UNIVERSIDADE FEDERAL DE VIÇOSA
Campus de Rio Paranaíba
Sistemas de Informação –
SIN 392 – Introdução ao Processamento Digital de Imagens – 2018-1
Prof. João Fernando Mari

Trabalho – Processamento de Imagens – 10 + 10 pontos

Implementar e testar os principais métodos de processamento de imagens relacionados aos tópicos estudados na disciplina.

- Fundamentos de imagens digitais
  - Operações aritméticas sobre imagens (aritm.py)
    - **`aritm.py <img_entrada_1> <img_entrada_2> <img_saida>`**
    - Obs.: Escolher uma operação aritmética (soma, subtração, multiplicação ou divisão).
  - Operações lógicas sobre imagens (logic.py)
    - **`logic.py <img_entrada_1> <img_entrada_2> <img_saida>`**
    - Obs.: Escolher uma operação lógica (AND ou OR).
- Transformações de intensidade
  - Negativo de uma imagem (neg.py)
    - **`neg.py <img_entrada> <img_saida>`**
  - Transformação gama (gama.py)
    - **`gama.py <img_entrada> <img_saida> <gama>`**
    - <gama> é um número float, maior do que 0.
  - Alargamento de contraste (cont.py)
    - **`cont.py <img_entrada> <img_saida>`**
  - Construção de histogramas (hist.py)
    - **`hist.py <img_entrada>`**
    - Não gera imagem de saída.
  - Equalização de histograma (eq.py)
    - **`eq.py <img_entrada> <img_saida>`**
- Filtragem espacial para suavização
  - Filtro de média (media.py)
    - **`media.py <img_entrada> <img_saida> <mask_size>`**
    - <mask_size> é um número inteiro. Exemplo: Se mask_size=3 então a máscara possui tamanho 3x3.
  - Filtro gaussiano (gaus.py)
    - **`gauss.py <img_entrada> <img_saida> <stdev>`**
    - <stdev> é o desvio padrão da Gaussiana.
  - Filtro da mediana (med.py)
    - **`med.py <img_entrada> <img_saida> <mask_size>`**
    - <mask_size> é um número inteiro. Exemplo: Se mask_size=3 então a máscara possui tamanho 3x3.
  - Filtro de máximo e mínimo (maxmin.py)
    - **`maxmin.py <img_entrada> <img_saida_min> <imgsaida_max> <mask_size>`**
    - <mask_size> é um número inteiro. Exemplo: Se mask_size=3 então a máscara possui tamanho 3x3.
    - Gera duas imagens de saída.
- Filtragem espacial para aguçamento
  - Laplaciano (lap.py)

        - **lap.py <img_entrada> <img_saida>**
        - Utilizar máscara laplaciana com centro -4.
    - Máscara de nitidez e *high-boost* (nit.py)
        - **nit.py <img_entrada> <img_saida>**
    - Gradiente (grad.py)
        - **grad.py <img_entrada> <img_saida>**
        - Utilizar o gradiente de Sobel.
- Segmentação – Detecção de bordas
    - Efeitos da suavização da detecção de bordas (bordas_s.py)
        - **bordas_s.py <img_entrada> <img_saida> <mask_size>**
        - Utilizar o filtro da média.
        - Utilizar o gradiente de Sobel.
    - Efeitos da suavização e limiarização na detecção de bordas (bordas_l.py)
        - **bordas_l.py <img_entrada> <img_saida> <mask_size>**
        - Obs.: Utilizar o filtro da média.
        - Obs.: Utilizar o gradiente de Sobel.
        - Obs.: Utilizar o limiar de 20% da maior intensidade da imagem.
- Segmentação – Limiarização
    - Limiarização iterativa (lim_it.py)
        - **lim_it.py <img_entrada> <img_saida> <T_ini>**
        - <T_ini> é um número tipo float. É o chute inicial do threshold.
        - Considerar delta-T mínimo como 0.001
    - Limiarização utilizando o método de Otsu (otsu.py)
        - **otsu.py <img_entrada> <img_saida>**
    - Efeito da suavização na limiarização (lim_s.py)
        - **lim_s.py <img_entrada> <img_saida> <maks_size>**
        - Utilizar o filtro da média.
        - Utilizar o método de Otsu.

Cada tópico deve ser implementado na forma de um programa, utilizando a linguagem Python e as bibliotecas de computação cientifica do Python, como NumPy, SciPy, Matplotlib e Scikit-image.

Sugestão: Instale o Anaconda Python: https://www.anaconda.com/download/
Escolha a versão para Python 3.6.

Os programas devem ser executados via linha de comando e os nomes das imagens de entrada e de saída devem ser passados como argumento na chamada do programa, assim como eventuais parâmetros. Nenhum tipo de interação com o usuário deve ocorrer durante a execução dos programas. Por exemplo:

Filtro gaussiano:
```
$ python gauss.py  im01.png img01_out.png 3.0
```

Em que 3.0 é o valor do sigma para o filtro gaussiano.

- Podem utilizar as imagens disponíveis no site do livro texto da disciplina para os testes.
    - http://tiny.cc/v5fmuy

- **T-1** : Entregar o trabalho no formato de relatório, contendo:
    - Um único arquivo no formato PDF.
    - Para cada programa, uma breve descrição do método/algoritmo utilizado.
    - Para cada programa, detalhar como o método/algoritmo foi utilizado.

    - Figuras com as imagens antes, durante (se possível) e após o processamento.
        - Obter a imagem utilizando <ALT + PRINT SCREEN> das figuras plotadas.
        - Plotar todas as figuras de uma vez no final de cada programa. Ou seja, incluir um único comando 'plt.show()' no final de cada programa.
    - Um pequeno texto analisando os resultados de cada programa.
    - Utilizar o modelo de artigos da SBC.
        - http://tiny.cc/y6fmuy
    - Data de entrega: 06 de julho de 2018 (PVAnet).
    - 10 pontos.

- **T-2** : Códigos fonte em um arquivo .RAR ou .ZIP:
    - Incluir todos os códigos fonte em uma única pasta compactada.
    - Data de entrega: 06 de julho de 2018 (PVAnet).
    - 10 pontos.

- Grupos de no máximo 4 alunos.

- Material auxiliar:
    - Slides da disciplina (parte teórica).
    - Aulas Práticas no PVAnet.
    - Scipy Lecture Notes:
        - http://tiny.cc/fagmuy

Bom trabalho!!!

Anexo A: Sequência de chamadas que serão utilizadas para um dos testes da avaliação do trabalho:

A imagem 'noisy_fingerprint.tif' pode ser encontrada no arquivo 'AulaPratica_(2018-1)', no PVAnet.

```
python aritm.py noisy_fingerprint.tif noisy_fingerprint_ARITM.tif
python logic.py noisy_fingerprint.tif noisy_fingerprint_LOGIC.tif
python neg.py noisy_fingerprint.tif noisy_fingerprint_NEG.tif
python gama.py noisy_fingerprint.tif noisy_fingerprint_GAMA.tif 0.5
python cont.py noisy_fingerprint.tif noisy_fingerprint_CONT.tif
python hist.py noisy_fingerprint.tif
python eq.py noisy_fingerprint.tif noisy_fingerprint_EQ.tif
python media.py noisy_fingerprint.tif noisy_fingerprint_MEDIA.tif 5
python gauss.py noisy_fingerprint.tif noisy_fingerprint_GAUSS.tif 15
python med.py noisy_fingerprint.tif noisy_fingerprint_MED.tif 5
python maxmin.py noisy_fingerprint.tif noisy_fingerprint_MIN.tif noisy_fingerprint_MAX.tif 3
python lap.py noisy_fingerprint.tif noisy_fingerprint_LAP.tif
python nit.py noisy_fingerprint.tif noisy_fingerprint_NIT.tif
python grad.py noisy_fingerprint.tif noisy_fingerprint_GRAD.tif
python bordas_s.py noisy_fingerprint.tif noisy_fingerprint_BORDAS_S.tif 5
python bordas_l.py noisy_fingerprint.tif noisy_fingerprint_BORDAS_L.tif 5
python lim_it.py noisy_fingerprint.tif noisy_fingerprint_LIM_IT.tif <T_ini>
python otsu.py noisy_fingerprint.tif noisy_fingerprint_OTSU.tif
python lim_s.py noisy_fingerprint.tif noisy_fingerprint_LIM_S.tif 5
```