

Atividade #05

Vale nota, individual ou em dupla, observar prazo e instruções de entrega no moodle

5.1) MSSIM image quality metric (IQM)

Conforme mencionado durante a aula e nas 'Notas de aula e prática MATLAB', uma das métricas objetivas (IQM) do tipo *full-reference* mais adotadas para a medida de qualidade de imagem é o índice *structural similarity (SSIM)* [WBo]. Está disponível no MATLAB a partir da versão 2014a, função `ssim` da IPT (<http://www.mathworks.com/help/images/ref/ssim.html>).

Não é necessário ler o paper original ou acessar outro conteúdo além do visto em aula para realizar esta atividade, mas se desejar, o artigo original está disponível em <https://ece.uwaterloo.ca/~z70wang/publications/ssim.pdf>. Você vai perceber que não é nenhum bicho de sete cabeças e que você tem conhecimento para digerir o paper de boas 😊 Também é possível saber mais sobre o método a partir de outras fontes. Neste caso, fique à vontade para ver os recursos sugeridos abaixo ou googlar.

- Jim Nilsson and Tomas Akenine-Möller, **Understanding SSIM**, *arXiv*, 2006.13846, eess.IV, 2020. Disponível em <https://arxiv.org/abs/2006.13846> 80+ citações segundo o Google Scholar
- Z. Wang and A. C. Bovik, **Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures**, in *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98-117, Jan. 2009, doi: 10.1109/MSP.2008.930649. Disponível em <https://ieeexplore.ieee.org/abstract/document/4775883> 3490+ citações segundo o Google Scholar
- Alan C. Bovik, *The Essential Guide to Image Processing*, Academic Press, 2009. 2nd ed, 2009. Capítulo 21, itens 21.3.1 e 21.3.2. Disponível na EBSCO da UTFPR: para fazer login, entrar em <http://portal.utfpr.edu.br/biblioteca/bibliotec>, clicar em 'Bases de Dados' e depois em 'Coleção Acadêmica de E-books (EBSCO)'
- Wikipedia, **Structural similarity** Disponível em https://en.wikipedia.org/wiki/Structural_similarity

Nesta atividade, você deve elaborar um script MATLAB para reproduzir a parte da 'Demonstration' do método em <https://ece.uwaterloo.ca/~z70wang/research/ssim>, em que são calculados o MSE e o índice SSIM para as seis imagens do Einstein. Estas seis imagens necessárias para a atividade estão disponíveis no arquivo *IF69D_ATV_05_arquivos.zip* (einstein.gif é a original).

Em <https://ece.uwaterloo.ca/~z70wang/research/ssim/#usage> há o código do índice SSIM, que foi usado para a 'Demonstration' mostrada no site. Se preferir, é possível também usar a função `ssim` do MATLAB com os parâmetros default. Por exemplo: `qs = ssim(Ia, Ib);`. Para obter o MSE (mean squared error), a função do MATLAB é a `immse`. Por exemplo: `qe = immse(Ia, Ib);`.

Na forma de comentários no seu script, anote os valores do MSE e do índice SSIM obtidos para cada imagem e marque a resposta correta:

1. A 'Demonstration' foi reproduzida com sucesso?

- (a) Não, o código não funcionou.
- (b) Não, o MSE foi melhor que o SSIM.
- (c) Sim. Embora os resultados possam não ser numericamente idênticos, são muitíssimos parecidos.
- (d) Sim. O SSIM representa a imagem através de wavelets e captura com fidelidade perfeita a qualidade das imagens

2. Por que o índice SSIM é melhor que o MSE neste experimento?
- (a) Ele não é melhor nem pior, é só uma maneira computacionalmente mais eficiente de calcular o MSE.
 - (b) Porque o MSE apresenta valores praticamente iguais para qualidades notavelmente diferentes das imagens, enquanto o SSIM captura essas diferenças, além de apresentar valores compatíveis com a noção de qualidade que seria atribuída por uma pessoa (SVH).
 - (c) Porque o SSIM, conforme explicado anteriormente, utiliza wavelets para capturar com fidelidade a qualidade das imagens objetivamente.
 - (d) Porque o SSIM exige menos cálculos que MSE, além de ser mais fácil de implementar em Python.

Nome do .m: atv05_01.m

5.2) (OPCIONAL) Filtro passa-baixas Gaussiano separável

Implemente o filtro Gaussiano na versão separável “na unha”. Crie um MATLAB live script para explicar os detalhes e incluir equações.

Recursos sugeridos:

- Sobre um MATLAB live script: no nosso material da Aula 03 tem um exemplo legal de MATLAB live script. É o arquivo *a03_07.mlx*, em *IF69D_NAP_03_arquivos.zip*.
- Wilhelm **Burger** and Mark J. **Burge**, **Digital Image Processing: An Algorithmic Introduction**, 3a Ed., Springer, 2022. Tópico 4.3.3 Separability of Linear Filters, Subtópico Separable Gaussian Filters, página 99. Site do livro: <https://imagingbook.com/>
- **Stan Birchfield**, **Image Processing and Analysis**, Cengage Learning, 2018. Tópico 5.2.3 Separability, página 224. Site do autor: <https://cecas.clemson.edu/~stb/>

5.3) (OPCIONAL) Descrição formal do bilateral filter

Vimos que o kernel w de um filtro bilateral é composto pela multiplicação ponto-a-ponto de dois kernels, denominados *spatial kernel* s (também chamado de *domain kernel* d) e *range kernel* r . s é uma função Gaussiana 2D convencional com σ_s . r é uma função (Gaussiana com σ_r) que, ao invés de considerar a distância do centro para as bordas do kernel, considera a similaridade (subtração) entre o nível de cinza do pixel que está sendo filtrado (pixel central) e os demais pixels pertencentes ao kernel.

Elabore um MATLAB live script que inclua as equações para a obtenção do kernel w do bilateral filter, implemente o kernel na “unha” e teste com imagens pequenas e bem simples.

Recursos sugeridos:

- Sobre um MATLAB live script: no nosso material da Aula 03 tem um exemplo legal de MATLAB live script. É o arquivo *a03_07.mlx*, em *IF69D_NAP_03_arquivos.zip*.
- Material da Aula 05, arquivo *bilateral_filter_test_point.m*, em *IF69D_NAP_05_arquivos.zip*.
- Wilhelm **Burger** and Mark J. **Burge**, **Digital Image Processing: An Algorithmic Introduction**, 3a Ed., Springer, 2022. Tópico 17.2.4 Bilateral Filter with Gaussian Kernels, página 507. Site do livro: <https://imagingbook.com/>
- **Stan Birchfield**, **Image Processing and Analysis**, Cengage Learning, 2018. Tópico 5.5.3 Bilateral Filtering, página 249. Site do autor: <https://cecas.clemson.edu/~stb/>

5.4) (OPCIONAL) Anisotropic diffusion

“Since the pioneering work by Perona and Malik*, anisotropic diffusion has seen continued interest in the image processing community and research in this area is still strong today.” [BB3] Tópico 17.3 Anisotropic Diffusion Filters, página 517.

* Pietro **Perona** and Jitendra **Malik**, **Scale-Space and Edge Detection Using Anisotropic Diffusion**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 12, Issue 7, 1990. Disponível em <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=56205> 17500+ citações segundo o Google Scholar.

O paper* é classicção. É bem escrito e nos ensina que um edge-preserving filters pode ser usado não só para atenuar ruído, mas também para obter o chamado *scale-space* de uma imagem. Abaixo segue um excerto no qual os autores usam como exemplo a imagem de uma árvore na floresta para explicar o que é o scale-space... vale a pena ler 😊

"

[...] We now examine the adequacy of the standard scale-space paradigm for vision tasks which need "semantically meaningful" multiple scale descriptions. Surfaces in nature usually have a hierarchical organization composed of a small discrete number of levels. At the finest level, a tree is composed of leaves with an intricate structure of veins. At the next level, each leaf is replaced by a single region, and at the highest level there is a single blob corresponding to the treetop. There is a natural range of resolutions (intervals of the scale-space parameter) corresponding to each of these levels of description. Furthermore at each level of description, the regions (leaves, treetops, or forests) have well-defined boundaries. In the standard scale-space paradigm the true location of a boundary at a coarse scale is not directly available in the coarse scale image [...] The reason for this spatial distortion is quite obvious - Gaussian blurring does not "respect" the natural boundaries of objects. Suppose we have the picture of a treetop with the sky as background. The Gaussian blurring process would result in the green of the leaves getting "mixed" with the blue of the sky, long before the treetop emerges as a feature (after the leaves have been blurred together) [...]

"

Googleie para encontrar, analise e rode (testes com imagens) uma implementação do filtro anisotropic diffusion.

Recursos sugeridos:

- Daniel Lopes (2024). Anisotropic Diffusion (Perona & Malik) (<https://www.mathworks.com/matlabcentral/fileexchange/14995-anisotropic-diffusion-perona-malik>), MATLAB Central File Exchange. Retrieved October 31, 2024.

Referências

- [WBo] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, Eero P. Simoncelli, **Image quality assessment: from error visibility to structural similarity**, IEEE Transactions on Image Processing, vol. 13, no. 4, p. 600-612, 2004. Disponível em <https://ieeexplore.ieee.org/document/1284395> 47400+ citações segundo o Google Scholar
- [BB3] Wilhelm Burger and Mark J. Burge, Digital Image Processing: An Algorithmic Introduction, 3a Ed., Springer, 2022. Site do livro: <https://imagingbook.com/>