

Introdução ao Processamento de Imagens
Trabalho Individual 2
Semestre: 2020-2

Questão 1:

Faça um programa que utiliza como entrada a imagem morf_test.png. O programa deve entregar uma imagem binária como saída, com o fundo branco e as letras pretas. Teste, e mostre os resultados das seguintes operações no relatório:

2.1 Binarizar a imagem diretamente

2.2 Propor um algoritmo MORFOLÓGICO EM NIVEIS DE CINZA de forma de eliminar (ou reduzir) as variações no fundo, e binarizar a imagem depois. Explique no relatório o algoritmo utilizado, mostre a imagem após o processamento morfológico, e a imagem binarizada.

DICA: Para achar o limiar de binarização na parte 2.1 utilize a função graythresh no matlab ou cvThreshold no openCV.

Questão 2:

Faça um programa que realize os seguintes passos:

1. Ler a imagem “cookies.tif”
2. Binarizar a imagem de tal forma que sejam identificadas as duas “cookies” (escolher o limiar apropriado, de forma de diferenciar as cookies do fundo).
3. Eliminar por completo a “cooky” mordida, deixando pelo menos parte da cooky completa, na imagem binarizada. Usando algoritmo morfológicos EM IMAGEM BINARIA. Indicar as operações morfológica utilizadas e elementos estruturantes no relatório. No relatório mostrar também a imagem resultante.
4. Recuperar a forma inicial da “cooky” completa na imagem resultante do passo anterior. Indicar as operações morfológica utilizadas e elementos estruturantes no relatório. No relatório mostrar também a imagem resultante.
5. A partir da imagem original e utilizando a imagem resultante do passo anterior como máscara, obter uma imagem final em níveis de cinza com somente a “cooky” completa.

Questão 3:

Faça um programa que segmenta a imagem img_cells.jpg. Inicialmente, o programa deve seguir os seguintes passos:

- 1 - Binarizar a imagem, onde as células são pretas e o fundo é branco (utilizando métodos similares à questão 1)
- 2 - Se necessário, utilize uma função para preencher espaços desconectados (tipo `bwareopen` no matlab).
- 3 - Se necessário faça um preenchimento de buracos (para preencher buracos pode ser necessário usar o negativo a imagem, nesse caso faça o negativo novamente antes de passar para o passo 4).
- 5 - compute a segmentação watersheed, comente seus resultados. Pode ser melhor se antes do watersheed primeiro calcular a função de distância (função `bwdist` no Matlab, ou `distanceTransform` no openCV).

No relatório deve estar a imagem final. E imagens dos passos realizados.