

Lab.02 –Segmentação de imagens por crescimento de regiões

1. Introdução

Neste experimento, os alunos devem implementar uma rotina para extrair o pano de fundo e as regiões sobressalentes serão consideradas “candidatas a embrião”, uma vez que as imagens correspondem a lâminas contendo embriões.

2. Descrição do programa

O programa em questão deverá abrir uma imagem em escala de cinza de entrada e serão realizadas cinco (05) inicializações distintas, a partir de diferentes regiões de interesse (ROIs) tanto pela localização e de tamanho. Deverão ser registradas as seguintes informações da ROI: forma [quadrada, retangular, elipsoidal, circular...], área [em pixels], histograma, informações sobre a intensidade [min, max, média, mediana, variância, moda].

Após chamar o *plugin* no ImageJ, deverá ser aberta uma janela de diálogo do tipo *WaitForUserDialog* para que o usuário selecione uma ROI correspondente ao pano de fundo, isto é, uma região que não contenha embriões. Espera-se que quanto maior for a ROI, melhores serão as estimativas das estatísticas, o que tende a melhorar o desempenho de segmentação.

Deverá ser construído um *PlugInFilter* denominado *Region_Growing_Grupo_XX.java*, onde XX é o número do grupo e ter a seguinte estrutura:

a. Classe *Region_Growing_Grupo_XX*:

i. Deve implementar a interface *PlugInFilter*;

Métodos obrigatórios:

ii. `public int setup` - com a função de checar o tipo da imagem, iniciar os atributos da classe e só funcionar para imagens *8 bit grayscale*;

- iii. `public void run` - chamará os métodos de segmentação da imagem que serão criados em outra classe, denominada *RegionGrowing*;
- b. Classe *RegionGrowing*:

Métodos obrigatórios:

- i. `public Roi drawROI()` - Abre uma janela do tipo *WaitForUserDialog* e pede para o usuário selecionar uma ROI na imagem.
 - ii. `public RoiStatistics calculateROIStatistics(Roi r)` - Recebe a ROI gerada no método anterior e chama o método da classe *RoiStatistics* que calcula as estatísticas solicitadas da ROI, as quais devem ser atributos de um objeto, a ser descrita a seguir e criada por vocês;
 - iii. `public ColorProcessor createBackgroundSegmentation(ImageProcessor ip)` - Recebe o *ImageProcessor* da imagem original e faz-se a varredura nela para verificar quais pixels pertencem ao plano de fundo segundo o critério estabelecido nesse laboratório. Deve retornar um objeto *ColorProcessor* contendo um array de pixels de duas cores distintas: BRANCA para o background e PRETA para os candidatos a embriões;
- c. Classe *RoiStatistics*

Atributos:

- a. Todas as estatísticas da ROI e dados da mesma como, por exemplo, a posição do vértice mais à esquerda e superior para uma ROI retangular junto com a largura e a altura da mesma. É necessário ter dados que permitam reconstruir a ROI a posteriori.
- b. Todos os atributos devem ser **privados** e devem ter **getters** e **setters** correspondentes;

Métodos:

- c. `public void calculateROIStatistics(Roi r)` - Calcula as estatísticas descritas no início do texto. Pode chamar métodos específicos para calcular cada estatística individualmente. Como elas serão atributos dessa classe, pense em como realizar

essa ação de forma ótima para não precisar gerar varreduras desnecessárias na imagem;

A imagem binária deve ser gerada e mostrada ao final da execução. Após isso, deve-se chamar o *pluginFilter* responsável pela rotulação (*labeling*) dos candidatos a embrião. Para tal, os alunos podem utilizar o plugin desenvolvido pelo professor na aula do dia 04/10/19 e disponibilizado no github ou criar a sua própria implementação.

Será necessário extrair as seguintes informações dos componentes conexos [candidatos a embrião]:

- Posição do centróide, área, perímetro, circularidade, dados da distribuição de intensidade [min, max, média, mediana, desvio padrão, moda] e uma variável booleana [*True or False*] que indica se o candidato a embrião pertence à borda da imagem;
- Vocês devem propor um critério baseado nos atributos dos candidatos que retire eventuais *falsos positivos*, isto é, regiões que não são embriões mas que foram selecionadas pelo método de segmentação e rotuladas como candidatas na imagem de saída;
- Devem ser plotadas duas imagens coloridas adicionais: a primeira com todos os componentes rotulados antes da filtragem baseada no critério do item anterior e outra removendo os componentes que atendem ao critério de exclusão e os pertencentes à borda.
- Todas as estatísticas -- da ROI e dos embriões -- devem ser mostradas numa tabela de resultados [*ResultsTable*] e exportadas num arquivo *.csv* [valores separados por vírgulas], o qual pode ser gerado a partir do um objeto *ResultsTable*;

Para a segmentação do pano de fundo utilize os seguintes critérios:

- Vizinhança espacial: Conectividade-8
- Critério de intensidade: $|I(x,y) - \bar{I}_{ROI}| < 3 \cdot \sigma_{ROI}$

Vocês devem compilar os resultados para as cinco inicializações distintas e comparar o desempenho do método com base nas diferentes inicializações e segundo o mesmo critério de exclusão. Esses itens devem ser discutidos na apresentação, que deve trazer esses dados e fomentar a discussão.

3. Avaliação

Os alunos deverão fazer uma apresentação no formato de um relatório simplificado do experimento contendo os seguintes itens:

- a) Breve descrição do problema;
- b) Descrição do algoritmo implementado;
- c) Problemas encontrados durante a execução e limitações do algoritmo;
- d) Discussão
- e) Conclusões
- f) Referências Bibliográficas
- g) Anexos: Código-Fonte Comentado

É permitido que os grupos tenham interação entre si contudo, não será permitido que dois ou mais grupos entreguem programas iguais ou que implementem o mesmo algoritmo.

Os grupos deverão ser compostos por dois alunos.

Grupos:

01 - Gabriel Del e Emanuelle; 02 - Welainy e Ítalo; 03 - Luana, Erick e Bianca; 04- Gabriel Miranda e Pedro; 05- Vitória e Raquel