

# Projeto 2

## MO443 - Introdução ao Processamento de Imagem Digital

Caio Augusto Alves Nolasco - RA:195181

### 1. Introdução e especificação do problema

O objetivo do trabalho 2 é aplicar diferentes filtros sobre uma imagem monocromática, alterando os valores de cinza de cada pixel da representação matricial da imagem, a partir de diferentes funções definidas pelas máscaras de cada filtro. O novo valor do pixel depende do seu valor inicial e do valor dos seus oito vizinhos. Cada pixel é trocado pela soma dos produtos (de elemento por elemento) dos valores dos pixels cobertos pela máscara e os valores atuais da imagem.

### 2. Pressupostos

O programa em Python recebe uma imagem de entrada para poder realizar a aplicação dos filtros. Sobre essa imagem são feitos os seguintes pressupostos:

- A imagem é monocromática;
- A imagem está salva no disco do computador que executa o código;
- A profundidade da imagem é igual a 1.

Vale ressaltar que o programa também pressupõe que as máscaras que serão aplicadas são todas matrizes de tamanho 3x3 e 5x5.

### 3. Decisões tomadas

O programa é feito baseado em decisões de design preestabelecidas para seu funcionamento. Elas são:

- O diretório da imagem original é fornecido por linha de comando em tempo de execução;
- A imagem resultante é salva em um diretório informado pelo usuário por linha de comando em tempo de execução;
- A imagem resultante é salva em um arquivo PNG com o nome “filterhX.png” no diretório destino, em que “hX” representa o filtro aplicado para essa imagem, definidos pelo enunciado do projeto e explorados mais adiante neste relatório;
- Os testes foram realizados sobre a imagem “city.png” disponibilizada no site da disciplina [1].

### 4. Padrões de implementação

Com exceção do caso específico da combinação dos filtros h3 e h4, os filtros são aplicados de maneira similar. Primeiro, a imagem é aberta usando a função `imread()` da biblioteca OpenCV2 [2], guardada como um Array da biblioteca Numpy em uma variável. Então, é criada uma matriz bidimensional como um Array da biblioteca Numpy, com os

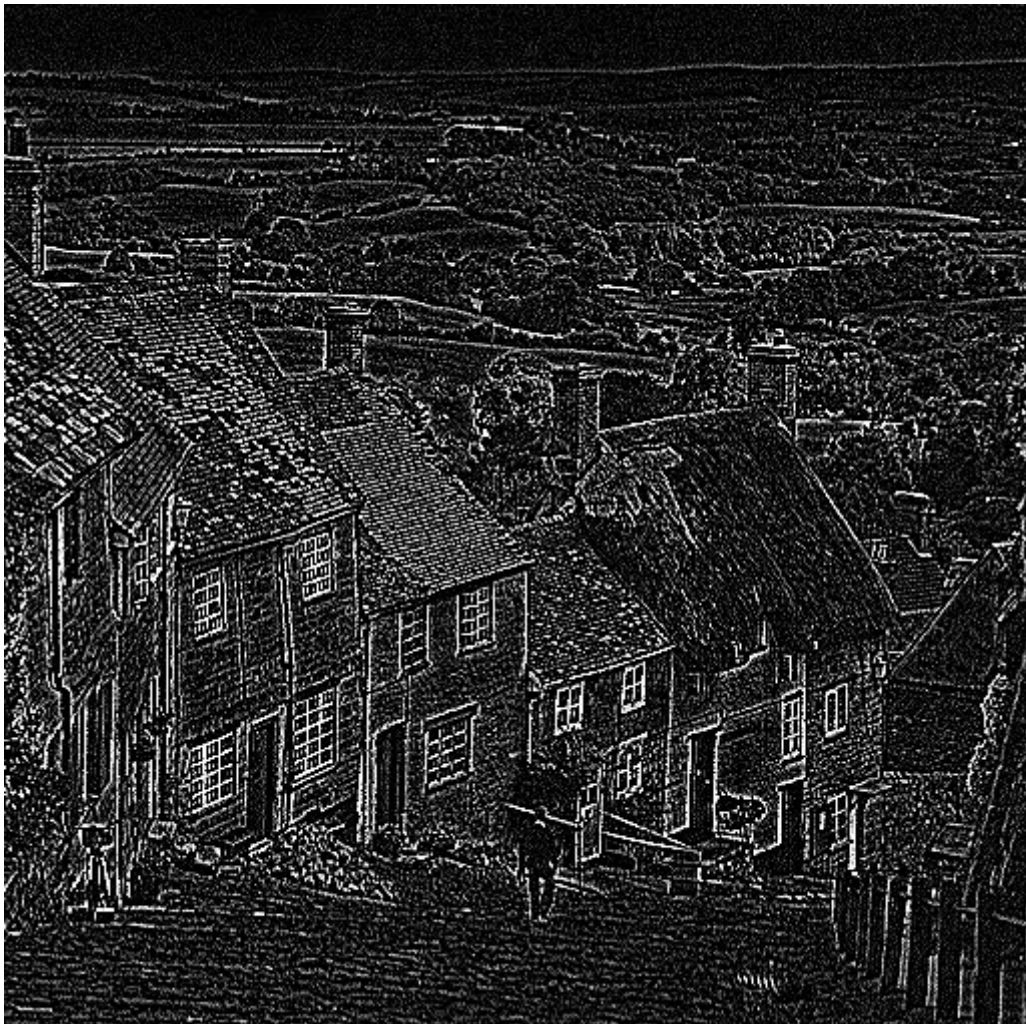
valores referentes à máscara pertencente ao determinado filtro, especificadas no enunciado do projeto. É invocada a função `filter2D()` de OpenCV2 [3], com a imagem em objeto Array e máscara criada como parâmetros. A função é responsável por aplicar a máscara sobre os valores originais da matriz, por convolução. O retorno de `filter2D()` é salvo em outra variável, que por fim é salva como uma imagem PNG por meio de `imwrite()` de OpenCV2 [4].

## 5. Resultado dos filtros

Filtro h1)

$$h1 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

O filtro h1 dá maior ênfase ao pixel sendo alterado na etapa corrente da aplicação da máscara em relação a seus vizinhos. O resultado desta alteração, quando aplicada na imagem inteira, é o realce das bordas da imagem, dando maior destaque a regiões com maior disparidade de valores de cinza com seus vizinhos.



Filtro h2)

$$h2 = 1/256 * \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

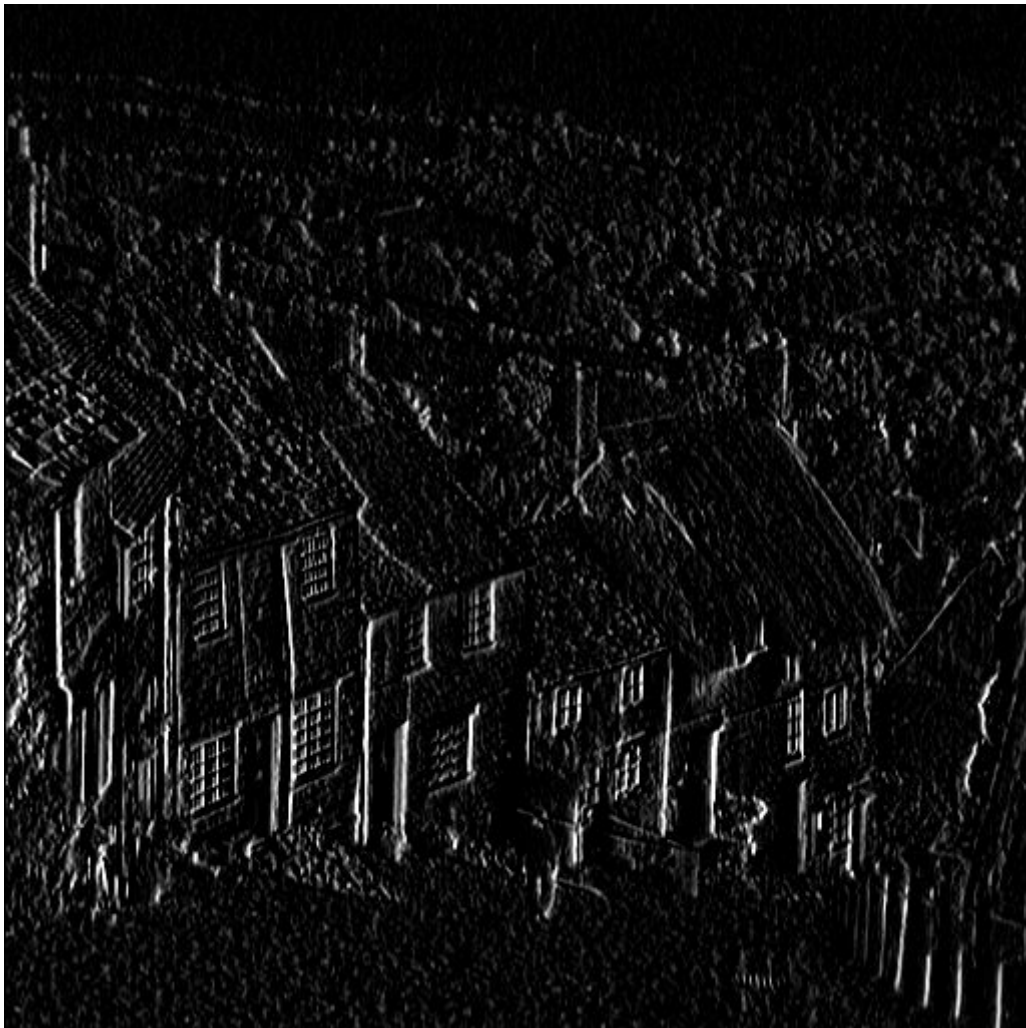
O filtro h2, similarmente a h2, dá um peso maior ao pixel em alteração em relação a seus vizinhos. Este filtro é o filtro Gaussiano, que resulta em suavização da imagem, reduzindo a intensidade de bordas da imagem, diminuindo detalhes e ruído.



Filtro h3)

$$h3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

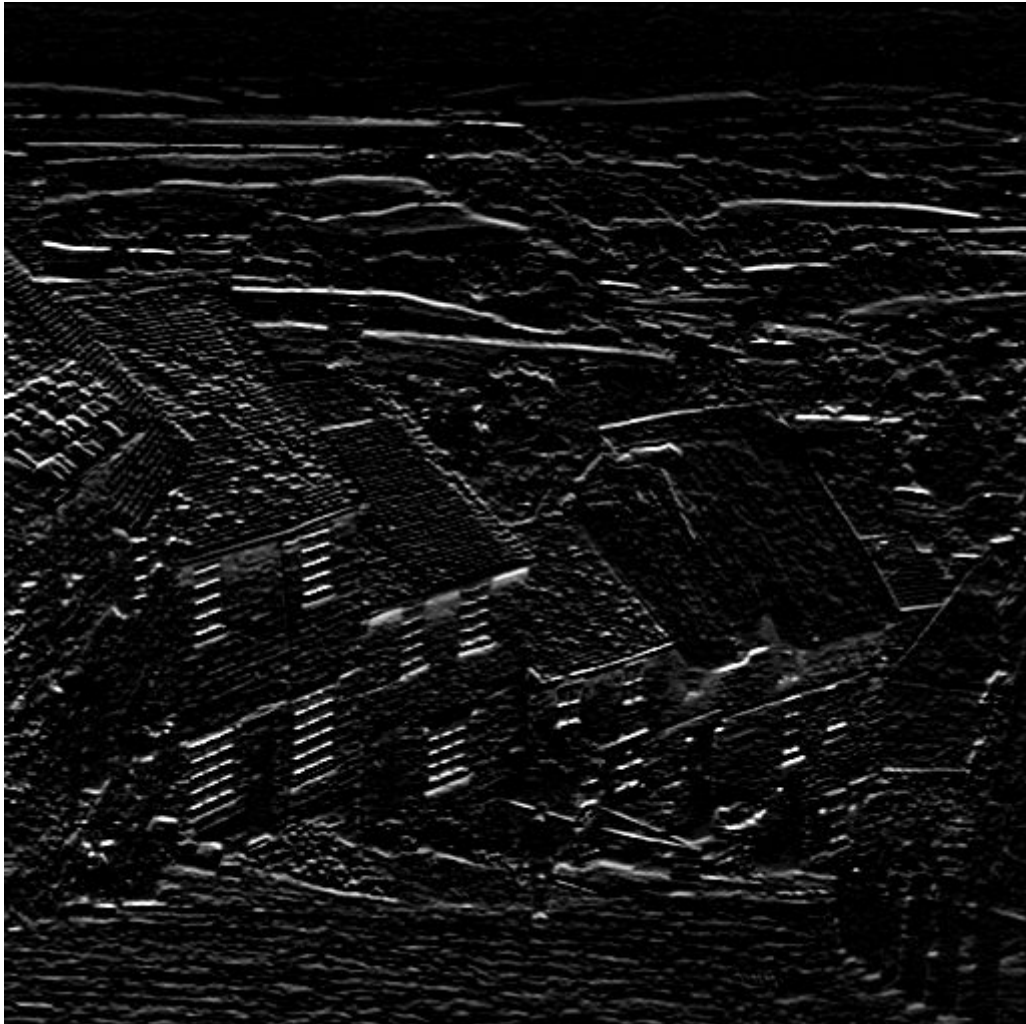
O filtro h3 é ideal para detecção de bordas. Ao inferir pesos maior a um dos lados da máscara (neste caso, o direito) em relação ao lado oposto, o filtro detecta mudanças bruscas no valores de cinza de um pixel e seus vizinhos, realçando as bordas verticais da imagem.



Filtro h4)

$$h4 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

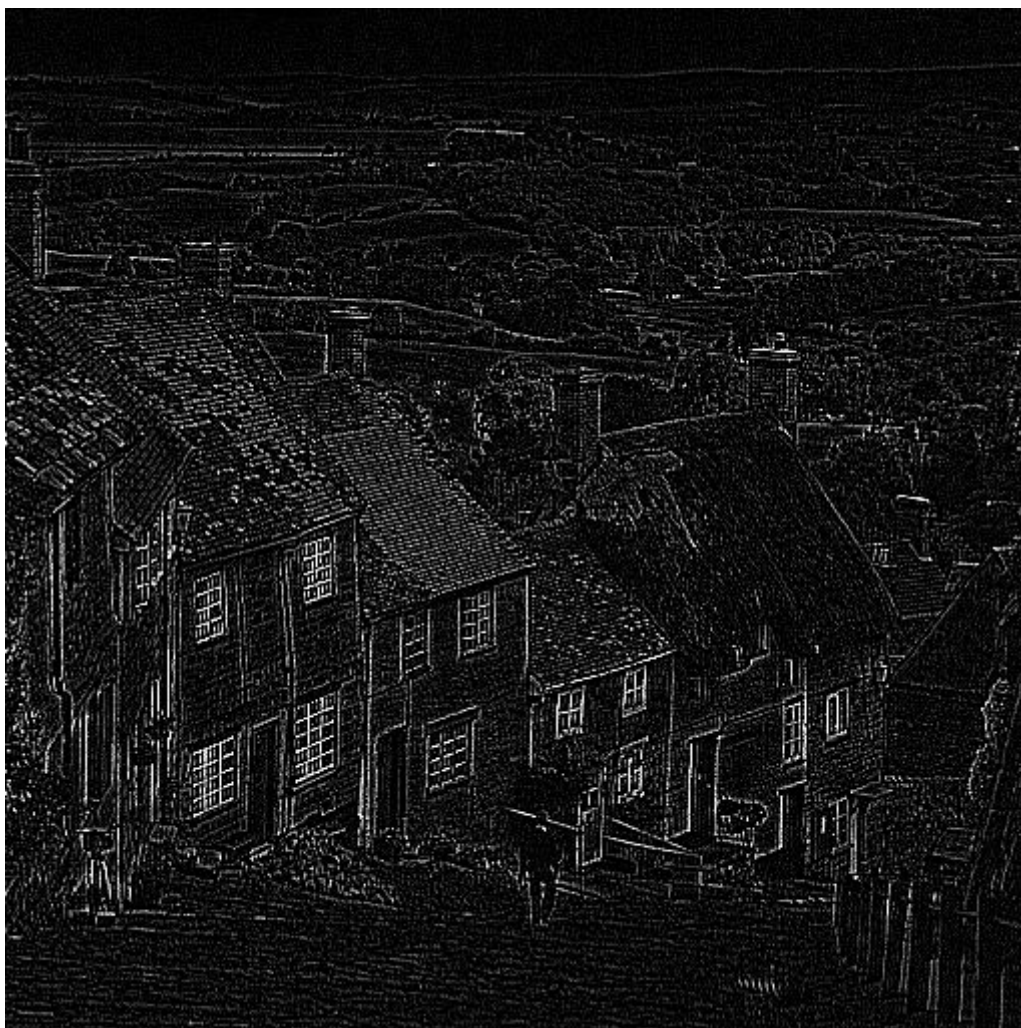
O filtro h4 alcança resultados muito similares ao filtro h3, só que neste caso são as bordas horizontais que são realçadas, já que a máscara dá um peso maior ao vizinhos inferiores de cada pixel.



Filtro h5)

$$h5 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

O filtro h5 funciona como um espécie de combinação dos filtros h3 e h4. O objetivo é realçar todas as bordas, independente de sua direção, por meio do ressalto de cada pixel em relação ao seus vizinhos.



Filtro h6)

$$h6 = 1/9 * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

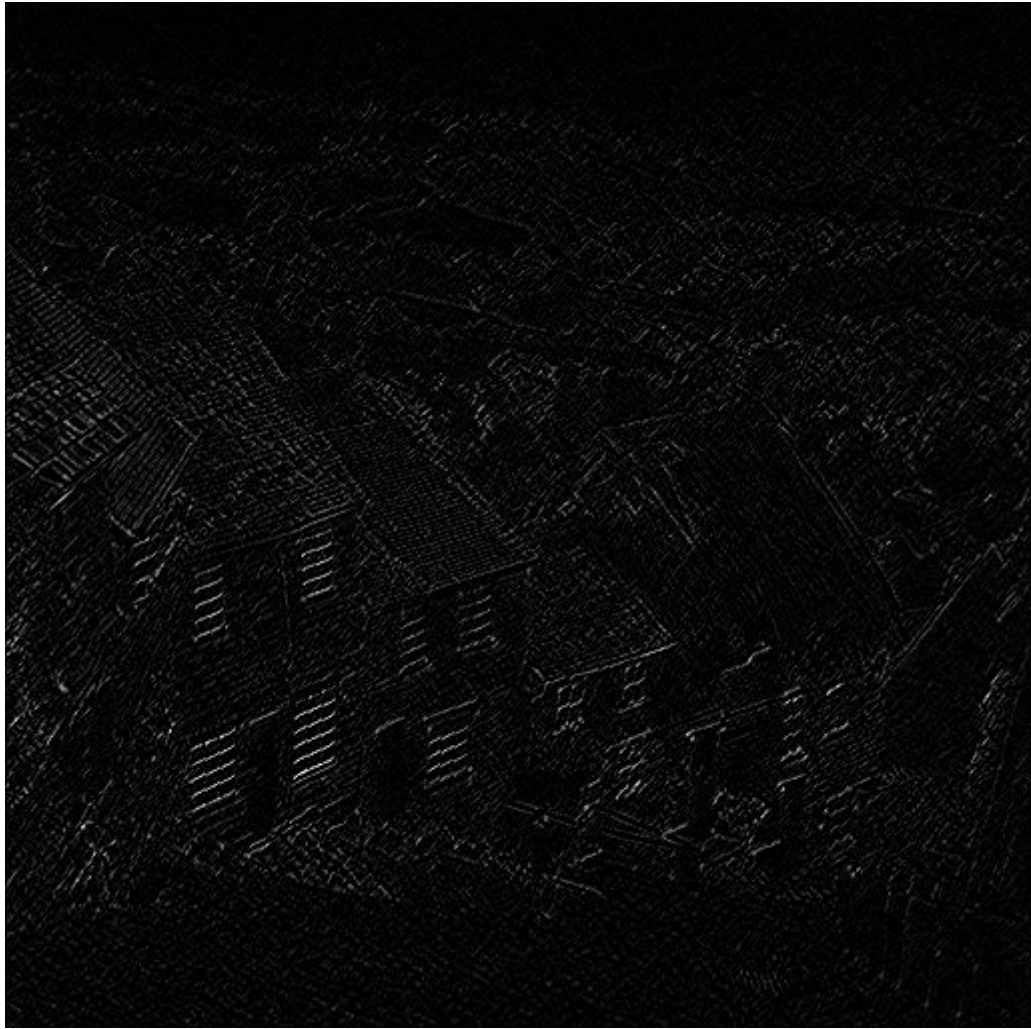
O filtro h6 aplica um efeito suavizador na imagem, aplicando a cada pixel o valor da média entre ele e seus vizinhos, como uma forma de se aproximar da suavização Gaussiana de h2



Filtro h7)

$$h7 = \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

O filtro h7 realça na imagem as linha inclinadas para a direita, atribuindo pesos maiores aos vizinhos que estão na mesma linha inclinada em detrimento aos demais vizinhos de cada pixel.



Filtro h8)

$$h8 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

Contrariamente ao efeito do filtro h7, o filtro h8 realça as linha inclinada para a esquerda, usando o mesmo princípio da máscara de h7, mas com ênfase nos vizinhos correspondentes a diagonal da matriz.





#### Combinação dos filtros h3 e h4)

Os filtros h3 e h4 são combinados realizando a média geométrica de cada pixel das imagens resultantes de h3 e h4. Como o filtro h3 realça as bordas verticais da imagem, e o filtro h4 as bordas horizontais, a combinação dos dois resulta em uma imagem com as bordas ressaltadas de forma geral.

