

Notas de aula e prática MATLAB #02

Arquivos necessários [disponíveis no zip ou no MATLAB]

1. gDSC04422.tif
2. cameraman.tif [MATLAB *built-in*]
3. cameraman2.tif [http://www.ogemarques.com/wp-content/uploads/2014/11/Tutorial_Images.zip]
4. earth1.tif [http://www.ogemarques.com/wp-content/uploads/2014/11/Tutorial_Images.zip]
5. earth2.tif [http://www.ogemarques.com/wp-content/uploads/2014/11/Tutorial_Images.zip]
6. contrast.jpg [<http://www.isi.uu.nl/Research/Gallery/DSA/>]
7. mask.jpg [<http://www.isi.uu.nl/Research/Gallery/DSA/>]
8. obj_bw.tif
9. msk_bw.tif
10. msk_gray_bw.png
11. obj_gray.png [<http://www.scotsman.com/edinburgh-evening-news/latest-news/source-of-legionnaires-outbreak-could-be-revealed-within-days-1-2350825>]

Operações aritméticas entre imagens

[<https://www.mathworks.com/help/images/image-arithmetic.html>]

imabsdiff	Absolute difference of two images
imadd	Add two images or add constant to image
imapplymatrix	Linear combination of color channels
imcomplement	Complement image
imdivide	Divide one image into another or divide image by constant
imlincomb	Linear combination of images
immultiply	Multiply two images or multiply image by constant
imsubtract	Subtract one image from another or subtract constant from image

2.1) Operações aritméticas (média de imagens)

Imagens adquiridas em condições de pouca iluminação podem apresentar níveis altos de ruído. Isto pode ser claramente observado na prática se fotografarmos um objeto em um ambiente pouco iluminado, utilizando a câmera digital em uma configuração de alta sensibilidade (alto ISO) e o zoom óptico. A imagem *gDSC04422.png* foi adquirida nestas condições, com uma máquina comum, modelo Sony DSC-W50. Observe que a imagem é na verdade um grid de 16 fotografias, adquiridas uma após a outra em um curto intervalo de tempo. Este tipo de aquisição é feita automaticamente pela máquina e é identificado pela opção *Multi Burst*.

Utilize o método da média das imagens para suavizar o ruído da imagem *gDSC04422.png*. Mostre uma imagem original ao lado de uma imagem processada. Código Jedi para extrair cada imagem do grid de imagens. Resposta disponível em a02_01.m.

[Matlab array manipulation tips and tricks, tópico 6.1.3

<https://www.ee.columbia.edu/~marios/matlab/Matlab%20array%20manipulation%20tips%20and%20tricks.pdf>]:

```
gmb = imread('gDSC04422.png');  
m = size(gmb,1);  
n = size(gmb,2);  
p = m/4;  
q = n/4;  
X = gmb;  
Y = reshape( X, [ p m/p n ] );  
Y = permute( Y, [ 1 3 2 ] );  
Y = reshape( Y, [ p q m*n/(p*q) ] );  
% Y(:, :, 1) é uma das 16 imagens,  
% Y(:, :, 2) é outra...
```

2.2) Operações aritméticas (insubtract, imabsdiff)

[[OM] Tutorial 6.1, após Question 2]

Use a função `insubtract` para encontrar a diferença entre as imagens *cameraman.tif* e *cameraman2.tif*. Mostre as imagens de entrada e a de saída. Note que não é possível visualizar todos os detalhes do prédio. Isto acontece porque, na subtração, alguns pixels resultaram em números negativos que foram truncados em 0 pela função `insubtract`. Então, trabalhe também a subtração das imagens em `double`. Resposta disponível em a02_02.m.

2.3) Operações aritméticas (multiplicação)

[[OM], Tutorial 6.1, step 15]

Image multiplication can be used for special effects such as an artificial 3D look. By multiplying a flat image with a gradient, we create the illusion of a 3D textured surface.

Create an artificial 3D planet by using the `immultiply` function to multiply the *earth1.tif* and *earth2.tif* images." Não normalize a saída (`mat2gray` ou normalização na unha depois do `immultiply`). Resposta disponível em a02_03.m.

2.4) Operações aritméticas (subtração)

Angiography or arteriography is a medical imaging technique used to visualize the inside of blood vessels and organs of the body, with particular interest in the arteries, veins and the heart chambers. This is traditionally done by injecting a radio-opaque contrast agent into the blood vessel and imaging using X-ray based techniques [<http://en.wikipedia.org/wiki/Angiography>]."

O processo de obtenção da imagem é o seguinte:

1. Adquire-se uma imagem de raio-x dos vasos a serem visualizados. Esta imagem é chamada de *mask image*.
2. Injeta-se o contraste no paciente.
3. Adquire-se novamente a imagem de raios-x. Esta imagem é chamada de *contrast image*.
4. A subtração das imagens *contrast* e *mask* ($contrast - mask$) é o resultado final, no qual a maior parte do background é eliminado e os vasos estão mais visíveis.

Obtenha o resultado do exame de angiografia das imagens *contrast.jpg* e *mask.jpg* [<http://www.isi.uu.nl/Research/Gallery/DSA/>]. Mostre as imagens de entrada e a imagem de saída. Resposta disponível em a02_04.m.

Operações lógicas entre imagens da classe logical

Sintaxe1) `A & B` Sintaxe2) `and(A,B)` Find logical AND [<https://www.mathworks.com/help/matlab/ref/and.html>]
Sintaxe1) `A | B` Sintaxe2) `or(A,B)` Find logical OR [<https://www.mathworks.com/help/matlab/ref/or.html>]
`C = and(A,B)` Find logical exclusive-OR [<https://www.mathworks.com/help/matlab/ref/xor.html>]
Sintaxe1) `~A` Sintaxe2) `not(A)` Find logical NOT [<https://www.mathworks.com/help/matlab/ref/not.html>]

2.5) Operações lógicas (imagens logical)

Realize as operações lógicas requisitadas e mostre os resultados. Utilize as imagens *A = obj_bw.png* e *B = msk_bw.png*.

1. `not A`
 2. `not B`
 3. `A and B`
 4. `A or B`
 5. `A xor B`
 6. `A and (not B)`. Esta é a operação *subtração de conjuntos* $A - B$.
- Resposta disponível em a02_05.m.

Operações bitwise (bit-a-bit) e manipulação de bits

[<https://www.mathworks.com/help/matlab/bit-wise-operations.html>]

bitand	Bit-wise AND
bitor	Bit-wise OR
bitxor	Bit-wise XOR
bitcmp	Bit-wise complement
bitget	Get bit at specified position
bitset	Set bit at specific location
bitshift	Shift bits specified number of places
swapbytes	Swap byte ordering

2.6) Operações lógicas (imagens uint8)

[Imagem *obj_gray.png* extraída de <http://www.scotsman.com/edinburgh-evening-news/latest-news/source-of-legionnaires-outbreak-could-be-revealed-within-days-1-2350825>]

Utilize uma operação lógica bit-a-bit (entre imagens uint8) entre as imagens *msk_gray_bw.png* e *obj_gray.png* para selecionar o disco de Petri da imagem *obj_gray.png*. Nesta aplicação, a imagem *msk_gray_bw.png* pode ser chamada de 'imagem máscara'. Observe que a imagem *msk_gray_bw.png* é do tipo lógico e o exercício pede operações bit-a-bit sobre imagens uint8. Resposta disponível em a02_06.m.

Common Geometric Transformations

[<https://www.mathworks.com/help/images/geometric-transformations.html>]

imcrop	Crop image
imcrop3	Crop 3-D image
imresize	Resize image
imresize3	Resize 3-D volumetric intensity image
imrotate	Rotate image
imrotate3	Rotate 3-D volumetric grayscale image
imtranslate	Translate image
impyramid	Image pyramid reduction and expansion

Generic Geometric Transformations

[<https://www.mathworks.com/help/images/spatial-referencing.html>]

imwarp	Apply geometric transformation to image
affineOutputView	Create output view for warping images
fitgeotrans	Fit geometric transformation to control point pairs
findbounds	Find output bounds for spatial transformation
fliptform	Flip input and output roles of spatial transformation structure
makeresampler	Create resampling structure
maketform	Create spatial transformation structure (TFORM)
tformarray	Apply spatial transformation to N-D array
tformfwd	Apply forward spatial transformation
tforminv	Apply inverse spatial transformation

2.7) Operações geométricas (função imresize)

imresize: <https://www.mathworks.com/help/images/ref/imresize.html>



Assista ao vídeo no
post desta aula



Use a função *imresize* para redimensionar a imagem *cameraman.tif* para o dobro do tamanho original, usando interpolação nearest neighbour e bicubic. Note que na interpolação bicubic o artefato das bordas parecerem "recortadas" (jaggadness ou pixelation) é menor, se comparado ao do nearest neighbour. Resposta disponível em a02_07.m.

2.8) Operações geométricas (função `imrotate`)

`imrotate`: <https://www.mathworks.com/help/images/ref/imrotate.html>



Assista ao vídeo no
post desta aula



Use a função `imrotate` para rotacionar a imagem gerada por `checkerboard()` 30° no sentido horário, usando interpolação `nearest neighbour` e `bicubic`. Note que na interpolação `bicubic` o artefato das bordas parecerem “recortadas” (*jaggadness* ou *pixelation*) é menor, se comparado ao do `nearest neighbour`. Resposta disponível em `atv02_08.m`.

Referências

[OM] Oge Marques, Practical image and video processing using MATLAB, Wiley, 2011.