



Espera-se do projeto:

A clara identificação do problema a ser resolvido. Descrição completa da solução proposta e resultados. Identificação objetiva e avaliação dos resultados.

O que deve ser entregue?

- Relatório feito pelos alunos, em 3 páginas como mínimo e 5 como máximo. Detalhes sobre o relatório, ler o arquivo "LER ANTES DE FAZER OS RELATÓRIOS" no Moodle.
- Código da solução, comentado e com descrição de uso e extensão.

Questão 1

Em sistemas automáticos de inspeção de circuitos impressos é necessário verificar a quantidade e diâmetro dos buracos para verificar se os componentes eletrônicos podem ser inseridos sem dificuldades. Faça um programa que use como entrada a imagem 'pcb.jpg', e mediante operações morfológicas conte o número de buracos e indica seu diâmetro em pixels.

DICA: uma solução possível seria utilizando as funções *imfill* e *regionprops* do Matlab (ou equivalente em OpenCV).

No relatório desta parte deve estar:

1.1 Mostrar a imagem original, e a imagem depois de cada um dos operadores morfológicos utilizados.

1.2 O resultado do programa indicando a quantidade de buracos e diâmetro.

Questão 2

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

2.1 Tentar criar uma imagem que seja somente o fundo (mediante operações morfológicas), e subtrair essa imagem da original.

2.2 Aplique filtros prévios para tentar melhorar o resultado de 2.1 e 2.2

2.3 No melhor resultado obtido aplique operações morfológicas binárias (tipo abertura, fechamento) para tentar evitar símbolos desconectados ou ruídos.

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

Questão 3

Faça um programa que segmente 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 a questão 2.1)

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).

4 - calcule a função de distancia (função `bwdist` no Matlab, ou `distanceTransform` no openCV).

5 - compute a segmentação watersheed, comente seus resultados.

No relatório deve estar presentes as imagens depois de cada um dos passos.