

Instruções

1. Esta avaliação deve ser feita de duas à 4 pessoas.
2. Data de entrega: **19/09/2024 até 18:59**. Trabalhos não podem ser entregues em atraso.
3. Esta avaliação tem por objetivo consolidar o aprendizado sobre reconhecimento de imagem e processamento digital de imagens.
4. A implementação deverá ser desenvolvida em Python. O uso de funções de filtragem ou operações pontuais prontas de bibliotecas não será aceito. Todos os códigos implementados estão passíveis de ser explicados, independente da origem.
5. O sistema deve ser entregue funcionando corretamente.
6. Deve ser apresentado uma a solução:
 - Identificação do autor e do trabalho.
 - Enunciado do projeto
 - Explicação e contexto da aplicação para compreensão do problema tratado pela solução
 - Desenvolvimento
 - Códigos importantes da implementação.
 - Resultados obtidos com a implementação
 - Análise e discussão sobre os resultados finais (inclusive de problemas)
7. Deve ser disponibilizado os códigos da implementação juntamente com a relatório (salvo o caso da disponibilidade em repositório aberto do aluno, que deve ser fornecido o link). O repositório deve estar aberto do momento da entrega em diante, sendo que o professor não responsabiliza caso o projeto não esteja disponível para consulta no momento da correção, sendo do(s) aluno(s) essa responsabilidade de manter disponível.
8. A não entrega e apresentação do trabalho acarretará nota zero para todos os integrantes. Todos os integrantes deverão ter conhecimento de todas as partes da implementação. O integrante que não apresentar, terá sua nota zerada no quesito apresentação. Todos os alunos deverão responder os questionamentos na apresentação, porém fica a cargo do professor escolher um representante que explicar pelo grupo.
9. Postar Slide+Relatório+Respositório/Colab/Código no material da disciplina.
10. Cada individual/dupla/trio terá de até 10 minutos para apresentar seu desenvolvimento. O processamento estará passível de ser testada com qualquer imagem no momento da apresentação.

Descrição do projeto a ser desenvolvido

Como forma a combinar os conceitos de operações pontuais, filtragem espacial e métricas, vocês terão que:

Projeto 1:

Primeira Parte:

Implementar o Filtro de Difusão Anisotrópica (FDA) em Python, usando como referência o pseudocódigo fornecido (equações). O algoritmo do Filtro de Difusão Anisotrópica (FDA) permite realizar um esmaecimento da imagem para retirar o ruído presente em uma imagem de entrada, porém ele irá auxiliar na manutenção das bordas presentes nas estruturas da cena. Diferente de filtros que possuem os seus parâmetros de filtragem estáticos, esse algoritmo possui parâmetros dinâmicos, ou seja, são gerados a cada iteração de janelamento na imagem, adaptando-se para o contexto de janela observado

na imagem. Essa característica nos permite incluí-lo dentro do contexto de filtros dinâmicos (passo anterior a modelos de aprendizado de máquina). O filtro, além de uma complexidade maior que os filtros mais tradicionais, requer parâmetros de configuração: τ , que varia de 0 à 0,5, número de iterações e fator λ , que é um número que melhor se adequa ao contexto de aplicação, porém > 0 .

- <https://ieeexplore.ieee.org/document/8351279>
- <https://revistas.unifacs.br/index.php/rsc/article/view/5112/3286>
- <https://www.mia.uni-saarland.de/weickert/Papers/book.pdf>

As funções principais para a implementação são referentes a seguintes equações:

$$M = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & P & p_6 \\ p_7 & p_8 & p_9 \end{bmatrix}$$

$$W = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & wC & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

$$W = \begin{bmatrix} (w_1(\lambda, \tau, x-1, y+1)) & (w_2(\lambda, \tau, x, y+1)) & (w_3(\lambda, \tau, x+1, y+1)) \\ (w_4(\lambda, \tau, x-1, y)) & (wC) & (w_6(\lambda, \tau, x+1, y)) \\ (w_7(\lambda, \tau, x-1, y-1)) & (w_8(\lambda, \tau, x, y-1)) & (w_9(\lambda, \tau, x+1, y-1)) \end{bmatrix}$$

$$w(\lambda, \tau, \alpha, \beta)_i = \frac{1 - e^{-8 \cdot \tau \cdot e^{-\left(\frac{\sqrt[5]{|\alpha - \beta|}}{\lambda}\right)}}}{8}$$

$$wC = 1 - \sum_{i=1}^8 w_i$$

$$P = M * W$$

Com essas equações e para um janelamento de 3x3, é necessário calcular o valor ω com base nos valores dos 8 pixels vizinhos ($\omega_1, \omega_2, \omega_3, \omega_4, \omega_6, \omega_7, \omega_8, \omega_9$) em relação ao pixel central, sendo α o valor do pixel central e β o valor do pixel vizinho. Esses 8 valores são então somados e, posteriormente, subtraídos de 1 para gerar o ωC . Após a geração dos parâmetros, é necessário multiplicar os ω_i e ωC com os valores dos pixels da janela. Esse processo se repete por toda a imagem. Após a convolução finalizar em toda a imagem, é obtida a imagem de saída da primeira iteração e iniciado a próxima iteração.

Após implementar o filtro, avaliar o desempenho dele com:

- Imagem de entrada com ruído (pode usar biblioteca)
- Mensurar com métrica apropriada e com a imagem original (sem ruído): PSNR e MSE.
- Comparar com o filtro gaussiano 3x3.

Segunda Parte:

Com os dois filtros de esmaecimento desenvolvidos, você deverá aplicar técnicas de aguçamento:

- A. Detecção de borda com Sobel e fazer uma análise visual de qual filtragem manteve melhor as bordas: FDA ou Gaussiano
- B. Aplicar Unsharp Masking e Highboost Filtering nas imagens após a filtragem de esmaecimento e refazer o processo de em A.

Pontuação extra – máximo 1,0 ponto:

Uma técnica que permite ganho de desempenho no processamento é utilizando LookUp Table (LUT) – no português tabelas de consulta. Essa técnica consiste em armazenar valores previamente calculados em uma tabela (ou vetor de dados) que contém, geralmente, no máximo 512 valores. No FDA, a função abaixo

$$w(\lambda, \tau, \alpha, \beta)_i = \frac{1 - e^{-8 \cdot \tau \cdot e^{-\left(\frac{\sqrt[5]{|\alpha - \beta|}}{\lambda}\right)}}}{8}$$

Pode ser substituída por uma LUT, dado que as variáveis alfa (α) e beta (β) podem no máximo assumir 256 valores (que podem ser usados para endereçar uma posição de vetor) e o valor de lambda (λ) pode ser definido por um valor padrão (ex: 15). Com isso, implemente a técnica de LUT para permitir ganho de desempenho do algoritmo FDA e avaliar (temporalmente) a versão modificada com a versão original para medir o ganho de desempenho.