

Trabalho 1 de Introdução ao Processamento de Imagem Digital

Nome: Paulo Junio Reis Rodrigues - RA: 265674

22/10/2020

1 Introdução

O objetivo deste trabalho é realizar a implementação de alguns filtros de imagens no domínio espacial. A filtragem de uma imagem é baseada em uma aplicação local, onde cada pixel é alterado, levando em consideração o seu valor e o valor de seus vizinhos. Para que isso ocorra, é necessário utilizar um método chamado convolução, onde, junto com uma máscara, pode ser feita a filtragem em uma imagem.

Junto com este relatório, está sendo enviado o arquivo `trabalho_1_265674.zip` que possui dentro dele todos os arquivos que irão ser citados neste relatório.

2 Programa

Todos os programas foram implementados usando Python 3.8.5, com as bibliotecas Numpy 1.19.2 e OpenCV 4.4.0.42.

2.1 Como executar

O programa pode ser executado através do script `trabalho_1.py`. Quando executado, deve ser enviado como argumento uma imagem de entrada. Logo abaixo é mostrado um exemplo de execução do script:

```
python trabalho_1.py imagens/house.png
```

2.2 Entrada

A única entrada que o programa necessita é uma imagem monocromática.

2.3 Saída

As saídas do programa são 14 imagens no formato *PNG*, uma para cada filtro aplicado. Ao longo do relatório será explicado cada saída e seu respectivo filtro.

3 Parâmetros Utilizados

Todas as imagens utilizadas para execução do programa estão presentes no diretório `imagens/`, todas estão disponíveis em https://www.ic.unicamp.br/~helio/imagens_png/.

As saídas do script executado, terá o nome do trabalho(`trabalho_1`) e logo depois o filtro utilizado, exemplo: `trabalho_1_H1.png`, e todas estas saídas estarão no diretório `outputs/`.

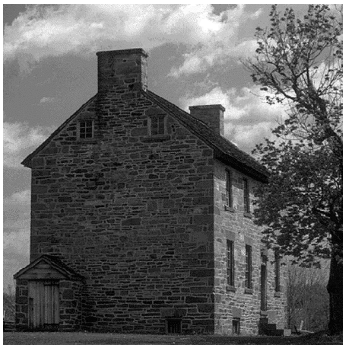
4 Solução do exercício

A solução do trabalho utiliza a função *imread* do *OpenCV* para a leitura das imagens a partir do diretório informado. Todas as imagens estão sendo lidas via escala de cinza. Para realizar a filtragem das imagens é utilizado o método *Filter2D* da biblioteca *OpenCV*, que realizar uma

correlação para filtrar a imagem, como neste trabalho os filtros informados estão relacionados a uma operação de convolução, para que o resultado seja o mesmo, foi feita uma rotação de 180 graus com todos os filtros utilizando o operador *Slice* do *Python*, para que os filtros possam ser utilizados na correlação. Além disso, o método acima utiliza várias técnicas para tratamento de bordas, para este trabalho, será utilizado o tratamento de replicação de pixels, que consiste em replicar todos os pixels das bordas horizontais e verticais, para criar uma camada maior das bordas, para que no processo de correlação não ocorra nenhum erro.

4.1 Primeiro filtro

A primeira máscara é um filtro de passa-alta, isso significa que ele está destacando alguns pontos de transições entre regiões diferentes, conhecidas como bordas. Porém esse filtro possui bastante ruído, mas é nítido que a imagem gerada está mostrando bastante bordas na cor branca.



(a) Imagem original

$$h_1 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

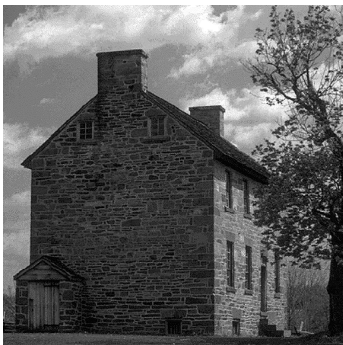
(b) Filtro 1



(c) Imagem filtrada

4.2 Segundo filtro

Já o segundo filtro é conhecido como gaussiano, que tem como objetivo suavizar a imagem. A imagem de saída está claramente mais suave, e alguns ruídos foram retirados, por exemplo o ruído do tronco da árvore direita, foi bastante retirado. Porém, algumas das bordas da imagem foram obstruídas, como por exemplo, algumas folhas das árvores ficaram grudadas. Esta suavização é gerada por uma média ponderada dos pixels vizinhos, onde o peso de cada vizinho decresce com a distância do pixel central.



(a) Imagem original

$$h_2 = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

(b) Filtro 2

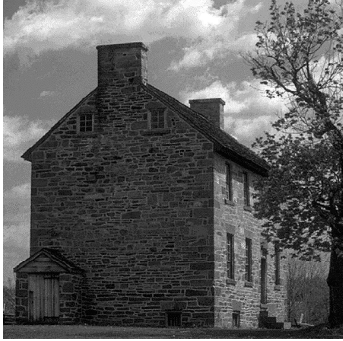


(c) Imagem filtrada

4.3 Terceiro e quarto filtro

Os próximos dois filtros são conhecidos como filtros de Sobel, o filtro 3 encontra linhas da imagem na vertical e o filtro 4 encontra linhas na horizontal. Estes filtros, possuem menos ruídos do que o primeiro filtro passa alta, e as bordas encontradas parecem um pouco mais grossas comparada com o primeiro filtro. A soma dos dois filtros, deixa a imagem resultante com linhas verticais e horizontais. Para realizar esta soma, foi utilizado o cálculo fornecido pelo professor $\sqrt{(h_3)^2 + (h_4)^2}$,

onde pára realizá-lo, primeiramente as duas imagens foram aumentadas para unidade de 16 bits considerando somente valores positivos, e logo após de todos os cálculos serem feitos, os valores foram arredondados utilizando o operador *round* da biblioteca *Numpy* e convertidos de volta para a unidade de 8 bits.



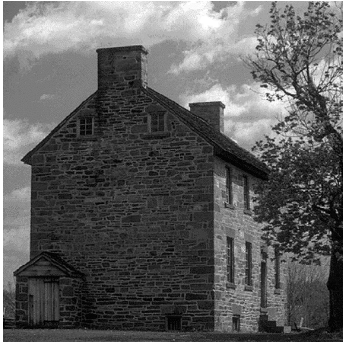
(a) Imagem original

$$h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

(b) Filtro 3



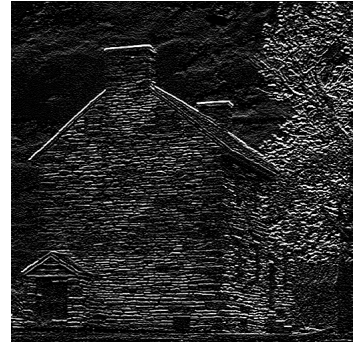
(c) Imagem filtrada



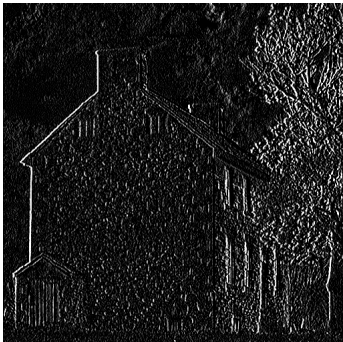
(a) Imagem original

$$h_4 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

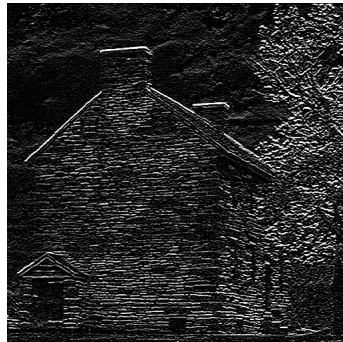
(b) Filtro 4



(c) Imagem filtrada



(a) Imagem filtrada 3



(b) Imagem filtrada 4



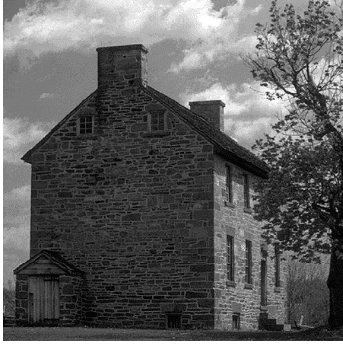
(c) Filtro somado $\sqrt{(h_3)^2 + (h_4)^2}$

4.4 Quinto filtro

A próxima máscara novamente é um filtro passa alta, porém diferente do primeiro filtro, este possui menos ruído, porém ainda possui mais ruído do que os dois filtros de Sobel. Contudo ele ainda continua destacando as transições entre regiões diferentes.

4.5 Sexto filtro

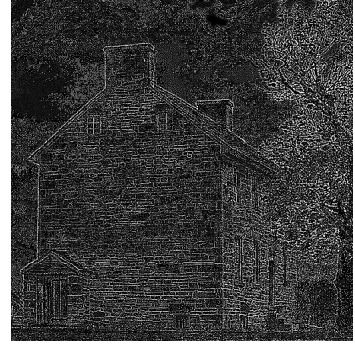
A próxima máscara é um filtro passa baixa, onde, nesse caso é feita uma média simples com todos os pixels sobrepostos pelo o filtro, assim gerando uma suavização na imagem e retirando alguns



(a) Imagem original

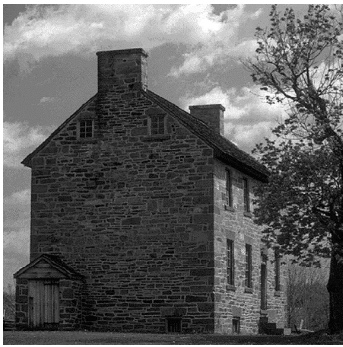
$$h_5 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

(b) Filtro 5



(c) Imagem filtrada

ruídos. Em comparação com o filtro 2 gaussiano, a suavização dessa máscara foi bem parecida com a aplicação do filtro 2.



(a) Imagem original

$$h_6 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

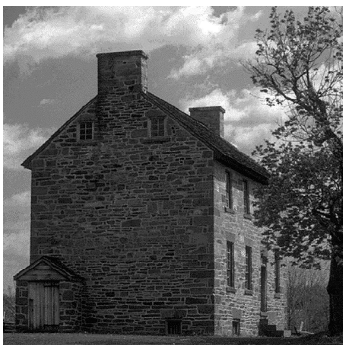
(b) Filtro 6



(c) Imagem filtrada

4.6 Sétimo filtro

O próximo filtro ao ser utilizado, está destacando retas na diagonal, mais precisamente as retas de 45 graus, uma maneira de encontrar facilmente este efeito, é olhando na textura gerada na parede da casa, onde é feito alguns rabiscos diagonais. Algumas bordas no telhado perto dos 45 graus foram bem destacadas, porém a imagem formada apresenta bastante ruído.



(a) Imagem original

$$h_7 = \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

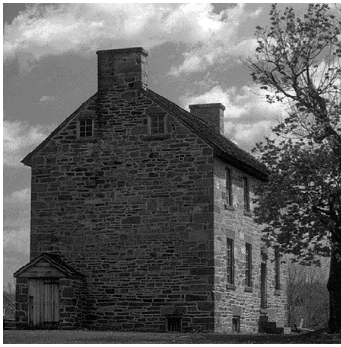
(b) Filtro 7



(c) Imagem filtrada

4.7 Oitavo filtro

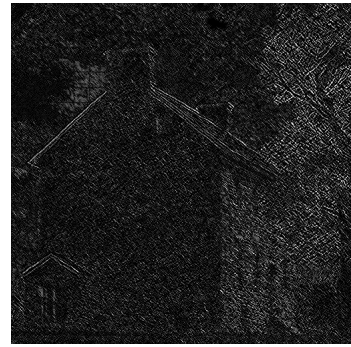
Este filtro é parecido com o filtro anterior, porém nele é encontrado retas diagonais com ângulo de 135 graus. Ainda é bem nítido a visualização dos rabiscos diagonais na parede da casa e algumas bordas no telhado foram novamente destacadas, porém, agora com um ângulo diferente.



(a) Imagem original

$$h_8 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

(b) Filtro 8



(c) Imagem filtrada

4.8 Nono filtro

Esta máscara é outro filtro passa baixa, porém a média é realizada somente nos pixels da diagonal principal, fazendo assim que a suavização seja bem maior do que todas já apresentadas, deixando a imagem bastante borrada e com muito pouco ruído, porém algumas bordas foram obstruídas.



(a) Imagem original

$$h_9 = \frac{1}{9} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

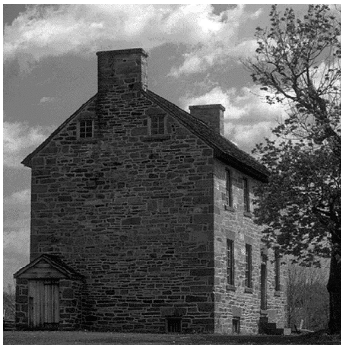
(b) Filtro 9



(c) Imagem filtrada

4.9 Décimo filtro

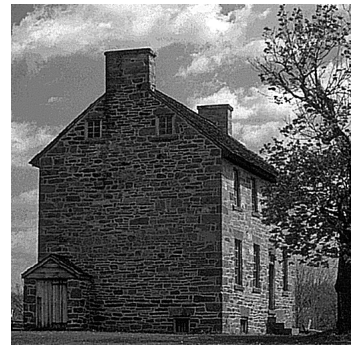
O próximo filtro obteve um efeito totalmente diferente dos outros já analisados, ele deixou a imagem mais nítida, porém o filtro gerou alguns ruídos na imagem. Ao utilizá-lo, as bordas de toda imagem ficaram mais destacáveis, dando essa sensação de nitidez.



(a) Imagem original

$$h_{10} = \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & 2 & 8 & 2 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

(b) Filtro 10

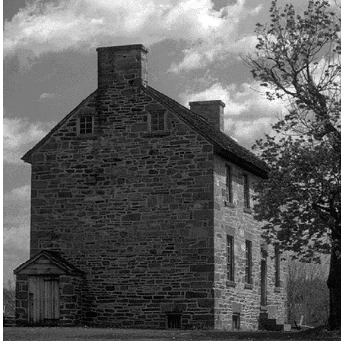


(c) Imagem filtrada

4.10 Décimo primeiro filtro

O último filtro gerou uma imagem com algumas bordas, porém pelos testes com outros 3 filtros (variações do filtro 11), foi observado que as bordas geradas foram somente algumas na parte supe-

rior esquerda dos objetos da imagem. Aparentemente onde há uma grande variação do contraste em direção aos valores negativos do filtro, é gerada um relevo na cor branca na imagem.



(a) Imagem original

$$h_{11} = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

(b) Filtro 11

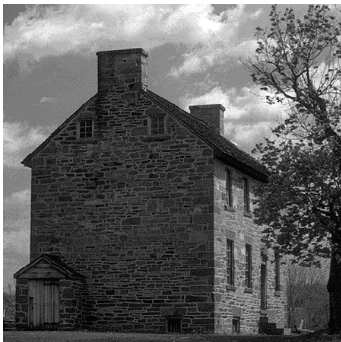


(c) Imagem filtrada

4.10.1 Variações do décimo primeiro filtro

Para uma compreensão melhor do filtro anterior, foram feitos mais 3 filtros para este trabalho, sendo eles o h12, h13 e o h14. Cada um deles foram obtidos através de rotações do filtro 11. O Filtro 12 foi obtido pela rotação de 90 graus do filtro 11, nele ao ser utilizado, a imagem ainda continuava a formar bordas na esquerda de objetos da imagem, porém agora na parte inferior.

O Filtro 13, foi obtido pela rotação de 180 graus do filtro 11, ao ser utilizado, a imagem obtida foi uma com bordas destacada na direita na parte inferior dos objetos, já a máscara 14, foi obtida com a rotação de 270 graus do filtro 11, ao utilizá-la, foi notado que as bordas continuavam na parte direita dos objetos, porém agora na parte superior. Esse feito parece muito que a detecção de relevo está se movimentando, a medida que a máscara é rotacionando, seguindo os valores negativos dos filtros.



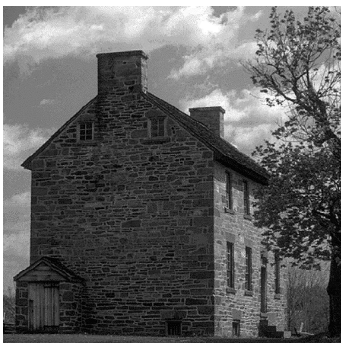
(a) Imagem original

$$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

(b) Filtro 12



(c) Imagem filtrada



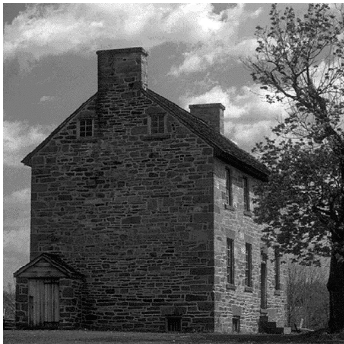
(a) Imagem original

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$

(b) Filtro 13



(c) Imagem filtrada



(a) Imagem original

0	-1	-1
1	0	-1
1	1	0

(b) Filtro 14



(c) Imagem filtrada