FGA0015 – Processamento Digital de Imagens (PDI – 102458) Prof. Marcus Chaffim (*chaffim@unb.br*)

PROJETO 4 – Realce de Imagens no Domínio Espacial

Faça o download das imagens de teste *flores.tif, Fig0310(b)(washed_out_pollen_image).tif* e *Fig0354(a)(einstein_orig).tif* , disponíveis na seção "Recursos para Projetos". Para cada uma das três imagens faça o que se descreve a seguir.

1. Calcule o histograma da imagem, o histograma normalizado ("*pdf"*, a função massa de probabilidade) e a função massa de probabilidade acumulada ("*cdf"*). Multiplique a *cdf* pelo nível de cinza máximo (255 para imagens de 8 bits) para obter o que será denotada por *cdf_Img*.

2. Aplique o alargamento de contraste usando uma função sigmoide $T(r) = 1/\left[1 + e^{-a(r-m)}\right]$. Use o nível de intensidade referente ao valor **central** da *excursão* dos níveis de cinza da *cdf_Img* como o ponto de inflexão *m* da função *T*(*r*). Considere como essa *excursão*, denotada por *cdf_Img_up*(*r*), os valores de *cdf_Img* (*r*) que excluem os extremos, i.e. 0 < *cdf_Img* (*r*) < *L*-1 (sendo $L = 2^k$, para a imagem de *k* bits de profundidade). Use as funções `polyfit` e `polyval` em *cdf_Img_up* para estimar sua regressão de 1º grau. Teste ao menos três diferentes valores do parâmetro *a* (a inclinação da sigmoide), incluindo a inclinação obtida com o auxílio do procedimento de regressão sobre *cdf_Img_up*.

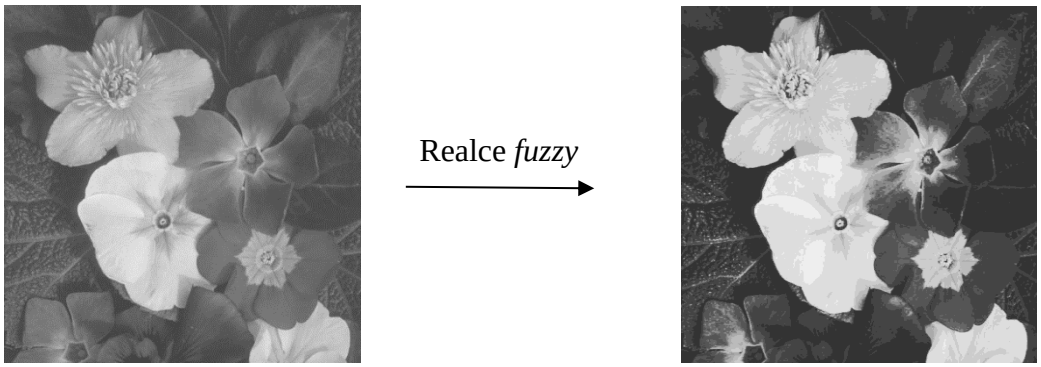
3. Para os itens 2), 4) e 6) devem ser mostrados também os histogramas (função `imhist`) antes e **depois** dos respectivos realces.

4. Implemente o algoritmo de **equalização de histograma** para imagens em tons de cinza. O algoritmo deve ser desenvolvido passo a passo. Não é permitido usar funções encapsuladas prontas de qualquer linguagem de programação como, por exemplo, `histeq` ou similares.

5. Leia a Seção *3.8 – Utilização de **técnicas fuzzy** para transformações de intensidade e filtragem espacial* do livro *Processamento de Imagens Digitais (Gonzalez e Woods) – 3ª edição*.
    a. Consulte também as REFERÊNCIAS SOBRE IMPLEMENTAÇÕES COMPUTACIONAIS ao final do documento.

6. Desenvolva um sistema que implemente o algoritmo de **realce de imagem de baixo contraste utilizando Lógica *Fuzzy*.** O sistema deve receber o valor de intensidade de cada pixel, realizar a transformação e substituir o valor original pelo valor sugerido pelo sistema de inferência *fuzzy* projetado. Os passos sugeridos são os seguintes:
    a. Crie três funções de pertinência para a variável de entrada **pixel_original**: ESCURO, CINZA, CLARO;
    b. Crie três funções de pertinência para a variável de saída de **pixel_saida**: MAIS ESCURO, CINZA, MAIS CLARO;
    c. Considera as seguintes regras: se um pixel é ESCURO, então faça-o MAIS ESCURO; se um pixel é CINZA, então faça-o CINZA; se um pixel é CLARO, faça-o MAIS CLARO;

    O aluno é livre para sugerir as funções de pertinência que acreditar serem mais convenientes, bem como propor a configuração dos parâmetros do sistema. O resultado esperado deve ser semelhante ao o mostrado a seguir:



Realize experimentos e crie um relatório no formado *.PDF de acordo com o modelo LaTEX/Word disponibilizado no ambiente moodle da disciplina, seção "Recursos para Projetos". Descreva as várias funções modeladas, os parâmetros adotados/testados e os resultados mais eficazes, assim como outros detalhes pertinentes. Anexe ao relatório o código fonte desenvolvido em GNU OCTAVE, MATLAB ou PYTHON 3. Todos os arquivos devem ser compactados para um único arquivo *.ZIP, que deverá ser submetido pelo link apropriado, disponibilizado no ambiente moodle da disciplina.

REFERÊNCIAS SOBRE IMPLEMENTAÇÕES COMPUTACIONAIS

[1] Documentação do MATLAB (site da MathWorks) sobre Lógica Fuzzy. Disponível em

a. Fuzzy Inference Process
https://www.mathworks.com/help/fuzzy/fuzzy-inference-process.html
b. Mamdani and Sugeno Fuzzy Inference Systems
https://www.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html
c. Build Fuzzy Systems at the Command Line
https://www.mathworks.com/help/fuzzy/working-from-the-command-line.html
d. Fuzzy Logic Image Processing
https://www.mathworks.com/help/fuzzy/fuzzy-logic-image-processing.html

[2] GNU Octave (para acessar o pacote de Lógica Fuzzy: `pkg load fuzzy-logic-toolkit`)

a. Octave Fuzzy Logic Toolkit - A mostly MATLAB-compatible fuzzy logic toolkit for Octave.
https://octave.sourceforge.io/fuzzy-logic-toolkit/overview.html
b. Exemplo de aplicação em linha de comando (Risco cardíaco em função de níveis de LDL e HDL):
https://github.com/utgarda/octave-fuzzy-logic-toolkit/blob/master/inst/heart_disease_demo_1.m
https://github.com/utgarda/octave-fuzzy-logic-toolkit/blob/master/inst/heart_disease_demo_2.m

[3] Python - Scikit-Fuzzy

a. Fuzzy Control Primer:
https://pythonhosted.org/scikit-fuzzy/userguide/fuzzy_control_primer.html
b. Fuzzy Control Systems - The Tipping Problem:
https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem_newapi.html
c. The Tipping Problem - The Hard Way:
https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem.html
d. Exemplo (Número de clientes estimado na loja de sorvetes de acordo com a temperatura do dia):
https://github.com/JDWarner/scikit-fuzzy/blob/scipy2013/scikit-fuzzy_demo.ipynb