

Exercício 3 – Filtragem de Imagens – Entregar até dia 07/10/2019

PESO 2

Depois de você ter trabalhado no desafio proposto pelo hospital, a vida acabou seguindo seu rumo. Em busca de novas oportunidades, você observou que havia muitos softwares de processamento de imagens, mas nenhum que reunisse uma grande quantidade de filtros já bem implementados com uma guide no software Matlab. Dessa maneira, você como um bom desenvolvedor, lembrou que já havia implementado muitos filtros, em seus vários anos de experiência. Além disso, um grande amigo seu, juntou-se a você para definir alguns passos para o desenvolvimento desse módulo.

1. Geração de Ruídos nas Imagens:

- **1°. Escolha:** Uma Imagem da pasta **Imagens_sem_Ruido.zip**.
- **2°. Acrescente Ruído Gaussiano:** O ruído possui **distribuição Gaussiana** com variância normalizada e alterando de 0.01 à 0.05 (passo de 0.01, ou seja cinco imagens). Para isso, utilize **imnoise**.
- **3°. Acrescente Ruído Salt & Pepper:** O ruído possui valores de 0.01 até 0.10 (passo de 0.02, ou seja cinco imagens no total). Para isso, utilize **imnoise**.

2. Restauração e Filtragem:

- **Agora você deve implementar os seguintes filtros:**
 - a) **Filtro Média Simples:** máscara 3x3, 7x7 e 9x9
 - b) **Filtro Mediana:** máscara 3x3, 7x7 e 9x9
 - c) **Filtro em Frequência Gaussiano:** $f_c = 3\%$, 6% e 10% da frequência máxima
 - d) **Desenvolva um filtro seu utilizando todos os conhecimentos até aqui empregados. Seu filtro deve ser melhor que os resultados aqui mostrados pelos filtros já implementados nessa tarefa (a avaliação deste resultado terá um peso maior que os demais na sua nota).**
 - e) Filtros com preservação de bordas de Tomita e Tsuji, Nagao e Matsuyama e Somboonkaew et al.

Obs: Você deverá implementar todos os filtros que usar, ou seja, é proibido o uso de filtros pré-definidos do MatLab. Você poderá utilizar (se for necessário), apenas as funções relacionadas ao cálculo da transformada de Fourier para ajudá-lo. Use convolução com FFT sempre que possível, pois o cálculo é muito mais rápido. Cuidado com erro de *wraparound*.

3. Avaliação dos Filtros:

- Aplique cada uma das métricas de qualidade já implementadas nos Matlab para realizar a avaliação sobre o conjunto de imagens geradas
 - a) **immse:**
 - b) **srn:**
 - c) **psnr:**
 - d) **ssim.**

4. Relatório e Entrega:

- **O relatório deverá conter no mínimo:**
 - a) Você deverá gerar tabelas para cada filtro, nível de ruído e avaliação de cada métrica.

- b) Deverá ser discutido todos os avanços em cada filtro, tipo de ruído e qualidade, considerando as medidas de qualidade extraídas quando comparados os resultados com a imagem inicial.
- c) Nome da entrega: **Relatorio_{Nome_Aluno}.pdf**
- **A entrega deverá ser realizada da seguinte maneira:**
 - a) Códigos (apenas a funções utilizadas para realizar a tarefa sem relação a estrutura do guide) e a função de chamada para completar os dados relativos ao exercício. Nome da entrega: **Funcoes_sem_guide_{Nome_Aluno}.zip**
 - b) Relatório Final:
Nome da entrega: **Relatorio_{Nome_Aluno}.pdf**
 - a) Planilhas e Plots: **Planilhas_Plots_{Nome_Aluno}.zip**

Exemplo de Entrega no Classroom:

Exercicio_de_Filtragem_Regina_Celia.zip

Nesse arquivo deverá conter a seguinte estrutura:

- **Funcoes_sem_guide_Regina_Celia.zip**
- **Planilhas_Plots_Regina_Celia zip**
- **Relatorio_Regina_Celia.pdf**

Obs1. Trabalhe com todas as imagens em *double*.

Obs2. Realize comentários em todo o código. Deixe esteticamente arrumado e “limpo” todo o código.

Obs3. Verifique se seu código está sendo reproduzido de maneira correta.

Obs4. Faça um relatório claro! Gere saídas que possam ser bem visíveis e com títulos. Salve todas as saídas.

Obs5. Aproveite do Matlab, gere as tabelas de modo bem estruturado e de fácil visualização.