

Trabalho 1 - MO443 - Introdução ao Processamento de Imagem Digital

Aluno: Gabriel Bianchin de Oliveira RA: 230217

Abril 2019

1 Introdução

O objetivo desse trabalho é implementar os filtros propostos pela atividade e analisar os resultados obtidos.

Os filtros foram divididos em filtros em domínio espacial e domínio de frequência. Foram utilizados quatro tipos de filtros no domínio espacial, chamados de h1, h2, h3 e h4, além de uma combinação entre os filtros h3 e h4 pela equação $\sqrt{(h3)^2 + (h4)^2}$. Para o domínio de frequência, foi utilizado o filtro Gaussiano.

2 Código

O código foi implementado em Python 3.6.7, com as bibliotecas OpenCV 4.0.0 e NumPy 1.13.3.

2.1 Como Executar

Para a execução do código, deve-se executar o script `codigo.py`. O script recebe como argumento uma imagem de entrada, o filtro escolhido e o nome da imagem de saída. Um exemplo de execução é mostrado a seguir:

`python3 codigo.py imagens/baboon.png h1 baboon-h1.png`

As opções de filtros utilizados como parâmetro são:

- **h1**: aplica o filtro h1
- **h2**: aplica o filtro h2
- **h3**: aplica o filtro h3
- **h4**: aplica o filtro h4
- **h3+h4**: aplica a combinação dos filtros h3 e h4 pela equação $\sqrt{(h3)^2 + (h4)^2}$
- **gauss**: aplica o filtro Gaussiano. Durante a execução, será necessário informar via terminal o valor de sigma e o tamanho do filtro.

Caso nenhuma das opções válidas for informada, será enviado uma mensagem, via terminal, informando que não foi possível realizar a aplicação do filtro.

2.2 Entrada

A entrada de dados consiste em uma imagem e uma opção de filtro. As imagens utilizadas para testes estão na pasta imagens, que foram obtidas pelo diretório http://www.ic.unicamp.br/~helio/imagens_png

2.3 Saída

A saída de dados consiste em uma imagem resultante da aplicação do filtro escolhido na imagem informada como entrada. A imagem será salva na pasta saída.

3 Filtros

3.1 Filtro h1

A Figura 1 mostra o filtro h1. O filtro h1 é um filtro Laplaciano do Gaussiano, realizando a filtragem do ruído para a detecção de bordas. Após a aplicação do filtro, ocorre o realçamento de bordas e linhas da imagem.

$$h_1 = \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & -1 & 0 & 0 \\ \hline 0 & -1 & -2 & -1 & 0 \\ \hline -1 & -2 & 16 & -2 & -1 \\ \hline 0 & -1 & -2 & -1 & 0 \\ \hline 0 & 0 & -1 & 0 & 0 \\ \hline \end{array}$$

Figura 1: Filtro h1

3.2 Filtro h2

A Figura 2 mostra o filtro h2. O filtro h2 é um filtro passa-baixa, que calcula a média ponderada dos pixels próximos ao pixel central, dando mais peso para os pixels mais próximos do centro. Após a aplicação do filtro, ocorre a suavização e borramento da imagem. Devido ao borramento, alguns detalhes são perdidos.

$$h_2 = \frac{1}{256} \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 6 & 24 & 36 & 24 & 6 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}$$

Figura 2: Filtro h2

3.3 Filtro h3

A Figura 3 mostra o filtro h3. O filtro h3 representa o operador de Sobel para a detecção de bordas perpendiculares ao eixo y.

$$h_3 = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

Figura 3: Filtro h3

3.4 Filtro h4

A figura 4 mostra o filtro h4. O filtro h4 representa o operador de Sobel para a detecção de bordas perpendiculares ao eixo x.

$$h_4 = \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Figura 4: Filtro h4

3.5 Filtro Gaussiano

O filtro Gaussiano utilizado nesse trabalho é o passa-baixa, já que visa suavizar a imagem sem interrupções bruscas. O filtro Gaussiano utiliza o valor de sigma

(σ), que indica quanto uma imagem irá suavizar, sendo que valores maiores suavizam mais a imagem do que valores menores. O exemplo do filtro Gaussiano pode ser visto na Figura 5¹.

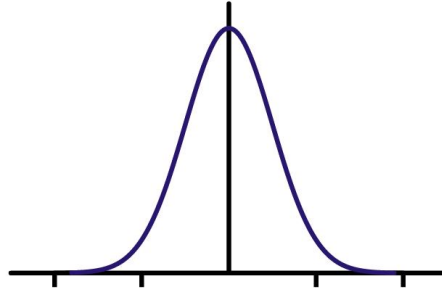


Figura 5: Filtro Gaussiano

4 Resultados

4.1 Leitura de dados e escolha de filtro

A leitura da imagem de entrada é feita pela função **cv2.imread**, da biblioteca OpenCV², que carrega a imagem em níveis de cinza e transforma em uma matriz de dimensões m e n , onde m representa as linhas e n representa as colunas da matriz. Em seguida, é realizada a aplicação do filtro escolhido. A aplicação de cada um dos filtros é mostrada a seguir. Para exemplificar a aplicação do filtro, foi utilizada a imagem baboon.png. A imagem original é mostrada na Figura 6.

4.2 Filtros no domínio espacial

Os subtópicos a seguir apresentam o resultado da filtragem no domínio espacial da Figura 6.

4.2.1 Aplicação do filtro h1

Para a aplicação do filtro h1, foi utilizada a função **cv2.filter2D**, que realiza a convolução da imagem com o filtro.

Devido h1 ser um filtro Laplaciano do Gaussiano, a imagem resultante terá as bordas realçadas e os pixels que possuírem valores próximos tenderão um novo valor próximo de zero (pixel com cor preta).

A Figura 7 mostra a aplicação do filtro h1 na imagem original baboon.png (Figura 6).

¹<https://en.wikipedia.org/wiki/Gaussianfilter>

²<https://docs.opencv.org/3.0-beta/index.html>

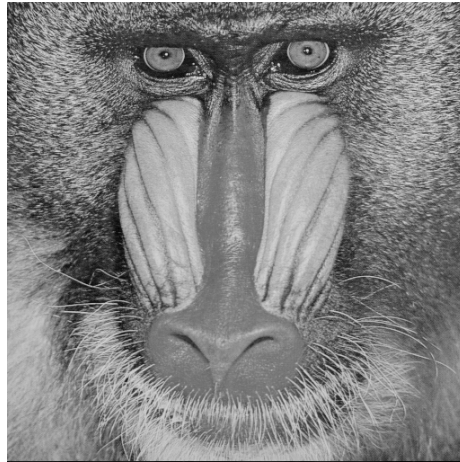


Figura 6: Imagem original

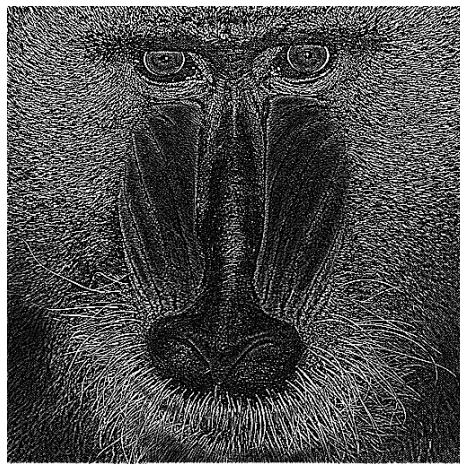


Figura 7: Resultado da aplicação do filtro h1

4.2.2 Aplicação do filtro h2

Para a aplicação do filtro h2, foi utilizada a função **cv2.filter2D**, que realiza a convolução da imagem com o filtro.

Devido h2 ser um filtro passa baixa, a imagem resultante terá uma suavização e apresentará uma borração.

A Figura 8 mostra a aplicação do filtro h2 na imagem original baboon.png (Figura 6).

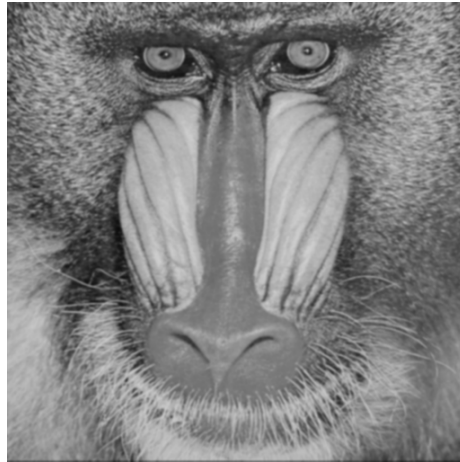


Figura 8: Resultado da aplicação do filtro h2

4.2.3 Aplicação do filtro h3

Para a aplicação do filtro h3, foi utilizada a função **cv2.filter2D**, que realiza a convolução da imagem com o filtro.

Devido h3 ser um operador de Sobel para detectar bordas no eixo y, a imagem resultante terá bordas no eixo y.

A Figura 9 mostra a aplicação do filtro h3 na imagem original baboon.png (Figura 6).

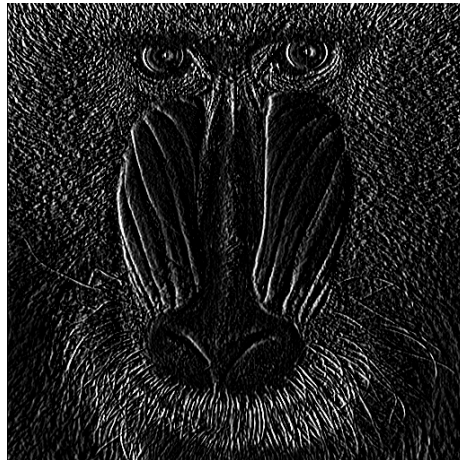


Figura 9: Resultado da aplicação do filtro h3

4.2.4 Aplicação do filtro h4

Para a aplicação do filtro h4, foi utilizada a função **cv2.filter2D**, que realiza a convolução da imagem com o filtro.

Devido h4 ser um operador de Sobel para detectar bordas no eixo x, a imagem resultante terá bordas no eixo x.

A Figura 10 mostra a aplicação do filtro h4 na imagem original baboon.png (Figura 6).

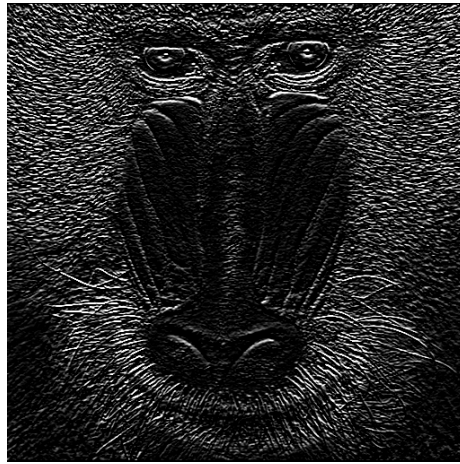


Figura 10: Resultado da aplicação do filtro h4

4.2.5 Aplicação do filtro h3 e h4

Para a aplicação conjunta dos filtros h3 e h4 foram realizadas as aplicações dos filtros h3 e h4 e em seguida a utilização da equação $\sqrt{(h3)^2 + (h4)^2}$, resultando na imagem final. A aplicação dos dois filtros pela equação resultará na magnitude da imagem.

A Figura 11 mostra a aplicação dos filtro h3 e h4 na imagem original baboon.png (Figura 6).

4.3 Filtros no domínio de frequência

O subtópico a seguir apresenta o resultado da filtragem no domínio de frequência da Figura 6.

4.3.1 Aplicação do filtro Gaussiano

Para a aplicação do filtro Gaussiano, é necessário realizar a transformada de Fourier, levando a imagem para o domínio da frequência. Para realizar a trans-

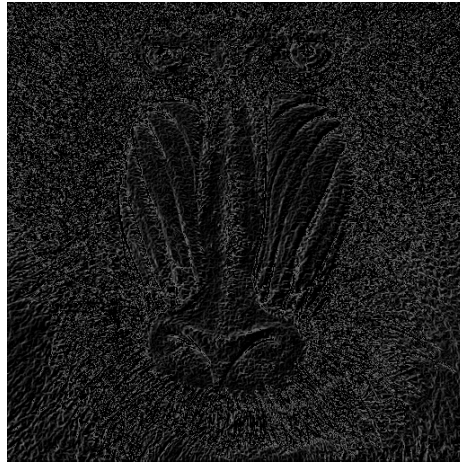


Figura 11: Resultado da aplicação dos filtros h3 e h4

formada e deixar a componente de frequência-zero no centro do espectro, foram utilizadas as funções **np.fft.fft2** e **np.fft.fftshift** da biblioteca NumPy³.

A criação do filtro Gaussiano depende do valor de sigma, que controla a suavização da imagem, e do tamanho do filtro.

Após a aplicação do filtro no domínio da frequência, é realizada a transformada inversa, para que a imagem volte para o domínio espacial.

A Figura 12 mostra a aplicação do filtro Gaussiano de tamanho 50, a Figura 13 mostra a aplicação do filtro Gaussiano de tamanho 100 e a Figura 14 mostra a aplicação do filtro Gaussiano de tamanho 200 na imagem original baboon.png (Figura 6). Ressalto que, quanto maior o sigma, mais suavização ocorre.

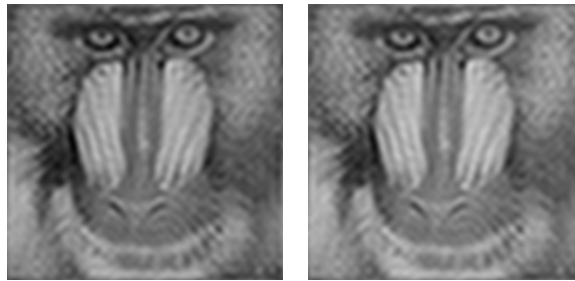


Figura 12: Filtro de tamanho 50. A imagem a esquerda possui $\sigma = 1.0$ e a imagem a direita possui o $\sigma = 2.0$

³<http://www.numpy.org/>

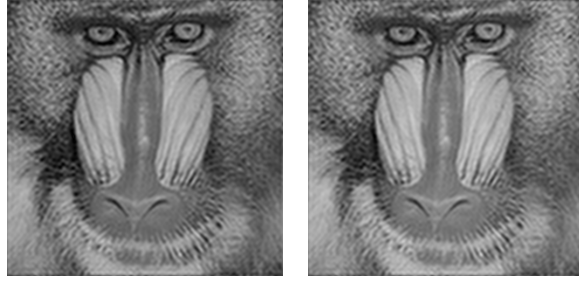


Figura 13: Filtro de tamanho 100. A imagem a esquerda possui $\sigma = 1.0$ e a imagem a direita possui o $\sigma = 2.0$

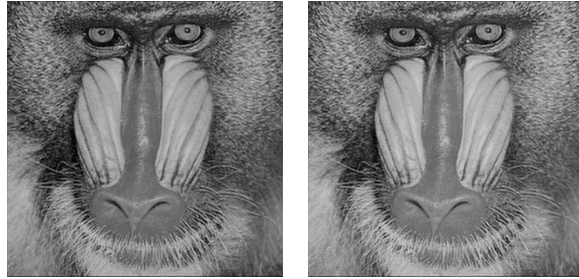


Figura 14: Filtro de tamanho 200. A imagem a esquerda possui $\sigma = 1.0$ e a imagem a direita possui o $\sigma = 2.0$