



UFOP

Universidade Federal
de Ouro Preto

Trabalho de Implementação

**Aluno em Graduação da Universidade
Federal de Ouro Preto do curso Ciência da**

Computação:

Halliday Gauss Costa dos Santos.

Matrícula: 18.1.4093.

Área: Processamento de Imagens.

Questão 1)

Código da função que remove o ruído da imagem:

```
Editor - C:\Users\halli\Desktop\7Periodo\PDI\Listas\Lista7\removeRuido.m
removeRuido.m  X +
1  function removeRuido()
2
3      % recebe uma imagem
4
5  -   img = imread("C:\Users\halli\Desktop\7Periodo\PDI\Listas\Lista7\coins.png");
6
7  -   imgbw = im2bw(img, 0.5); % garantir que ela eh binaria
8
9  -   figure; imshow(im2double(img), []); % mostrar imagem original binaria
10
11
12      % inserir ruído
13  -   x = rand(size(imgbw));
14
15  -   d1 = x <= 0.05;
16
17  -   d2 = x >= 0.95;
18
19  -   imgbw(d1) = 0;
20
21  -   imgbw(d2) = 1;
22      %-----
23
24  -   figure; imshow(im2double(imgbw), []); % mostrar imagem ruidosa
25
26      %-----
```

```
Editor - C:\Users\halli\Desktop\7Periodo\PD\Listas\Lista7\removeRuido.m
removeRuido.m  x  +
26  %-----
27  -   sp = ones(3); % elemento estruturante 1
28
29  -   abertura = imopen(imgbw, sp); % operacao de abertura
30
31  -   figure; imshow(im2double(abertura), []); % mostrar imagem apos abertura
32
33  -   fecho = imclose(abertura, sp); % operacao de fecho
34
35  -   figure; imshow(im2double(fecho), []); % mostrar imagem apos fecho
36
37  %-----
38
39  %-----
40  -   sk = [0 1 0; 1 1 1; 0 1 0]; % elemento estruturante 2
41  |
42  -   abertura = imopen(imgbw, sk); % operacao de abertura
43
44  -   figure; imshow(im2double(abertura), []); % mostrar imagem apos abertura
45
46  -   fecho = imclose(abertura, sk); % operacao de fecho
47
48  -   figure; imshow(im2double(fecho), []); % mostrar imagem apos fecho
49
50  %-----
51
52  -   end
```

Após aplicar esse código será mostrado as seguintes imagens:

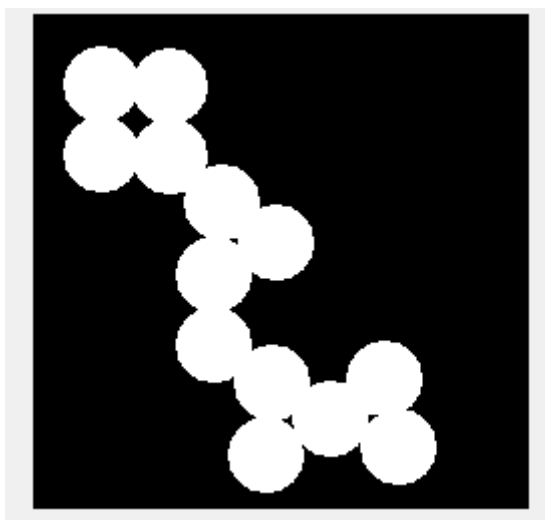


Imagem Original

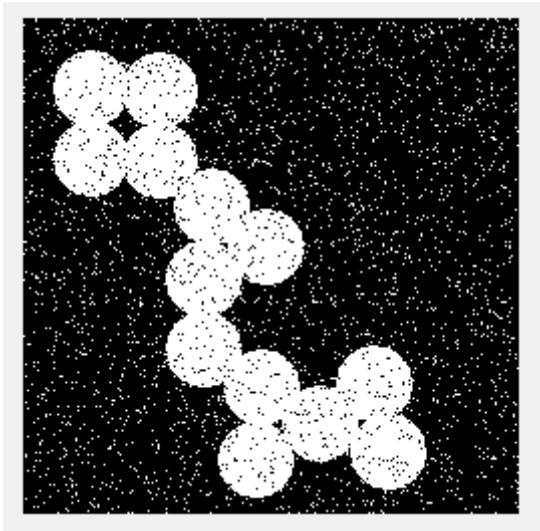


Imagem com Ruído

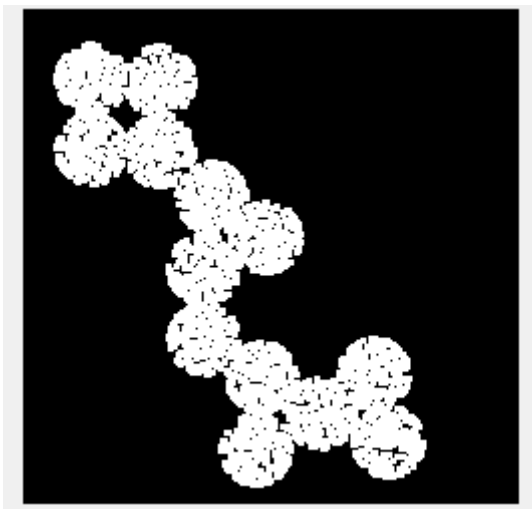


Imagem após Abertura utilizando o Elemento Estruturante sp

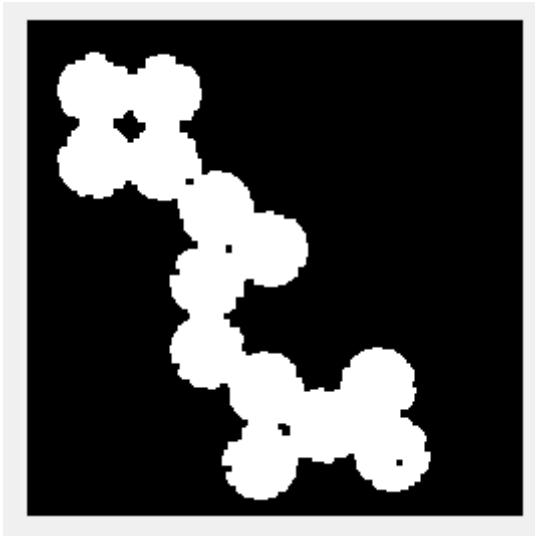


Imagem após Fecho utilizando o Elemento Estruturante sp

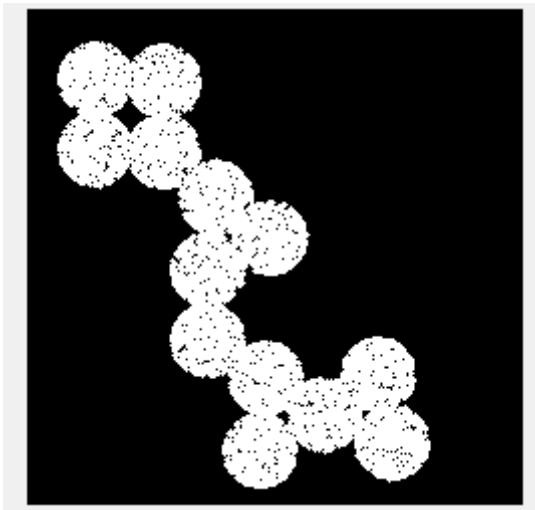


Imagem após Abertura utilizando o Elemento Estruturante sk

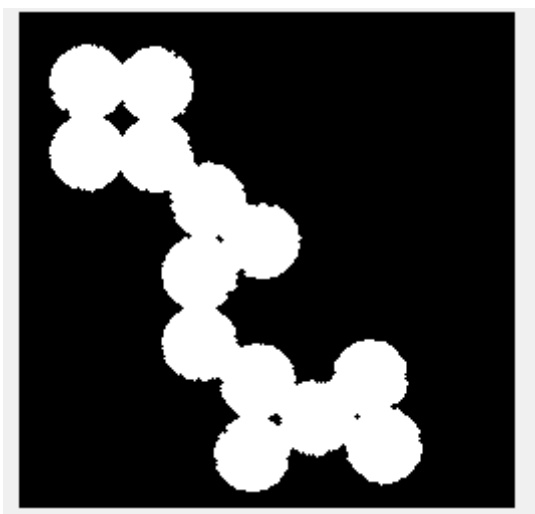


Imagem após Fecho utilizando o Elemento Estruturante sk

Questão 2)

Código para preenchimento de buracos em uma imagem binarizada

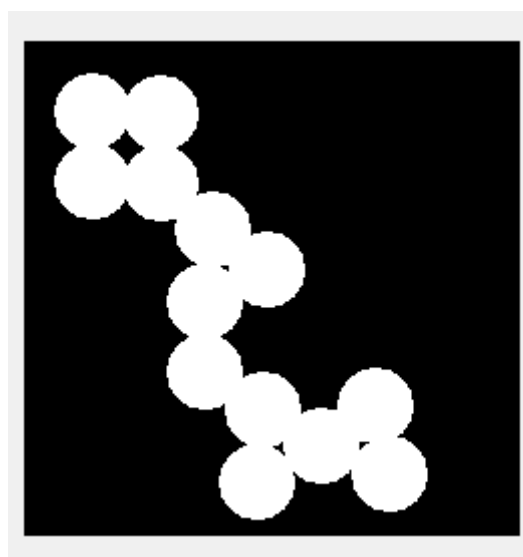
```
Editor - C:\Users\halli\Desktop\7Periodo\PDI\Listas\Lista7\preenchimento.m
removeRuido.m x preenchimento.m x +
1 function nimg = preenchimento(img)
2     [lin, col, ~] = size(img);
3
4     xk = zeros(lin, col);
5
6     imgc = imcomplement(img); % complemento da imagem
7
8     se = strel('diamond', 1); % elemento estruturante
9
10    dl = find(imgc == 0);
11
12    x = xk;
13
14    xk(dl(1)) = 1; % ligar 1 pixel
15
16    while xk ~= x % enquanto x da iteracao atual for diferente do x da iteracao anterior
17
18        x = xk
19
20        aux = imdilate(xk, se); % dilatacao utilizando o elemento estruturante
21
22        xk = min(aux, imgc); % interseccao
23    end
24
25    nimg = max(img, xk); % uniao com a imagem original
26 end
```

Questão 3)

A partir dos seguintes comandos:

```
>> img = imread('C:\Users\halli\Desktop\7Periodo\PDI\Listas\Lista7\coins.png')
>> imshow(img);
```

É mostrada a imagem binária abaixo que será utilizada para os testes de detecção de bordas:



a) Identificar Bordas Internas em imagens binárias:

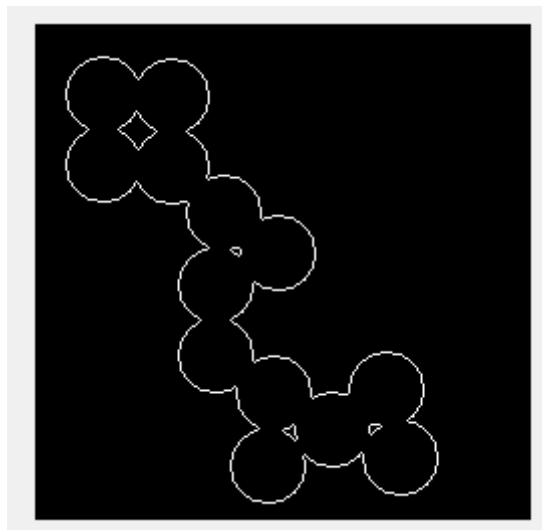
Código:

```
bordaInterna.m*  X  +  
1  function nimg = bordaInterna(img)  
2  -     se = strel('diamond', 1); % elemento estruturante  
3  -     erosao = imerode(img, se);  
4  -     nimg = img - erosao;  
5  -     imshow(nimg);  
6  -     end
```

Execução:

```
>> bordaInterna(img);
```

Resultado:



b) Identificar Bordas Externas em imagens binárias:

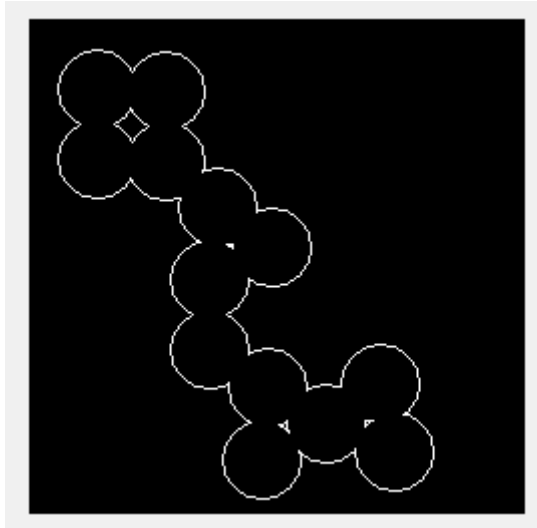
Código:

```
Editor - C:\Users\hami\Desktop\7º Período\PD1\Listas\Lista7\bordaExterna.m  
bordaExterna.m  X  +  
1  function nimg = bordaExterna(img)  
2  -     se = strel('diamond', 1); % elemento estruturante  
3  -     dilatacao = imdilate(img, se);  
4  -     nimg = dilatacao - img;  
5  -     imshow(nimg);  
6  -     end
```

Execução:

```
>> bordaExterna(img);
```

Resultado:



c) Identificar Bordas (Gradiente Morfológico) em imagens binárias:

Código:

```
gradienteMorfologico.m  x  +
1  function nimg = gradienteMorfologico(img)
2  -     se = strel('diamond', 1); % elemento estruturante
3  -     dilatacao = imdilate(img, se);
4  -     erosao = imerode(img, se);
5  -     nimg = dilatacao - erosao;
6  -     imshow(nimg);
7  - end
```

Execução:

```
>> gradienteMorfologico(img);
```

Resultado:

