

MAC5768 (MAC0417) Visão e Processamento de Imagens
Atividade no. 2

Prof. Dr. Paulo Miranda

¹Instituto de Matemática e Estatística (IME)
Universidade de São Paulo (USP)

1. Problema 01: Filtragem no domínio da frequência

Dada uma imagem com ruído periódico (arquivo “leopard_noise.png” da Figura 1), faça um programa em Python, utilizando os seus conhecimentos sobre filtragem no domínio da frequência, para remover o ruído presente na imagem.



Figura 1: Imagem com ruído periódico do arquivo “leopard_noise.png”.

Dicas:

- Estude o capítulo 4 do livro [1].
- Estude o tutorial do NumPy em:
<https://docs.scipy.org/doc/numpy/user/quickstart.html>
- Use o OpenCV-Python para a leitura da imagem.
- Utilize as implementações da transformada de Fourier e de sua transformada inversa presentes na biblioteca do NumPy ou OpenCV.

2. Problema 02:

Dada uma imagem corrompida por um ruído impulsivo, resultando em múltiplas pequenas regiões pretas (arquivo “fruit.png” da Figura 2), faça um programa em Python para eliminar os pequenos buracos da imagem, utilizando a árvore de componentes.

Dicas:

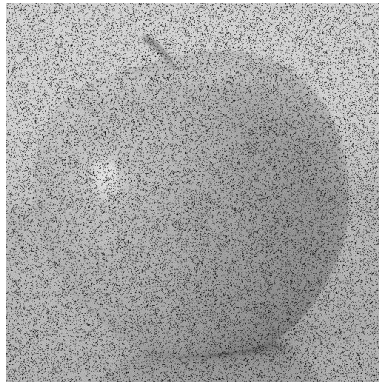


Figura 2: Imagem corrompida do arquivo “fruit.png”.

- Use o OpenCV-Python para a leitura da imagem.
- Utilize o toolbox do `siamxt` [2] para preencher as bacias da imagem de acordo com o atributo de área das bacias¹.

3. Problema 03:

Dada uma imagem de tomografia computadorizada do joelho (arquivo “knee.pgm” da Figura 3), faça um programa em Python que:

1. Calcula o Gradiente Morfológico da imagem, utilizando como elemento estruturante um disco de raio unitário.
2. Aplica um filtro de extinção no gradiente, utilizando o toolbox do `siamxt` [2] com vizinhança-8, de modo a conservar apenas as oito bacias mais relevantes de acordo com o atributo de volume.
3. Calcula os mínimos regionais da imagem filtrada do item anterior.
4. Rotula com números crescentes os diferentes mínimos regionais, utilizando a função “`cv2.connectedComponents`” do OpenCV-Python.
5. Calcula a Transformada de Watershed do OpenCV-Python, usando os mínimos regionais rotulados como marcadores, de modo a dividir a imagem em oito regiões.

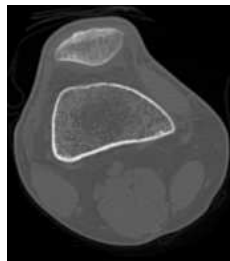


Figura 3: Imagem de tomografia computadorizada do joelho do arquivo “knee.pgm”.

¹<https://github.com/rmsouza01/siamxt>

4. Problema 04:

Faça um programa em Python, para selecionar da melhor forma possível os caracteres alfanuméricos em preto da imagem “revista_fapesp.png” (Figura 4).

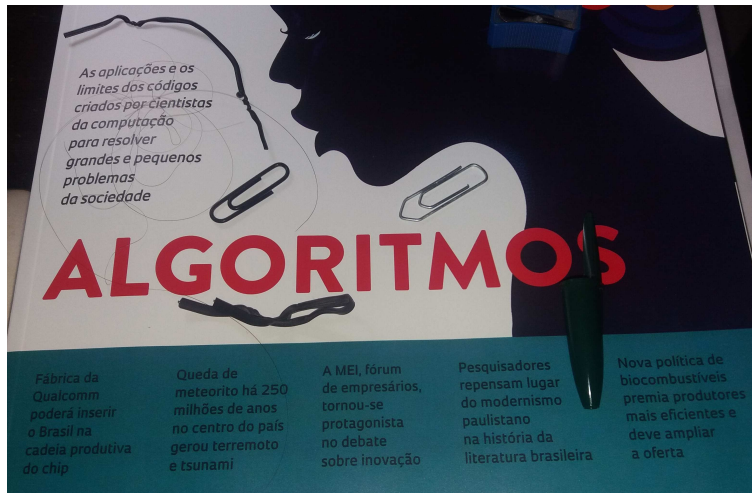


Figura 4: Capa de revista do arquivo “revista_fapesp.png”.

Dicas:

- Use o OpenCV-Python para a leitura da imagem.
- Utilize filtros morfológicos para remover os fios de cabelo da imagem.
- Teste o toolbox do `siamxt` [2] para projetar um filtro dos atributos da árvore de componentes para selecionar as letras.

5. Informações de entrega:

Vocês devem entregar, além dos códigos dos programas em Python, um pequeno relatório descrevendo o trabalho realizado, com ilustrações de cada passo. Os arquivos devem ser entregues na forma de um arquivo .zip no PACA.

Referências

- [1] R.C. Gonzalez and R.E. Woods. *Processamento Digital de Imagens, terceira edição*. Pearson, Prentice Hall, São Paulo, SP, 2010.
- [2] Roberto Souza, Letícia Rittner, Rubens Machado, and Roberto Lotufo. `iamxt`: Max-tree toolbox for image processing and analysis. *SoftwareX*, 6:81 – 84, 2017.