

# PROCESSAMENTO DE IMAGENS

## Lendo e visualizando imagens no MATLAB

### I Imagens no MATLAB

Qualquer que seja o formato da imagem, o MATLAB manipula as imagens como se fossem matrizes. Uma imagem em níveis de cinza é representada por uma matriz onde cada elemento corresponde ao valor do pixel. Uma imagem colorida pode ser representada de 2 formas: (i) ou por uma matriz onde cada elemento é um índice que aponta para uma tabela com 3 elementos, os valores de R, G e B para aquele pixel, chamado de “colormap”; (ii) ou por 3 matrizes diferentes, R, G e B. Ver o Tutorial do manual do “Image Processing Toolbox” do MATLAB.

Cada matriz pode ser ou do tipo UINT8 ou do tipo DOUBLE. As do tipo UINT8 usam apenas 1 byte para representar cada pixel, enquanto as do tipo DOUBLE usam 8 bytes. Entretanto, as do tipo UINT8 são úteis apenas para visualização de imagem. **Operações aritméticas não são definidas para matrizes do tipo UINT8. Para processamento usa-se apenas as matrizes DOUBLE. Pode ser feita a conversão de tipos usando os comandos `double(X)` e `uint8(X)`. Os pixels das imagens representadas como matrizes DOUBLE estão em geral no intervalo  $[0,1]$ , enquanto as do tipo UINT8 no intervalo  $[0,255]$ .**

### II Formatos das imagens usadas e procedimentos de leitura

As imagens usadas vão estar disponíveis em

[http://www.smt.ufrj.br/~eduardo/imgproc2017/matlabimages\\_cursos/](http://www.smt.ufrj.br/~eduardo/imgproc2017/matlabimages_cursos/) Elas poderão ter, entre outras, as seguintes extensões: “.MAT”, “.BMP”, “.TIF”, “.JPEG” e “.GIF”.

**.MAT** : São lidas com `load <nome>`, o nome sem a extensão. A saída é composta de 2 partes: (i) uma imagem indexada na matriz `X` (de um modo geral, mas em algumas imagens vale a pena dar um comando `whos` para ver o nome da matriz); (ii) um “colormap”, de um modo geral com o nome `map`.

**Outros formatos :**

- **Imagens em níveis de cinza:** Usa-se o comando `X = imread('nome', 'formato')` ou `X = imread('nome.ext')`. Ele lê imagens em níveis de cinza diretamente em `X`, no formato UINT8. A sua vantagem é que os elementos da matriz `X` são representados com 1 byte (tipo de dados UINT8), ao invés dos 8 bytes nos casos das funções `load` e `imread`. Entretanto, operações aritméticas não são definidas para o tipo UINT8. Assim sendo, para processamento é necessário que as matrizes obtidas sejam convertidas para tipo DOUBLE (`Y = (double) X`).
- **Imagens coloridas:** Usa-se o comando `RGB = imread('nome', 'formato')` ou `RGB = imread('nome.ext')`. Ele lê as três componentes R, G, e B da imagem no array tridimensional `RGB`, no formato UINT8. Para processamento é necessário que as matrizes obtidas sejam convertidas para tipo DOUBLE (`Y = (double) X`).

### III Procedimentos de “Display” para os diferentes formatos de imagens

**Imagem indexada:** `imshow(X,map)`.

**Imagem em níveis de cinza:**

- `imshow(X)`: Dá o display da imagem na matriz `X`. Note que para uma imagem `UINT8` se mapeia 0 para preto e 255 para branco. Já no caso de uma imagem `DOUBLE` se mapeia 0 para preto e 1 para branco.
- `imagesc(X)`: Dá o display da imagem em níveis de cinza no array `X` do tipo `DOUBLE` mapeando o valor máximo para 1 e o valor mínimo para zero. Para se visualizar a imagem em níveis de cinza, deve-se usar após ele o comando `colormap(gray)`.
- `imshow('nome.ext')`: Dá o display da imagem diretamente do arquivo.

**Imagem RGB:** `imshow(RGB)`, onde `RGB` é um array tridimensional que contém a imagem.

**“Display” de varias imagens simultâneamente:** Usar o comando

`subplot(m,n,j)`, `imshow(...)`, onde a janela de “display” é dividida em  $m \times n$  subjanelas e a imagem em questão vai aparecer na janela `j`. ATENÇÃO: O “colormap” é comum, então deve se ter cuidado na hora de dar o “display” simultâneo de imagens indexadas (ver o manual do Image Processing Toolbox pp.1-30–1.32).

**Comando `imtool`:** Um comando útil, que além de visualizar uma imagem permite várias outras operações, incluindo a inspeção de valores usando o mouse. A sintaxe é `imtool(X)`, onde `X` é a matriz que contém a imagem.

### IV Procedimentos de conversão de uma imagem de um tipo para outro

**`ind2gray`:** Converte imagem indexada para grayscale, com valores no intervalo  $[0,1)$ .

**`gray2ind`:** Função inversa da `ind2gray`.

**`ind2rgb`:** Converte imagem indexada para RGB.

**`rgb2ind`:** Função inversa da `ind2rgb`.

**`rgb2gray`:** Converte imagem RGB para “grayscale”.