente divide-se  $H[k]$  pela quantidade de pixels da imagem e se representa probabilidades
- Aqui, cada  $H[k]$  representa a probabilidade de cada valor de pixel da imagem

$$P_r(r_k) = \frac{n_k}{n}$$

$$0 \leq r_k \leq 1$$

# Na tabela

<b>Nível de Cinza (<math>r_k</math>)</b>	<b><math>n_k</math></b>	<b><math>P_r(r_k)=n_k/n</math></b>
<b>0</b>	<b>1120</b>	<b>0,068</b>
<b>1/7</b>	<b>3214</b>	<b>0,196</b>
<b>2/7</b>	<b>4850</b>	<b>0,296</b>
<b>3/7</b>	<b>3425</b>	<b>0,209</b>
<b>4/7</b>	<b>1995</b>	<b>0,122</b>
<b>5/7</b>	<b>784</b>	<b>0,048</b>
<b>6/7</b>	<b>541</b>	<b>0,033</b>
<b>1</b>	<b>455</b>	<b>0,028</b>

# Normalização pelo Maior

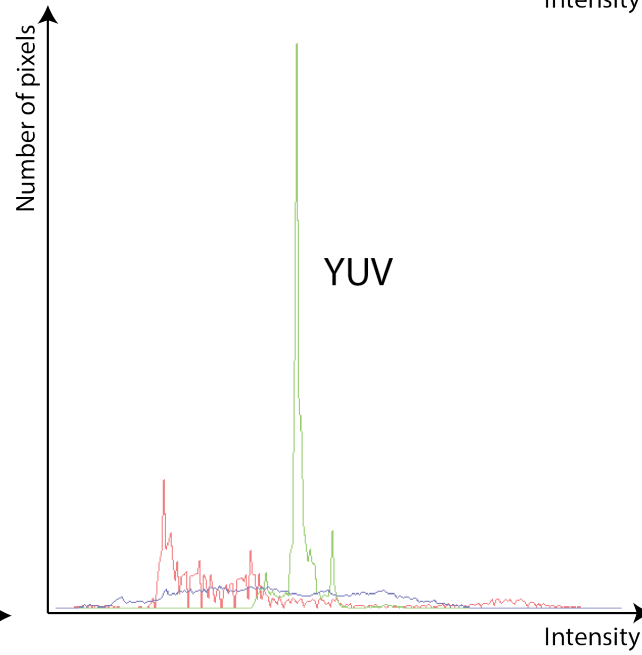
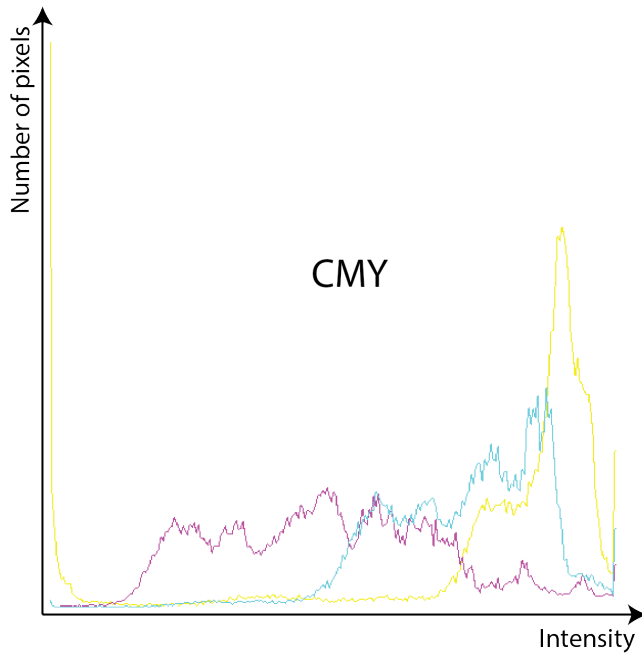
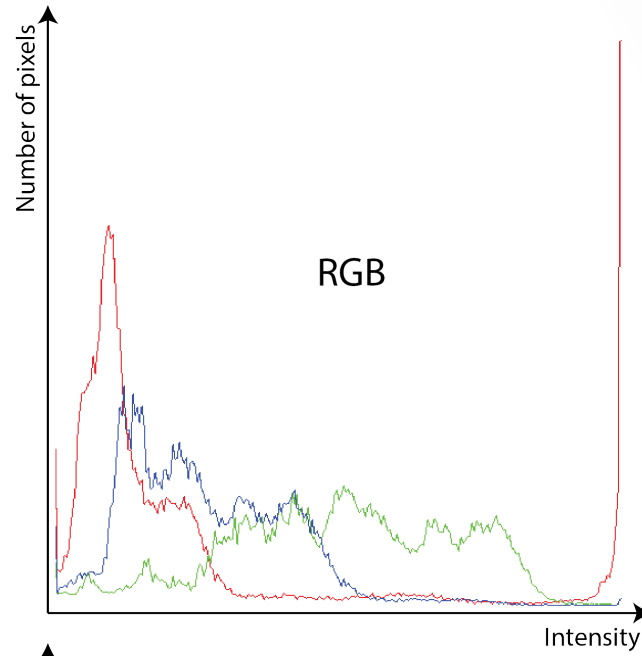
- Outra maneira de normalizar um histograma é:
  1. Obtenha o maior valor do histograma ( $\max(H(k))$ )
  2. Divida todos os elementos do histograma por esse maior valor
- histograma está normalizado entre [0 ... 1]

Com o opencv:

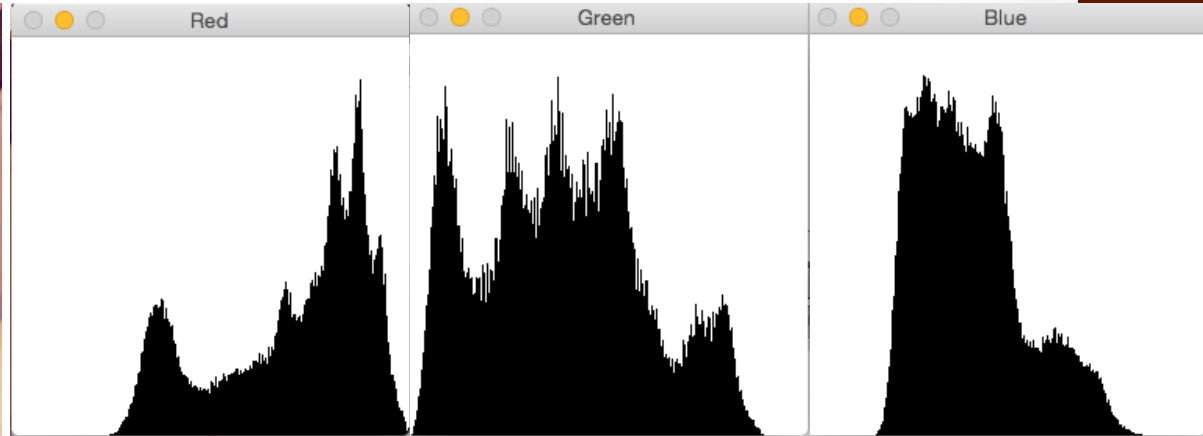
```
normalize(histogram,histogram,1.0);
```

# Histogramas também podem ser ND

- No caso de imagens multidimensionais, o histograma é calculado separadamente para cada canal
- A forma que as cores são distribuídas influenciam diretamente no resultado



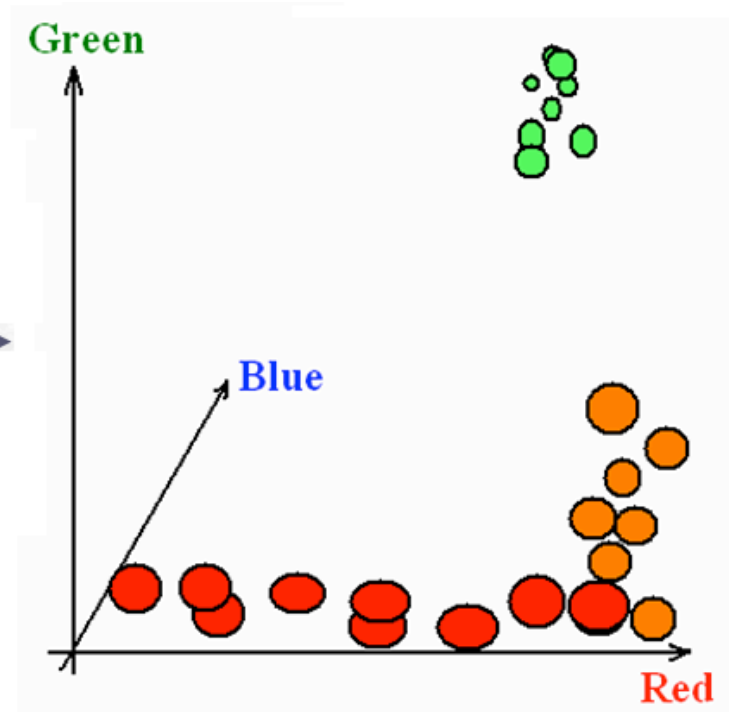
# HistogramaND no opencv



```
Mat b_hist, g_hist, r_hist; vector<Mat> bgr_planes;  
split( src, bgr_planes );  
calcHist( &bgr_planes[0], 1, 0, Mat(), b_hist, 1, &number_bins, NULL );  
calcHist( &bgr_planes[1], 1, 0, Mat(), g_hist, 1, &number_bins, NULL );  
calcHist( &bgr_planes[2], 1, 0, Mat(), r_hist, 1, &number_bins, NULL );
```

# E se considerarmos o histograma 3D?

- E utilizar todos os canais ao mesmo tempo?
- O que podemos supor é que o poder de discriminação seja melhor
- **O que fazer com o número de bins do histograma?**



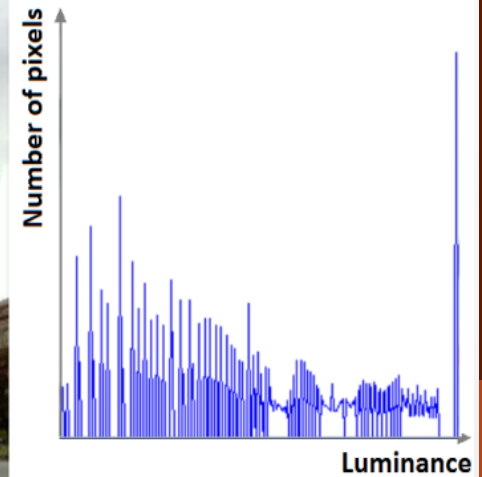
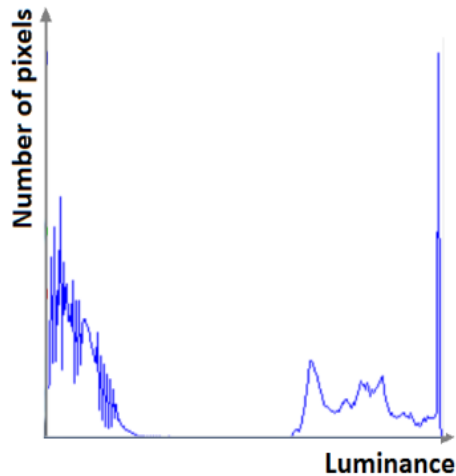
# Histograma 3D

- Utilize quantização para reduzir a quantidade de informação por canal
  - 6 bits = 262,144
  - 4 bits = 4,096
  - 2 bits = 64
  
- Faça você esse exemplo!



# Equalização de Histogramas

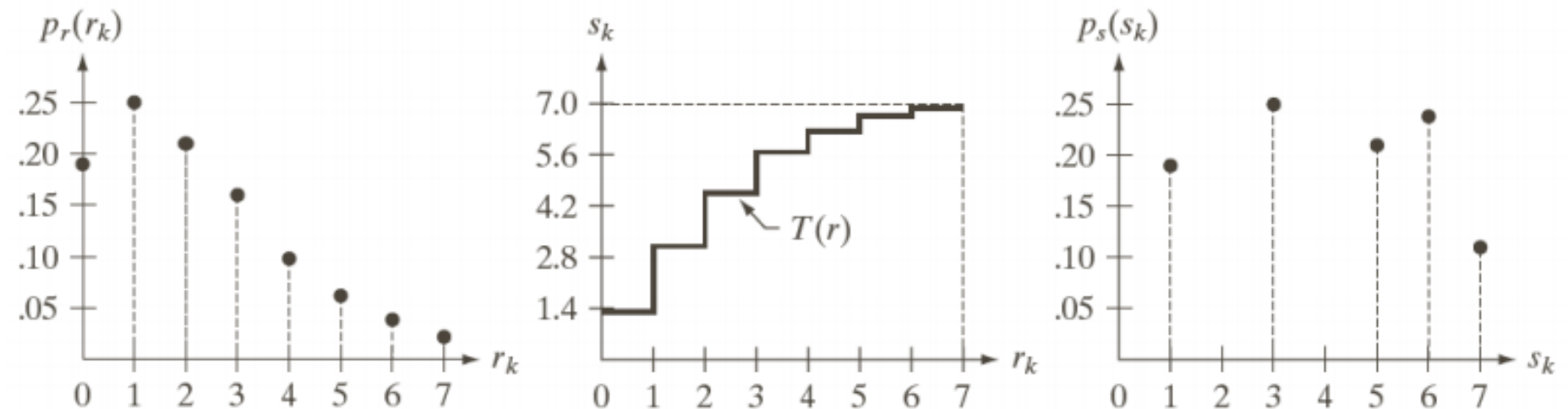
- Aplicado para imagens que possuam baixo contraste
- Realiza o processo espalhando ocorrências sobre todos os níveis de cinza



# Processo de Equalização

- O processo é feito através de uma função de transformação  $T$  (histograma acumulativo)
- $T$  acumula ocorrências de tonalidades anteriores sujeito ao fator de distribuição:

$$\frac{L-1}{M * N}$$



# Processo de Equalização

- Seja  $r$  definida entre  $[0, L-1]$
- A função  $T(r)$ :

$$T(r_k) = \frac{(L-1)}{M * N} \sum_{j=0}^k n_j$$

sendo  $k = 0, 1, 2, 3, \dots, L-1$

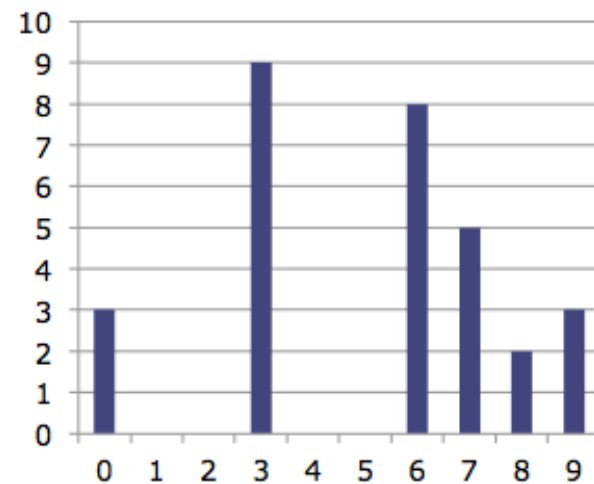
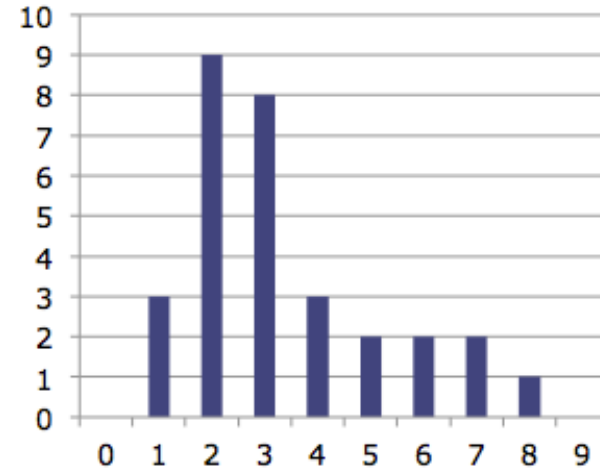
A função cria uma tabela de mapeamento de cores

Utilize uma função de arredondamento ou  $r_k$  deve ser um valor inteiro

# Processo de Equalização

- Para  $M*N = 30$  e 10 níveis de cinza

k	n	$\Sigma n$	T(rk)
0	0	0	0
1	3	3	0
2	9	12	3
3	8	20	6
4	3	23	7
5	2	25	7
6	2	27	8
7	2	29	9
8	1	30	9
9	0	30	9



# Equalização de Histogramas

- `equalizeHist(image,result);`



# USANDO HISTOGRAMAS



# Equalização de Histogramas

```
cvtColor(image, hls_image, COLOR_BGR2HLS);  
vector<Mat> channels;  
split(hls_image, channels);  
equalizeHist( channels[1], channels[1] );  
merge( channels, hls_image );
```



# Comparação de Histogramas

- Motivo
  - Para encontrar imagens similares (note na distribuição de tonalidades)
- Métrica para comparação
  - Correlação (uma das)
- Sempre realizado sobre histogramas normalizados



# Métricas de Correlação

- $D_{Correlation}(h_1, h_2) = \frac{\sum_i (h_1(i) - \bar{h}_1)(h_2(i) - \bar{h}_2)}{\sqrt{\sum_i (h_1(i) - \bar{h}_1)^2 \sum_i (h_2(i) - \bar{h}_2)^2}}$
- $D_{Chi-Square}(h_1, h_2) = \sum_i \frac{(h_1(i) - h_2(i))^2}{(h_1(i) + h_2(i))}$
- $D_{Intersection}(h_1, h_2) = \sum_i \min(h_1(i), h_2(i))$
- $D_{Bhattacharyya}(h_1, h_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{h}_1 \cdot \bar{h}_2 \cdot N^2} \sum_i \sqrt{h_1(i) \cdot h_2(i)}}$
- where
  - $N$  is the number of bins in the histograms,
  - $\bar{h}_k = \frac{\sum_i (h_k(i))}{N}$

# Query usando Histogramas

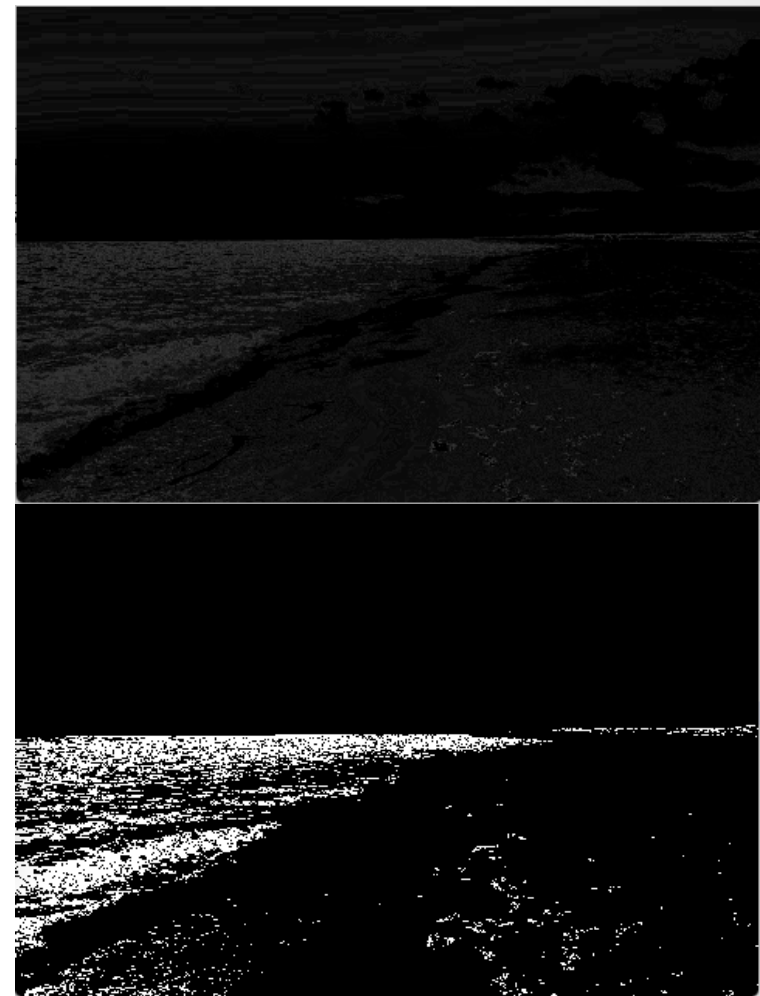
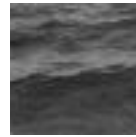


```
normalize( histogram1, histogram1, 1.0);  
normalize( histogram2, histogram2, 1.0);  
double matching= compareHist( histogram1, histogram2,  
CV_COMP_CORREL);
```

# Matching usando Histograma

- Técnica usada para encontrar um conjunto de correspondências semelhantes entre si
- Passos
  1. Obtenha um conjunto representativo (templates)
  2. Calcula o histograma desses representantes
  3. Normaliza o histograma para o maior valor ser 1
  4. Usa projeção inversa (backprojection) do histograma sobre a imagem que se deseja encontrar

**O resultado é um mapa de probabilidades  $P$  que indica a similaridade do histograma com a imagem**



```
calcHist( &imageROI, 1, 0, Mat(), histogram, 1, &number_bins, NULL);  
normalize( histogram, histogram, 1.0);  
calcBackProject(&image, 1, 0, histogram, result, &ranges, 255.0, true);
```

# Colorida



```
calcHist( &imageROI, 1, channel_numbers, Mat(), histogram,  
image.channels(), number_bins, channel_ranges );  
normalize( histogram, histogram, 1.0);  
calcBackProject(&image,  
1,channel_numbers,histogram,result,channel_ranges,255.0);
```



# Outro exemplo com HLS



```
calcHist( &hls_samples_image, 1, channel_numbers, Mat(),  
         histogram,image.channels(),number_bins,channel_ranges);  
normalize( histogram, histogram, 1.0);  
Mat probabilities = histogram.BackProject( hls_image );
```