

PRÁTICA 6

Prof. Dr. Evandro Luis Linhari Rodrigues

Além da solução usando Matlab, todos os exercícios deverão ser implementados também usando Python e OpenCV.

Segmentação de Imagens – Parte 2.

1) Thresholding (limiarização)

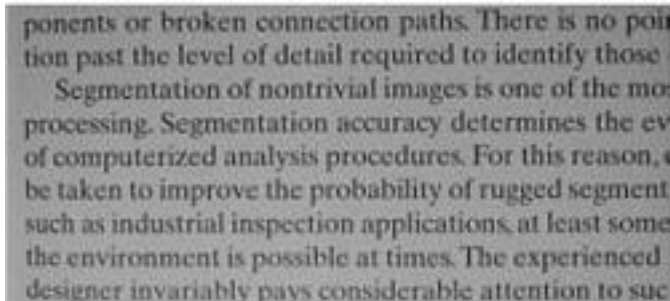


Figura 1 - texto.bmp

O MatLab possui uma função que calcula o Threshold Automático de uma imagem através do método de Otsu. Esta função tem a seguinte sintaxe:

$$T = \text{graythresh}(I)$$

Onde I é a imagem de classe uint8, uint16, ou double e T é o nível de limiar calculado entre [0, 1]

T_1: Implementar o Algoritmo de Threshold Automático (Gonzalez;Woods,2002) dado em classe, para a imagem da Figura 1:

```
f = imread('texto.bmp');
hf = imhist(f);
figure, imshow(f), figure, plot(hf)
T1 = 0.5*(double(min(f(:)))+double(max(f(:))));
done = false;
while ~done
    g = f >= T1;
    T1next = 0.5*(mean(f(g)) + mean(f(~g)));
    done = abs(T1 - T1next) < 0.5;
    T1 = T1next;
end
T2 = T1/255;
s1 = im2bw(f,T2);
figure, imshow(s1)
```

E_1: Utilizar a função do Matlab (Método de Otsu) na imagem da Figura 1 e comparar os valores do Threshold calculados pelos dois métodos, bem como a qualidade das imagens geradas.

2) Crescimento de Região.

O Toolbox de Processamento de Imagens do MatLab não possui uma função para Crescimento de Região. O livro (Gonzalez;Woods;Eddins,2004) tem uma função para realizar o Crescimento de Região (*regiongrow.p*) cuja sintaxe é:

$$[g, NR, SI, TI] = \text{regiongrow}(f, S, T)$$

onde f é a Imagem de entrada, S o conjunto de Sementes e T o Threshold a ser aplicado na Imagem. Se S for uma Matriz, ela deve conter 1 nas coordenadas das Sementes e 0 no resto. Se S for um escalar ele define um valor de intensidade para as Sementes na imagem f .

Se T for uma Matriz, ele deve conter o valor de Threshold em cada coordenada de f . Se T for um escalar ele define um Threshold Global para a imagem.

O valor do Threshold T serve para verificar a diferença de intensidades entre o pixel 8-conectado e a Semente.

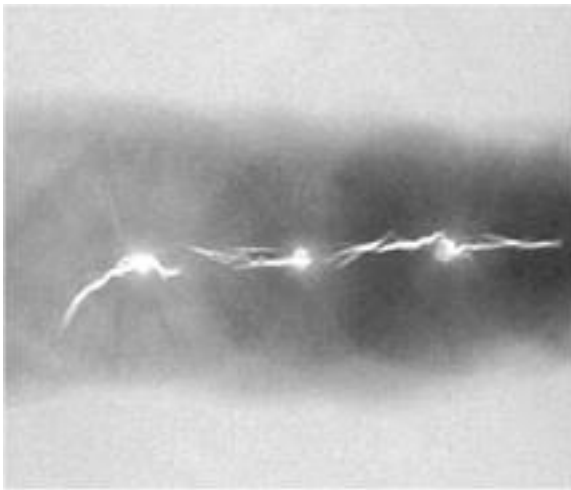


Figura 2 - Solda.bmp

E_2: Realizar o Crescimento de Região na Imagem da Figura 2, para segmentar o defeito na solda.

- a) Utilizar o valor da Semente $S = 255$ pois o defeito observado na figura parece conter pixels brancos, e o limiar $T=65$, ou seja, pixels 8-conectados à semente cuja diferença não exceda a T pertencerá à região.
- b) Variar o Valor de S e T baseado no histograma de f e analisar os resultados obtidos.

Obs. Copiar a função *regiongrow.p* para o diretório de trabalho.

3) Transformada Watershed.

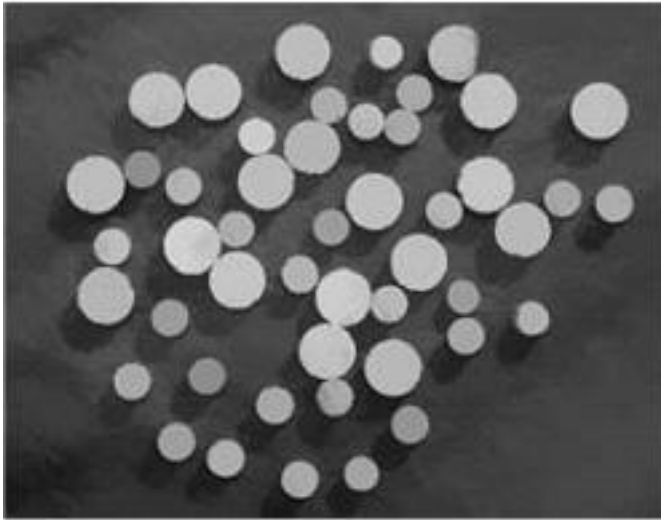


Figura 3 - objetos.bmp

E_3: Implementar o programa abaixo e explicar em cada linha o que foi realizado e o porque, para gerar a Transformada Watershed que segmenta a imagem da Figura 3.

```
f = imread('objetos1.bmp');  
g = im2bw(f, graythresh(f));  
gc = ~g  
D = bwdist(gc);  
L = watershed(-D);  
w = L == 0;  
g2 = g / w;  
subplot(2,2,1);imshow(f)  
subplot(2,2,2);imshow(g)  
subplot(2,2,3);imshow(gc)  
subplot(2,2,4);imshow(w)  
imview(g2)
```

4) Pirâmides de Resolução.



Figura 4 - ferramentas2.bmp

E_4: Para a imagem da Figura 4, gerar a Pirâmide de Resolução para $n = 4$ através da média, mostrando os quatro arranjos gerados. Aplicar o Detector de Bordas de Canny no arranjo $n = 1$, marcando as mesmas na figura original.