

## ATIVIDADE

### Assunto:

Processamento morfológico – algoritmos básicos – parte 1.

### Orientações:

A atividade deve ser executada individualmente e entregue através do ambiente *Google Classroom*.

### Nome completo:

**Raul Aquino de Araújo**

1. Utilize a ferramenta Octave Online (<https://octave-online.net>) para fazer o que se pede (a resposta deve ser dada em formato de relatório, onde o código-fonte criado para cada item deve ser seguido do resultado/imagem obtido):

- a. Com o intuito de demonstrar a extração de fronteiras, faça o que se pede:

- i. Carregue a imagem `licoln_from_penny.tif` (fornecida em anexo)

```
octave:6> I = imread('licoln_from_penny.tif');  
octave:7> imshow(I);
```



- ii. Declare um elemento estruturante 3x3 preenchido com valores iguais a 1

```
octave:8> se = strel("square", 3)  
se =  
Flat STREL object with 9 neighbors  
  
Neighborhood:  
1 1 1  
1 1 1  
1 1 1
```

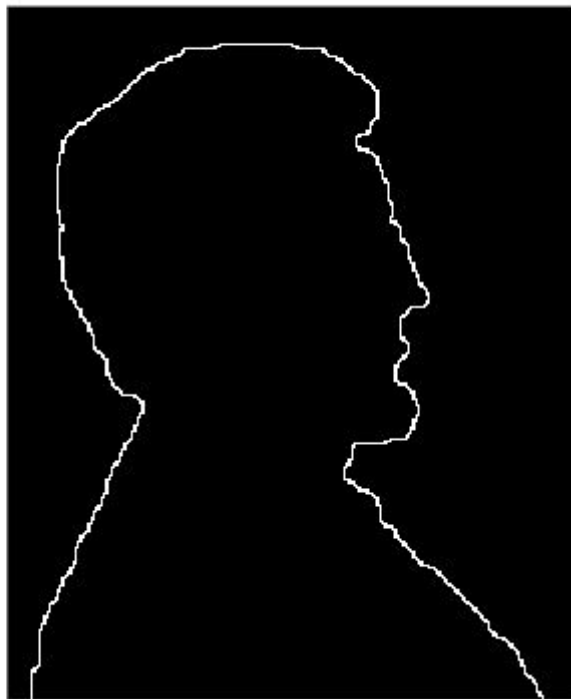
- iii. Realize a erosão da imagem original pelo elemento estruturante

```
octave:9> I2 = imerode(I, se);  
octave:10> imshow(I2);
```

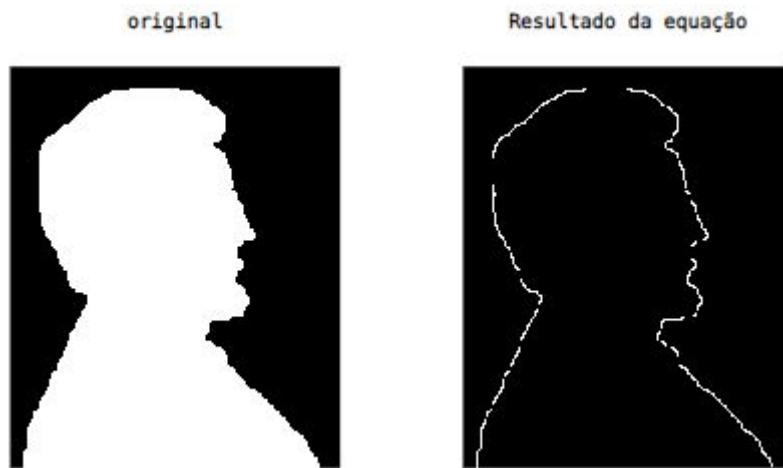


- iv. Crie uma imagem de resultado equivalente à subtração da imagem original pelo resultado da erosão

```
octave:11> I3 = I - I2;  
octave:12> imshow(I3);
```



- v. Exiba as imagens original e de resultado, comparando se o resultado obtido está similar ao demonstrado na Figura 9.14(b) do livro PDI-Gonzalez (pág. 424).



2. Utilize a ferramenta Octave Online (<https://octave-online.net>) para fazer o que se pede (a resposta deve ser dada em formato de relatório, onde o código-fonte criado para cada item deve ser seguido do resultado/imagem obtido):

a. Com o intuito de demonstrar como a operação de preenchimento de buracos pode ser útil para corrigir uma limiarização falha, faça o que se pede:

i. Carregue a imagem coins.png (fornecida em anexo)

```
octave:4> I = imread('coins.png');
octave:5> imshow(I);
```

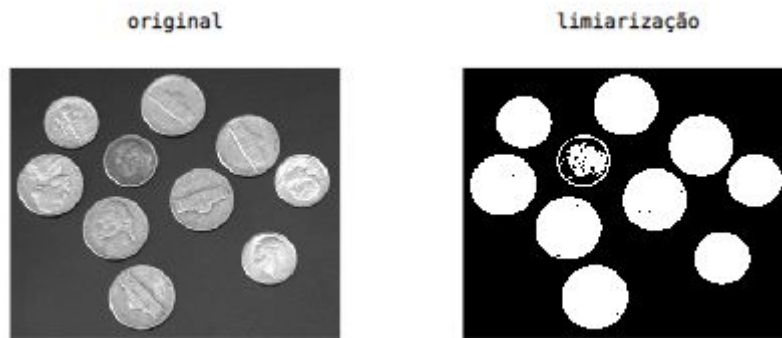


ii. Usando a função `graythresh`, calcule automaticamente o limiar `T`

iii. Usando a função `im2bw`, realize a limiarização da imagem original com o limiar `T`

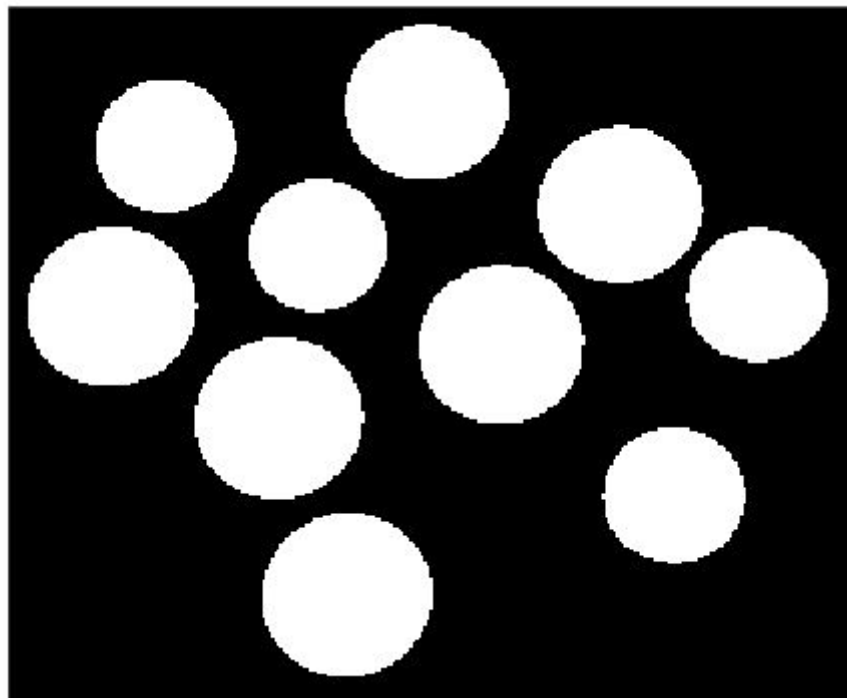
```
octave:6> T1 = graythresh(I)
T1 = 0.4941
octave:7> BW1 = im2bw(I, T1);
octave:8> figure(1),
subplot(1,2,1), imshow(I), title('original'),
subplot(1,2,2), imshow(BW1), title('limiarização');
```

iv. Compare a imagem original com a imagem segmentada, demonstrando a falha na segmentação de algumas moedas (pixels pretos na parte interna da moeda)

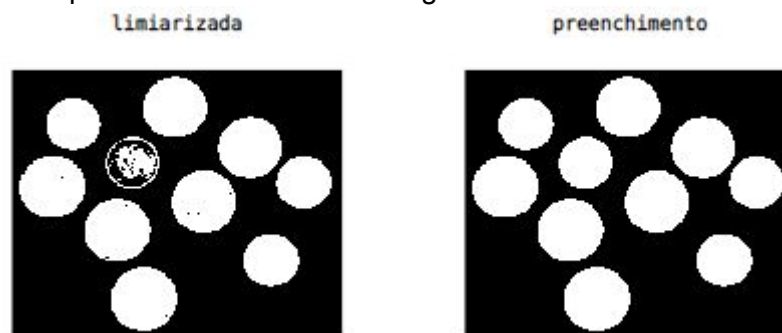


- v. Aplique o preenchimento de buracos na imagem limiarizada (use a função `imfill` com o parâmetro 'holes')

```
octave:16> PB = imfill(BW1,'holes');
octave:20> imshow(PB);
```



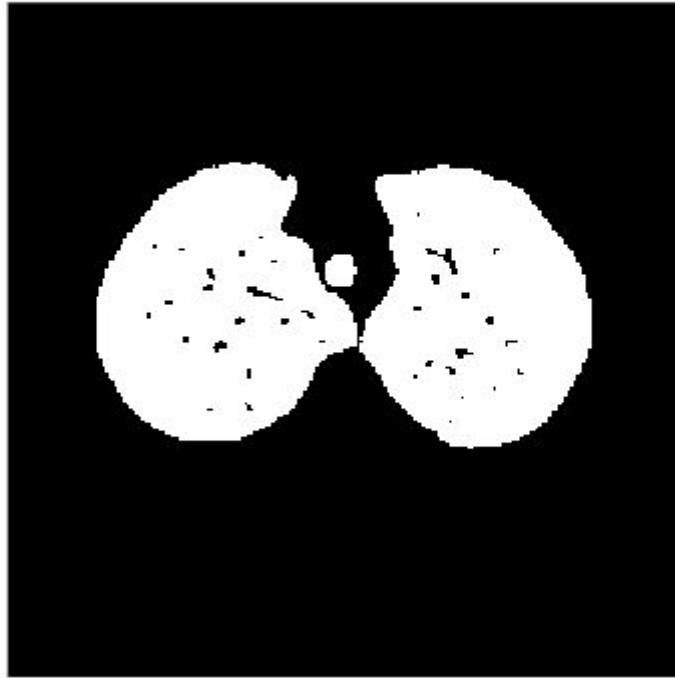
- vi. Compare a imagem segmentada com a imagem após o preenchimento de buracos e verifique se as falhas foram corrigidas



3. Utilize a ferramenta Octave Online (<https://octave-online.net>) para fazer o que se pede (a resposta deve ser dada em formato de relatório, onde o código-fonte criado para cada item deve ser seguido do resultado/imagem obtido):

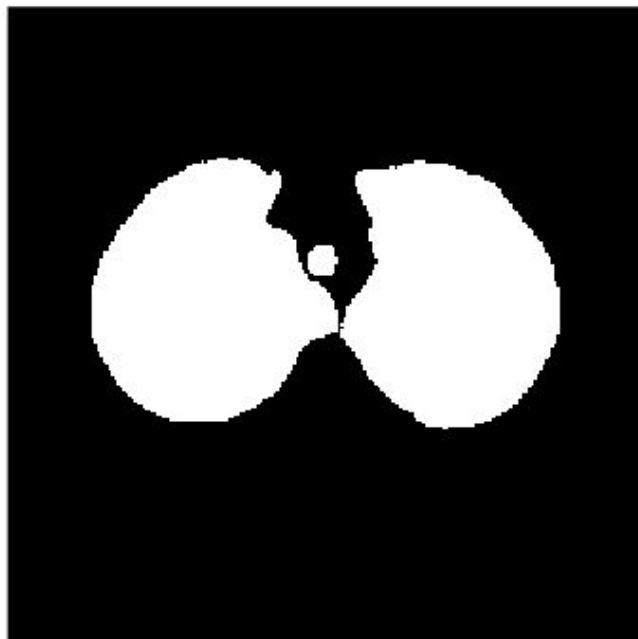
- a. Carregue a imagem `lung_seg.png` (fornecida em anexo)

```
octave:3> I = imread('lung_seg.png');  
octave:4> imshow(I);
```



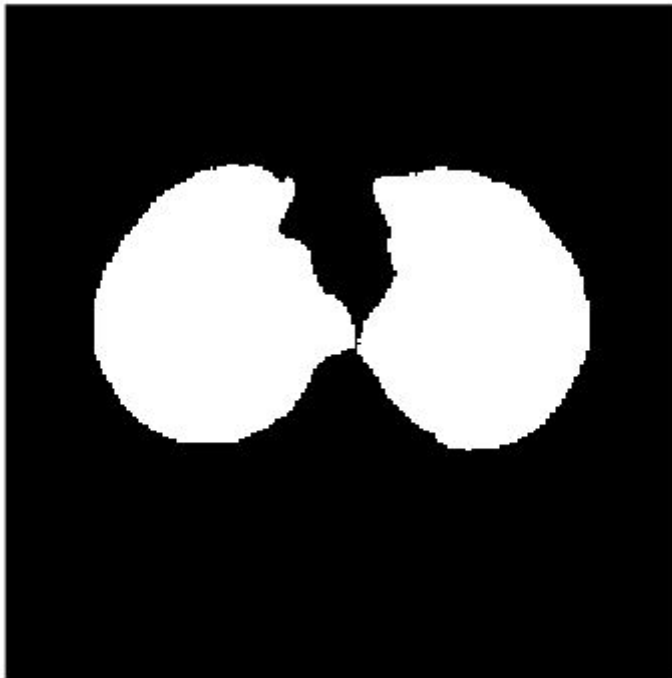
- b. Esta imagem apresenta uma fatia (*slice*) no plano axial de um exame de tomografia computadorizada do tórax, após segmentação do pulmão utilizando uma função de limiarização. É possível observar 3 regiões principais: o pulmão esquerdo, o pulmão direito e um terceiro elemento de menor tamanho, que representa a traqueia do paciente examinado. Há falhas na segmentação de ambos os pulmões. Estas regiões foram obtidas através da segmentação pois possuem menor densidade do que as demais regiões, uma vez que estas são compostas majoritariamente por ar
- c. O objetivo do processamento é eliminar a traqueia e preencher as falhas existentes na segmentação dos dois pulmões
- d. Para tanto, inicialmente aplique uma função de preenchimento de buracos e compare o resultado com a imagem original. Os buracos presentes nos dois pulmões devem ter sido eliminados.

```
octave:6> H = imread('lung_seg.png');  
octave:7> PB1 = imfill(H, 'holes');  
octave:8> imshow(PB1);
```



- e. Em seguida, através do uso da função `bwconncomp`, realize a análise dos componentes conectados e exclua o componente com menor área (traqueia)

```
octave:29> CC = bwconncomp(PB1);  
octave:30> CC.NumObjects  
ans = 3  
octave:33> numPixels = cellfun(@numel, CC.PixelIdxList);  
octave:34> [smallest, idx] = min(numPixels);  
octave:35> AUX = PB1;  
octave:37> AUX(CC.PixelIdxList{idx}) = 0;  
octave:38> imshow(AUX);
```

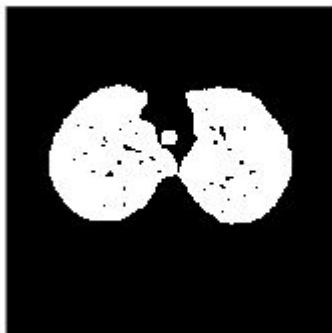


- f. Compare o resultado deste processamento com a imagem gerada após o preenchimento de buracos

```
octave:41> figure,  
subplot(1,2,1), imshow(H), title('original'),  
subplot(1,2,2), imshow(AUX), title('sem traqueia');
```

original

sem traqueia



Boa sorte!

Prof. Igor.