

# **Fundamentos de Processamento Gráfico**

## **Aula 3**

### **Introdução ao Processamento de Imagens**

**Profa. Fátima Nunes**  
**Prof. Helton Bíscaro**

- **Para esta aula você deve dominar:**
  - conceitos sobre matrizes e operações sobre matrizes;
  - conceitos básicos sobre estatística e matemática: média, mediana, somatórios, valores absolutos;
  - conceitos básicos sobre algoritmos: estruturas condicionais, estruturas de controle;
  - Implementação de algoritmos em qualquer linguagem de programação

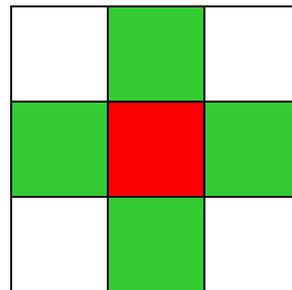
Sobre o pixel são definidas  
algumas relações básicas:  
vizinhança, conectividade,  
distância ...

# Questões

- O que é vizinhança de um pixel?

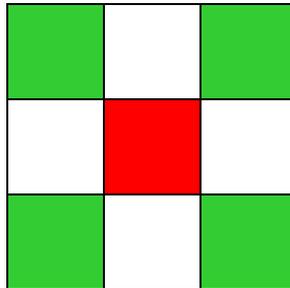
- **Vizinhança**

- seja  $p$ , um pixel nas coordenadas  $(x,y)$ . A vizinhança de 4 ( ou  $N_4(p)$  ) de um pixel é composta por seus vizinhos na horizontal e na vertical, cujas coordenadas são:  $(x+1,y)$ ,  $(x-1,y)$ ,  $(x,y+1)$ ,  $(x,y-1)$ .



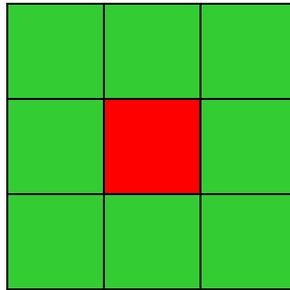
- **Vizinhança**

- A vizinhança diagonal ( ou  $N_D(p)$  ) de um pixel é constituída por seus vizinhos com coordenadas:  $(x+1,y+1)$ ,  $(x+1,y-1)$ ,  $(x-1,y+1)$ ,  $(x-1,y-1)$ .



- **Vizinhança**

- A vizinhança de 8 ( ou  $N_8(p)$  ) é o conjunto de todos os pixels vizinhos, ou seja:  $N_4(p) \cup N_D(p)$ .



# Questões

- Para que serve a mensuração da distância entre pixels?
- Como se faz?

- Distância entre pixels

- Geralmente é um valor mensurável e:

- $d(x,y) = 0$ , se  $x = y$ ;
- $d(x,y) = d(y,x)$ ;
- $d(x,y) + d(y,z) \geq d(x,z)$ .

- Diversas fórmulas empregadas para a definição de distância.

- São definidas e adaptadas fórmulas para aplicações específicas.

- **Distância entre pixels**

- Algumas das métricas mais conhecidas, aplicadas para dois pixels  $p=(x_1,y_1)$  e  $q=(x_2,y_2)$ :

- Distância Euclidiana:

$$d(p,q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- Distância “City Block”:

$$d(p,q) = |x_1 - x_2| + |y_1 - y_2|$$

- Distância “Chessboard”:

$$d(p,q) = \max \{ |x_1 - x_2|, |y_1 - y_2| \}$$

# Definições

- Qual o valor dessas distâncias para os pixels destacados em vermelho?

1	2	3
4	5	6
7	8	9

- $d(p, q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
- $d(p, q) = |x_1 - x_2| + |y_1 - y_2|$
- $d(p, q) = \max \{ |x_1 - x_2|, |y_1 - y_2| \}$

- Processamento de imagens ▶ três níveis.
  - **Processamento de baixo nível:** responsável pela remoção de dados indesejáveis e realce de dados importantes;
  - **Processamento em nível médio:** parte do processamento que identifica formas significantes. A esse processo damos o nome de “segmentação”.
  - **Processamento em alto nível:** responsável pela ligação da imagem com algum banco de conhecimento.

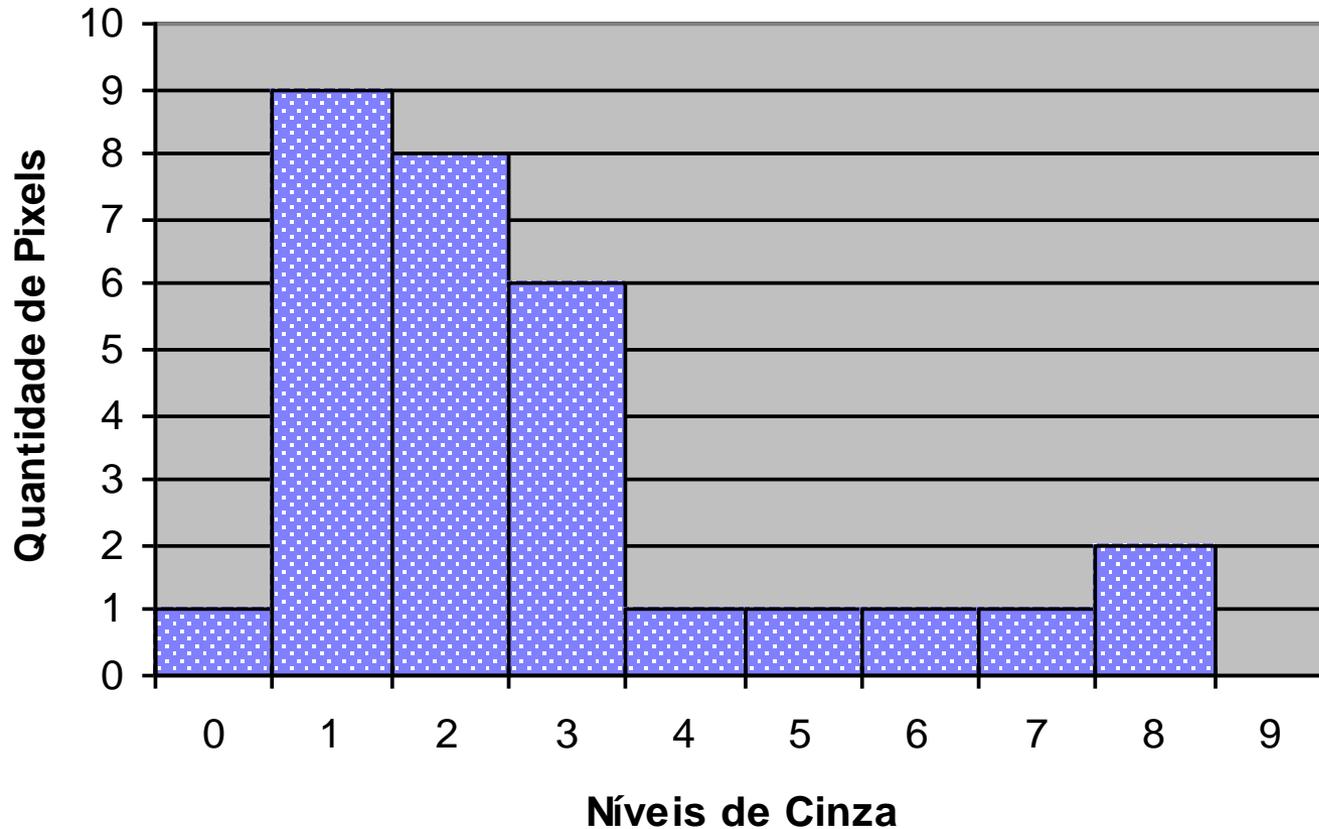
# Questões

- O que é histograma de uma imagem?

- **Histograma de uma imagem**

- Função que fornece a frequência de cada nível de cor na imagem.
- Valor do histograma em um nível de cinza
  - ▶  $H(k)$  ▶ quantidade de pixels da imagem com aquele nível de cinza.
- Útil para alterações globais na imagem.
- Impossível aplicá-lo em processamentos que necessitem de conhecimento sobre a localização de pixels.

## Exemplo de Histograma de uma imagem

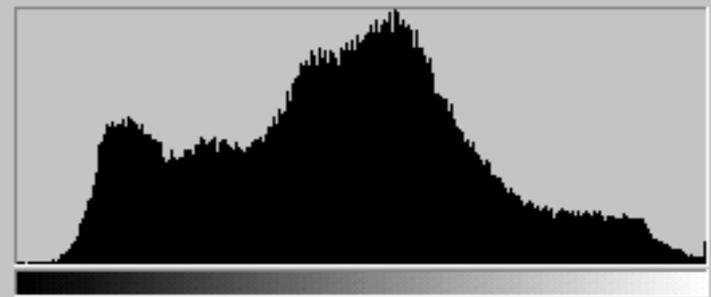


# Processamento de Baixo Nível

## Exemplo de Histograma de uma imagem



Channel: Gray



Mean: 119,43

Level: 100

Std Dev: 51,52

Count: 1162

Median: 122

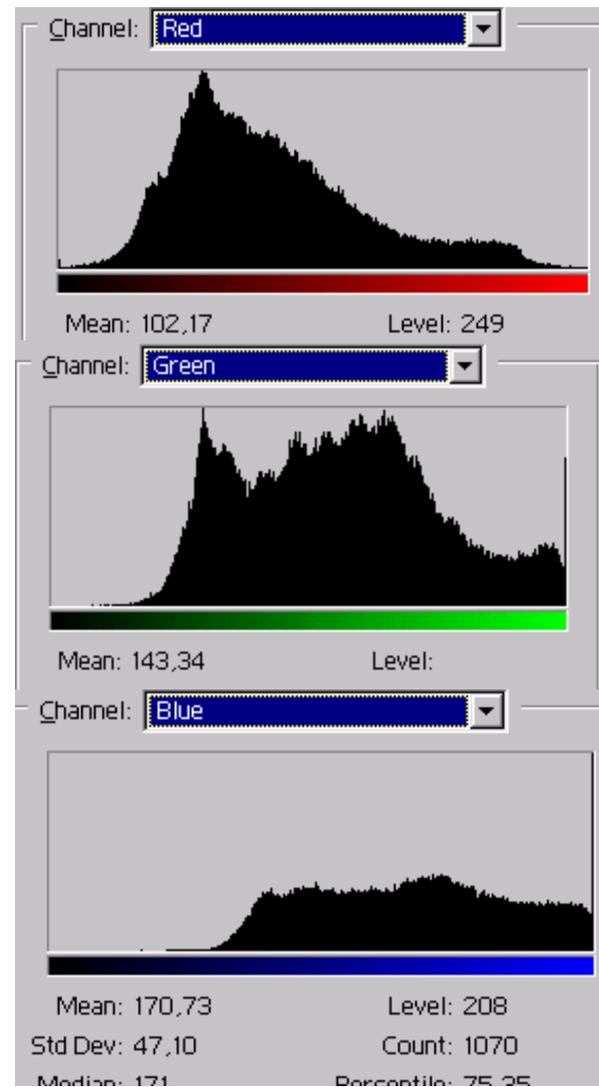
Percentile: 34,42

Pixels: 187500

Cache Level: 1

# Processamento de Baixo Nível

Exemplo de Histograma de uma imagem colorida



## Histograma de uma imagem

Algoritmo

## Histograma de uma imagem

Várias técnicas de processamento de baixo nível são aplicadas com base no histograma da imagem e/ou acarretam alterações no histograma.

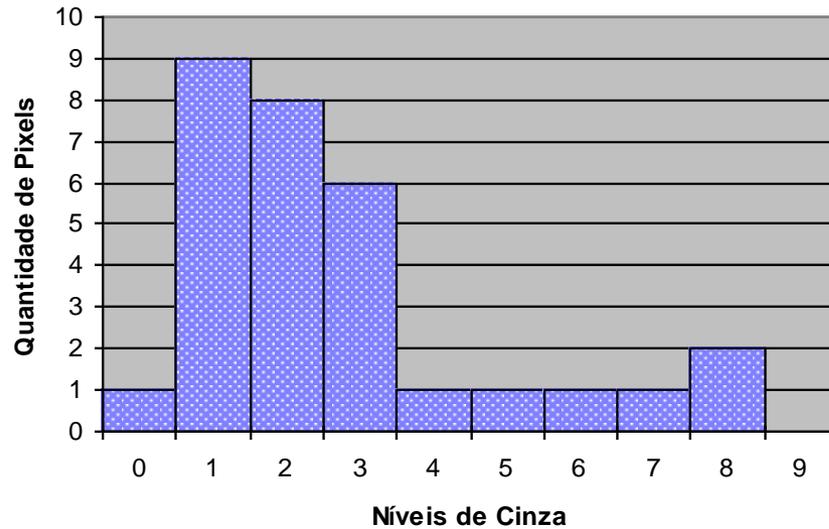
- **Alterações globais no brilho**

- Para tornar uma imagem mais clara ou mais escura ▶ soma ou subtração de uma constante em todos os pixels da imagem.
- Acarreta **alteração no histograma**

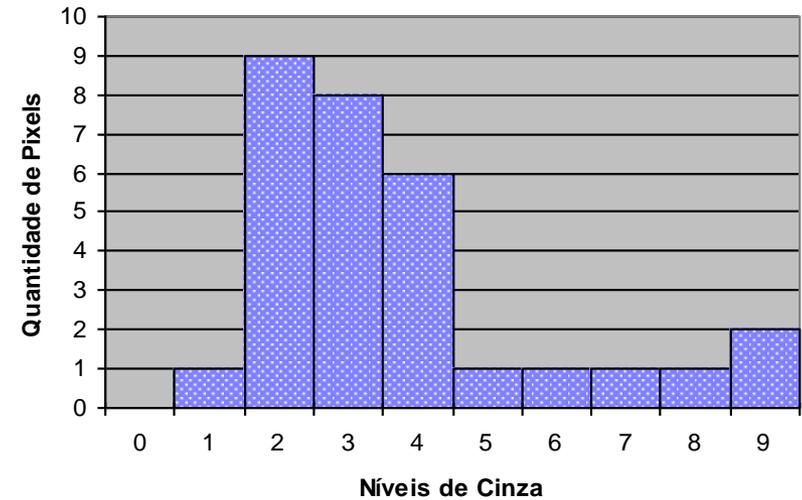
# Processamento de Baixo Nível

- Alterações globais no brilho

Histograma antes da alteração no brilho



Histograma com alteração na intensidade no brilho



# Processamento de Baixo Nível

## Alterações globais no brilho



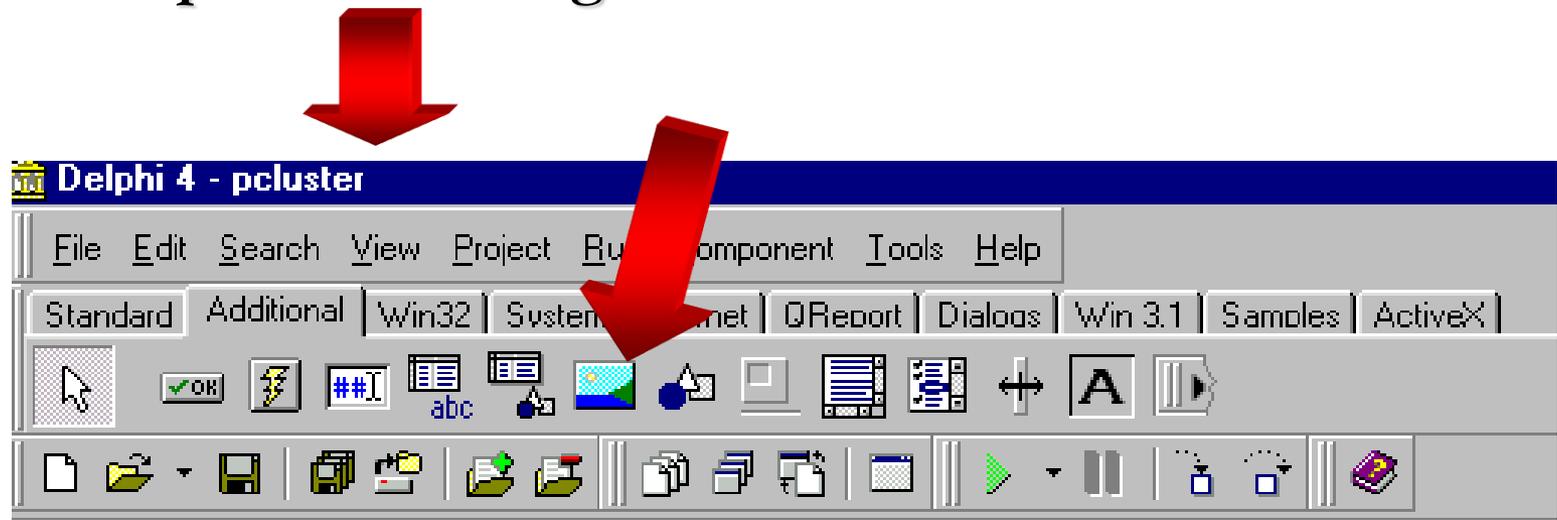
Mean: 119,43      Level: 100  
Std Dev: 51,52      Count: 1162  
Median: 122      Percentile: 34,42



Mean: 192,81      Level:  
Std Dev: 45,42      Count:  
Median: 199      Percentile:  
Pixels: 187500      Cache Level: 1

- ⇒ *Como manipular uma imagem?*
- ⇒ *Exemplo em linguagem Delphi*

⇒ **Componente TImage**



⇒ **Componente  
TImage**



The Object Inspector window displays the properties for a component named "AuxImagem: TImage". The "Properties" tab is selected, showing a list of properties and their values. The "Name" property is highlighted in blue.

Property	Value
Align	alNone
+Anchors	[akLeft,akTop]
AutoSize	True
Center	False
+Constraints	(TSizeConstraints)
Cursor	crDefault
DragCursor	crDrag
DragKind	dkDrag
DragMode	dmManual
Enabled	True
Height	341
Hint	
IncrementalDisp	False
Left	344
Name	AuxImagem
ParentShowHint	True
Picture	(None)
PopupMenu	
ShowHint	False
Stretch	True
Tag	0
Top	8
Transparent	False
Visible	True
Width	326

⇒ **Componente TImage**

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
10	10	0	0	0	0	0	0	0	0
10	20	20	10	0	0	0	0	0	0
10		25	25	25	20	0	0	0	0
20		25	25	25	30	30	30	0	0
20		20	20	30	40	40	50	40	0
15		20	30	40	40	50	80	80	0
30		30	30	30	50	50	50	50	0
10		20	20	20	40	40	40	50	10



```
Imagem.canvas.pixels[coluna,linha] := numero;
```

⇒ *Como são formadas as cores?*

⇒ *Combinação de três canais: R, G, B*

⇒ *R = RED (vermelho)*

⇒ *G = GREEN (verde)*

⇒ *B = BLUE (azul)*

Cor	R	G	B
	255	0	0
	0	255	0
	0	0	255
	250	125	50
	250	20	200
	100	100	100
	0	0	0
	255	255	255

⇒ *Como são formadas as cores?*

Cor	R	G	B
	255	0	0
	0	255	0
	0	0	255
	250	125	50
	250	20	200

Em linguagem Delphi:

```
Imagem.canvas.pixels[col,lin] :=  
RGB(250,20,200);
```

⇒ *Para percorrer a imagem: trecho de repetição.*

⇒ Exemplo:

```
for linha :=1 to QuantLinhas do
  for coluna := 1 to QuantColunas do
    begin
      valorpixel := GetRValue(Imagem.canvas.pixels[coluna,linha]);
      valorpixel := valorpixel + 10;
      Imagem.canvas.pixels[coluna,linha] :=
        RGB(valorpixel, valorpixel, valorpixel);
    end;
```

# Implementação

```
for linha :=1 to QuantLinhas do
  for coluna := 1 to QuantColunas do
    begin
      valorpixel := GetRValue(Imagem.canvas.pixels[coluna,linha]);
      valorpixel := valorpixel + 10;
      Imagem.canvas.pixels[coluna,linha] :=
        RGB(valorpi
```

0	0	0	0	0	0
0	0	0	0	0	0
10	10	0	0	0	0
10	20	20	10	0	0
10	20	25	25	25	20
20	20	25	25	25	30
20	20	20	20	30	40
15	20	20	30	40	40
30	30	30	30	30	50
10	10	20	20	20	40

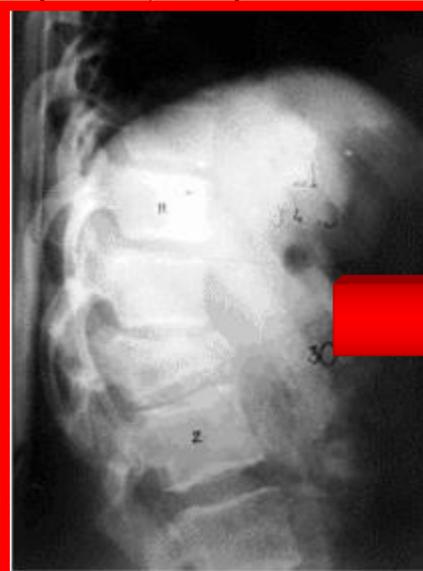


Imagem original

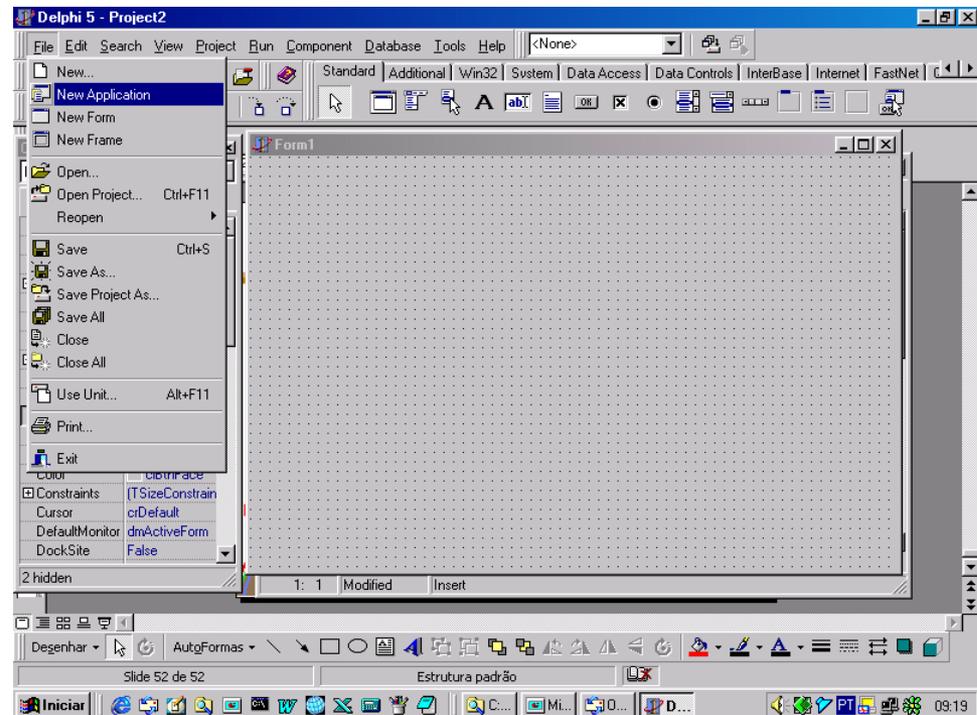
10	10	10	10	10
10	10	10	10	10
20	20	10	10	10
20	30	30	20	10
20	30	35	35	35
30	30	35	35	35
30	30	30	30	40
25	30	30	40	50
40	40	40	40	40
20	20	30	30	30



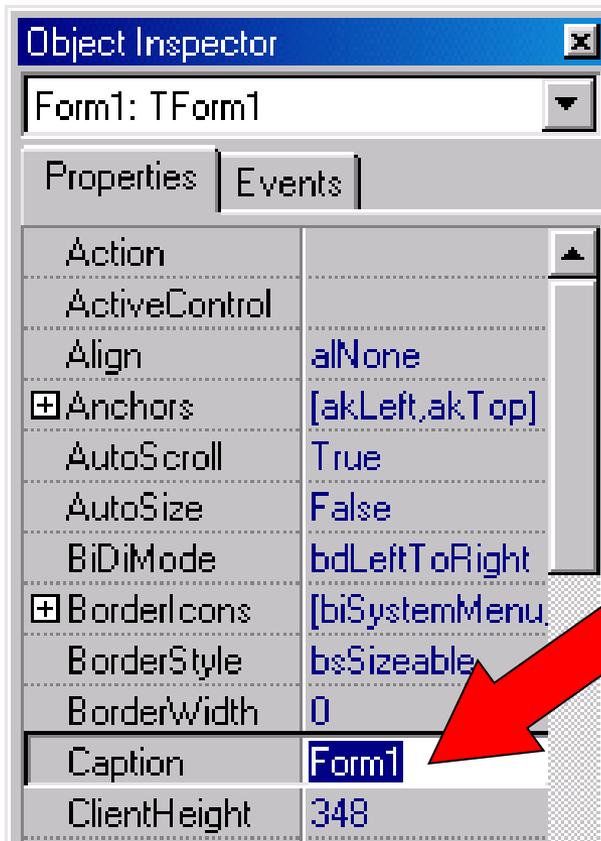
Imagem mais clara

- Em Delphi:

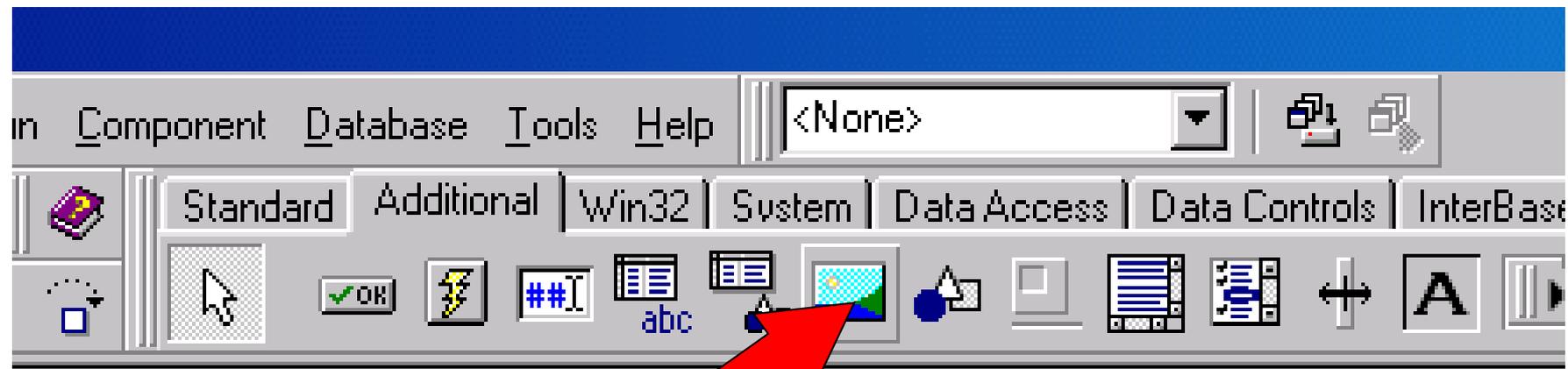
- Abrir uma nova aplicação: File/New Application
- Salvar a Unit como *principal.pas*: File/Save
- Salvar o projeto com o nome *prog1.dpr*: File/Save



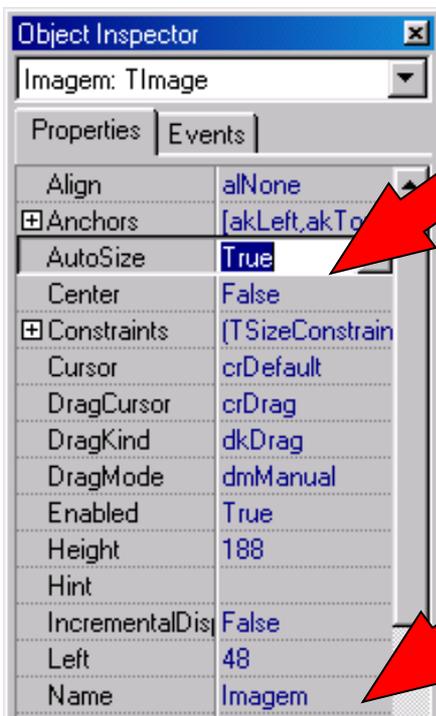
- Em Delphi:
  - Mudar o nome do Formulário para *Principal*



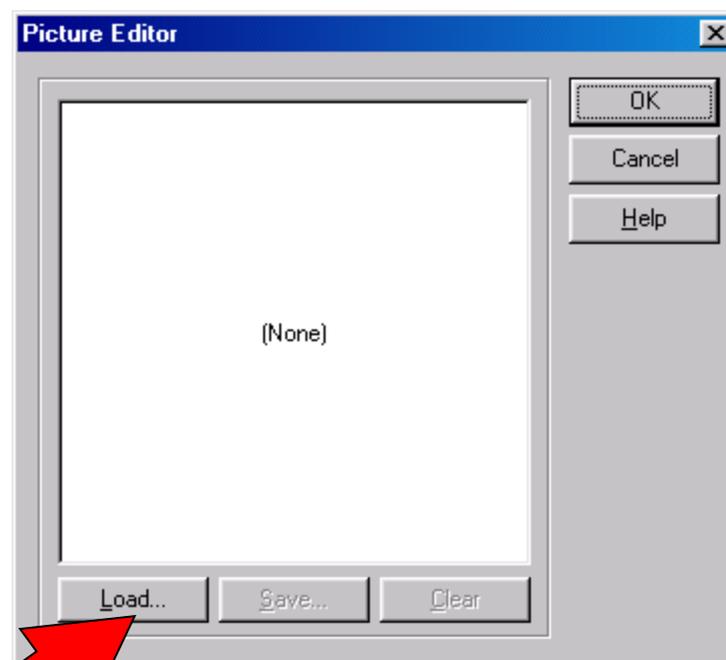
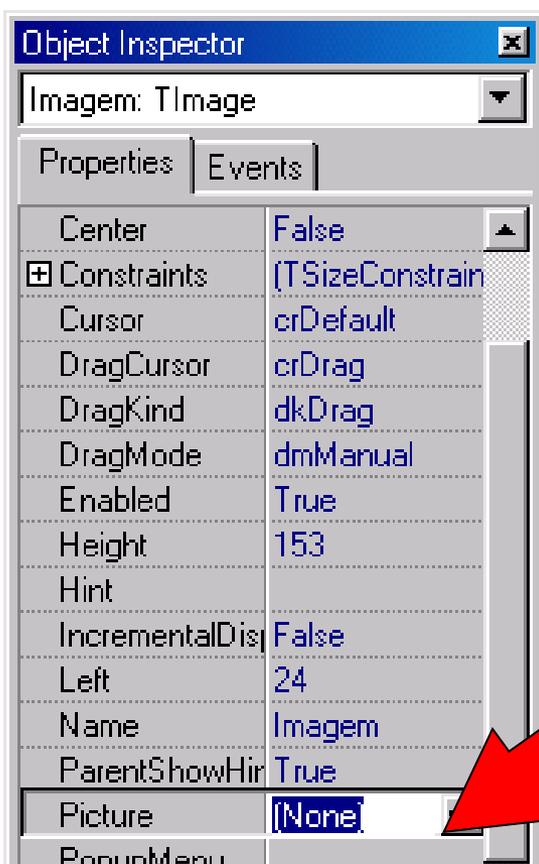
- Em Delphi:
  - Inserir um componente Timage: Barra de Ferramentas Additional



- Em Delphi:
  - Mudar o nome do componente para Imagem
  - Mudar o atributo AutoSize para True

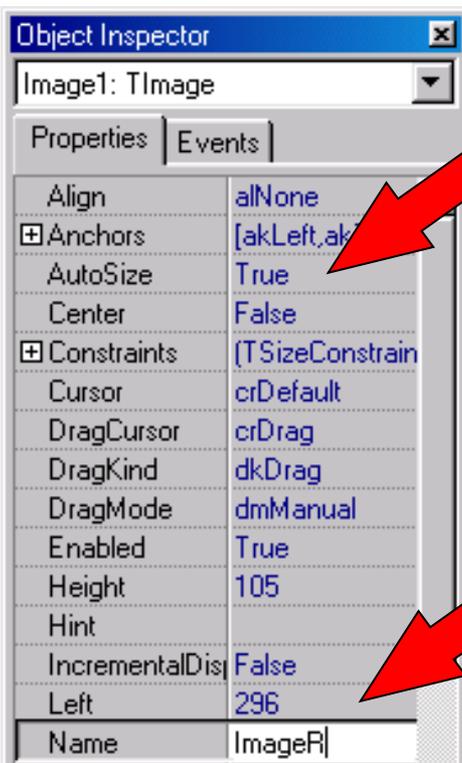


- Em Delphi:
  - Carregar a imagem foto1.bmp



## • Em Delphi:

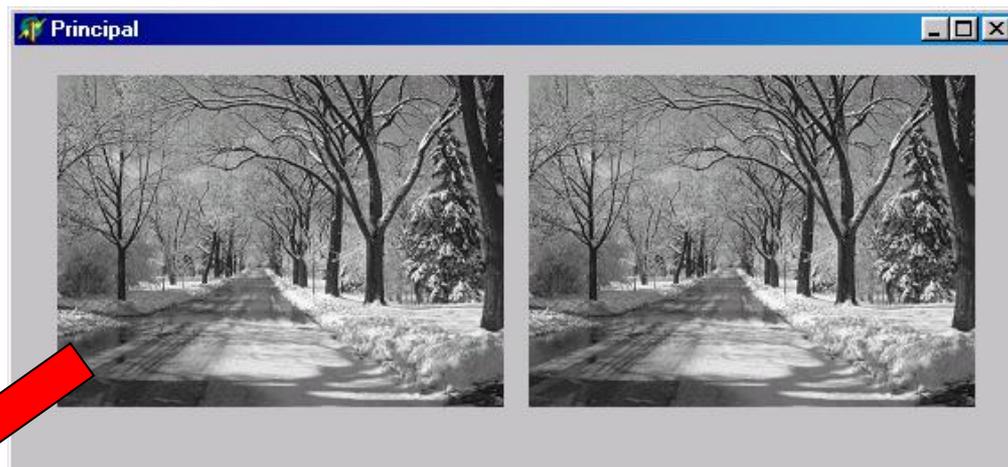
- Inserir outro componente TImage. Mudar o nome para ImagemR e mudar o atributo AutoSize para True.
- Carregar a mesma imagem no componente.



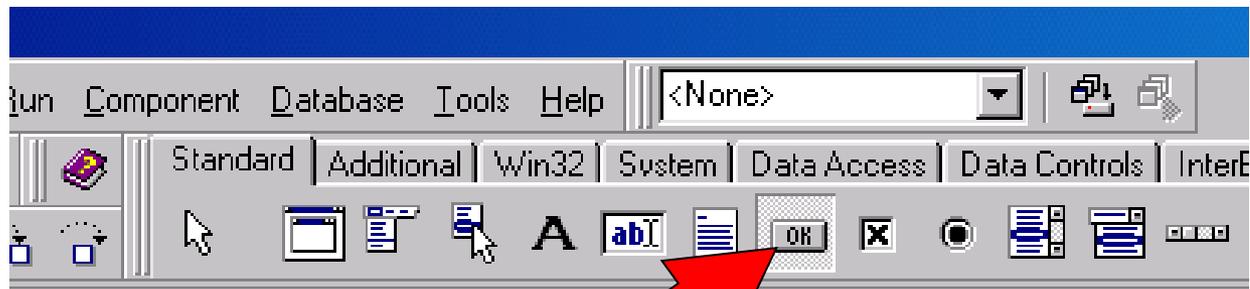
- Em Delphi:

- Inserir outro componente Timage. Mudar o nome para ImagemR e mudar o atributo AutoSize para True.
- Carregar a mesma imagem no componente.

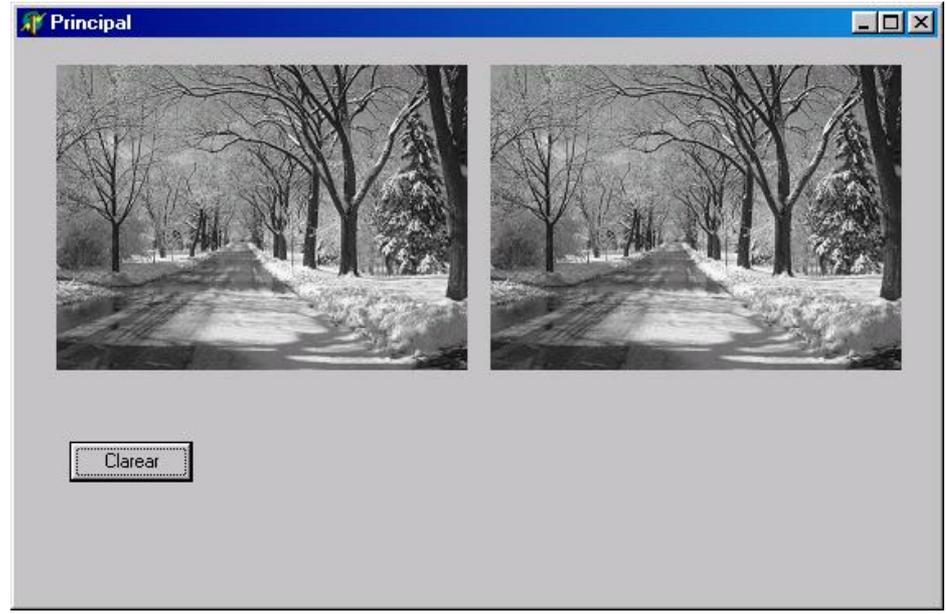
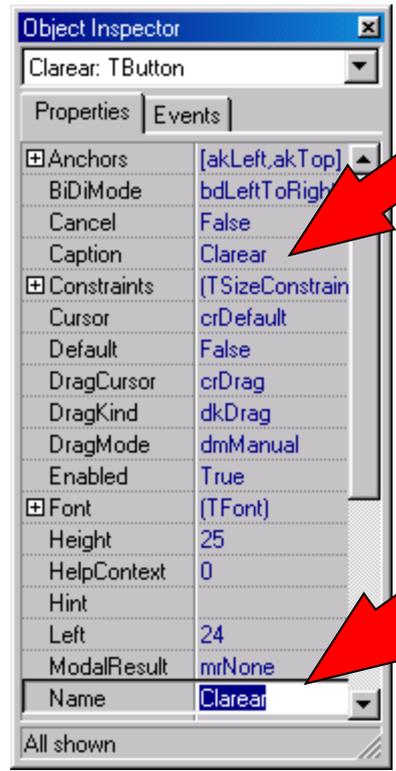
Properties	Events
Align	alNone
⊕ Anchors	[akLeft,ak
AutoSize	True
Center	False
⊕ Constraints	(TSizeConstrain
Cursor	crDefault
DragCursor	crDrag
DragKind	dkDrag
DragMode	dmManual
Enabled	True
Height	105
Hint	
IncrementalDis	False
Left	296
Name	ImagemR



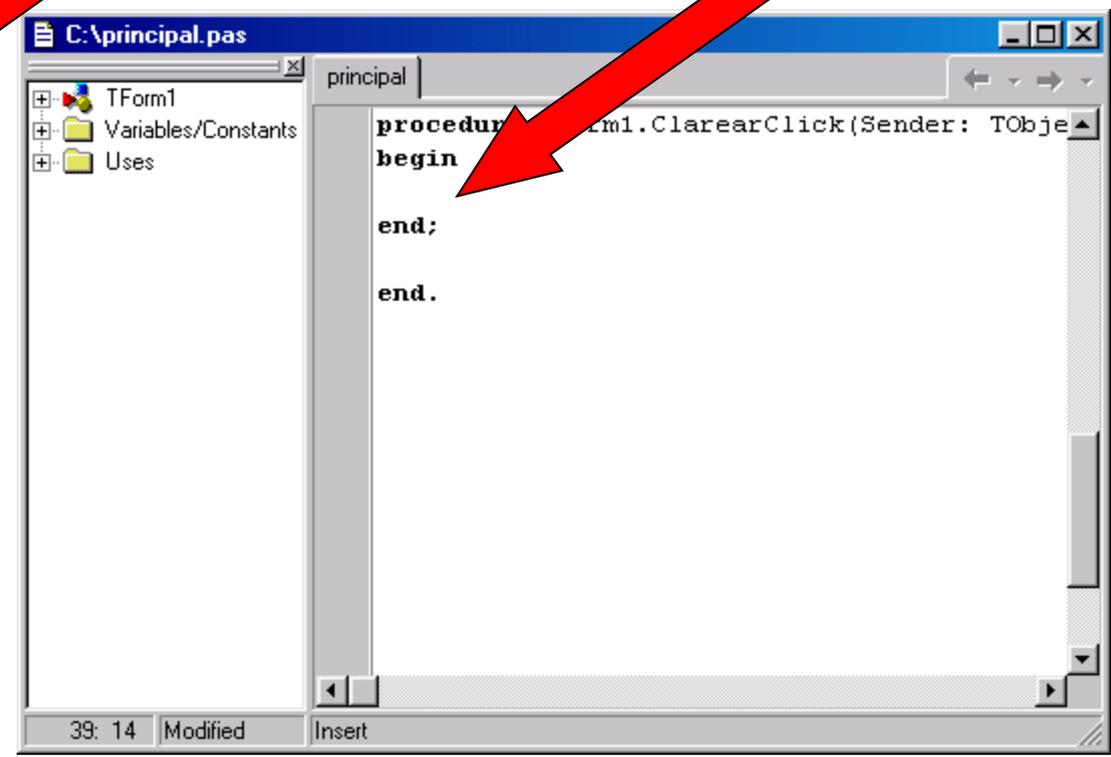
- Em Delphi:
  - Inserir um Botão: Barra de Ferramentas Standard



- Em Delphi:
  - Mudar o Nome e o Caption para *Clarear*



- Em Delphi:
  - Clique duplo no Evento **OnClick** do Botão *Clarear*

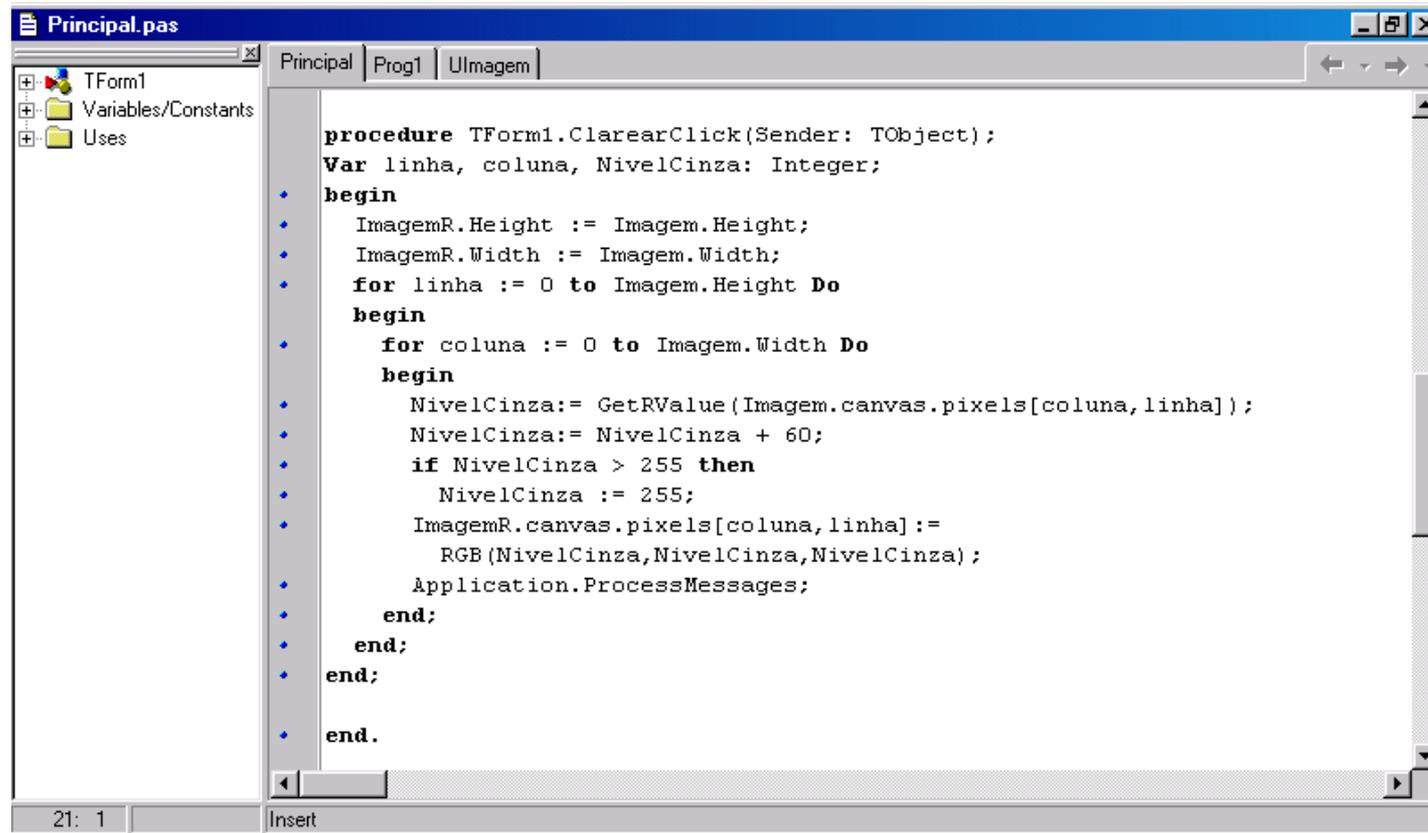


- **Em Delphi:**

- **Inserir o seguinte código no local indicado**

```
procedure TForm1.ClarearClick(Sender: TObject);
Var linha, coluna, NivelCinza: Integer;
begin
  ImagemR.Height := Imagem.Height;
  ImagemR.Width := Imagem.Width;
  for linha := 0 to Imagem.Height Do
  begin
    for coluna := 0 to Imagem.Width Do
    begin
      NivelCinza:= GetRValue(Imagem.canvas.pixels[coluna,linha]);
      NivelCinza:= NivelCinza + 60;
      if NivelCinza > 255 then
        NivelCinza := 255;
      ImagemR.canvas.pixels[coluna,linha]:=
        RGB(NivelCinza,NivelCinza,NivelCinza);
      Application.ProcessMessages;
    end;
  end;
end;
```

- Em Delphi:
  - Tela final abaixo. Executar o programa: F9



```
Principal.pas
Principal Prog1 Ulmagem

procedure TForm1.ClarearClick(Sender: TObject);
Var linha, coluna, NivelCinza: Integer;
begin
  ImagemR.Height := Imagem.Height;
  ImagemR.Width := Imagem.Width;
  for linha := 0 to Imagem.Height Do
  begin
    for coluna := 0 to Imagem.Width Do
    begin
      NivelCinza:= GetRValue(Imagem.canvas.pixels[coluna,linha]);
      NivelCinza:= NivelCinza + 60;
      if NivelCinza > 255 then
        NivelCinza := 255;
      ImagemR.canvas.pixels[coluna,linha]:=
        RGB(NivelCinza,NivelCinza,NivelCinza);
      Application.ProcessMessages;
    end;
  end;
end;
end.
```

# Processamento de Baixo Nível

- Processamento de imagens ▶ três níveis.
  - **Processamento de baixo nível:** responsável pela remoção de dados indesejáveis e realce de dados importantes;
  - **Processamento em nível médio:** parte do processamento que identifica formas significantes. A esse processo damos o nome de “segmentação”.
  - **Processamento em alto nível:** responsável pela ligação da imagem com algum banco de conhecimento.

# Processamento de Baixo Nível

- Processamento de imagens
  - **Domínio da frequência**
    - transformadas matemáticas
  - **Domínio espacial**
    - conjunto de pixels

# Processamento de Baixo Nível

- **Relembrando alguns conceitos**
  - média
  - mediana

- ***Suavização - Média da Vizinhança***
  - Nível de cinza de cada pixel ► média dos valores de cinza dos pixels de uma vizinhança pré-definida.
  - Técnica muito empregada para **eliminação de ruídos** na imagem.

- ***Suavização - Média da Vizinhança***

- Considerando a imagem  $f(x,y)$  com  $N \times M$  pixels, podemos definir a imagem gerada  $g(x,y)$  como:

$$g(x, y) = \frac{1}{M} \sum_{(p,q) \in S} f(p, q) \quad \text{para } x=0,1,\dots,N-1; \\ y=0,1,\dots,M-1$$

onde:

- $S$  é o conjunto de coordenadas de pontos na vizinhança do ponto  $(x,y)$ , incluindo o próprio  $(x,y)$ ;
- $M$  é o número total de pontos na vizinhança escolhida.

# Processamento de Baixo Nível

- *Suavização - Média da Vizinhança*
  - Qual é o efeito resultante na imagem?

- *Suavização - Média da Vizinhança*
  - Qual é o efeito resultante na imagem?
  - Apresenta **borramento** na imagem final obtida e consequente perda na definição de bordas.

## *Média da Vizinhança*

***Problema - borramento das bordas e outros detalhes que deveriam estar realçados.***



## *Média da Vizinhança* *Exemplo*



## *Média da Vizinhança*

*Para deixar regiões com grande variações de níveis de cinza inalteradas:*

$$g(x, y) = \begin{cases} \frac{1}{M} \sum_{(p,q) \in S} f(p, q), & \text{se } \left| f(x, y) - \frac{1}{M} \sum_{(p,q) \in S} f(p, q) \right| < T \\ f(x, y), & \text{caso contrário} \end{cases}$$

*T é um valor de limiar não negativo.*

## *Média da Vizinhança* *Algoritmo*

- ***Suavização - Mediana da Vizinhança***

- Cada pixel da imagem final é substituído pelo nível de cinza mediano em uma vizinhança do pixel.
- Método é eficiente quando a imagem contém ruídos grandes e o realce das bordas é importante.

- ***Suavização - Mediana da Vizinhaça***
- Nível mediano  $m$  de um conjunto de valores ▶ metade dos valores no conjunto são menores que  $m$  e a outra metade é constituída de valores maiores que  $m$ .
- Principal função do filtro mediano ▶ forçar pontos com intensidades muito diferentes a ser mais parecidos com seus vizinhos.
- Elimina picos da imagem.

## *Mediana da Vizinhança - Exemplo*



# Processamento de Baixo Nível

## *Média X Mediana*



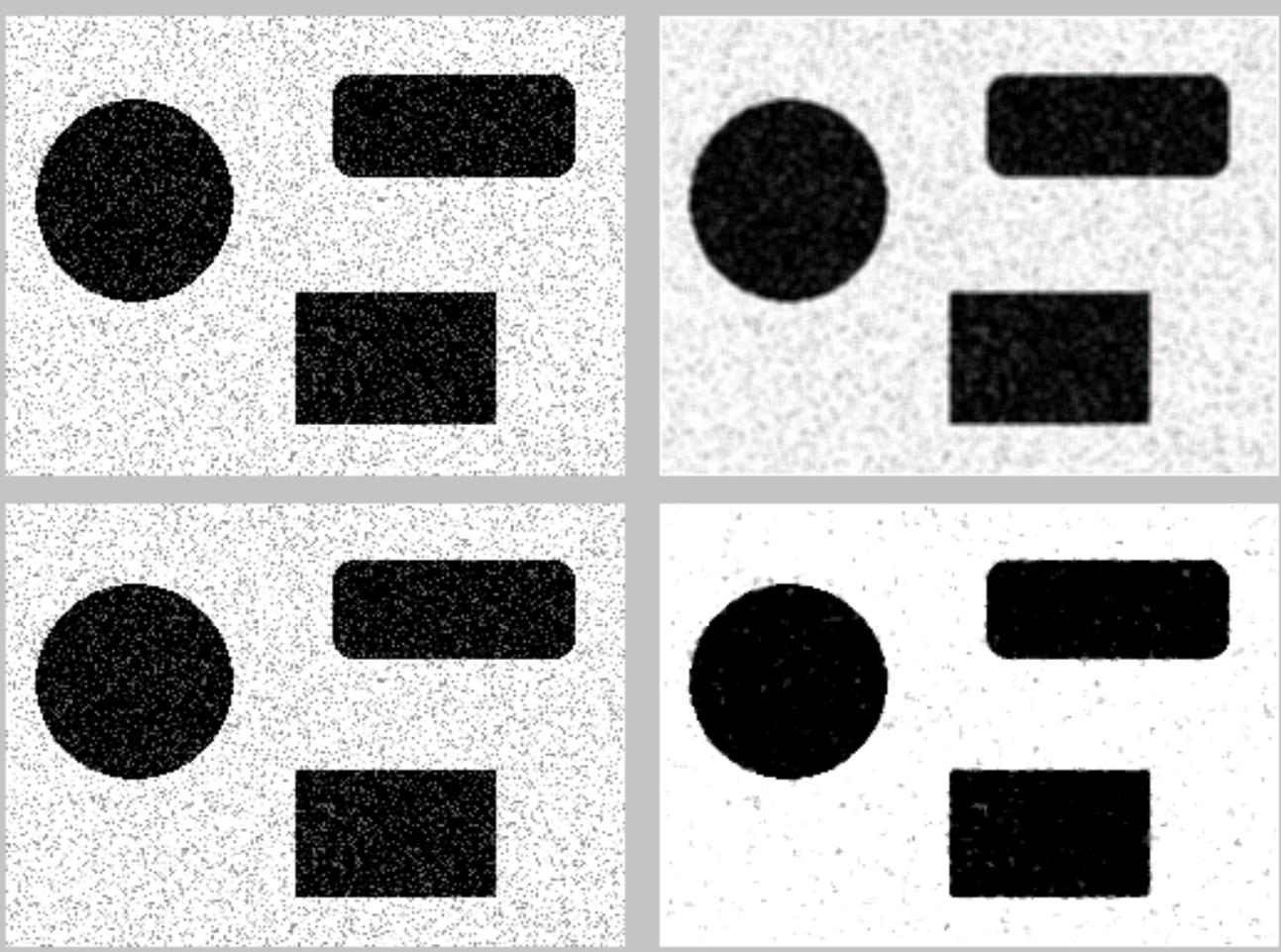
*Média*



*Mediana*

# Processamento de Baixo Nível

## *Média X Mediana*



*Média*

*Mediana*

## *Mediana da Vizinhança Algoritmo*

- ***Suavização - Média de múltiplas imagens***
- Considerando uma imagem com ruídos  $g(x,y)$ , formada pela adição de ruídos  $a(x,y)$  em uma imagem original  $f(x,y)$ :  
$$g(x,y) = f(x,y) + a(x,y)$$
- Objetivo deste procedimento: obter uma imagem suavizada resultante da média de várias imagens ruidosas.

# Processamento de Baixo Nível

Imagem Ruído 1

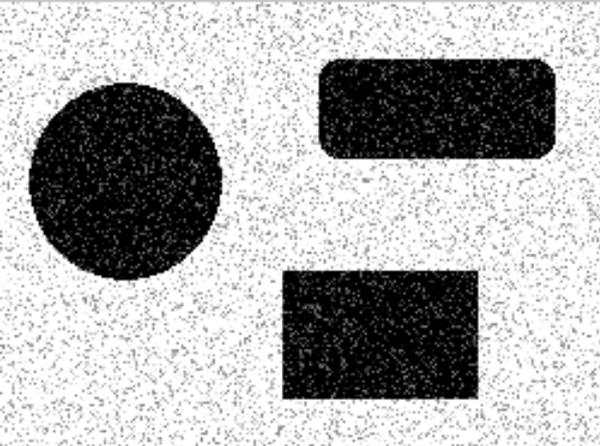


Imagem Ruído 2

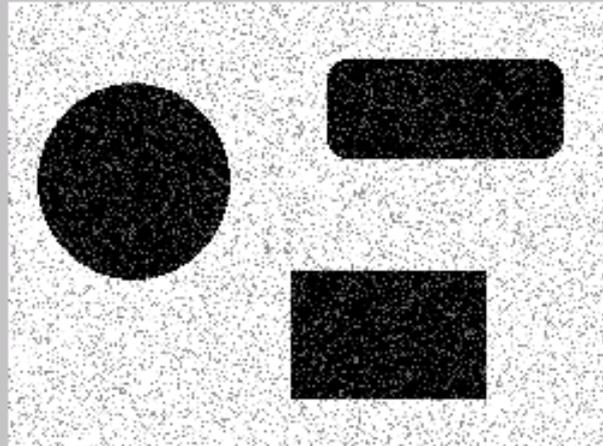
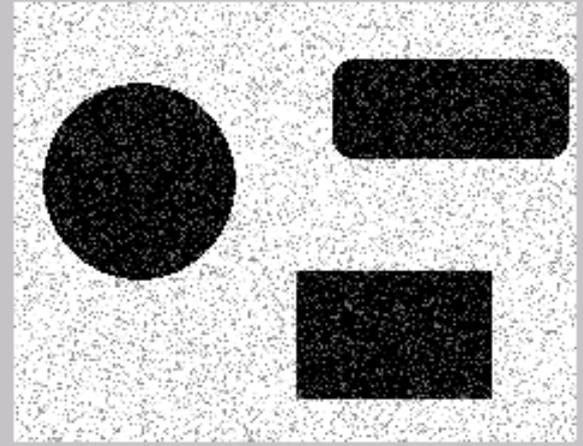
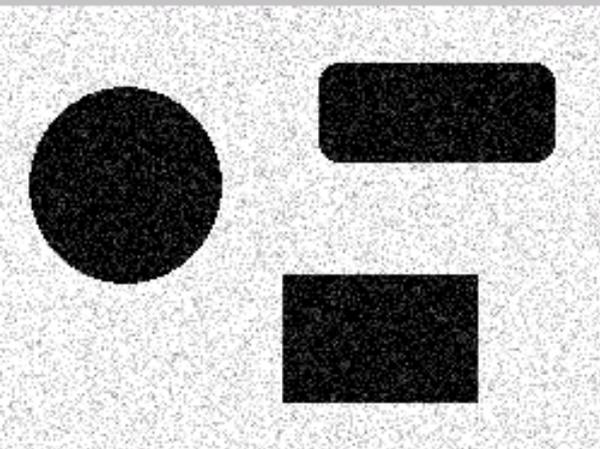


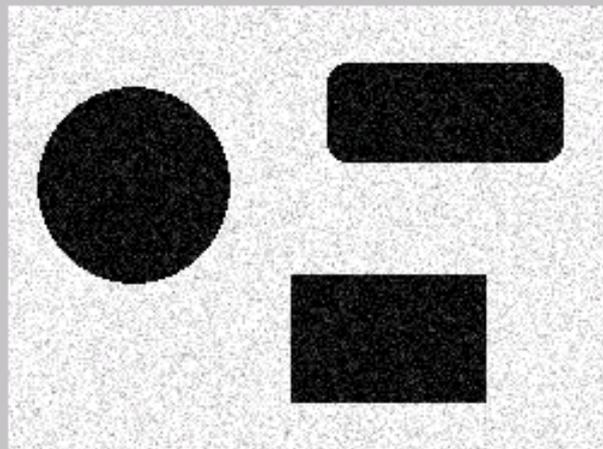
Imagem Ruído 3



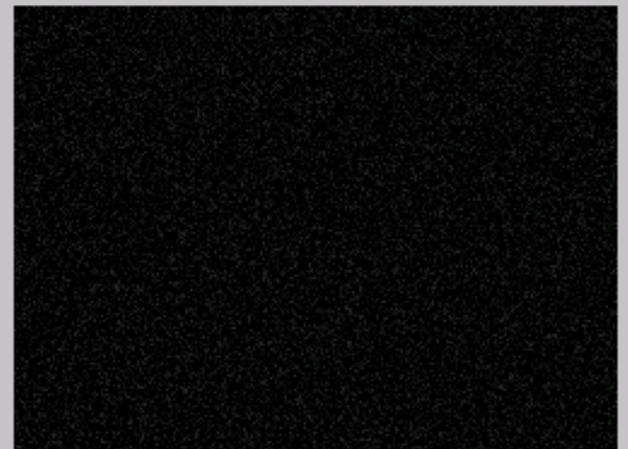
R1 - Média imagens 1 e 2



R2 - Média imagens 1, 2 e 3



Subtração R1 - R2



Média 2 Imagens

Media 3 Imagens

Subtração

*Média de múltiplas imagens*  
*Algoritmo*

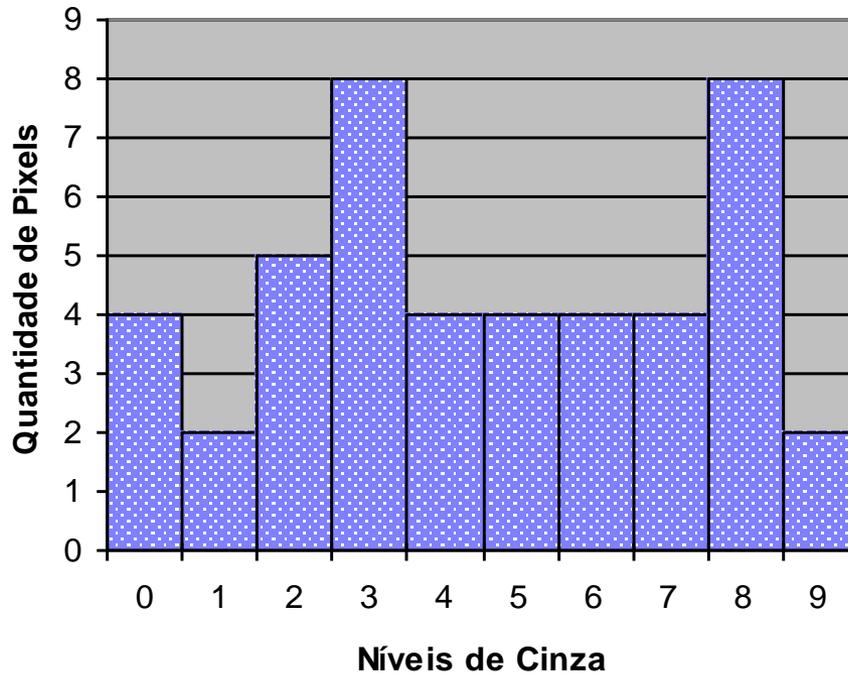
- **Realce de Imagens**
  - *Quantização*
  - *Splitting*
  - *Equalização*

- **Quantização**

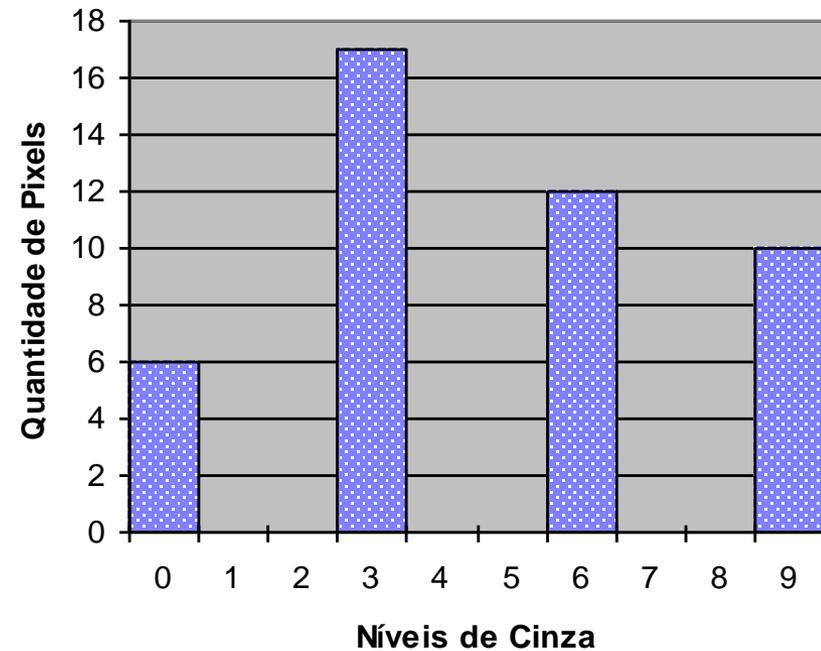
- Redução da quantidade de níveis de cinza diferentes na imagem.
- Útil para remover gradações indesejáveis na imagem.

- Quantização**

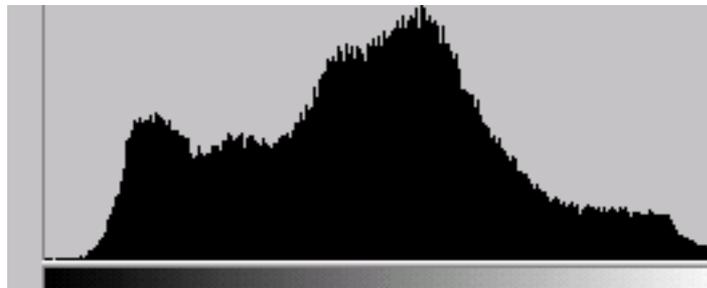
Histograma antes da Quantização



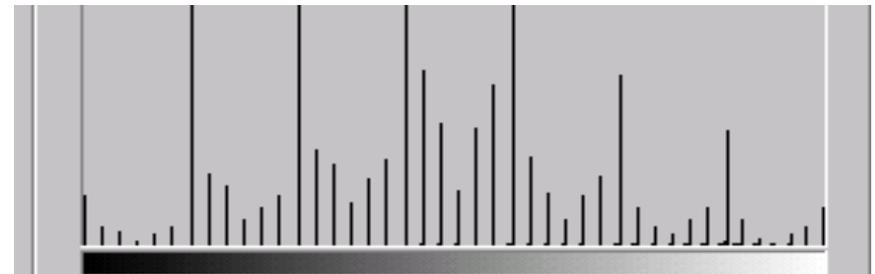
Histograma após Quantização



## Quantização – Algoritmo?



Mean: 119,43	Level: 100
Std Dev: 51,52	Count: 1162
Median: 122	Percentile: 34,42
Pixels: 187500	Cache Level: 1



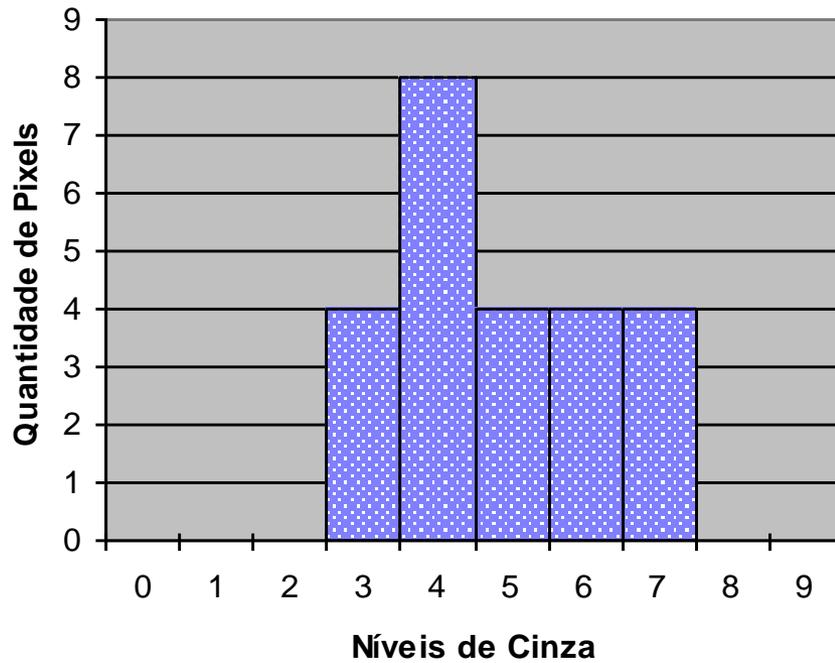
Mean: 117,55	Level: 135
Std Dev: 54,35	Count: 5117
Median: 117	Percentile: 60,17
Pixels: 187500	Cache Level: 1

- ***Splitting***

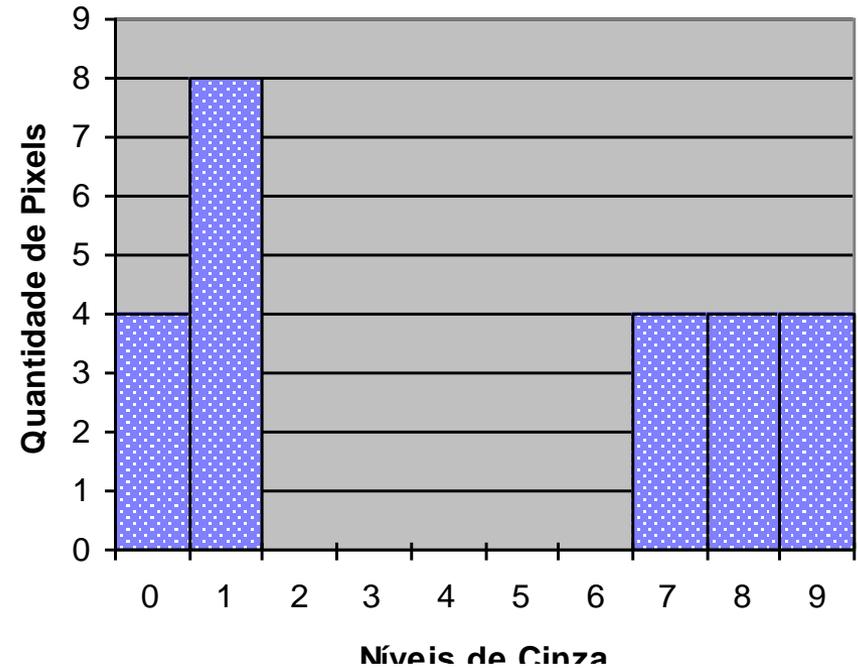
- Aumenta o contraste de uma imagem com base no seu histograma.
- Divide os pixels em dois grupos distintos de níveis de cinza

- Splitting***

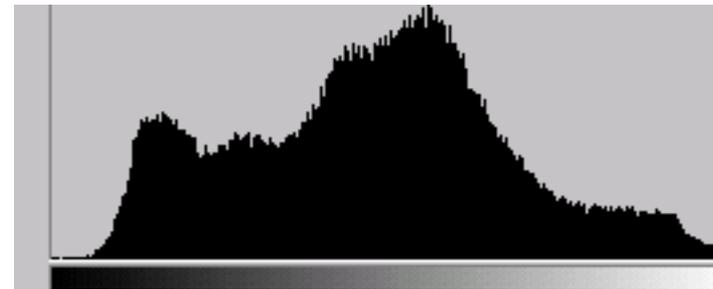
Histograma antes do "Splitting"



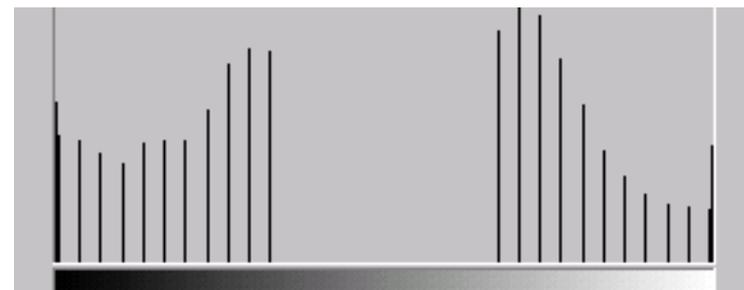
Histograma após "Splitting"



## Splitting

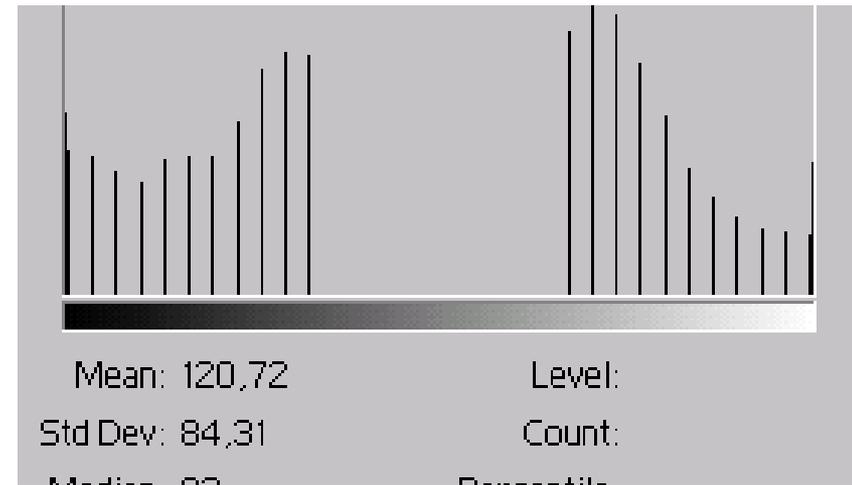
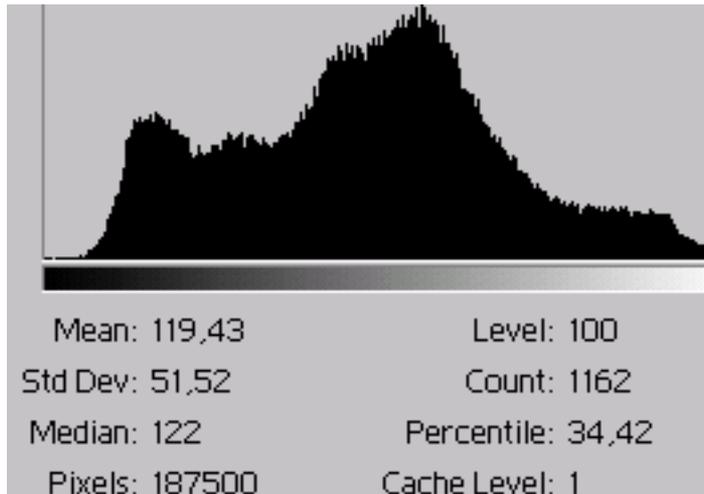


Mean: 119,43	Level: 100
Std Dev: 51,52	Count: 1162
Median: 122	Percentile: 34,42
Pixels: 187500	Cache Level: 1



Mean: 120,72	Level:
Std Dev: 84,31	Count:
Median: 99	Percentile:

## Splitting – Algoritmo?



- **Equalização**

- Um dos métodos mais utilizados para realce de contraste.
- Também conhecida como “**Linearização de Histograma**”
- **Finalidade** ▶ obter um histograma uniforme, através do espalhamento da distribuição dos níveis de cinza.
- Operação muito poderosa, conseguindo, muitas vezes recuperar imagens consideradas perdidas.

- ***Equalização***

- ◆ Exemplo de método:

- Dada uma Imagem de  $n \times m$  pixels e  $g$  níveis de cinza.
- Número ideal de pixels em cada nível:

$$I = \frac{n \times m}{g}$$

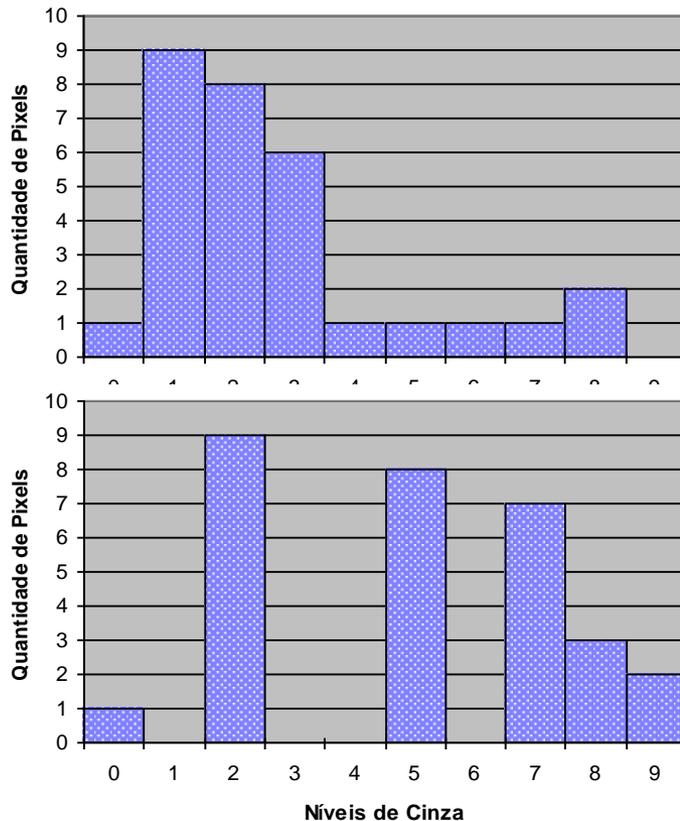
- **Equalização**

- Equalização pode ser realizada, encontrando o novo valor de nível de cinza  $q$  para um nível de cinza atual  $g$ .

$$q = \max \left\{ 0, \text{arred} \left( \frac{\sum_{j=0}^k n_j}{I} \right) - 1 \right\} \quad 0 \leq k \leq g$$

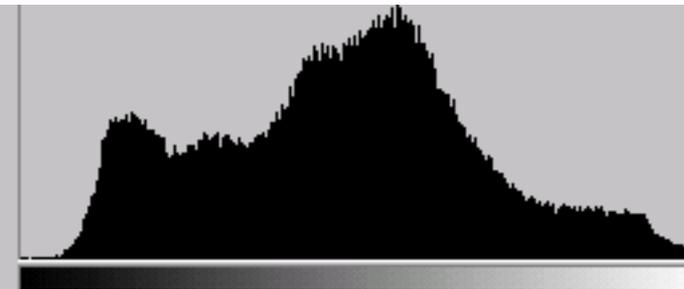
## • Equalização - exemplo

$$q = \max \left\{ 0, \text{arred} \left( \frac{\sum_{j=0}^k n_j}{I} \right) - 1 \right\} \quad 0 \leq k \leq g$$

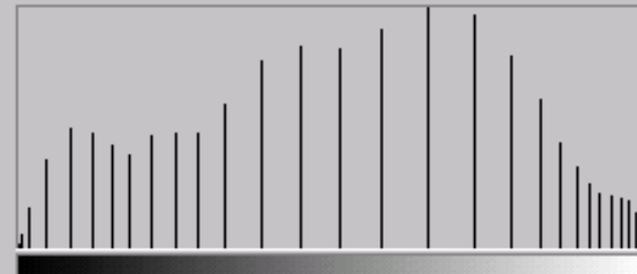


g	$\sum g$	$\sum$ acumulado	q
0	1	1	0
1	9	10	2
2	8	18	5
3	6	24	7
4	1	25	7
5	1	26	8
6	1	27	8
7	1	28	8
8	2	30	9
9	0	30	9

## *Equalização - aplicação do método visto*



Mean: 119,43                      Level: 100  
Std Dev: 51,52                      Count: 1162  
Median: 122                      Percentile: 34,42



Mean: 132,69                      Level:  
Std Dev: 73,81                      Count:  
Median: 131                      Percentile:  
Pixels: 47000                      Cache Level: 1

**Ver Exercício  
na plataforma e-disciplinas**

# **Fundamentos de Processamento Gráfico**

## **Aula 3**

### **Introdução ao Processamento de Imagens**

**Profa. Fátima Nunes**

**Prof. Helton Bíscaro**