

Comparação do uso de diferentes filtros para remoção do ruído *salt-and-pepper*

Gabriel de Oliveira Pontarolo

1 Introdução

O relatório a seguir faz uma análise do uso de diferentes filtros de suavização para fazer a remoção do ruído *salt-and-pepper* de uma mesma imagem base de entrada. Para tal análise, foram utilizados os filtros da média (*blur*), Gaussiano, mediana, empilhamento e bilateral.



Exemplo do ruído aplicado no nível 0.1

2 Metodologia

Para a avaliação dos métodos, foi utilizado a medida PSNR (peak signal-to-noise ratio) em relação ao tamanho do kernel nos filtros da média, Gaussiano e mediana, a quantidade de imagens para o filtro de empilhamento e os valores de *sigma* para o filtro bilateral. Cinco níveis de ruído diferentes foram utilizados, sendo esses 0.01, 0.02, 0.05, 0.07 e 0.1 na imagem abaixo para gerar o efeito do *salt-and-pepper*.



Imagen original utilizada para comparação

3 Implementação

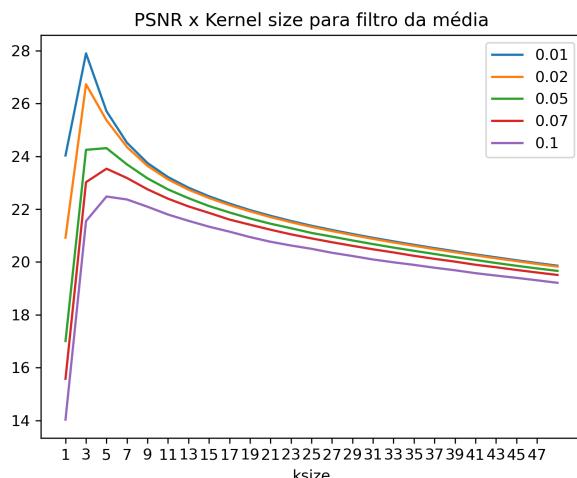
A implementação foi feita totalmente na linguagem *Python*. Foi utilizada a biblioteca *OpenCV*, que possui a implementação de grande parte dos filtros mais utilizados para processamento de imagens. Os gráficos que serão exibidos foram gerados utilizando a biblioteca *Pandas* em conjunto com *Matplotlib* a partir de arquivos *.csv* com os dados gerados. O código para gerar as tabelas, gráficos e a implementação da filtragem com o melhor filtro escolhido estão disponíveis em conjunto com este relatório.

4 Filtros

Serão exibidos a seguir os resultados obtidos a partir do uso de cada filtro.

4.1 Filtro da média (*blur*)

Para valores pequenos de ruído, é possível observar que o melhor kernel é 3x3, onde é possível obter um PSNR de 27.92. A partir de 0.07 de ruído, um kernel de 5x5 se mostra uma opção melhor com 24.71 de PSNR, comparado a 24.24 com o tamanho 3x3.



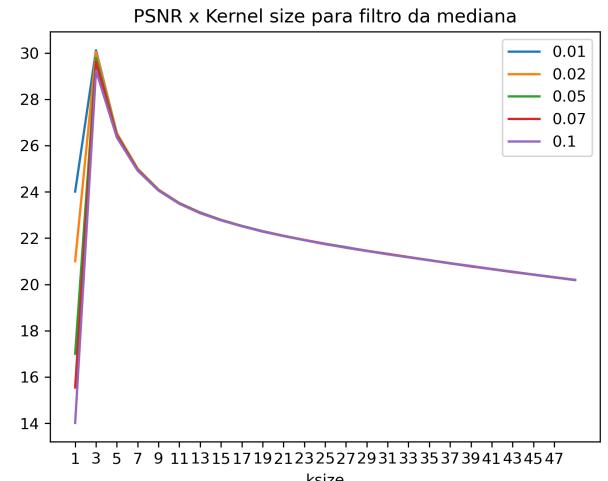
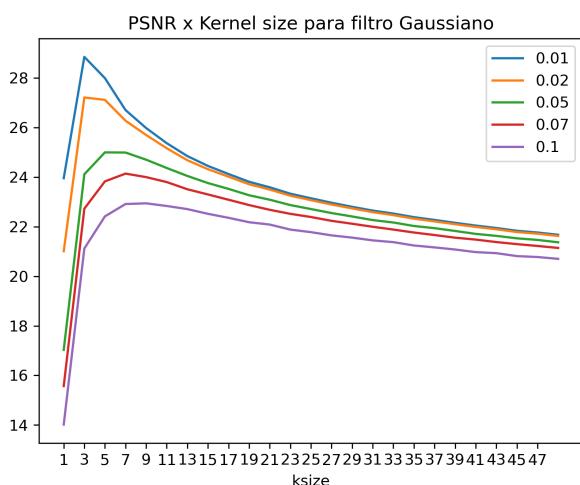
Resultado da filtragem com kernel 9x9 e ruído 0.1



Resultado da filtragem com kernel 5x5 e ruído 0.1

4.2 Filtro Gaussiano

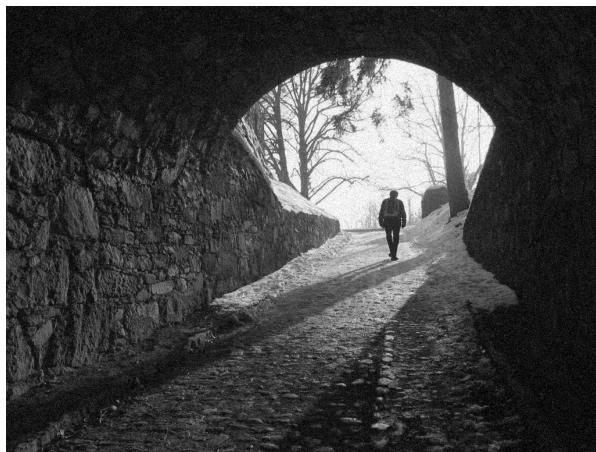
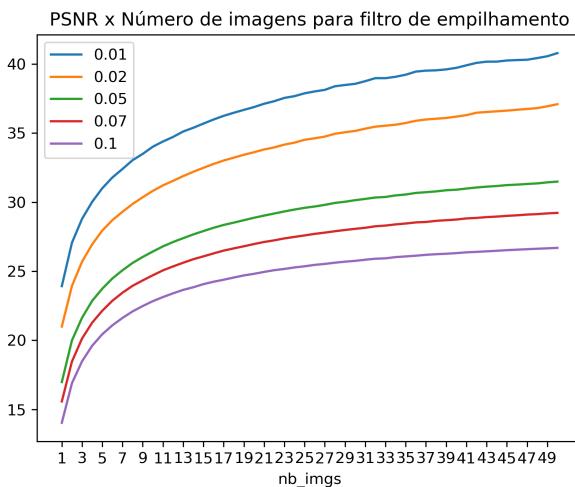
Similar ao filtro da média, é necessário um kernel maior para níveis mais altos de ruído, onde temos um pico de 28.95 de PSNR para um filtro 3x3 em 0.01 de ruído, e um pico de 22.96 para 0.1 de ruído, porém com um kernel de 9x9.



Resultado da filtragem com kernel 3x3 e ruído 0.1

4.3 Filtro de empilhamento (*stacking*)

Esse filtro possui uma propriedade particular em relação aos outros utilizados. Como ele faz uso de várias versões da mesma imagem com ruídos diferentes, quanto maior o número de imagens, maior o valor de PSNR. Porém, é possível observar que os valores crescem em uma taxa similar a uma função logarítmica e quanto maior o número de imagens, menos significativa é a diferença.

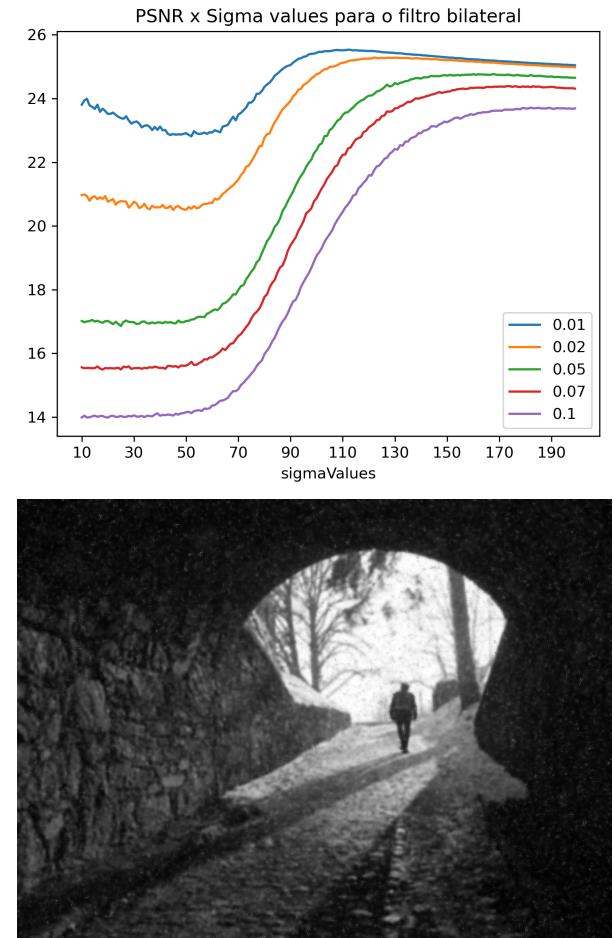


Resultado da filtragem com empilhamento de 20 imagens e ruído 0.1

4.3 Filtro bilateral^[1]

A premissa do filtro bilateral é realizar a suavização da imagem enquanto mantém as bordas realçadas. Ao invés de considerar dois pixels próximos fazendo uso apenas das dimensões espaciais da imagem, ele também leva o espaço de cores em consideração.

Para realizar os testes, foi fixado um $d = 9$ (como recomendado pela documentação do *OpenCV*^[1]) e variado os valores de *sigma* para cor e espaço. É possível notar que valores muito altos ou muito baixos de *sigma* causam uma redução no PSNR. Além disso, valores mais baixos de ruído atingem o pico antes, obtendo-se 25.53 de PSNR com erro igual 0.01 e *sigma* igual a 112, e 23.712 erro igual 0.1 e *sigma* igual a 182.



Resultado da filtragem com ambos os *sigmas* iguais a 180 e ruído 0.1

5 Conclusão

Embora os maiores valores de PSNR tenham sido obtidos com o filtro de empilhamento, o seu uso pode não ser viável em situações práticas do mundo real onde existe apenas uma versão de uma imagem com ruído. Com isso em mente, o filtro mais efetivo para remoção do ruído *salt-and-pepper* se mostrou o filtro da mediana, o qual obteve 29.19 de PSNR com o maior nível de ruído, e se mostrou

consistente com um mesmo tamanho de kernel sendo efetivo mesmo para níveis diferentes. Em contrapartida, o filtro menos efetivo foi o bilateral, com um PSNR de 23.712 com o maior nível de ruído e 25.53 com o menor.

6 Referências

[1] Filtragem Bilateral:

OpenCV Image Filtering:

https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html#ga9d7064d478c95d60003cf839430737ed

Bilateral Filtering for Gray and Color Images:

https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html