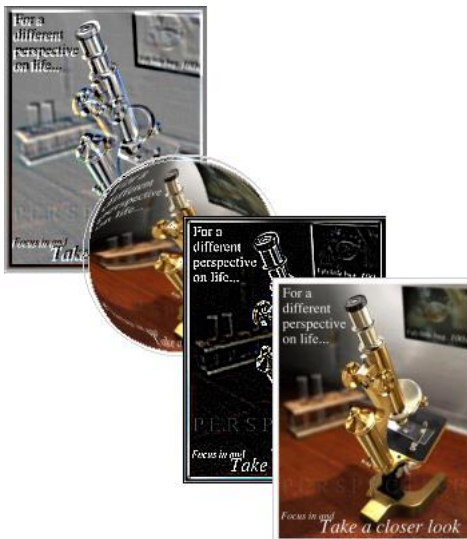


# Histograma

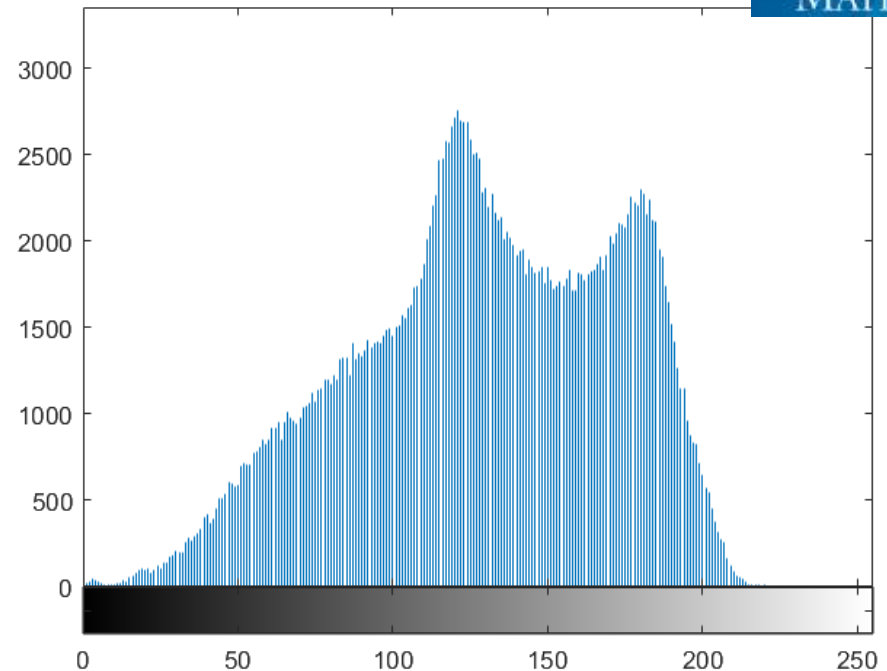
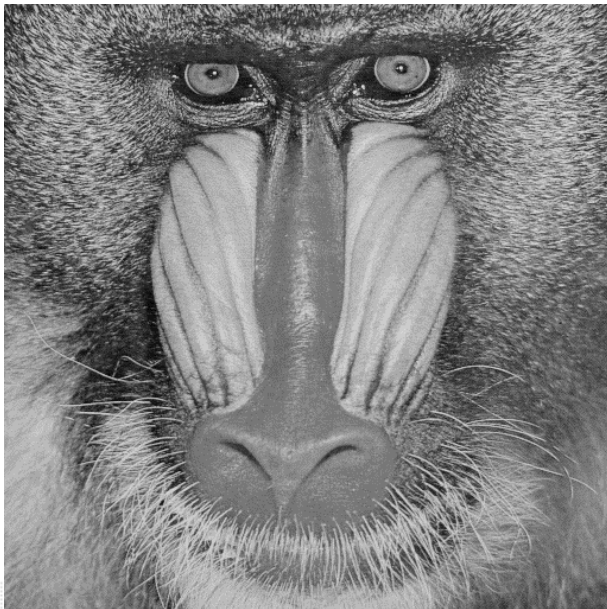
Carlos Alexandre Barros de Mello  
CIn/UFPE



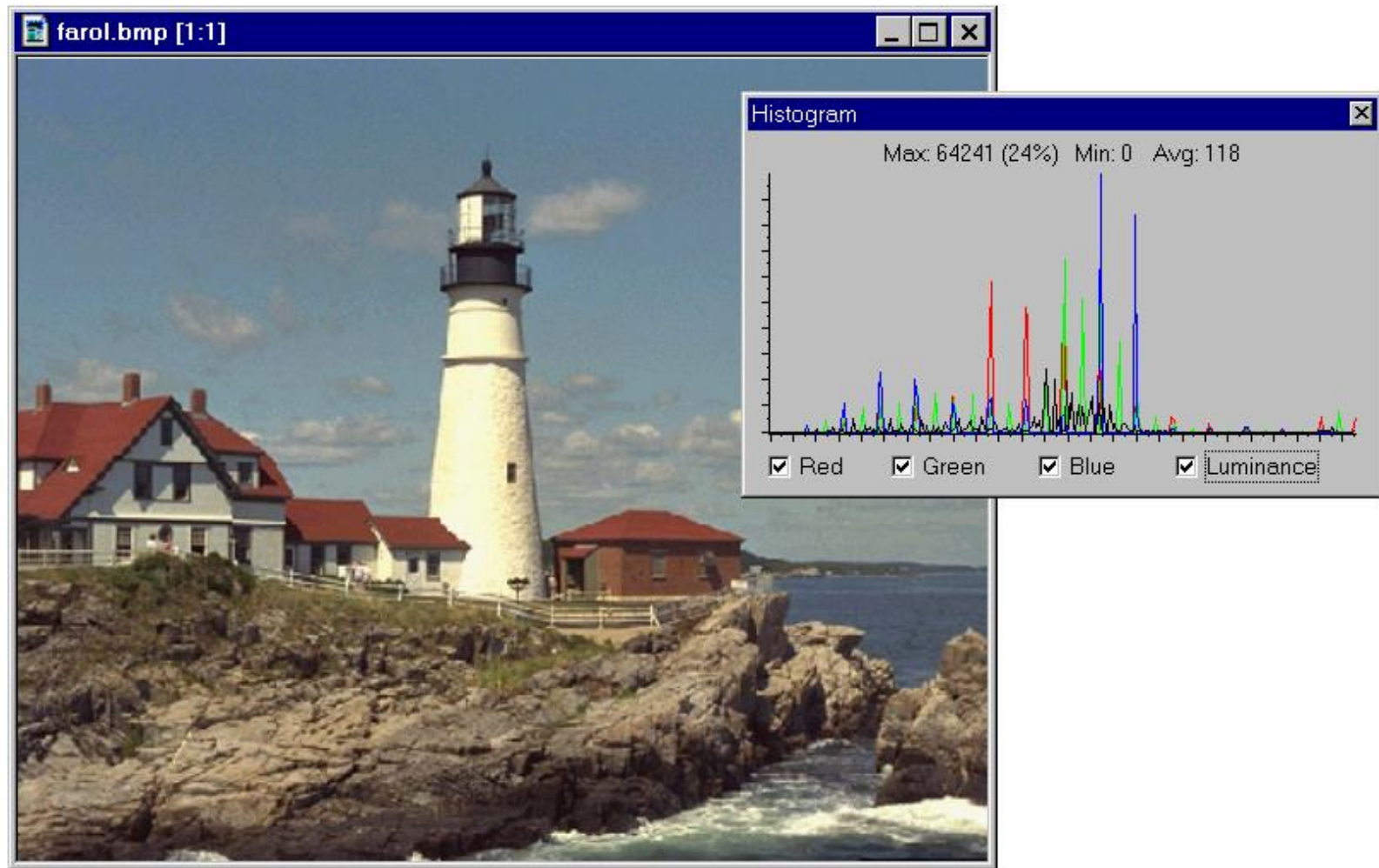
# Histograma

- O **Histograma** de uma imagem provê uma descrição global da aparência da imagem em termos de distribuição de cores

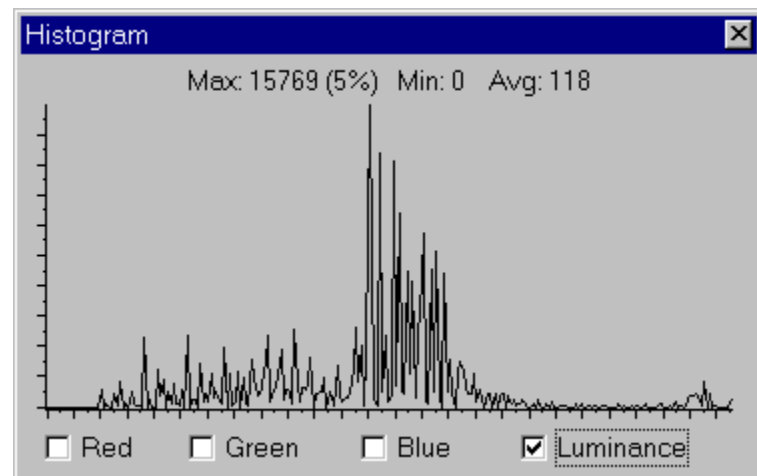
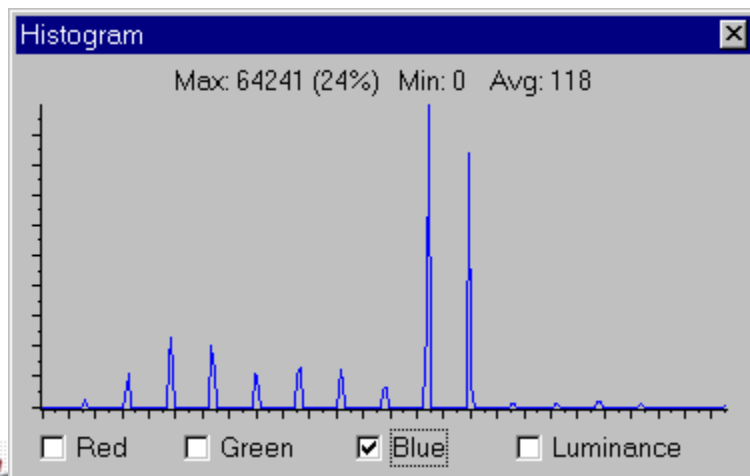
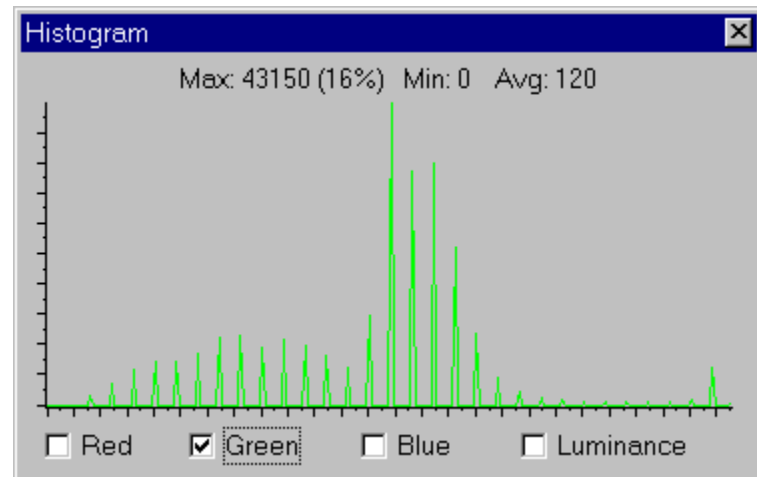
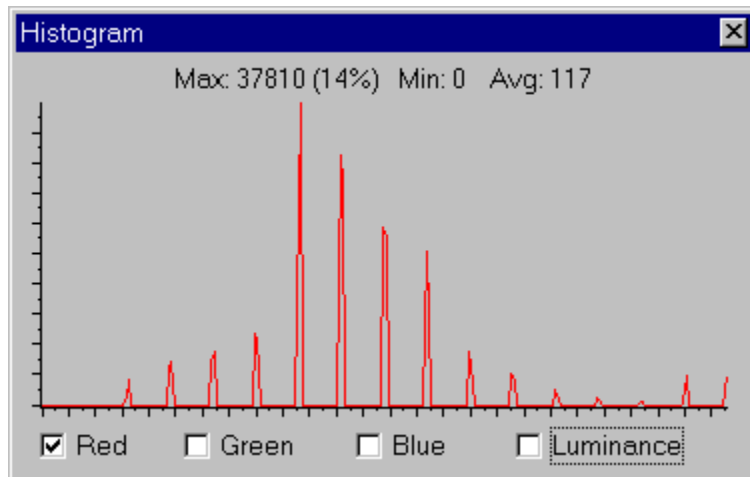
```
>> I = imread('baboon.bmp') >> figure, imhist(I,256)  
>> imshow(I)
```



# Histograma

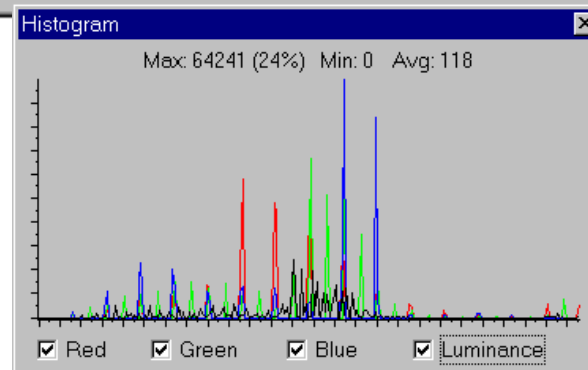
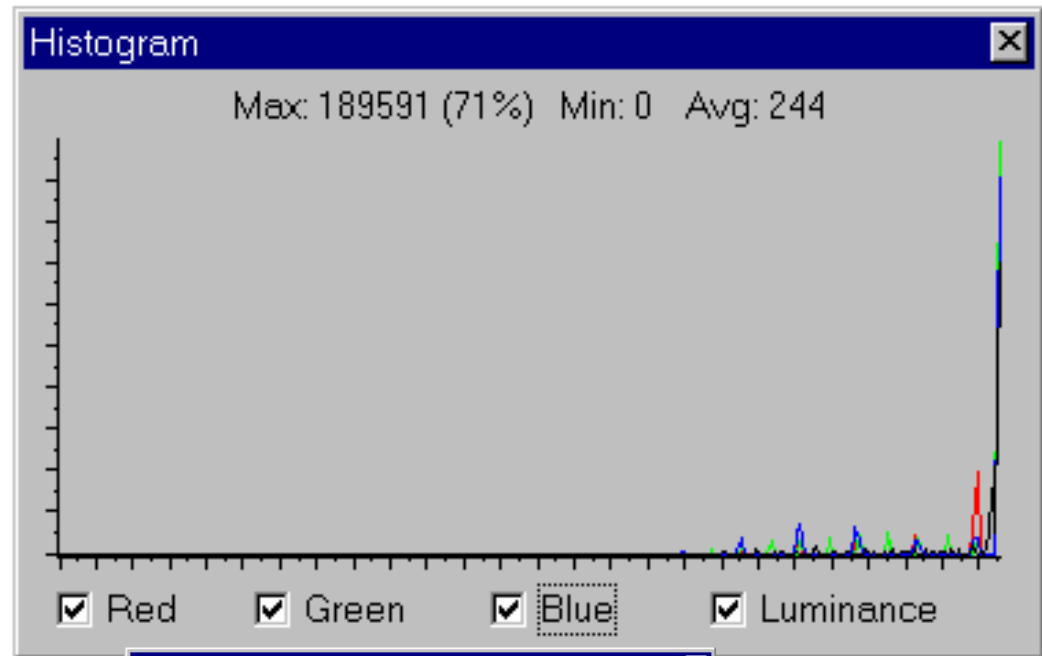


# Histograma



# Histograma

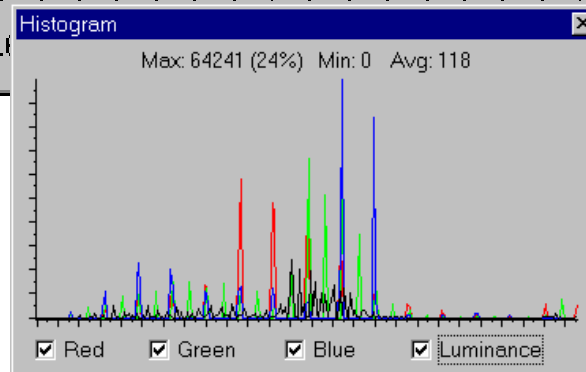
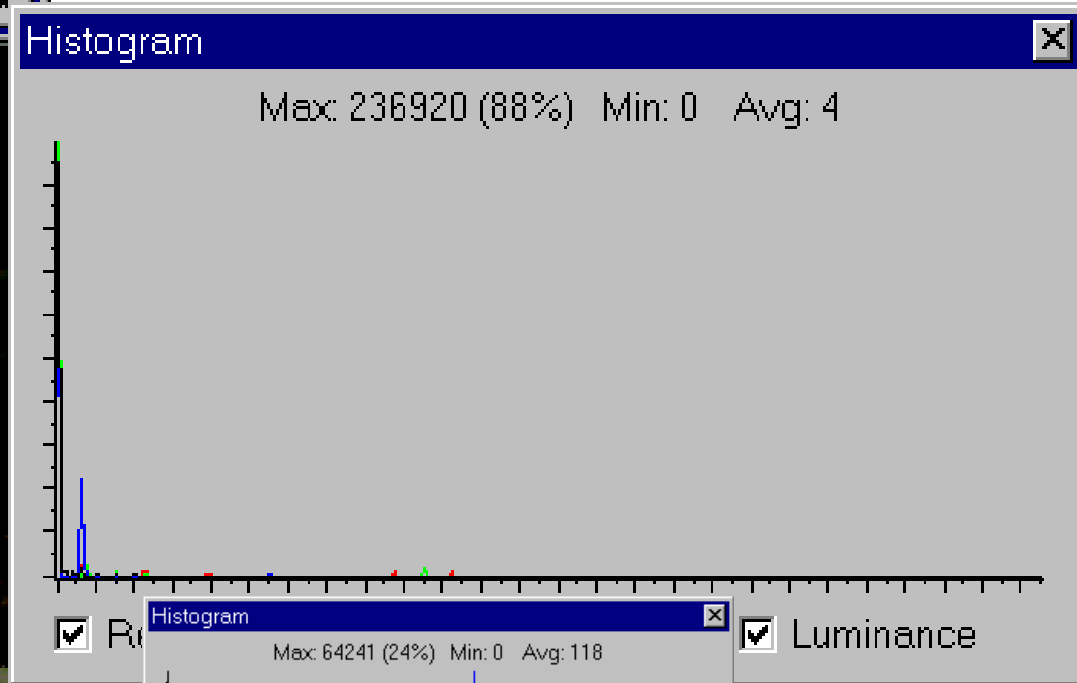
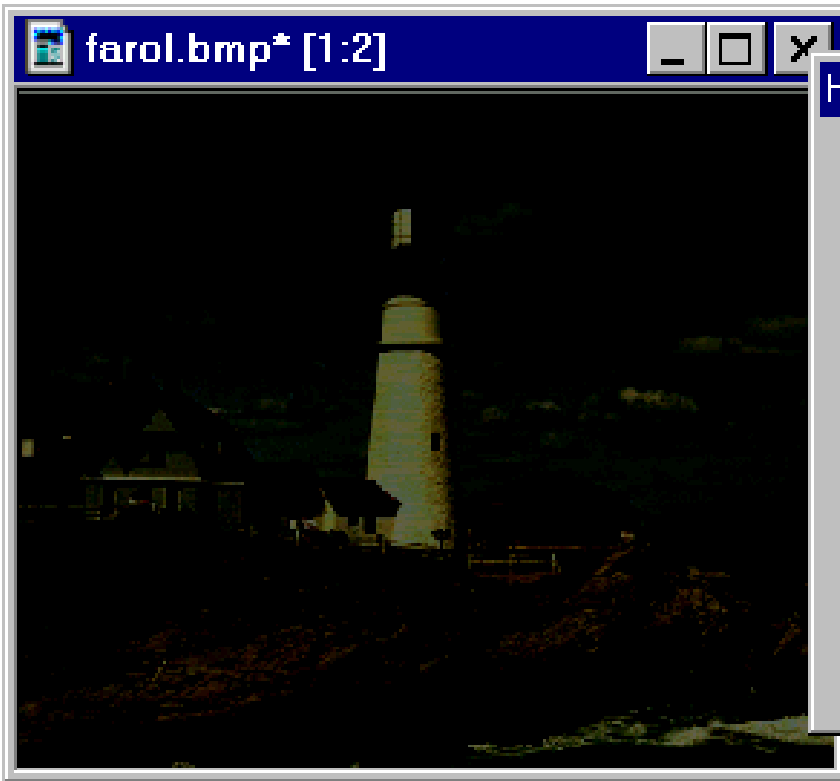
Imagem muito clara: histograma próximo do branco...



**Histograma original**

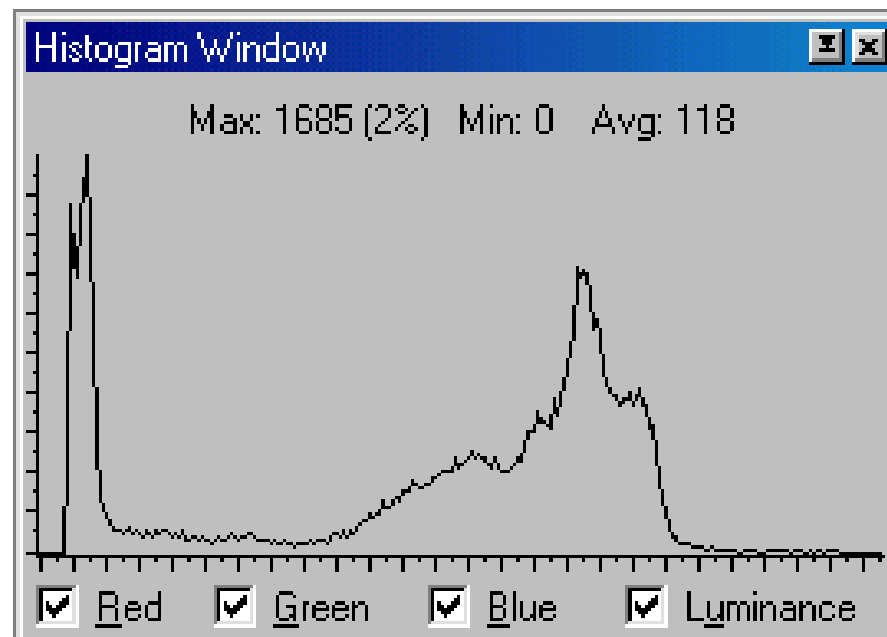
# Histograma

Imagem muito escura: histograma próximo do preto...



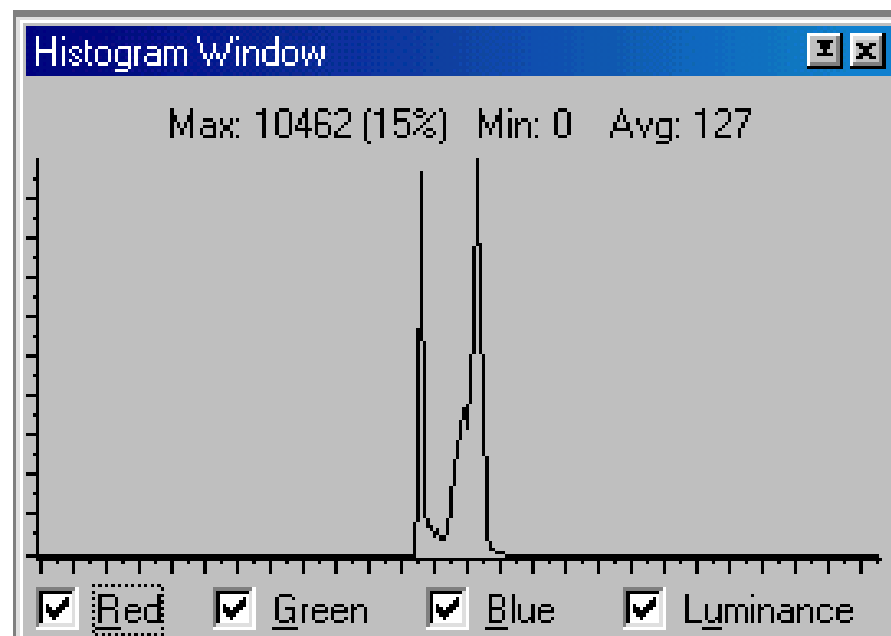
Histograma original

# Histograma e Contraste



**Contraste: nível de separação entre as cores**

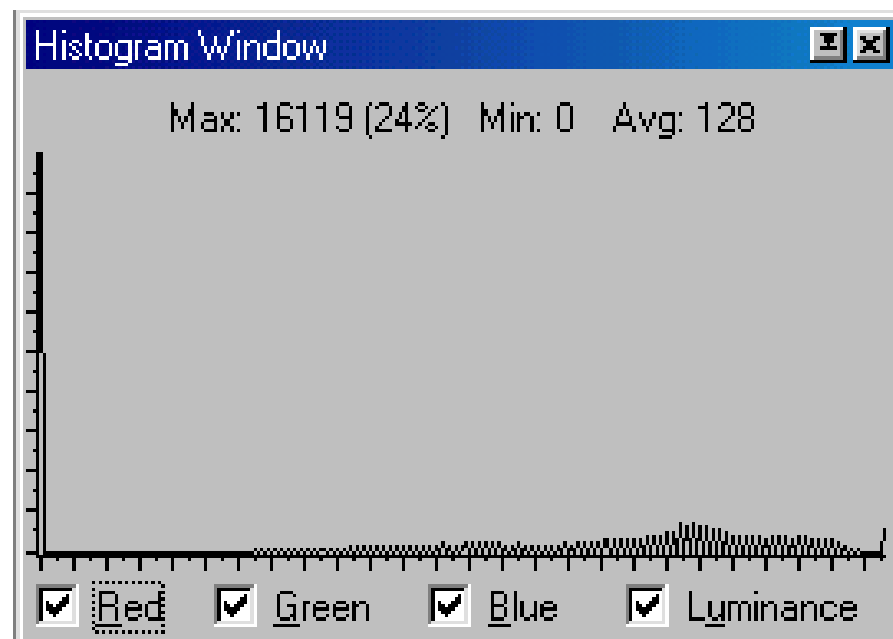
# Histograma e Contraste



**Imagem de (mais) Baixo Contraste**



# Histograma e Contraste



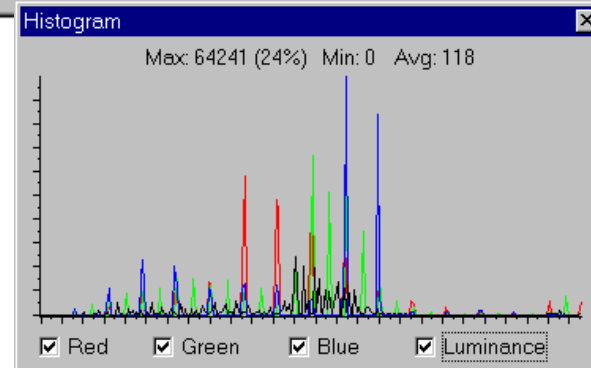
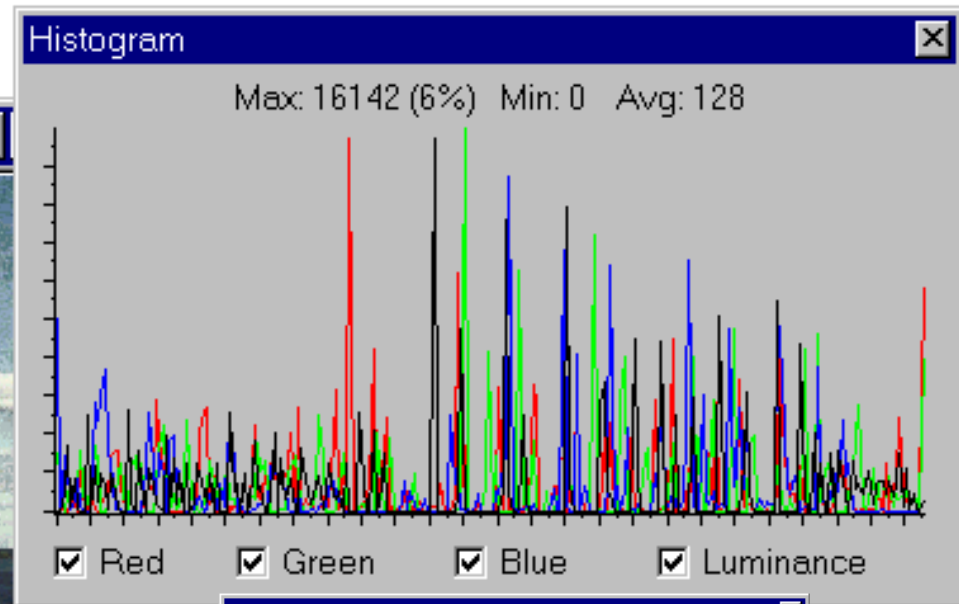
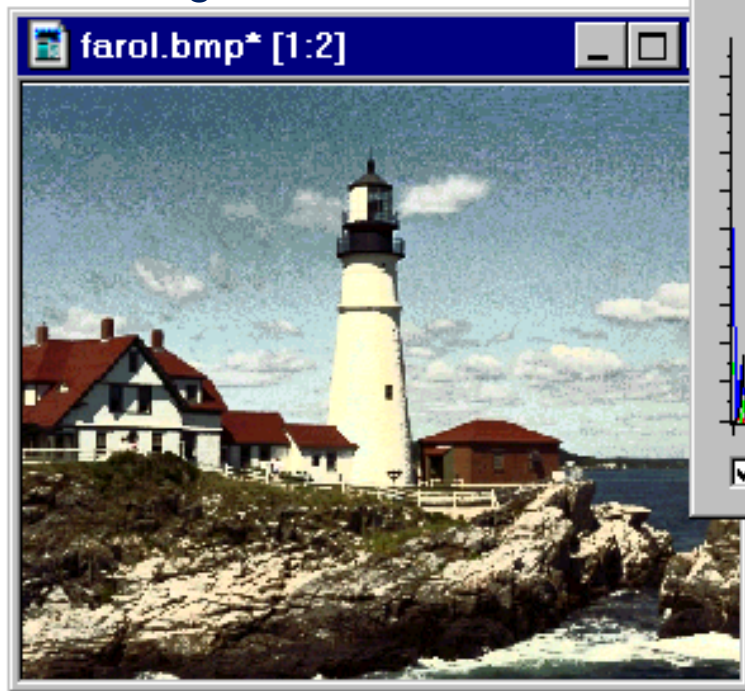
**Imagem de (mais) Alto Contraste**

# Histograma

## Operações

### Equalização

Aproximação para um histograma uniforme



# Histograma

## Operações

---

- Equalização:
  - Torna o pixel mais escuro mais próximo do preto e o mais claro mais próximo do branco
  - Distribui o restante das cores mais uniformemente nesse intervalo

# Histograma

## Operações

- Equalização:
  - Operação simples:
  - Para cada nível de brilho  $j$  na imagem original (e seu histograma), um novo valor  $k$  é atribuído, onde:

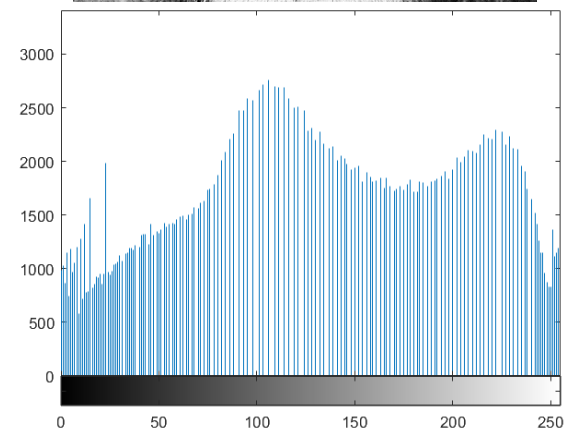
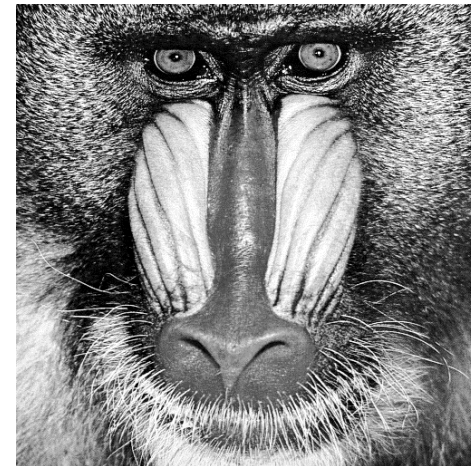
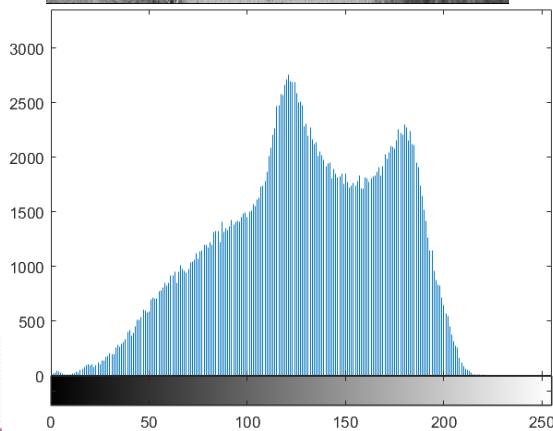
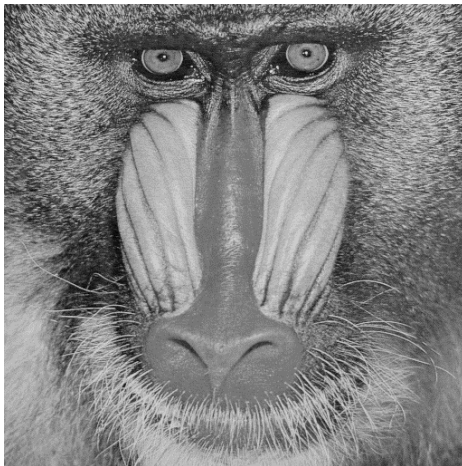
$$k = \sum_{i=0}^j \frac{N_i}{T}$$

- onde  $T$  é o número de pixels na imagem e  $N_i$  é o total de pixels na posição  $i$  do histograma

# Histograma

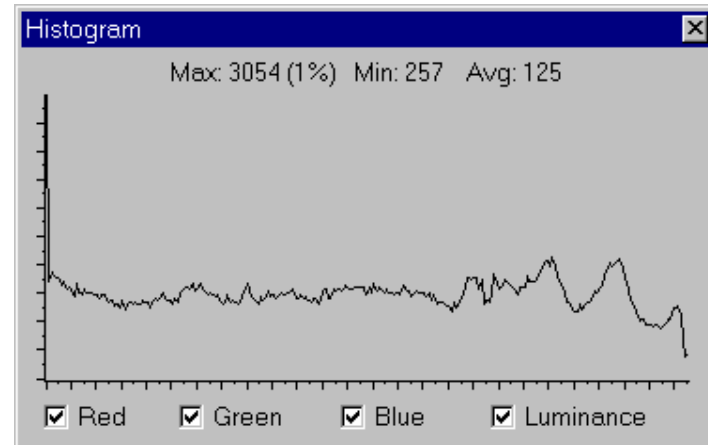
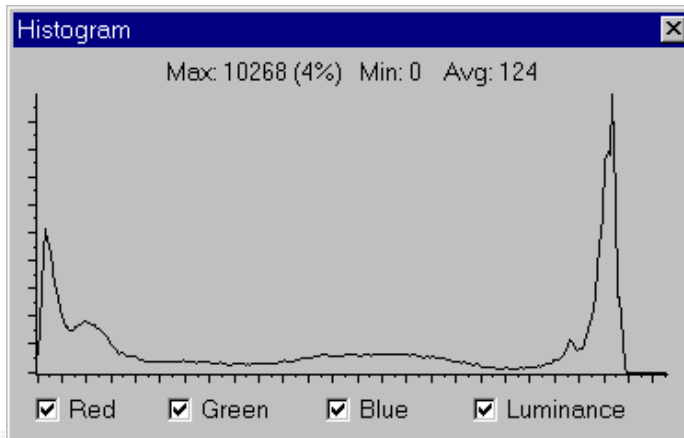
## Operações

- Equalização:



# Histograma

## Equalização - Exemplos

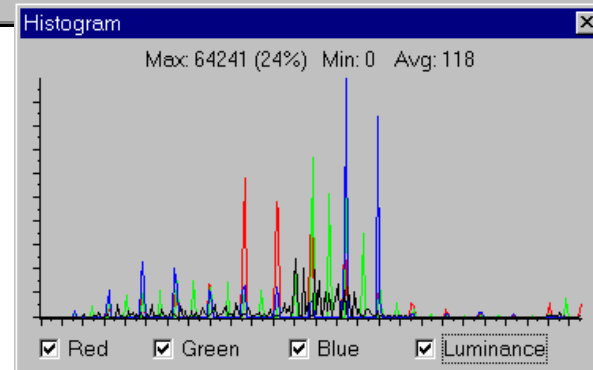
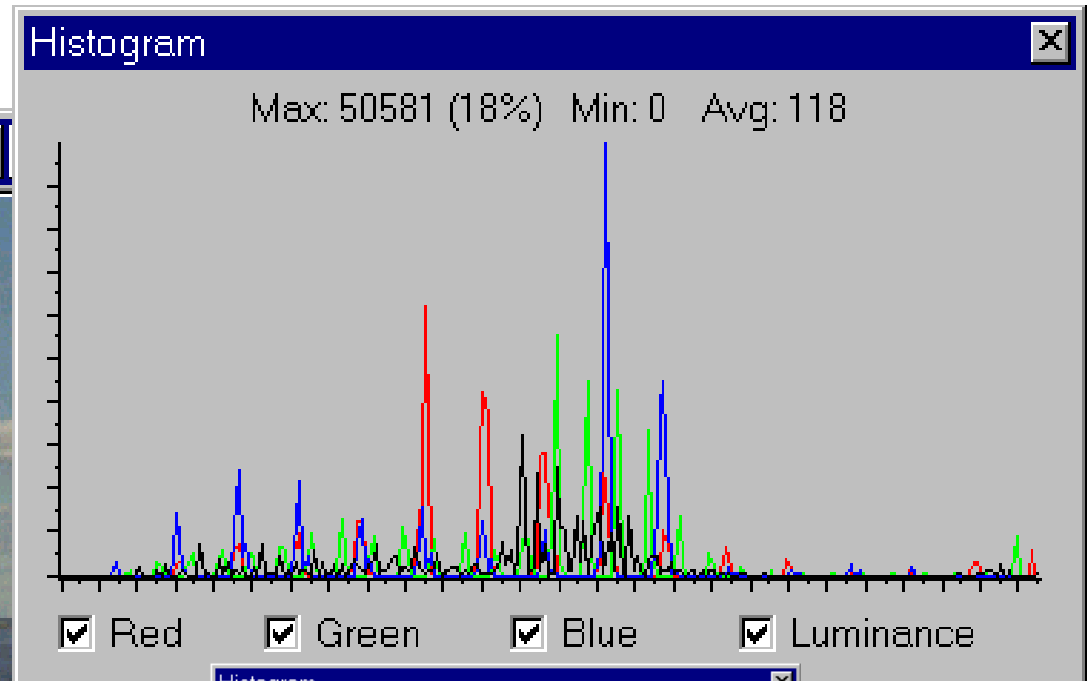




# Histograma

## Operações

### Stretch



# Histograma

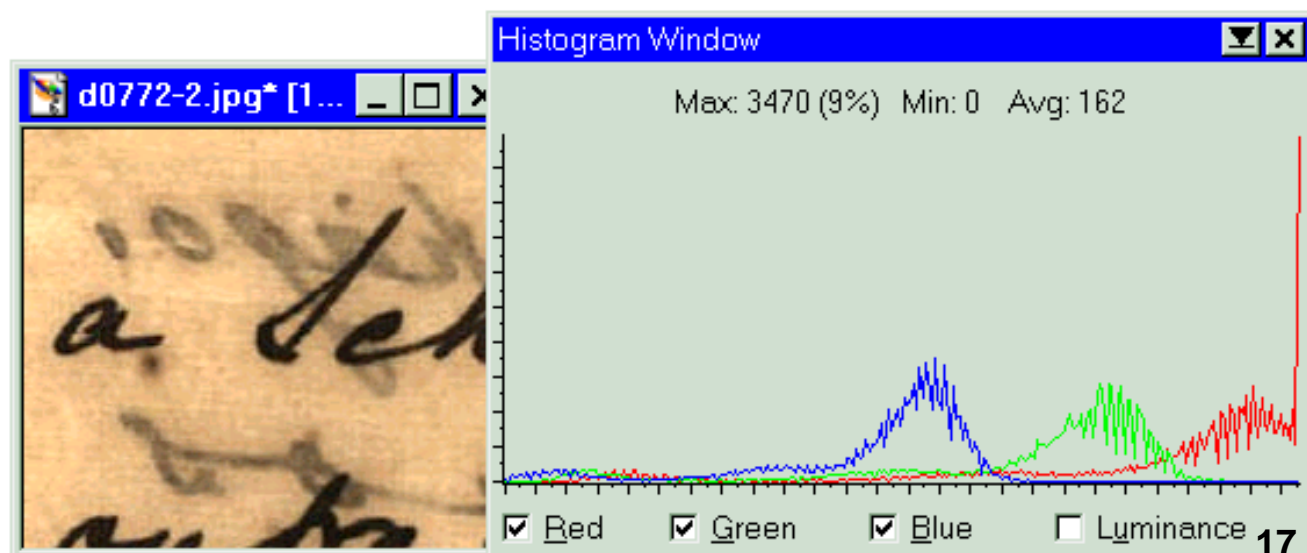
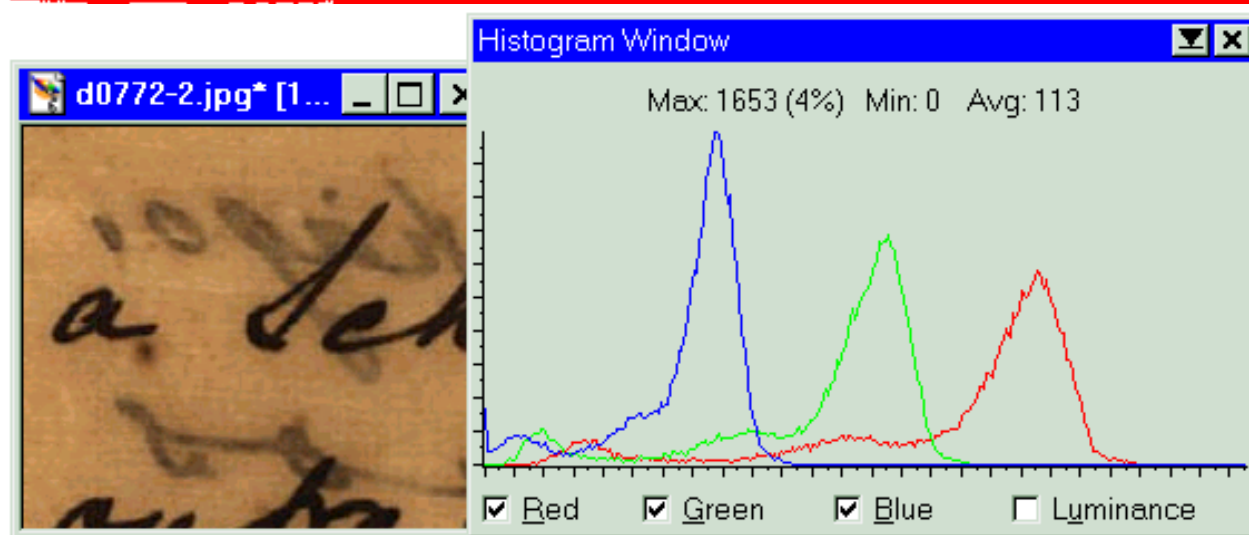
## Operações

- Stretch
  - Tenta espalhar mais o histograma pelo espectro
  - Não provoca muita diferença em imagens com um grande número de cores
    - Cores já espalhadas naturalmente



# Histograma

## Stretch - Exemplo

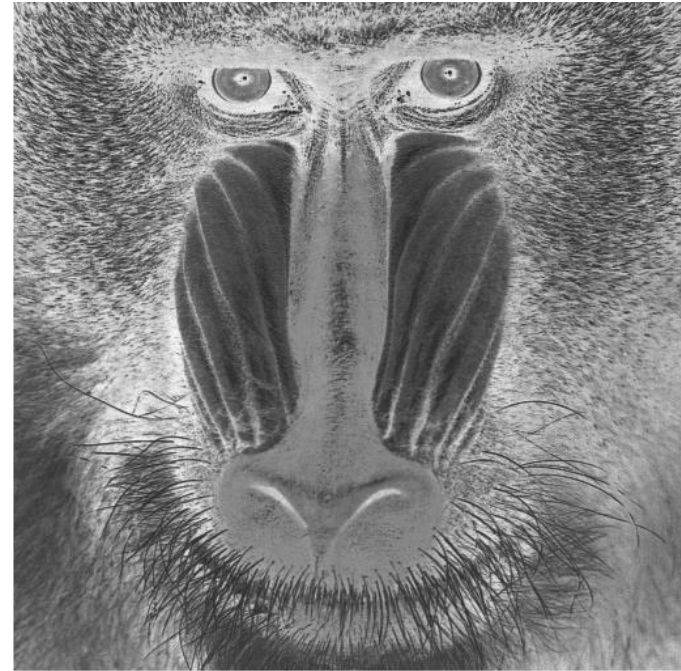
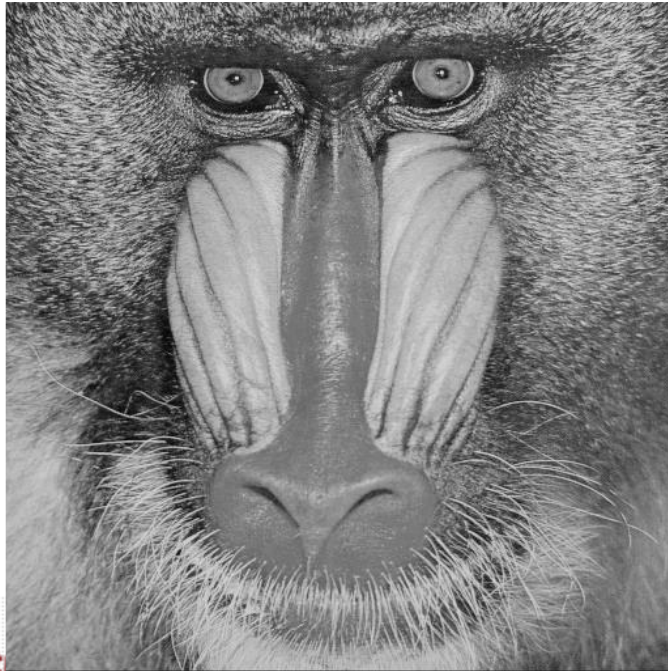


# Técnicas de modificação de histograma

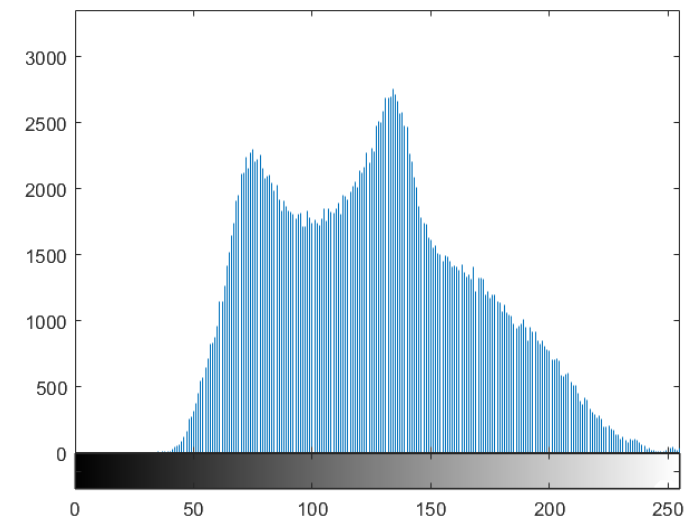
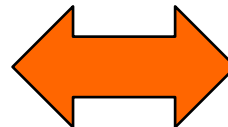
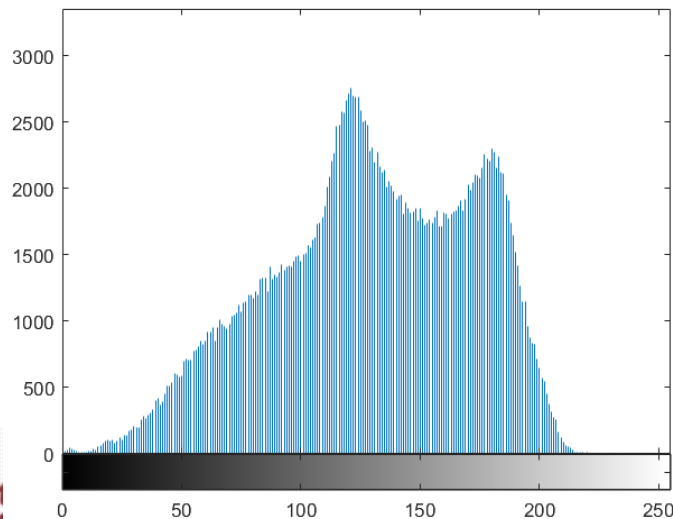
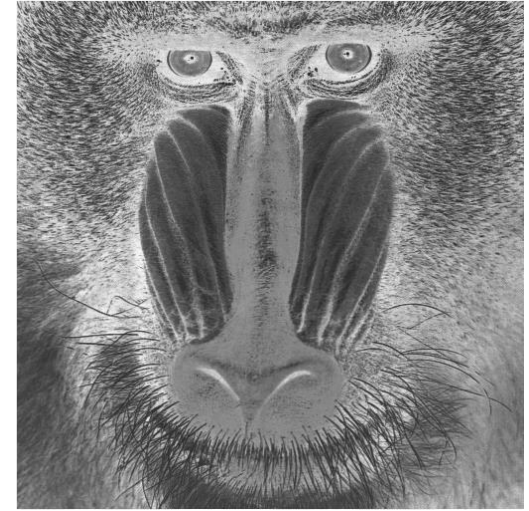
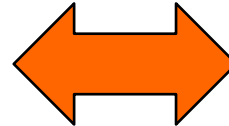
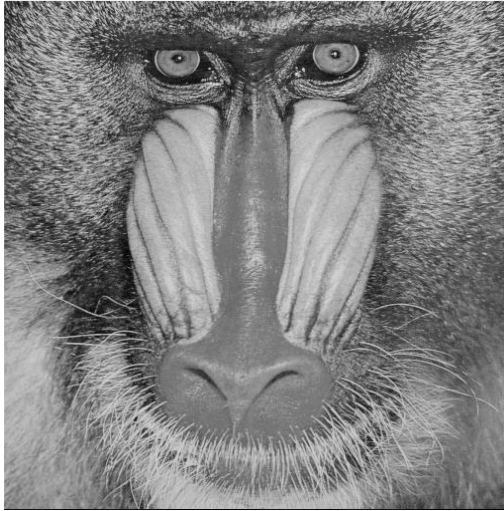
- São técnicas utilizadas para processar a imagem através da modificação do histograma
- Exemplos
  - Negativo (Complemento)
  - Binarização
  - Brilho
  - Expansão
  - Compressão

# Complemento

```
I = imread('baboon.bmp');  
I = 255 - I;  
imshow(I);  
figure, imhist(I);
```



# Complemento





# Complemento em Imagens Coloridas

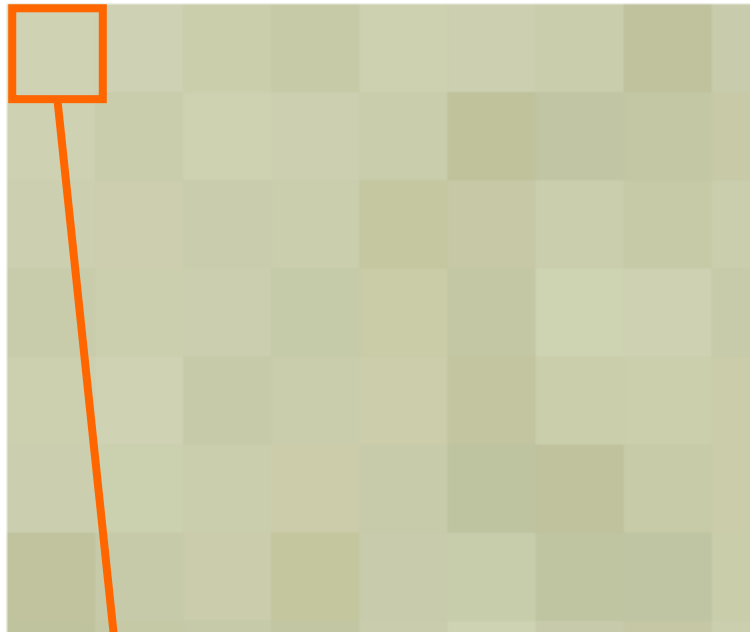


# Complemento em Imagens Coloridas



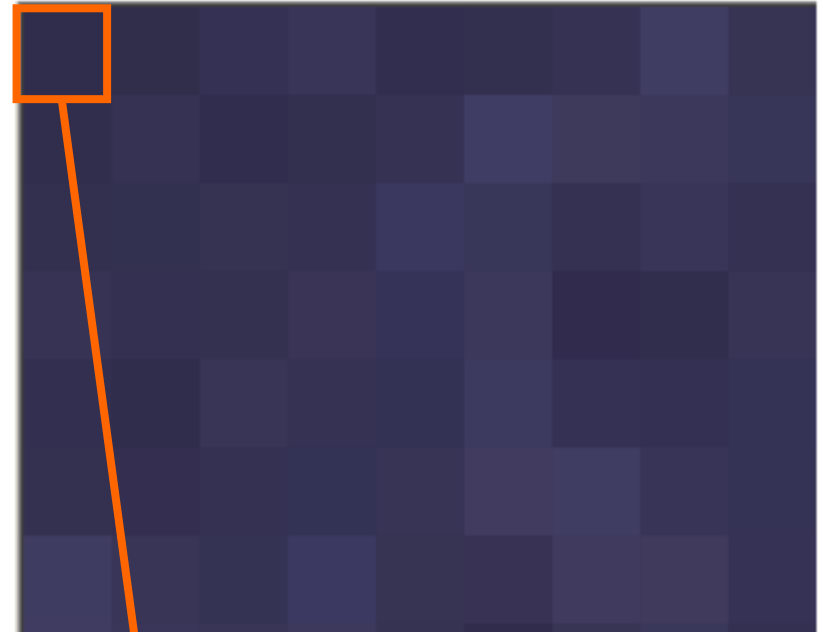
# Complemento em Imagens Coloridas

Imagem Complementar



(207, 210, 179)

Imagem Original



(48, 45, 76)

+

(255, 255, 255)

# Binarização

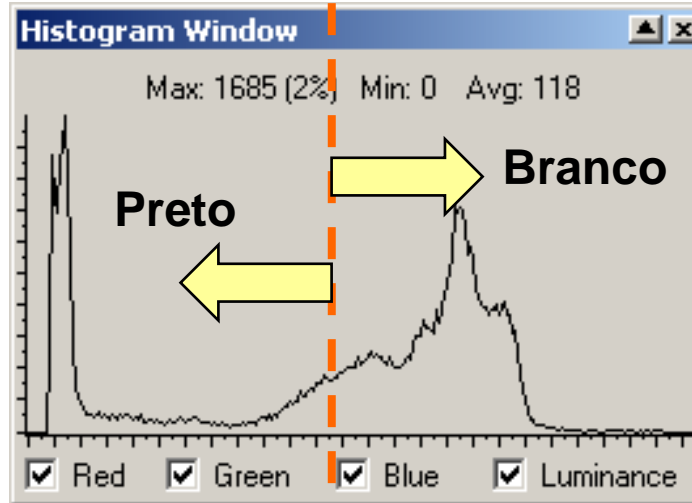
- Consiste em separar o histograma de uma imagem em regiões classificando-as como uma classe (0 – preto) ou outra (1 – branco)
- Em imagens preto e branco
  - quando o pixel apresentar um tom de cinza mais próximo do preto, ele é convertido para preto
  - quando apresentar um tom de cinza mais próximo do branco, ele é convertido para branco



# Binarização

- Algoritmo de Recorte

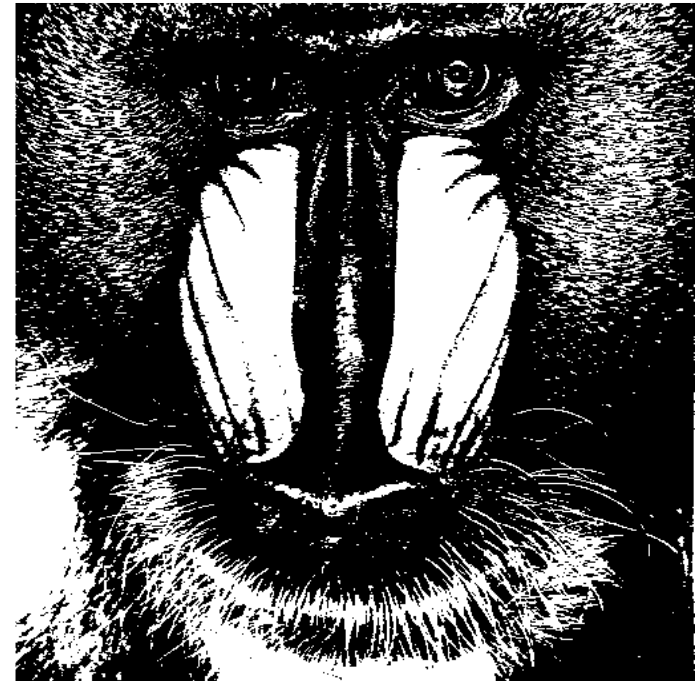
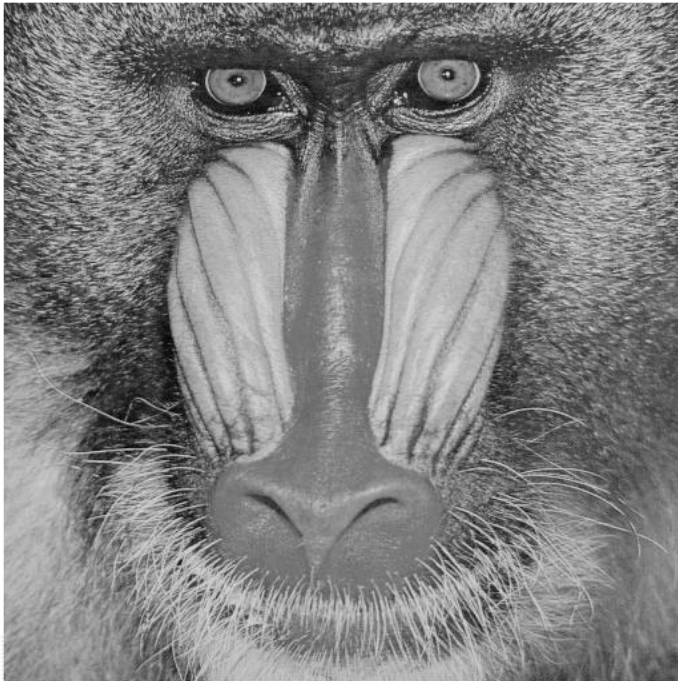
**Se  $\text{cor}(i) \leq 127$   
Então  $\text{cor}(i) = \text{Preto (0)}$   
Senão  $\text{cor}(i) = \text{Branco (255)}$**



**Valor de Corte = 127  
(Threshold, Limiar)**

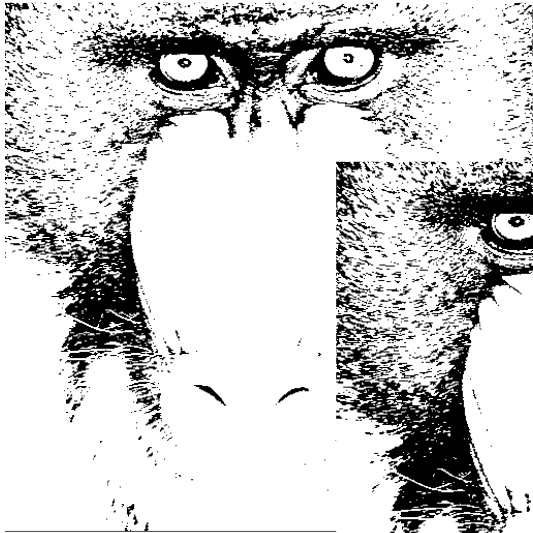
# Binarização

```
I = imread('baboon.bmp');  
I = (I > corte) % 150  
imshow(I);
```



# Binarização

Corte = 80



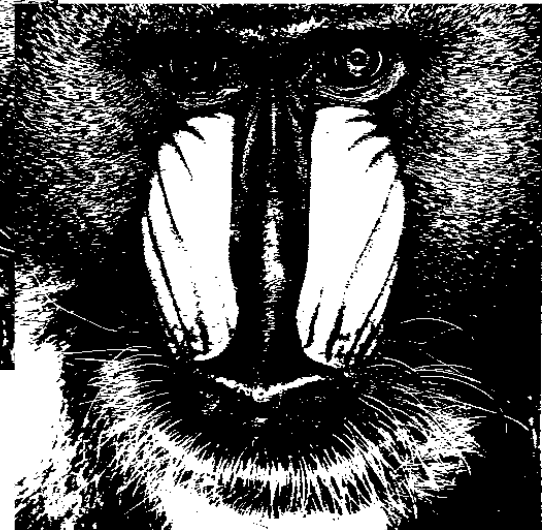
Corte = 100



Corte = 128

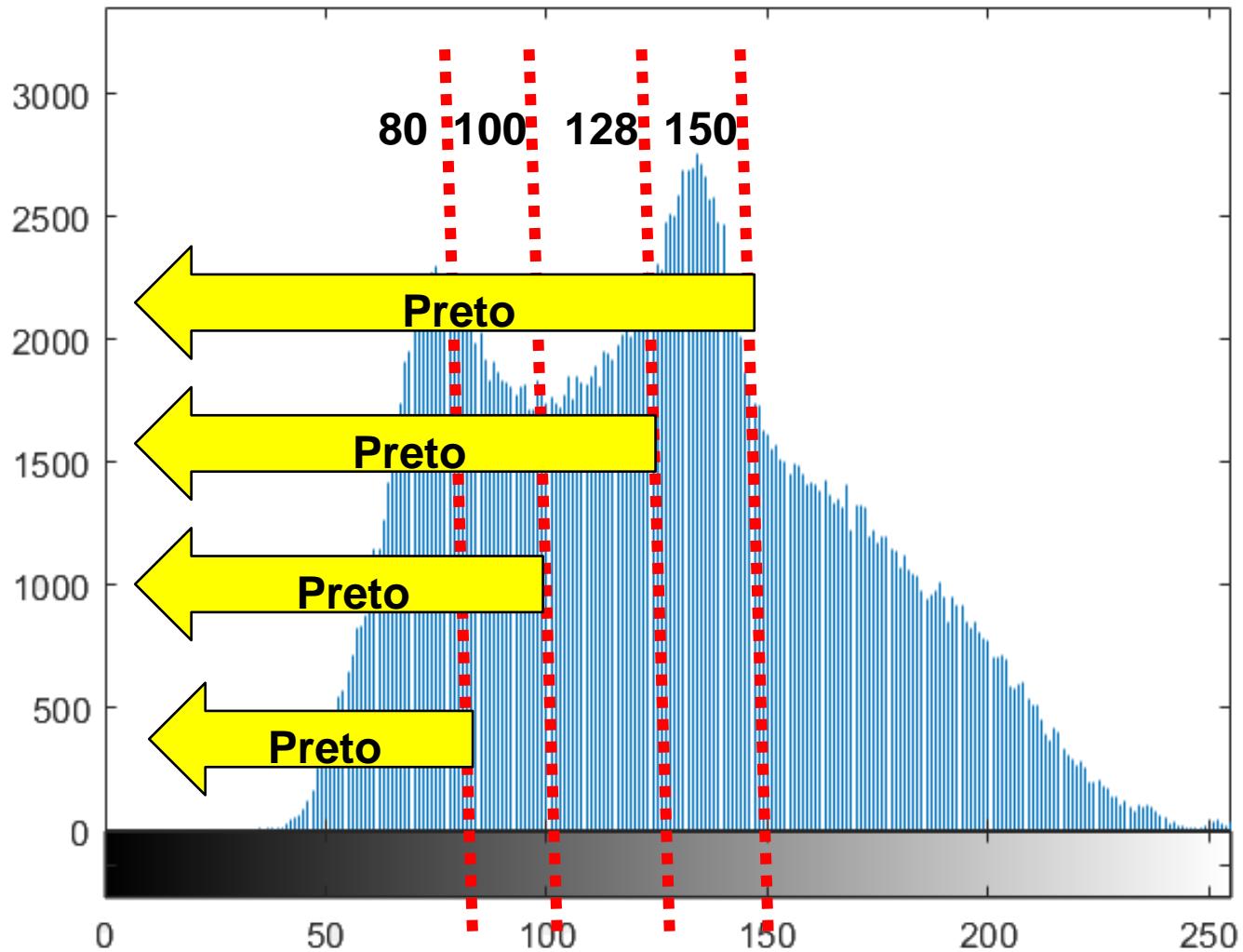


Corte = 150



Corte = limiar de binarização

# Binarização



# Brilho

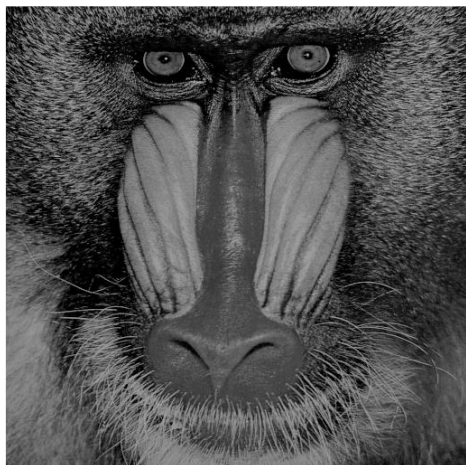
```
I = imread('baboon.bmp');  
I = I + lum;  
imshow(I);  
figure, imhist(I);
```



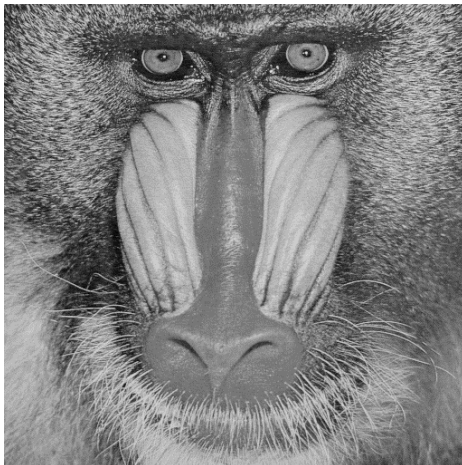


# Brilho

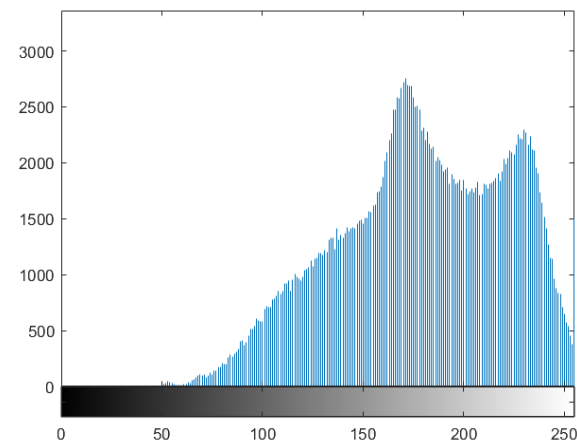
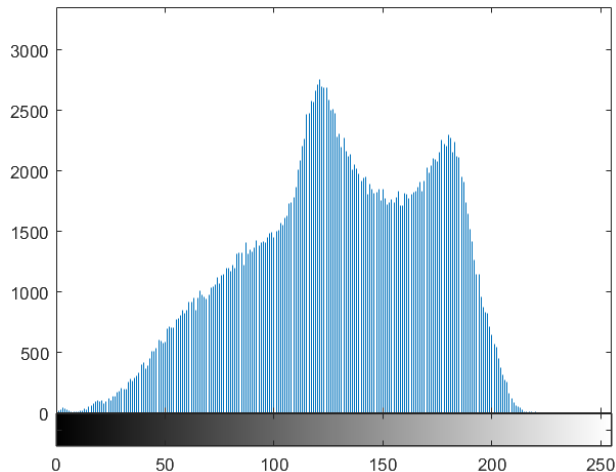
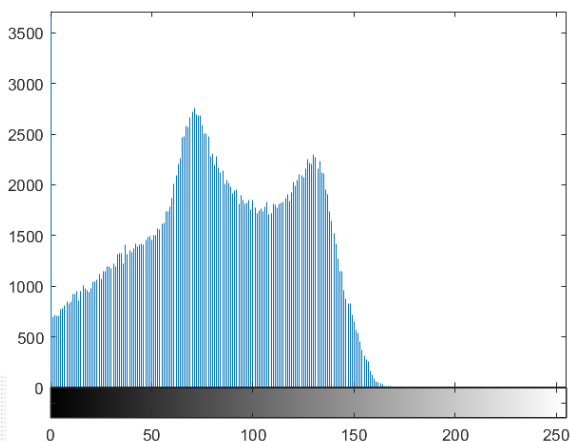
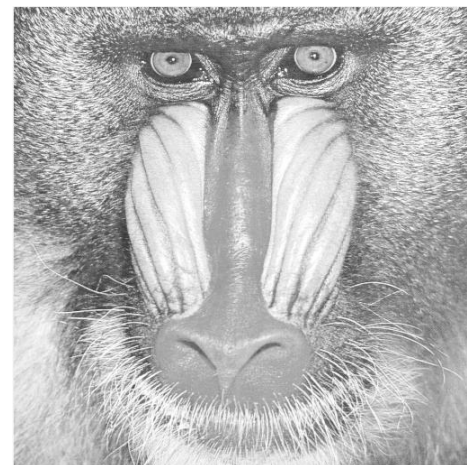
**Lum = -50**



**Lum = 0**



**Lum = +50**



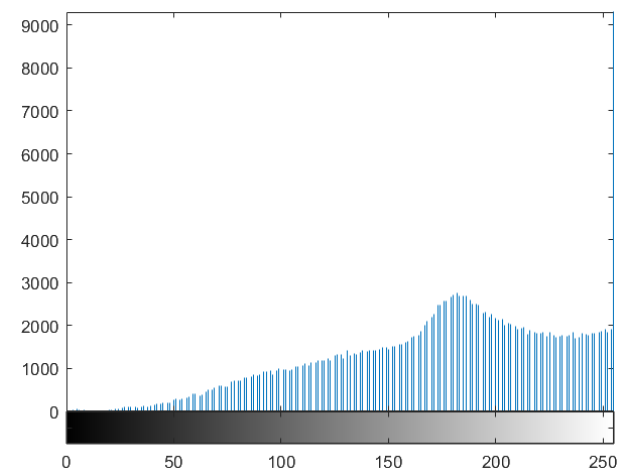
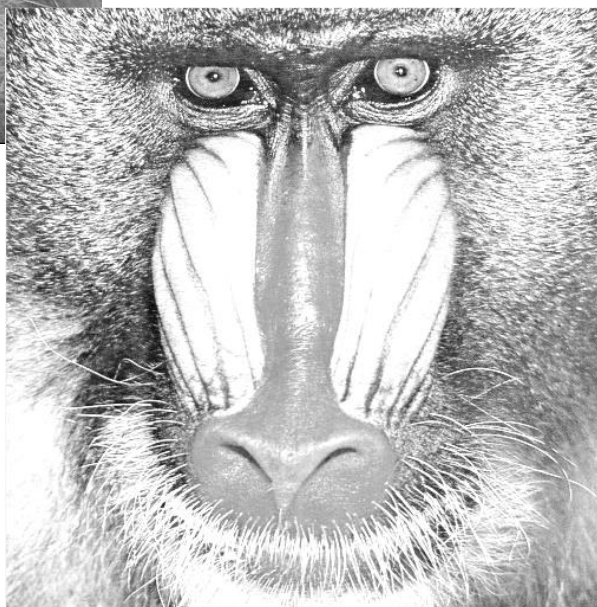
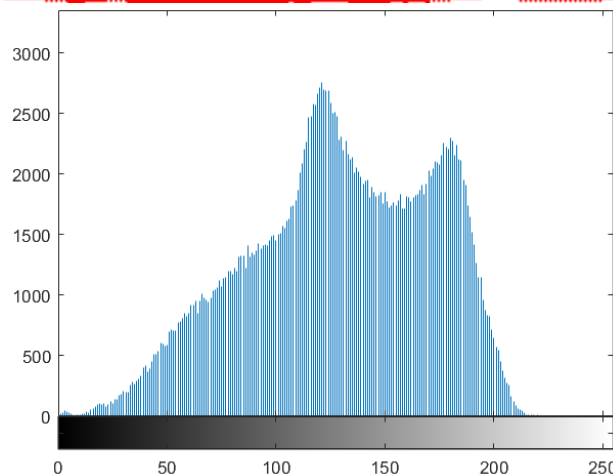
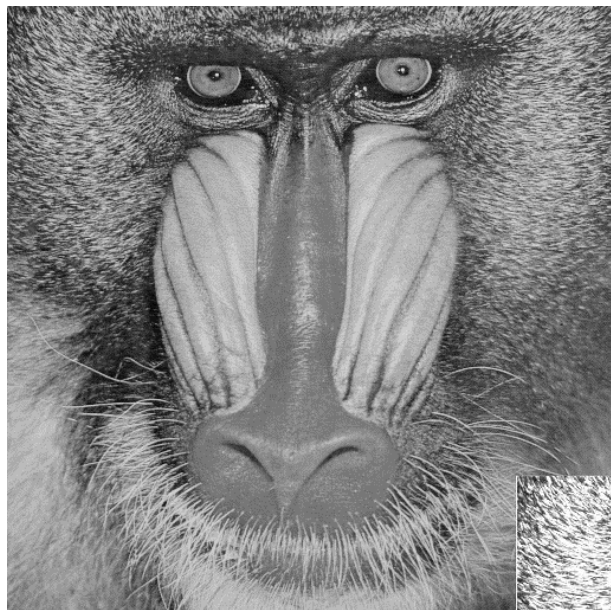
# Contraste

```
I = imread('baboon.bmp');  
I = contr*I;  
imshow(I);  
figure, imhist(I)
```



- Expansão de histograma: **contr** > 1
- Compressão de histograma: **0** < **contr** < 1

# Expansão de histograma

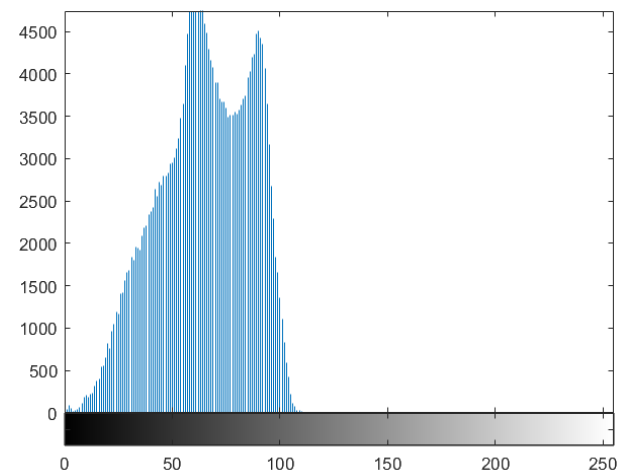
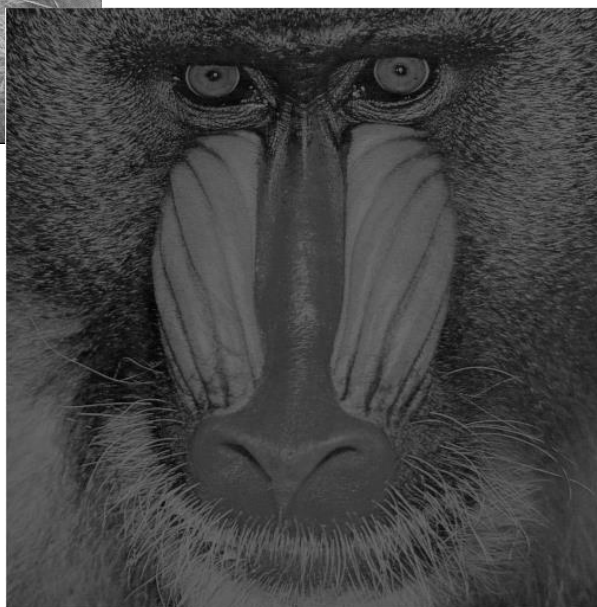
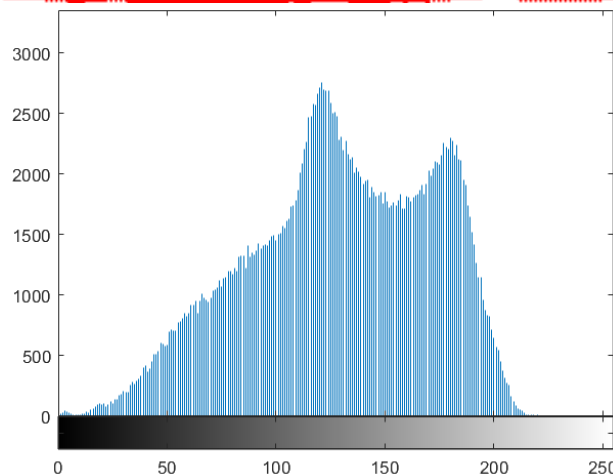
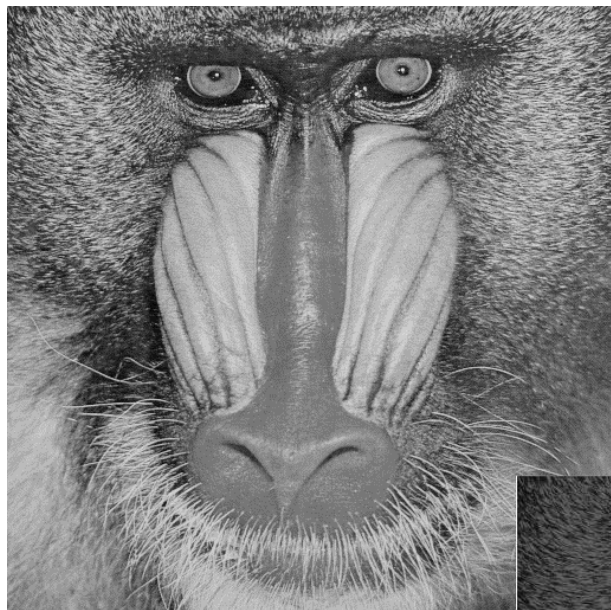


contr = 1,5

Aumenta o contraste



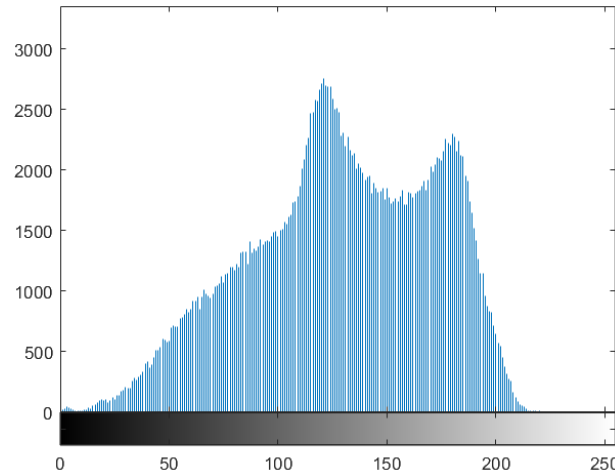
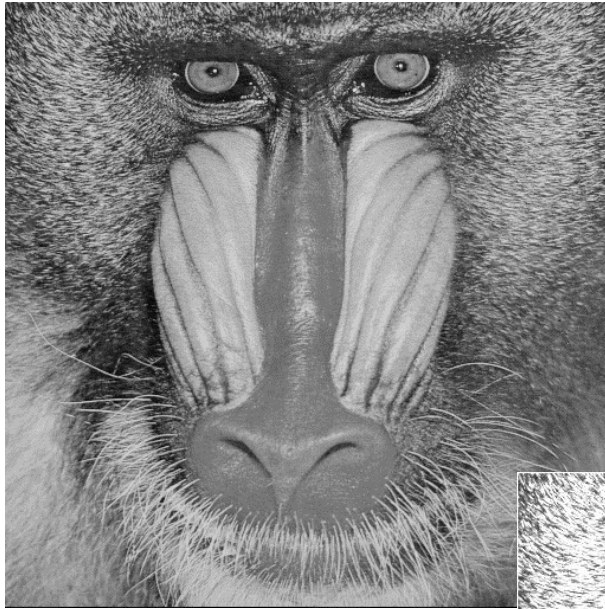
# Compressão de histograma



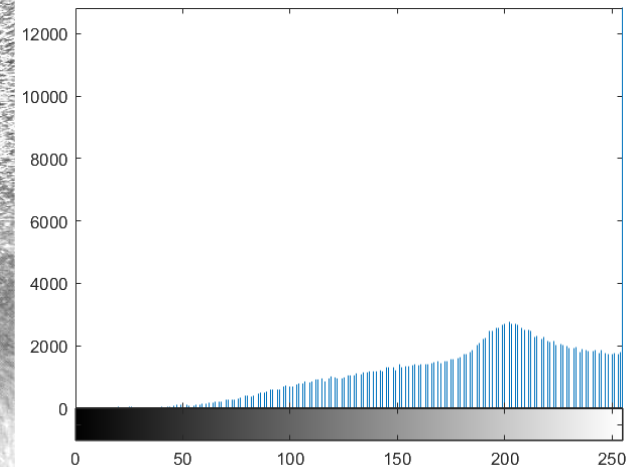
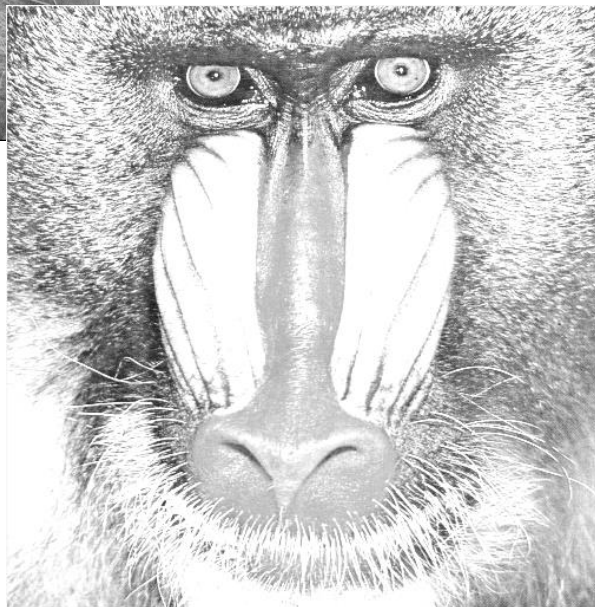
Contr = 0,5

Diminui o contraste

# Contraste + Brilho

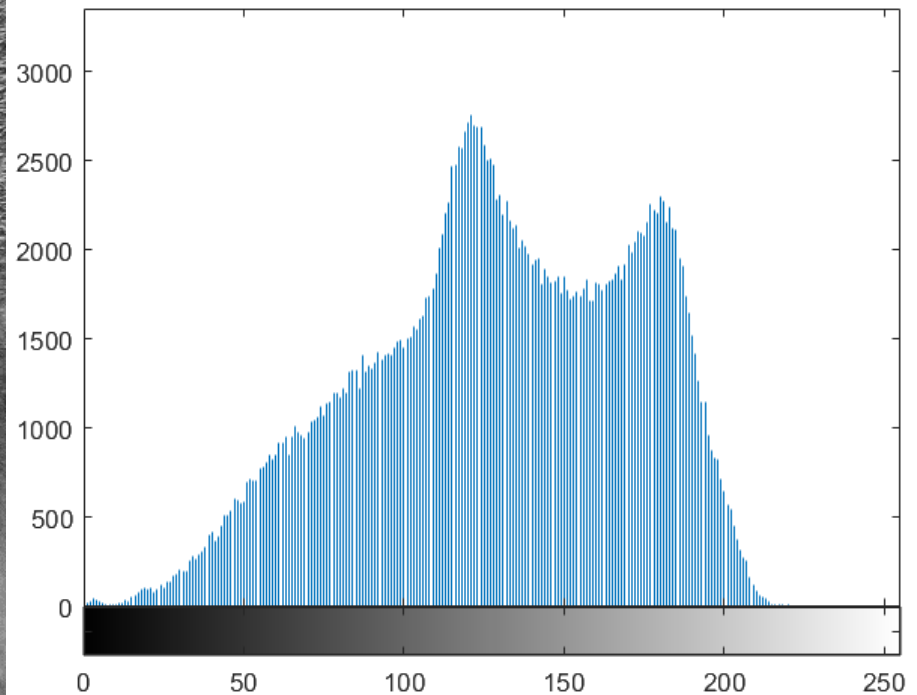
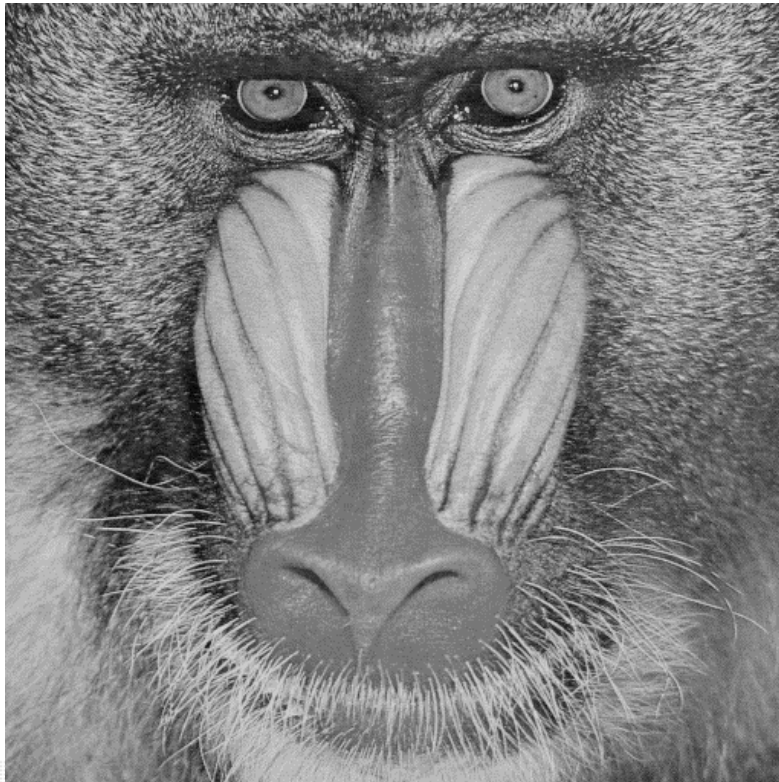


Contr = 1,2  
Brilho = +20



# Dois Picos do Histograma

- Operação comum em histograma
  - Encontrar os dois maiores picos do histograma



# Dois Picos do Histograma

- Dois maiores picos

- Primeiro pico:

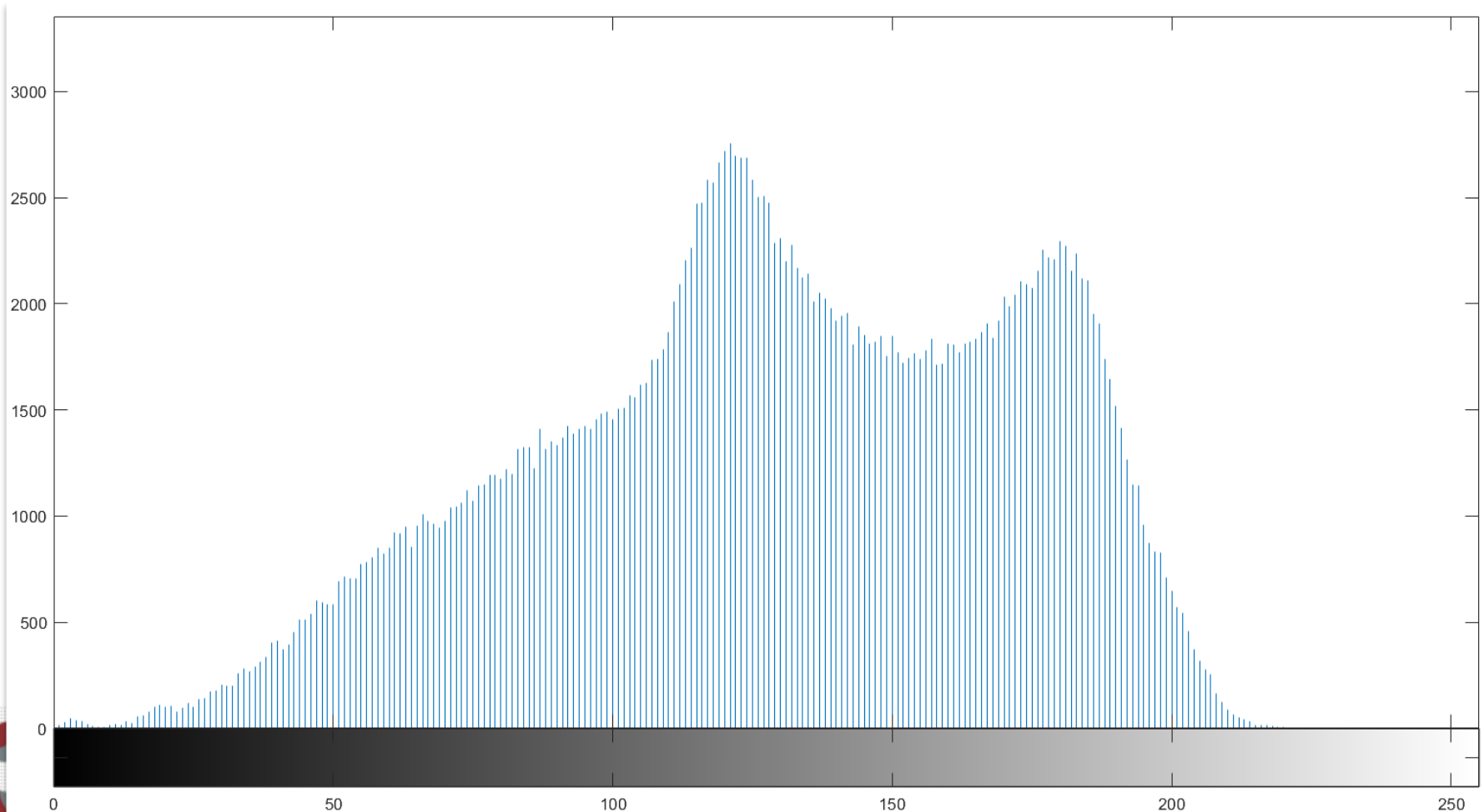
- Maior amplitude do histograma (posição  $j$ )

- Segundo pico:

- Não pode ser a segunda maior amplitude
  - Geralmente, está próxima da maior amplitude
- Solução:
- Segundo pico em:  $\max((k - j)^2 h[k])$ 
  - $h$  é o histograma
  - $0 \leq k \leq 255$  (para imagens com 256 tons de cinza)

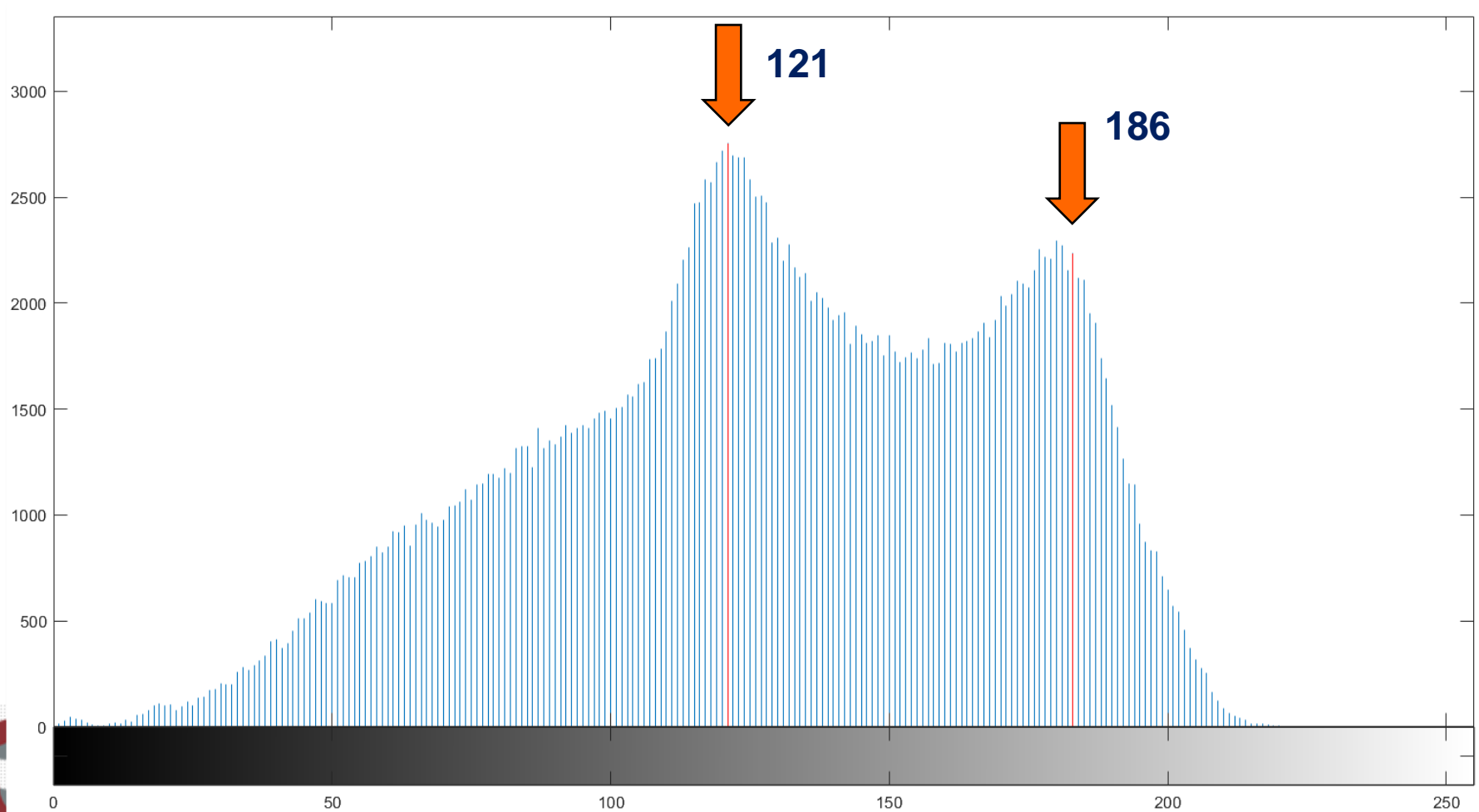
# Dois Picos do Histograma

- Dois maiores picos



# Dois Picos do Histograma

- Dois maiores picos





# Extração de Características

- A partir do histograma:
  - Média
  - Variância
  - Desvio Padrão
  - Entropia
  - Máximo/Mínimo
  - Maiores picos

# Extração de Características

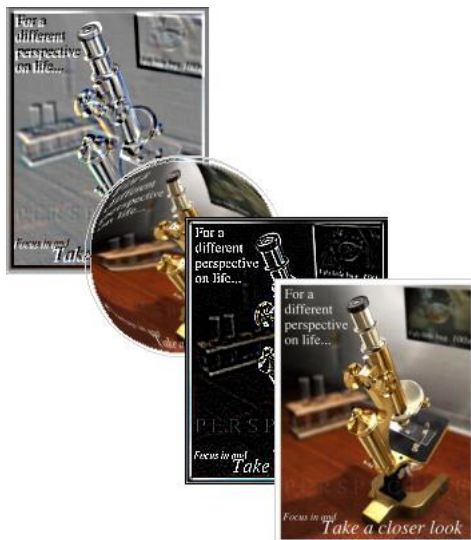
- Cálculo da Média
  - É a média do histograma, não a média dos valores (cores) que aparecem na imagem
  - Seja  $N$  o número de pixels da imagem
  - Seja  $h$  o histograma previamente calculado
  - Seja  $Prob$  a probabilidade de cada cor estar presente na imagem, ou seja,  $Prob = h[i]/N$
  - Média =  $\sum Prob(i).Cor(i)$



# Limiarização - *Thresholding*

---

Carlos Alexandre Mello



# Limiarização - *Thresholding*

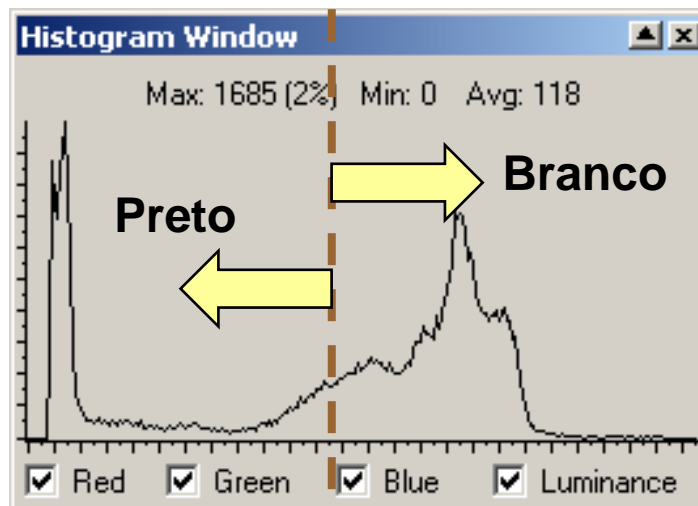
- Conversão de uma imagem para dois tons a partir de um dado ponto de corte (limiar)



# Limiarização

- Algoritmo de Recorte

**Se  $\text{cor}(i) \leq 127$   
Então  $\text{cor}(i) = \text{Preto}$   
Senão  $\text{cor}(i) = \text{Branco}$**

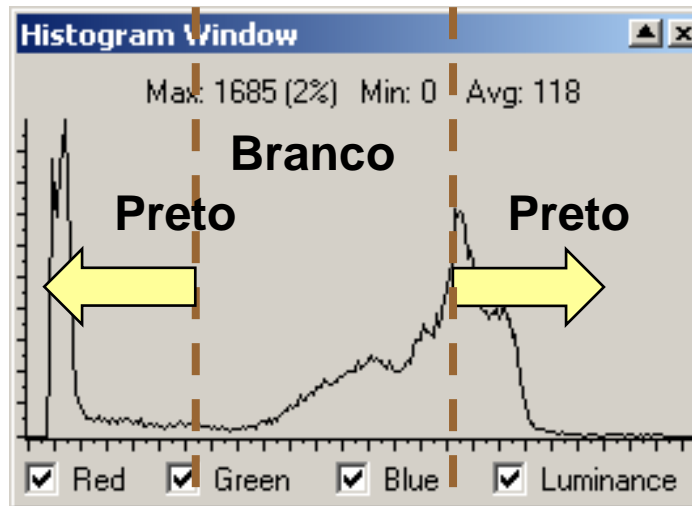


**Valor de Corte = 127  
(threshold, limiar)**

# Limiarização

- Algoritmo de Recorte (faixa de corte)

**Se  $50 < \text{cor}(i) < 100$   
Então  $\text{cor}(i) = \text{Branco}$   
Senão  $\text{cor}(i) = \text{Preto}$**



**Branco entre 50 e 100**

# Limiarização

- Tipos de Limiarização

- **Global**

- Um único valor de corte é definido para toda a imagem

- Vantagem: Velocidade
      - Desvantagem: Qualidade

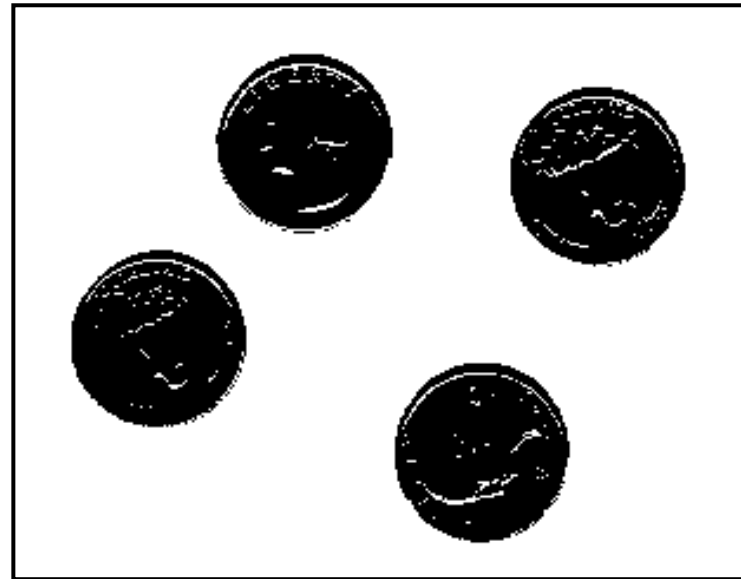
- **Local**

- Diferentes pontos de corte são definidos para diferentes regiões da imagem

- Vantagem: Qualidade
      - Desvantagem: Velocidade

# Limiarização

- Limiarização Ótima



**Objetos formados por tons bem diferentes do background:  
Fácil redução de cores com qualidade**

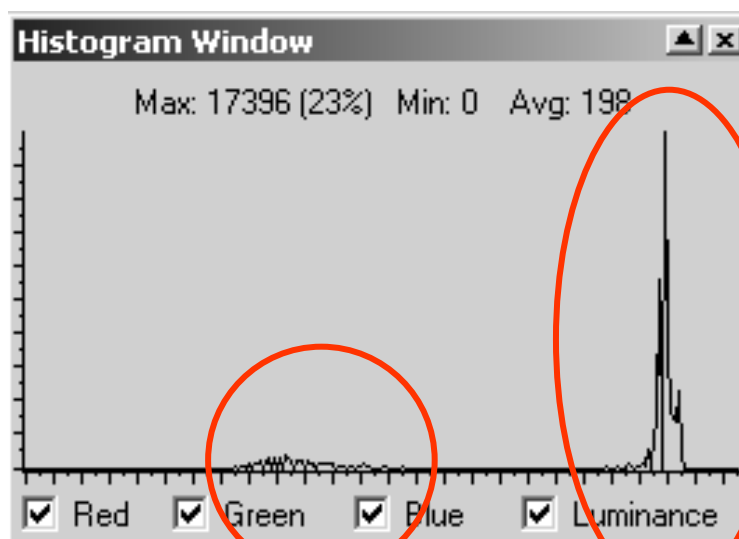


# Limiarização

- Limiarização Ótima



**Imagem original com cores bem distintas no Histograma**



**Região com os tons da moeda**

**Região com os tons do background**

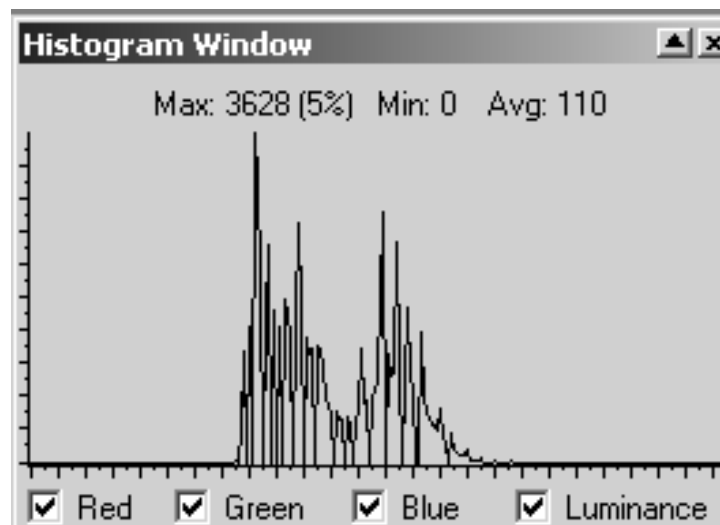
**Regiões bem separadas**

# Limiarização

- Problemas na Limiarização



**Problema: Imagem com baixo contraste**



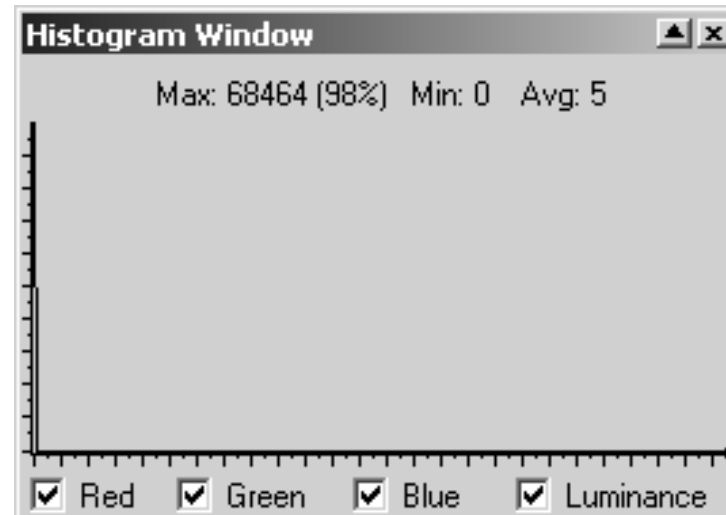
**É possível, por exemplo, separar automaticamente a menina do background?**

# Limiarização

- Problemas na Limiarização

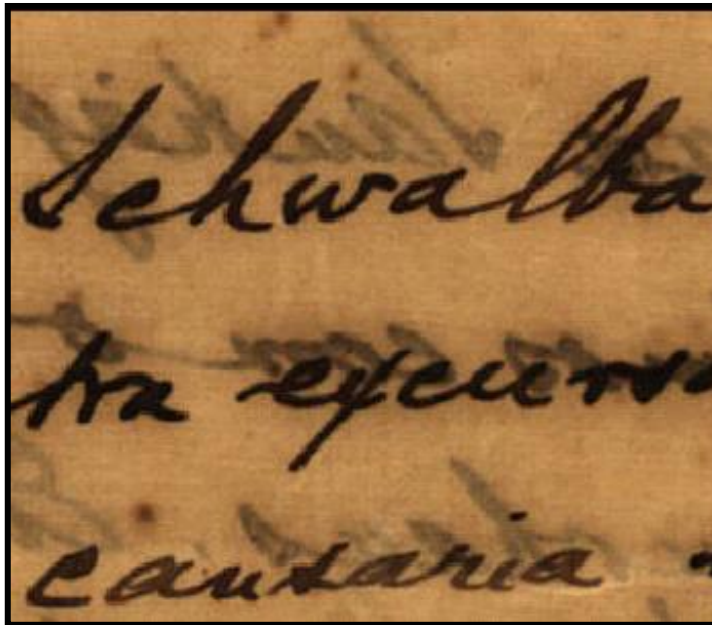


Resultado da Limiarização



# Limiarização

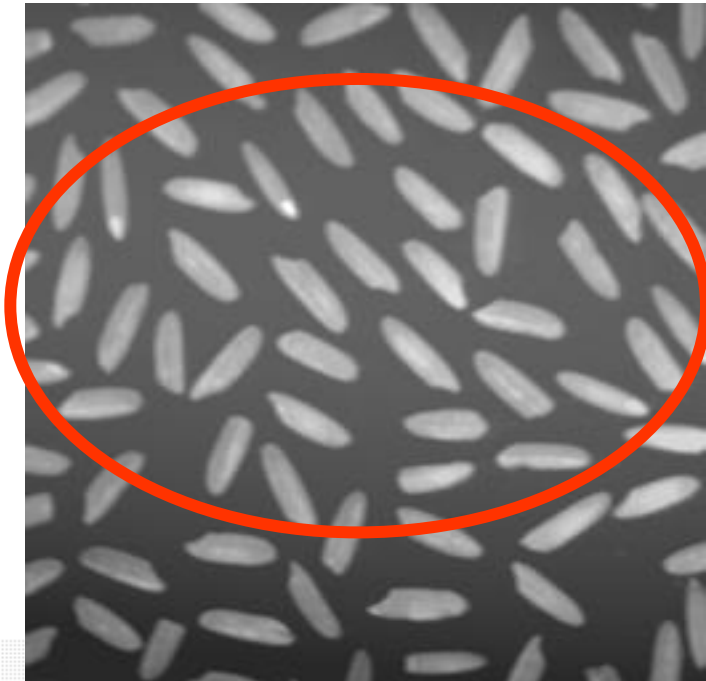
- Problemas na Limiarização



Algoritmo de threshold com  
valor de corte padrão

# Limiarização Local

- O valor de corte varia dentro de uma mesma imagem, dependendo de fatores da imagem
  - Exemplo: Imagem com diferentes focos de iluminação



Área iluminada

# Limiarização

- Os algoritmos também se dividem de acordo com as características procuradas na imagem para gerar o ponto de corte:
  - Histograma
  - Entropia
  - Agrupamento
  - Maximização/Minimização de funções
  - Atributos de objetos
  - Métodos espaciais



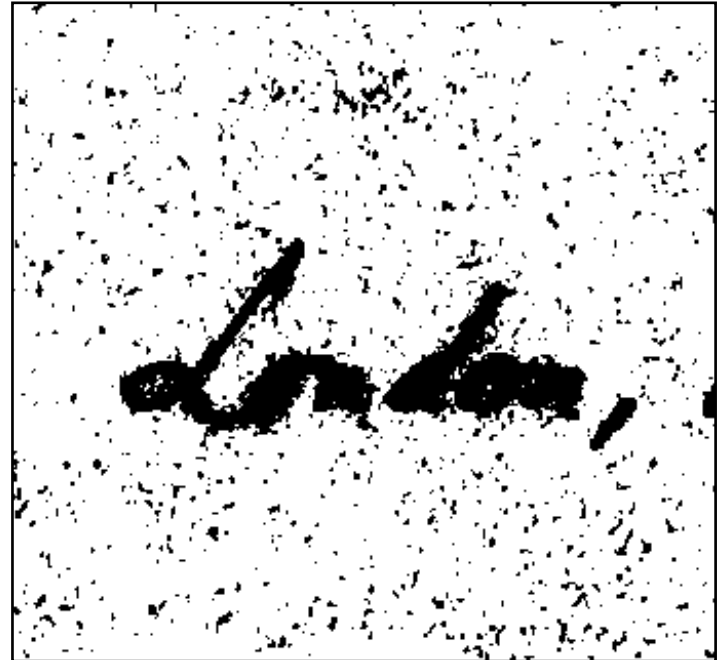
# Tom de Cinza Médio

- *Mean Grey Level*
- O ponto de corte é, simplesmente, o ponto médio entre a maior e a menor cor presentes no histograma
  - Não é a média do histograma como calculado antes

# Porcentagem de Preto

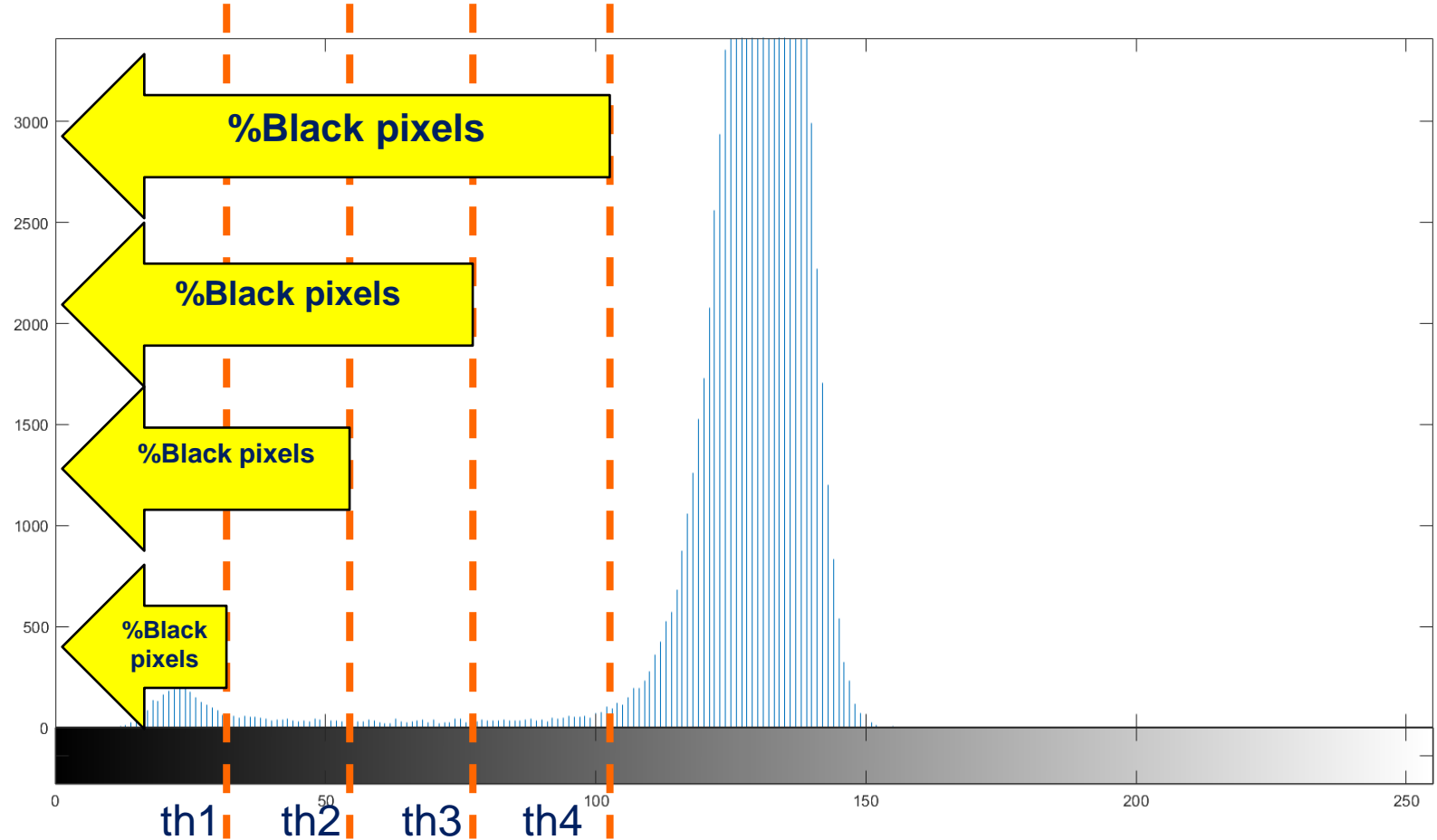
- Define-se *a priori* a quantidade de tons pretos esperados na imagem final
- O ponto de corte é variado até que essa porcentagem seja encontrada
- Exemplo: Valor esperado para imagens de **documentos** é de 10% de preto
- Busca e análise feitas no histograma da imagem

# Porcentagem de Preto



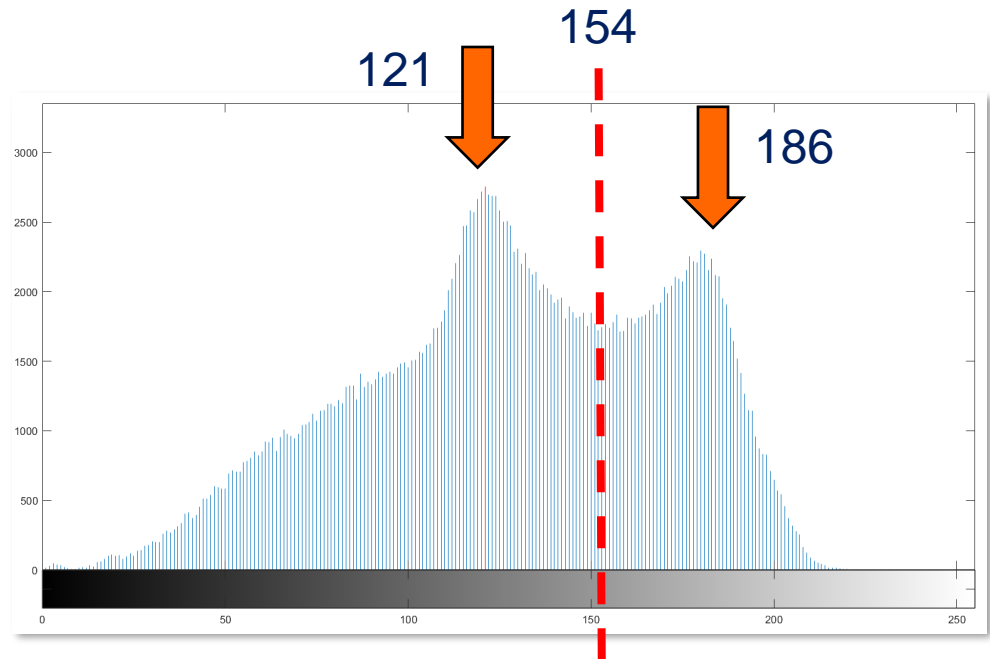
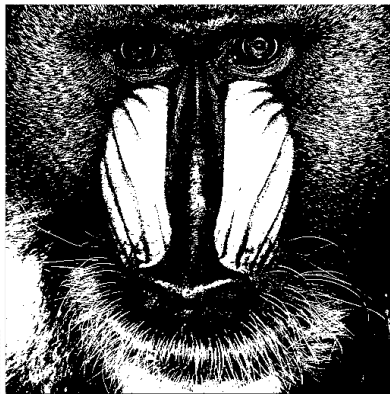
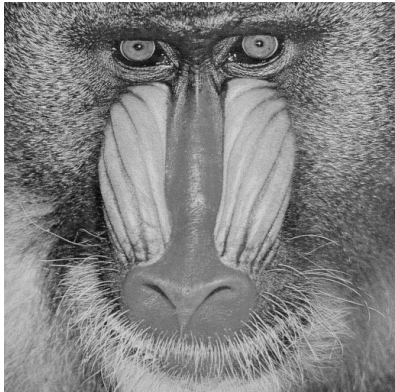
th = 119 (10% de preto)

# Porcentagem de Preto



# Dois Picos (*Two Peaks*)

- O ponto de corte é o ponto mais baixo entre dois picos
- Localização dos picos como apresentado anteriormente
- Exemplo anterior:



# Seleção Iterativa (Thrussel)

- Um *threshold* inicial é atribuído e ajustado através de leituras sucessivas da imagem
  - Considera 2 classes: *foreground* e *background*
  - *Threshold* inicial ( $T_0$ ): tom de cinza médio
  - Calcula então o tom de cinza médio abaixo e acima desse  $T_0$ :  $T_b$  e  $T_w$
  - Um novo *threshold* é estimado como  $(T_b + T_w)/2$
  - O processo continua até que nenhuma mudança ocorra no ponto de corte



# Otsu

- A operação de limiarização é considerada como sendo o particionamento dos pixels de uma imagem de  $L$  níveis de cinza em duas classes,  $C_0$  e  $C_1$ , que podem representar o objeto e o fundo, ou vice-versa, sendo que esta partição se dará no nível de cinza  $t$ 
  - Desta forma, teremos  $C_0 = \{0, 1, \dots, t\}$  e  $C_1 = \{t + 1, t + 2, \dots, L\}$

# Otsu

- Seja  $\sigma^2_W$  a variância inter-classe,  $\sigma^2_B$  a variância entre classes e  $\sigma^2_T$  a variância total:

$$\mu_b = \frac{\mu_t(t)}{w_0(t)} \quad \mu_w = \frac{\mu_t - \mu_t(t)}{1 - w_0(t)}$$

$$\sigma_b^2 = w_0(t)w_1(t)[\mu_b - \mu_w]^2 \quad \sigma_t^2 = \sum_{i=0}^{255} (i - \mu_t)^2 p(i)$$

$$\sigma_w^2(t) = \sigma_t^2 - \sigma_b^2$$

$$w_0(t) = \sum_{i=0}^t p(i) \quad \mu_t(t) = \sum_{i=0}^t i \cdot p(i) \quad \mu_T = \sum_{i=t+1}^{255} i \cdot p(i)$$

$$w_1(t) = 1 - w_0(t) \quad p(i) = \frac{cor(i)}{N}$$

# Otsu

- O valor ótimo do limiar pode ser encontrado pela maximização da função  $\eta(t) = \sigma_b^2(t) / \sigma_T^2$
- Ou seja, a relação entre a variância entre classes e a variância total
- O valor máximo de  $\eta$ , pode ser usado como medida para avaliar a separabilidade das classes  $C_0$  e  $C_1$  na imagem original ou a bimodalidade do histograma

# Otsu

- Esta é uma medida bastante significativa pois é invariante para transformações afins da escala de níveis de cinza, sendo unicamente determinada dentro do intervalo  $0 \leq \eta \leq 1$ 
  - O limite inferior (zero) é obtido quando e somente quando uma dada imagem tenha um único e constante nível de cinza, e o limite superior (um) é obtido quando e somente quando imagens de dois valores são dadas

# Otsu

- O método se caracteriza por sua natureza não paramétrica e não supervisionada de seleção de limiar e tem as seguintes vantagens:
  - O processo como um todo é muito simples;
  - Um limiar ótimo é selecionado de forma automática e estável, baseado em propriedades globais do histograma;
  - Viabiliza a análise de outros aspectos importantes, tais como estimativa dos níveis médios das classes, avaliação da separabilidade das classes, etc;

# Otsu

- O método se caracteriza por sua natureza não paramétrica e não supervisionada de seleção de limiar e tem as seguintes vantagens:
  - **O método é extremamente genérico**, podendo ser utilizado em outros casos de classificação não supervisionada no qual um histograma de alguma característica discriminativa que classifique objetos esteja disponibilizado.



# Usando Pixels de Borda

- Baseado no operador Laplaciano (**Filtro Passa Alta**)

0	1	0
1	-4	1
0	1	0

- Após a filtragem, a imagem é binarizada
- O histograma da imagem filtrada é calculado, considerando apenas aqueles pixels com “alto” valor de laplaciano
  - “alto” é um percentual pré-definido
  - A binarização é feita usando esse histograma

# Usando Pixels de Borda

- Funciona como uma mistura de dois outros algoritmos
  - Como no algoritmo de porcentagem de preto, procura o ponto de corte que passaria para preto o percentual pré-definido da imagem
  - Mas, no caso, esse ponto definirá apenas a parte do histograma que será processada
  - Nesse pedaço de histograma, usa o algoritmo de dois picos para achar o ponto de corte

# Média Móvel (*Moving Average*)

## P. Wellner

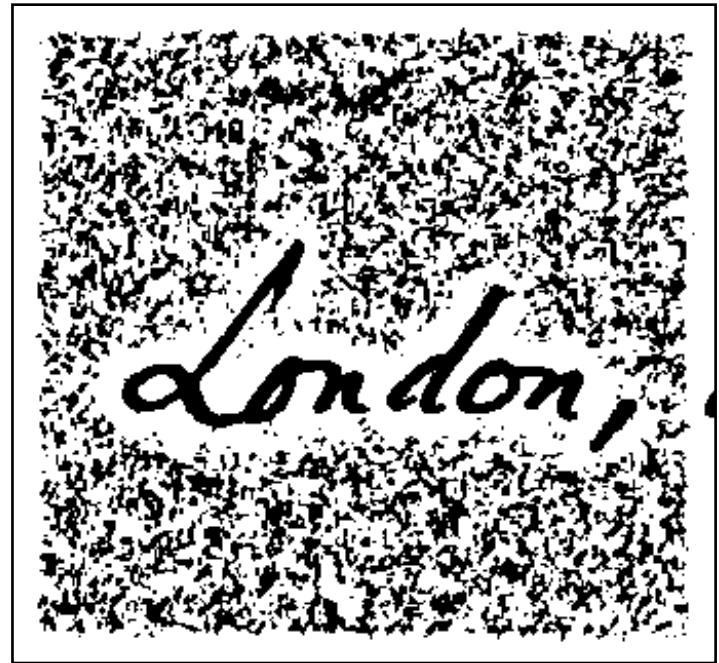
- Limiarização Local
- Define um ponto de corte para cada pixel
- Uma média móvel é apenas o tom de cinza médio dos últimos  $n$  pixels
- Nesse caso, a imagem pode ser considerada armazenada como um vetor
- Para o pixel  $i+1$  com cor  $cinza_{i+1}$ :
  - $M_{i+1} = M_i - M_i/n + cinza_{i+1}$

# Niblack

- Limiarização Local
- Imagem dividida em janelas quadradas ( $N \times N$ ) e cada janela é avaliada em separado
- Calcula-se média ( $m$ ) e desvio padrão ( $d$ ) de cada janela
- O ponto de corte ( $th$ ) é definido por:
  - $th = m + bias \cdot d$
  - $bias$  = peso atribuído
  - Não há definição de valor específico para esse *bias*

# Niblack

- Exemplo:



Janela = 31; Bias = -0.8

# Sauvola

- Melhoria no algoritmo de Niblack para imagens com problemas de iluminação
- Nova variável para modificar o efeito do desvio padrão
  - $R = 128$
- O ponto de corte é definido como:
  - $th = m + bias^*(d/R)$

# White

- Compara o tom de cinza de cada *pixel* da imagem com a média dos valores dos seus vizinhos numa janela
- Para peso igual a 2, se o *pixel* for mais escuro que a média, ele é classificado como objeto, senão é background
  - Se  $\text{média} > \text{cor\_pixel} \times \text{peso}$ , então  $\text{cor\_pixel} \leftarrow \text{Preto}$
  - Senão  $\text{cor\_pixel} \leftarrow \text{Branco}$



# Bernsen

- O limiar é escolhido como a média entre o mínimo e o máximo tom de cinza da vizinhança do *pixel*
- Se o contraste for menor que certo limiar dado, então o *pixel* é considerado da mesma classe que seus vizinhos, objeto ou *background*, dependendo do valor do limiar
  - O contraste é calculado como a diferença entre o máximo e o mínimo nível de cinza dado

# Sugestões de Leitura

- A Novel Contrast Enhancement Technique using Gradient-Based Joint Histogram Equalization
  - Circuits, Systems, and Signal Processing, 2021
  - <https://doi.org/10.1007/s00034-021-01655-3>
- A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems
  - Special Issue on Hybridization between Neural Computing and Nature Inspired Algorithms for Solving Multi-Criteria Decision-Making Problems, 2021
  - <https://doi.org/10.1007/s00521-020-04820-y>
- A novel Black Widow Optimization algorithm for multilevel thresholding image segmentation
  - Expert Systems with Applications, 2021
  - <https://doi.org/10.1016/j.eswa.2020.114159>
- An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation
  - Knowledge-Based Systems, 2021
  - <https://doi.org/10.1016/j.knosys.2021.107348>

# Referência

- Imprescindível para qualquer trabalho que envolva limiarização:
  - Sezgin, M.; Sankur, B., “Survey over image thresholding techniques and quantitative performance evaluation”. Journal of Electronic Imaging, 2004. Vol.13, p.146-165, 2004.

# Referência

- Histograma + Binarização no Colab
  - [https://colab.research.google.com/drive/1tp\\_W6XKvCh5-\\_a2KUaFtNnuVHCxpAFVz#scrollTo=5yzNrIUTVgQg](https://colab.research.google.com/drive/1tp_W6XKvCh5-_a2KUaFtNnuVHCxpAFVz#scrollTo=5yzNrIUTVgQg)