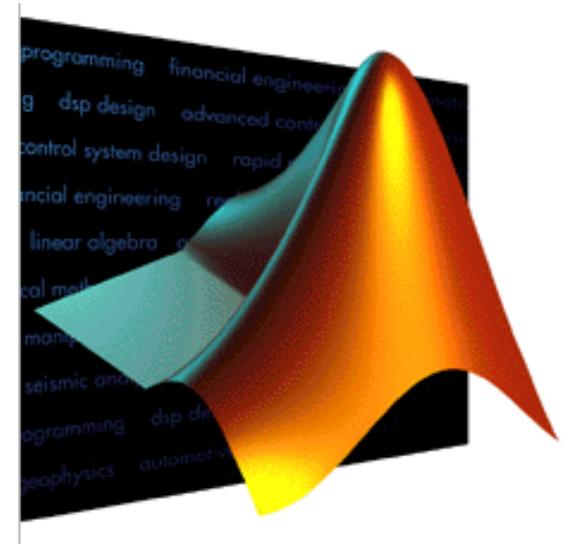


Processamento de Imagens com MATLAB

Carlos Alexandre Mello



Processamento de Imagens com MatLab

- **Toolboxes**
 - **Image Processing**
 - **Diretório:**
 - **toolbox/images/images**

Processamento de Imagens com MatLab

- **Comando imshow():**
 - **Visualização de imagens**
 - **Uso:**
 - `imshow(nome_do_arquivo ou variável)`
 - **Exemplo:**
 - `imshow('eight.tif')`
 - **Exemplo:**
 - `im = imread('eight.tif');`
 - `imshow (im);`

Processamento de Imagens com MatLab

- **E/S de arquivos de imagem**

- **imread**

- **Lê um arquivo de imagem**

- **Uso:**

- **A = imread(filename)**

- **Exemplo:**

- **A = imread('eight.bmp')**

Processamento de Imagens com MatLab

● E/S de arquivos de imagem

■ Leitura de Imagem BMP

○ `[im, map] = imread ('nome.bmp')`

- Imagem indexada
- Map armazena a paleta de cores

○ `im = imread ('nome.bmp')`

- Imagem em 24 bits
- `im` é uma matriz $m \times n \times 3$
- `R = im(:, :, 1); % Componente de vermelho`
- `G = im(:, :, 2); % Componente de verde`
- `B = im(:, :, 3); % Componente de azul`

Processamento de Imagens com MatLab

● Dimensões de uma imagem

■ Função `size`

○ `[lin, col] = size (im);`

- % Para imagens em tons de cinza e preto e branco

○ `[lin, col, plan] = size (im);`

- % Para imagens coloridas (`plan = 3`, indicando que é uma estrutura com 3 matrizes). Para imagens em tons de cinza ou preto-e-branco, se usado esse parâmetro, ele terá valor 1
- % Se não usado aqui, a imagem ficará com o número de colunas multiplicado por 3

Processamento de Imagens com MatLab

- **E/S de arquivos de imagem**

- **imwrite**

- **Escreve uma imagem para um arquivo gráfico**

- **Uso:**

- `imwrite(A, filename, FMT)`

- `FMT = formato`

- **Exemplo:**

- `imwrite (A, 'eight.bmp', 'bmp')`

Processamento de Imagens com MatLab

● Visualização

■ **image**

- **Mostra uma matriz C como uma imagem**
- **Uso:**
 - `image(C)`
- **Exemplo:**
 - `>> C = round(255*rand(255));`
 - `>> image(C);`
- **Se não especificada, é usada a paleta de cores default**

Processamento de Imagens com MatLab

- Conversão entre Resoluções de Cores
 - **dither**
 - Dithering de uma imagem pelo método de Floyd-Steinberg
 - Uso:
 - **X = dither (im)**
 - X = Imagem em preto-e-branco
 - im = Imagem em tons de cinza

Processamento de Imagens com MatLab

● Conversão entre Resoluções de Cores

■ **im2bw**

○ **Converte uma imagem para preto-e-branco**

○ **Uso:**

• **BW = im2bw(X, level)**

– **X = Imagem original**

– **level = Valor de corte (threshold): $0 \leq \text{level} \leq 1$**

○ **Exemplo:**

```
im = imread('lena.bmp');
```

```
bw = im2bw(im, 0.4);
```

```
imshow(bw);
```

Processamento de Imagens com MatLab

- **Conversão entre Resoluções de Cores**
 - **rgb2gray**
 - **Converte uma imagem RGB para uma imagem em tons de cinza**
 - **Uso:**
 - `I = rgb2gray(RGB)`
 - RGB = imagem original true color
 - **Exemplo:**

```
A = imread('flowers.tif');  
I = rgb2gray(A);  
imshow(I);
```

Processamento de Imagens com MatLab

● Conversão entre Tipos

■ **rgb2ind**

○ **Converte uma imagem RGB para uma imagem indexada**

○ **Uso:**

• **[X, NEWMAP] = rgb2ind (RGB, N)**

– **RGB = imagem 24 bits de entrada**

– **NEWMAP = paleta de cores final**

– **N = Número de Cores**

Processamento de Imagens com MatLab

● Tipos de Imagens

■ **isbw**

- Verdadeiro para imagens preto e branco

■ **isgray**

- Verdadeiro para imagens em tons de cinza

■ **isind**

- Verdadeiro para imagens indexadas

Processamento de Imagens com MatLab

● Operações Geométricas

■ **imresize**

○ Altera o tamanho de uma imagem

○ Uso:

- $B = \text{imresize}(A, M, \text{'method'})$
- Retorna uma matriz que é M vezes maior (ou menor) que a imagem A
- M pode ser também as novas dimensões: $M = [m \ n]$
- 'method' =
 - nearest = vizinho mais próximo
 - bilinear = interpolação bilinear
 - bicubic = interpolação bicúbica

Processamento de Imagens com MatLab

● Operações Geométricas

■ **imresize**

○ Exemplo:

- `>> A = imread('eight', 'tif');`
- `>> B = imresize (A, 0.5, 'nearest');`
- `>> imshow(B)`

- `>> A = imread('eight', 'tif');`
- `>> B = imresize (A, [100 50], 'nearest');`
- `>> imshow(B)`

Processamento de Imagens com MatLab

● Operações Geométricas

■ **imrotate**

○ Rotaciona uma imagem

○ Uso:

- **B = imrotate(A, Ângulo, 'method');**
- **Method = nearest, bilinear ou bicubic**

○ Exemplo:

- **>> A = imread ('eight.tif');**
- **>> B = imrotate (A, 45, 'nearest');**
- **>> imshow(B)**

Processamento de Imagens com MatLab

- Valores de Pixels e Estatísticas
 - **imhist**
 - Histograma de uma imagem
 - Uso:
 - **imhist(A)**: histograma de 256 cores
 - **imhist(A, N)**: histograma de N cores

Processamento de Imagens com MatLab

● Valores de Pixels e Estatísticas

■ **mean2**

○ Média de uma matriz

- **Uso:**

- `mean2(A)`

■ **std2**

○ Desvio padrão bidimensional

- **Uso:**

- `std2(A)`

Processamento de Imagens com MatLab

- Realce de imagem

- **histeq**

- Equalização de Histograma

- **imnoise**

- Adiciona ruído a uma imagem

- O ruído pode ser gaussiano, *salt & pepper* ou *speckle*

- Exemplo: $J = \text{imnoise}(A, \text{'gaussian'})$;

Processamento de Imagens com MatLab

● Filtragem

■ **imfilter**

○ Filtro digital para matriz

○ Uso:

- `imfilter(im, h)` - Filtra a imagem *im* usando o filtro definido pela matriz *h*

○ Bordas

- *X* (constante, se não definido, assume valor zero), *symmetric* (reflexão), *replicate* (cópia do elemento mais próximo), *circular* (periódico)

Processamento de Imagens com MatLab

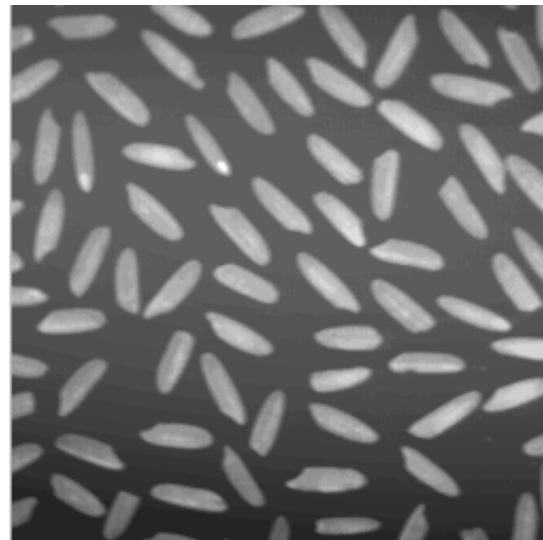
● Filtragem

■ **filter2**

○ Exemplo:

○ >> I=imread('rice.bmp');

○ >> imshow(I);

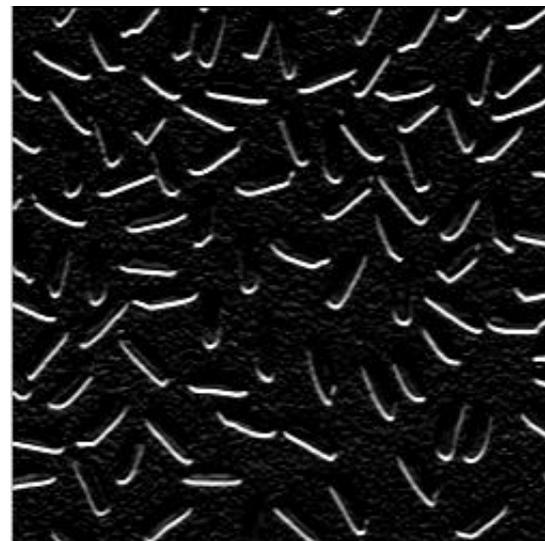


Processamento de Imagens com MatLab

● Filtragem

■ **filter2**

- » $h=[1\ 2\ 1; 0\ 0\ 0; -1\ -2\ -1];$
- » $I2 = \text{imfilter}(I, h);$
- » $\text{imshow}(I2)$



Processamento de Imagens com MatLab

● Filtragem

■ **nfilter**

○ Filtro não-linear

○ Exemplo:

» `I=imread('rice.bmp');`

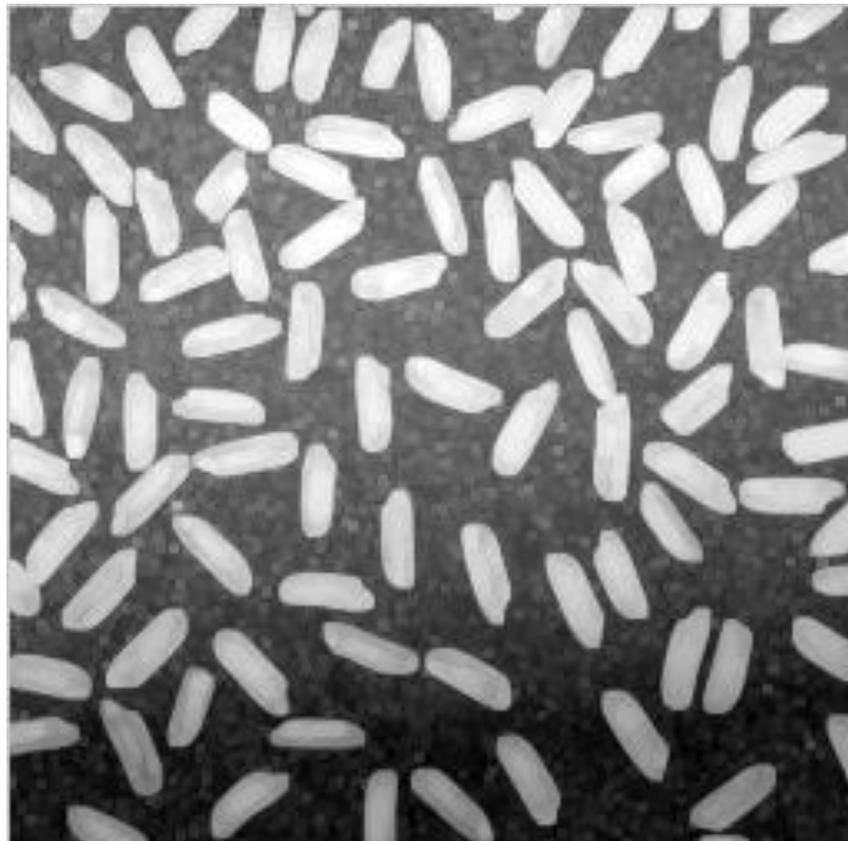
» `f=inline('max(I(:))');` % define uma função

» `I2=nfilter(I,[3 3],f);` % processa a vizinhança

» `imshow(I2)`

Processamento de Imagens com MatLab

- **Filtragem**
 - **nfilter**



Processamento de Imagens com MatLab

● Filtragem

■ Processamento em blocos

○ `B=blkproc(A,[M N], FUN)`

- Processa a imagem A aplicando sobre ela a função FUN a cada bloco MxN distinto de A

○ Exemplo:

» `I =imread('rice.tif');`

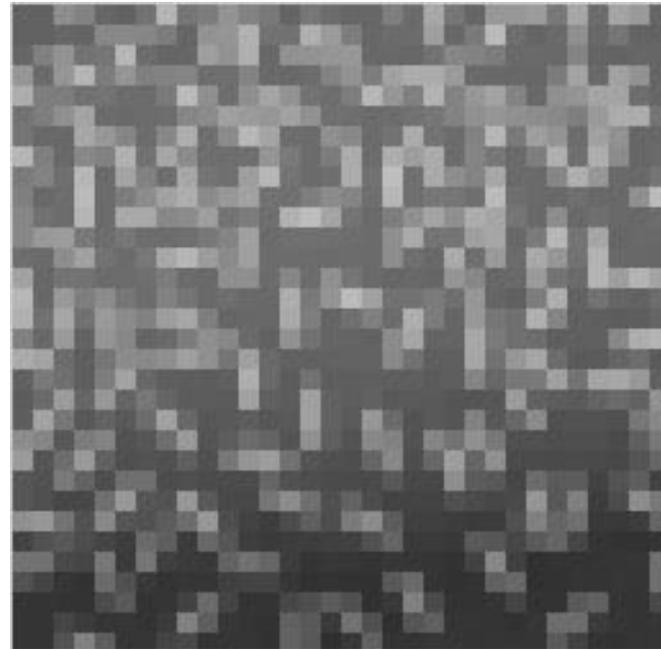
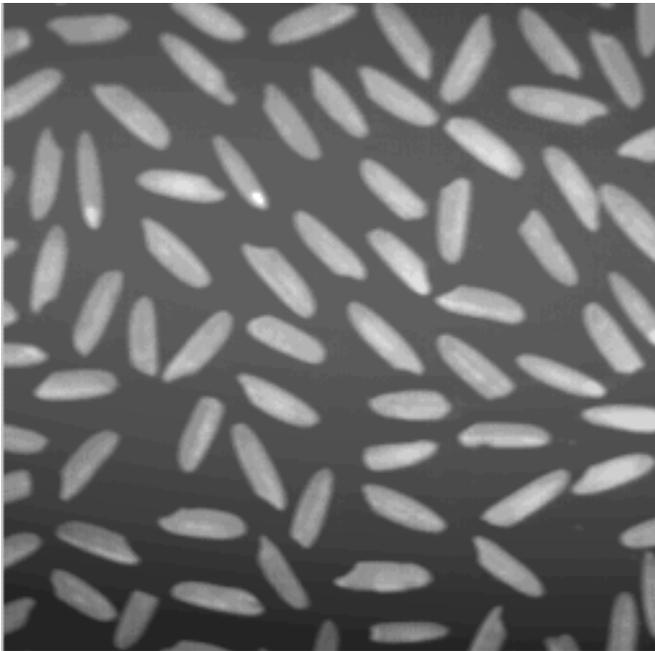
» `f=inline('uint8(round(mean2(x)*ones(size(x))))');`

» `I2=blkproc(I,[8 8],f);`

» `imshow(I2)`

Processamento de Imagens com MatLab

- **Filtragem**
 - **Processamento em blocos**



Processamento de Imagens com MatLab

● Filtragem

■ fspecial

○ Cria um filtro 2D de um tipo específico

- gaussian
- sobel
- prewitt
- laplacian
- log
- average
- unsharp

Processamento de Imagens com MatLab

● Filtragem

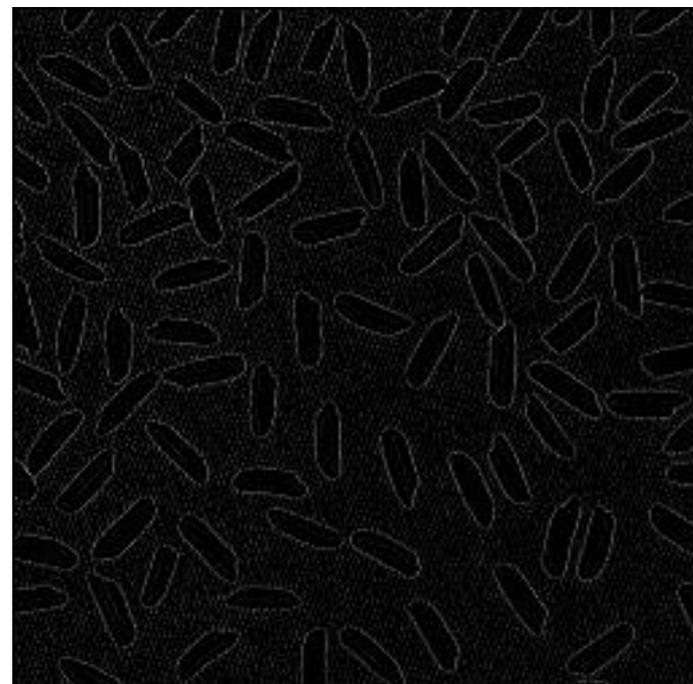
■ **fspecial**

○ Exemplo:

» `h=fspecial('laplacian');`

» `I2=imfilter(I, h);`

» `imshow(I2)`



Processamento de Imagens com MatLab

● Filtragem

■ **fspecial**

○ Exemplo:

» `h=fspecial('sobel');`

» `I2=imfilter(I, h);`

» `imshow(I2)`



Processamento de Imagens com MatLab

● Análise de Imagem

■ Edge:

○ Extração de bordas

○ Uso:

- `BW = edge(A, 'method');`

○ Exemplo:

- `A = imread('rice', 'tif');`
- `BW = edge (A, 'sobel');`
- `imshow(BW);`

∴ `method =` {
sobel
roberts
prewitt
log
zerocross

Processamento de Imagens com MatLab

● Manipulação de Paleta de Cores

■ **brighten**

○ Clareia ou escurece uma paleta de cores

○ **Uso:**

- `brighten(BETA)`
- Se $0 < \text{BETA} < 1$: A paleta é clareada
- Se $-1 \leq \text{BETA} < 0$: A paleta escurece

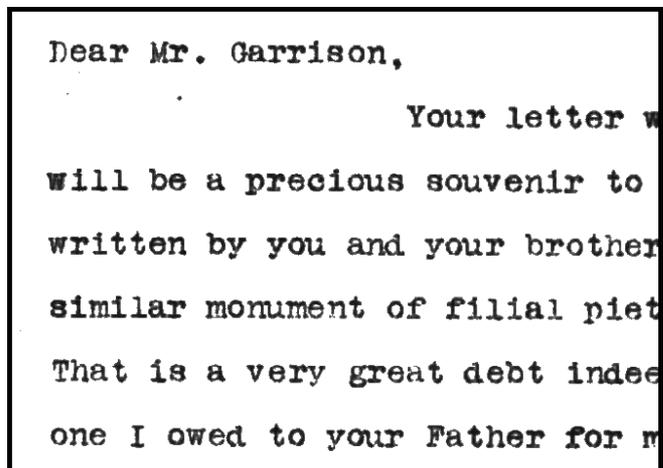
Processamento de Imagens com MatLab

- Imagens carregadas no formato uint8
- É preciso convertê-lo para outro formato para algumas operações
- Operações com imagens
 - Dadas duas imagens com as mesmas dimensões:

```
A = imread ('imagem1', 'ext');  
B = imread ('imagem2', 'ext');  
C = uint8(double(A) + double(B));  
imshow (C);
```

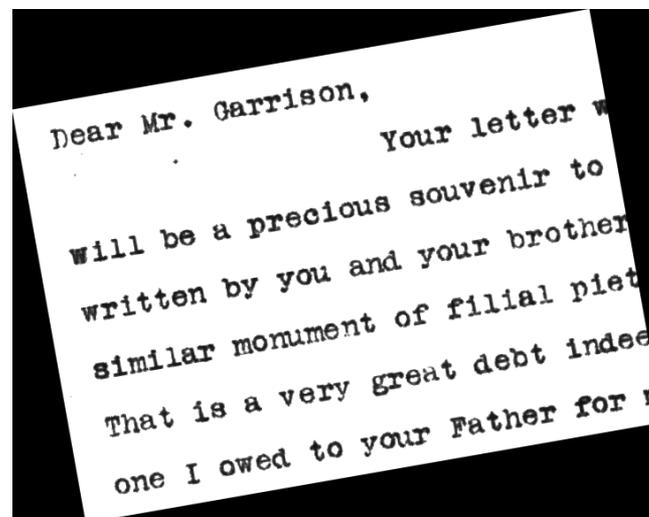
Processamento de Imagens com MatLab

- Transformada de Hough

A rectangular image showing a snippet of a letter. The text is centered and reads: "Dear Mr. Garrison, Your letter will be a precious souvenir to written by you and your brother similar monument of filial piet That is a very great debt indee one I owed to your Father for m".

Dear Mr. Garrison,
Your letter
will be a precious souvenir to
written by you and your brother
similar monument of filial piet
That is a very great debt indee
one I owed to your Father for m

Imagem Original

The same letter snippet as in the previous image, but rotated 10 degrees counter-clockwise. The text is slanted upwards and to the right.

Dear Mr. Garrison,
Your letter
will be a precious souvenir to
written by you and your brother
similar monument of filial piet
That is a very great debt indee
one I owed to your Father for m

Imagem Rotacionada 10°
sentido anti-horário

Processamento de Imagens com MatLab

● Transformada de Hough

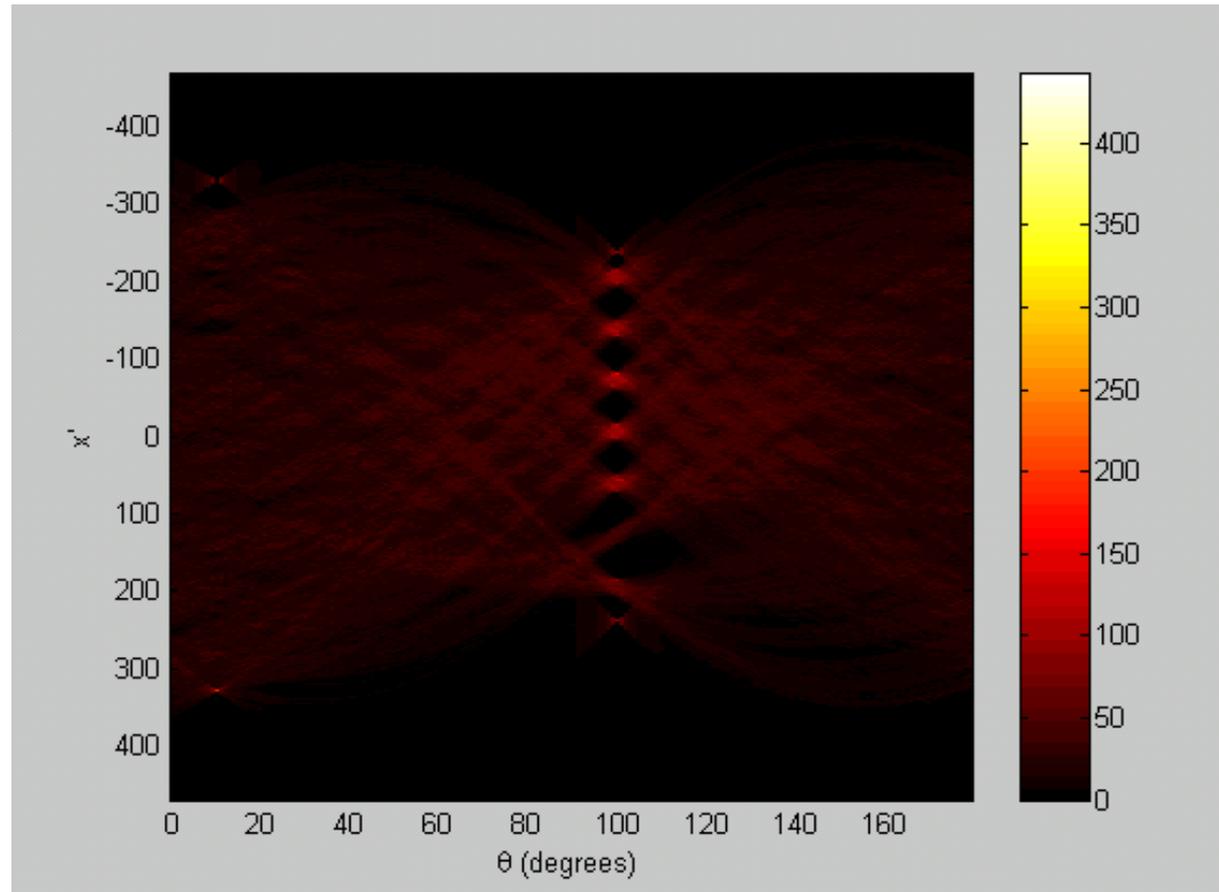
Função no MatLab

```
function y=posrot(x)
BW=edge(x);
theta=0:179;
[R,xp]=radon(double(BW),theta);
imagesc(theta,xp,R); colormap(hot);
xlabel('\theta (degrees)');ylabel('x'); colorbar;
maxr = max(max(R));
tam=size(R);
for i=1:tam(1)
    for j=1:tam(2)
        if (R(i,j)==maxr)
            posi=i; posj=j;
        end
    end
end
y=90-posj;
```

Processamento de Imagens com MatLab

- Transformada de Hough

Resultado= -11°



Processamento de Imagens com MatLab

● Operações morfológicas

■ **strel**

○ **Define um elemento estruturante**

- disk, circle, line, square, octagon, rectangle

○ **se = strel('disk', 15);**

```
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
```



Processamento de Imagens com MatLab

● Operações morfológicas

- **imerode**
- **imdilate**

○ Erosão e dilatação

```
>> se = strel('disk', 3);  
>> im = imread('blob1.bmp');  
>> bw1 = imerode(im, se);  
>> bw2 = imdilate(im, se);
```

Imagem original



Erosão



Dilatação



Processamento de Imagens com MatLab

- Operações morfológicas

- Para complementar uma imagem de duas cores:

- $im = \sim im;$

Imagem original

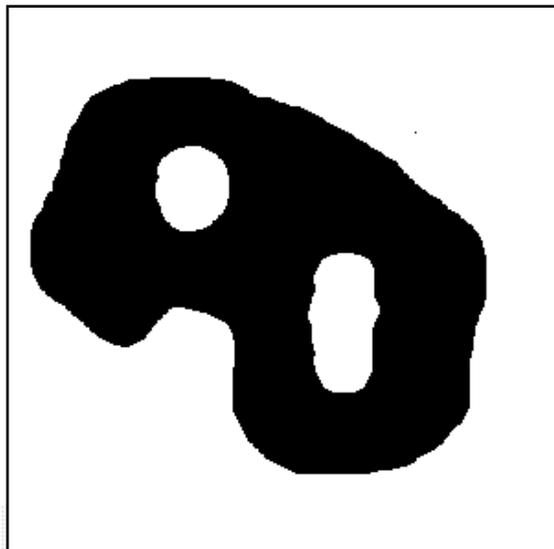


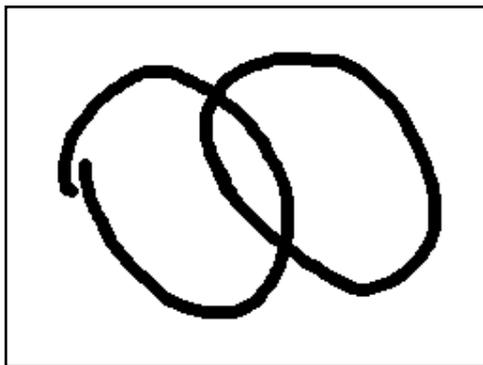
Imagem complementada



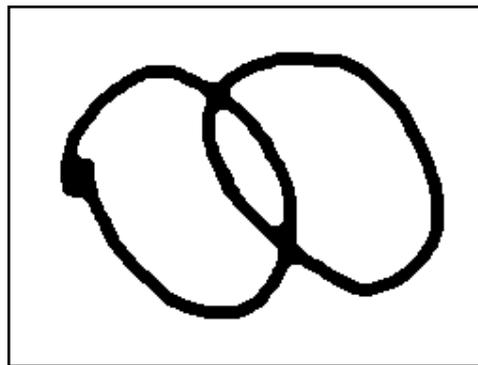
Processamento de Imagens com MatLab

- Operações morfológicas
 - Abertura e Fechamento

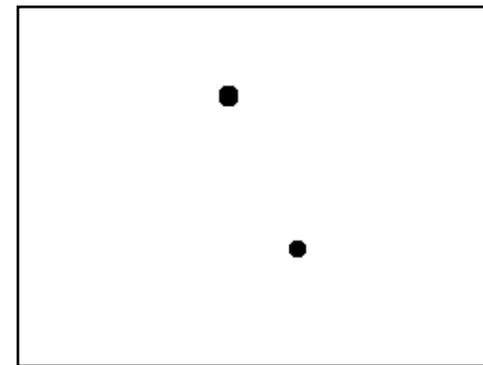
Imagem original



Abertura



Fechamento

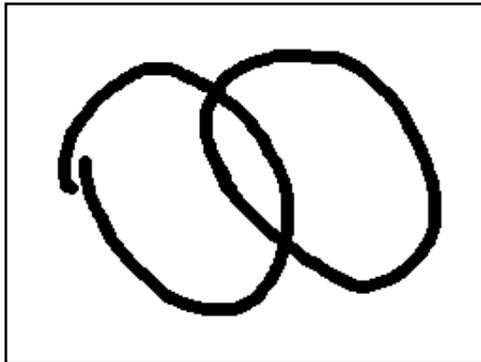


```
im = imread('zero-zero.bmp');  
se = strel('disk', 5);  
bw1 = imopen(im, se);  
bw2 = imclose(im, se);
```

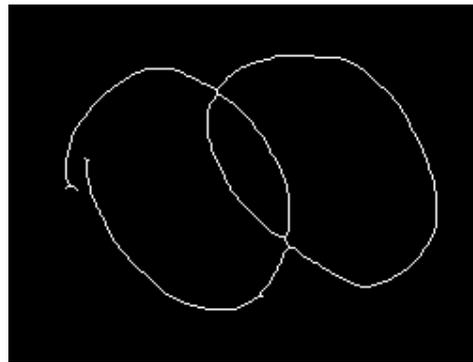
Processamento de Imagens com MatLab

- Operações morfológicas
 - Esqueletização e Afinamento

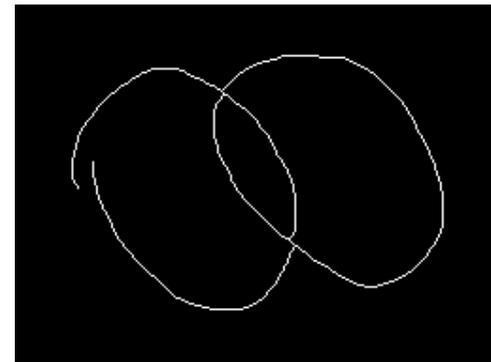
Imagem original



Esqueletização



Afinamento



```
im = imread('zero-zero.bmp');  
im = ~im;  
bw1 = bwmorph(im, 'skel', 'inf');  
bw2 = bwmorph(im, 'thin', 'inf');
```

Processamento de Imagens com MatLab

● Operações morfológicas

■ **bwmorph**

○ endpoints

- Encontra terminações em esqueletos

○ branchpoints

- Encontra pontos de cruzamento em esqueletos

○ e diversas outras funções

Processamento de Imagens com MatLab

● Operações morfológicas

■ **bwlabel**

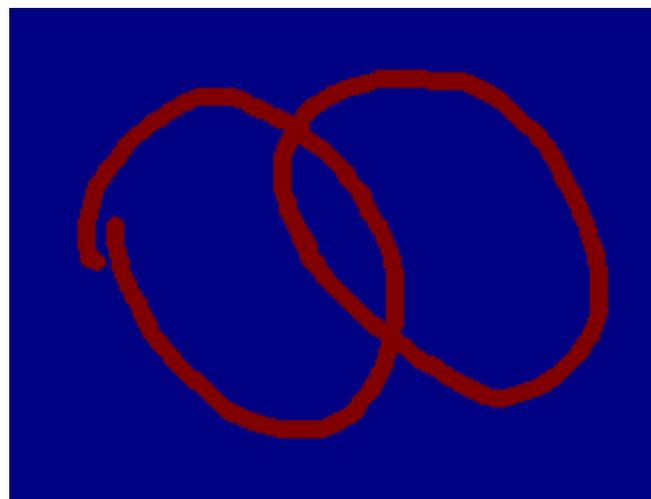
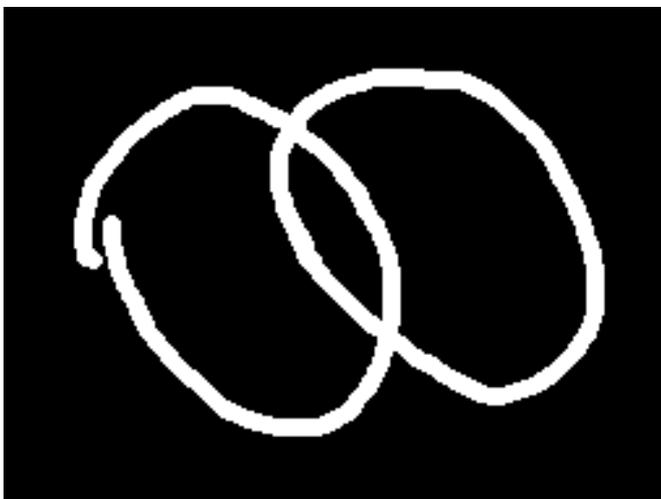
- Legenda os componentes conectados de uma imagem binária
- Os objetos são os elementos em branco
- Cada região em branco conectada recebe uma legenda de 1 a n
- É atribuída a legenda 0 para todas as regiões em preto

Processamento de Imagens com MatLab

● Operações morfológicas

■ **bwlabel**

- `[legenda, n] = bwlabel(im); % n = 1`
- `imagesc(legenda);`



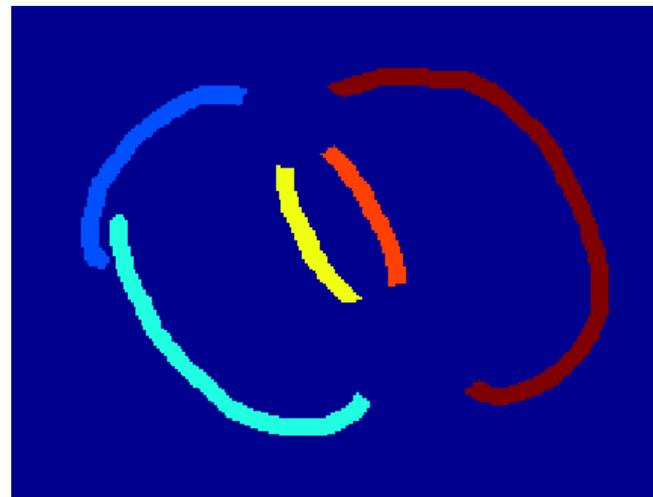
Processamento de Imagens com MatLab

- Operações morfológicas

- **bwlabel**

- `[legenda, n] = bwlabel(im); % n = 5`

- `imagesc(legenda);`



Processamento de Imagens com MatLab

- Criação de função

```
function im2 = teste(nome)
im = imread(nome);
[lin, col] = size(im);
for i = 1:lin
    for j=1:col
        .....
    end
end
imwrite (im2, 'saida.bmp', 'bmp');
```