Filtragem no Domínio Espacial

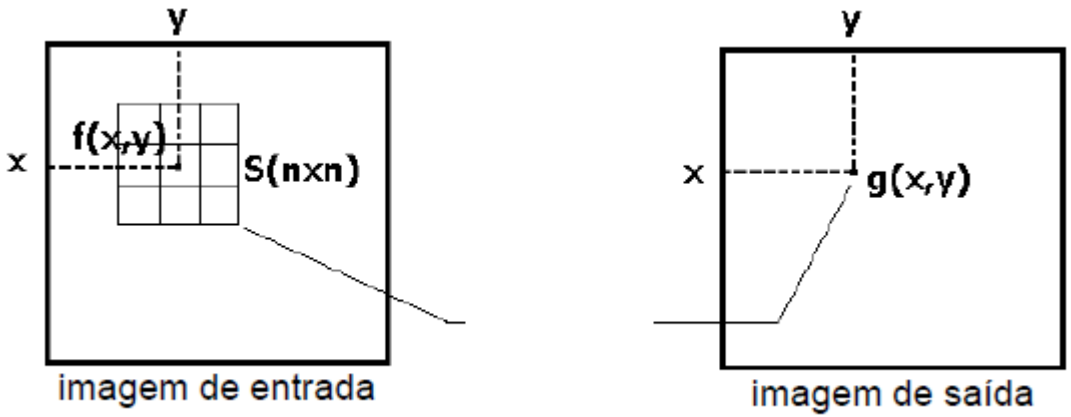
Aula 4

Pablo G. Cavalcanti

Aula originalmente produzida por: Prof. André Backes (UFU)

Definição

- Conhecidos como operadores locais ou filtros locais
- Combinam a intensidade de um certo número de pixels, para gerar a intensidade da imagem de saída.

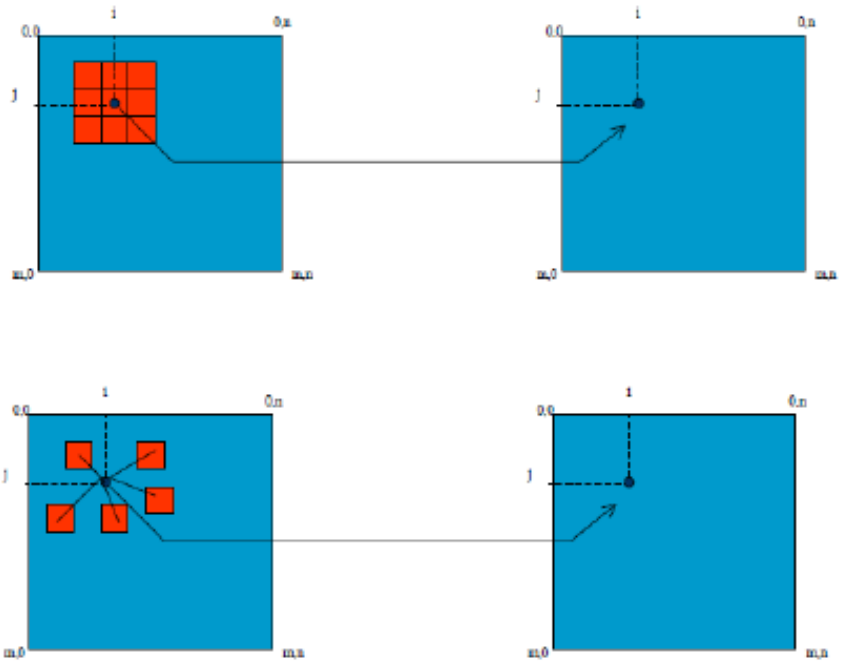


São os operadores locais mais utilizados em processamento de imagens, com diversas aplicações:

- Pré-processamento
- Eliminação de ruídos
- Suavização
- Segmentação

São técnicas baseadas na convolução de

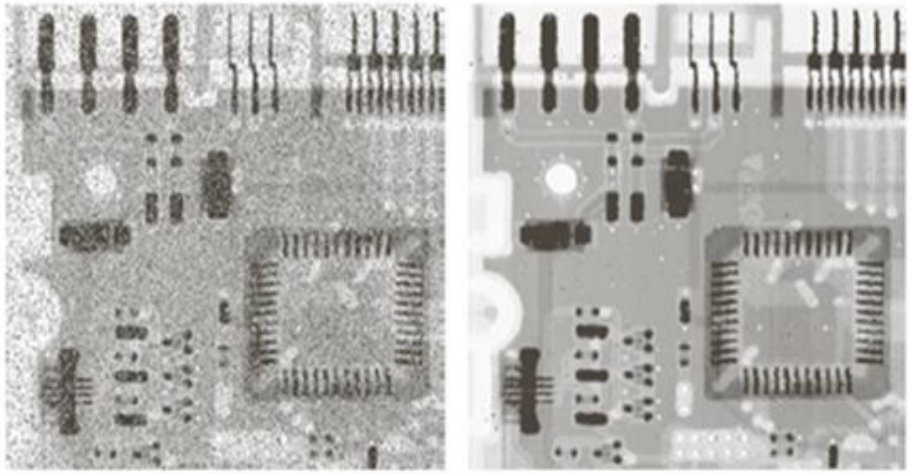
- templates
  - janelas, matrizes
- tuplas
  - conjunto de pixels



$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

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

Exemplo: remoção de ruído



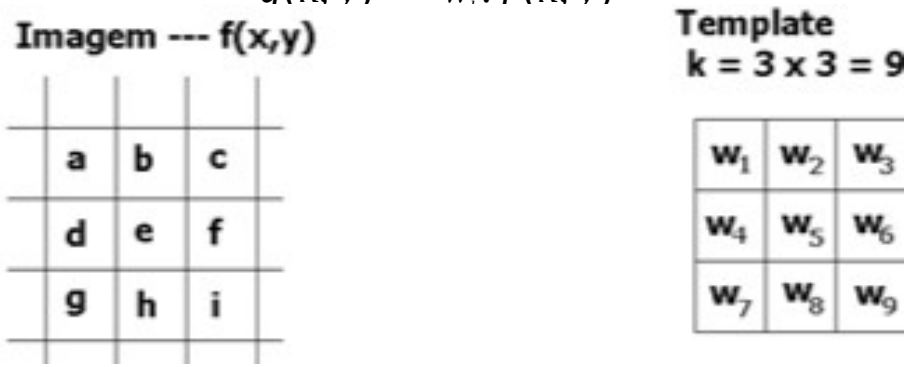
Processo de filtragem

- Cada elemento da máscara é multiplicado pelo valor do pixel correspondente na imagem ***f***

- A soma desses resultados é o novo valor do nível de cinza na nova imagem ***g***

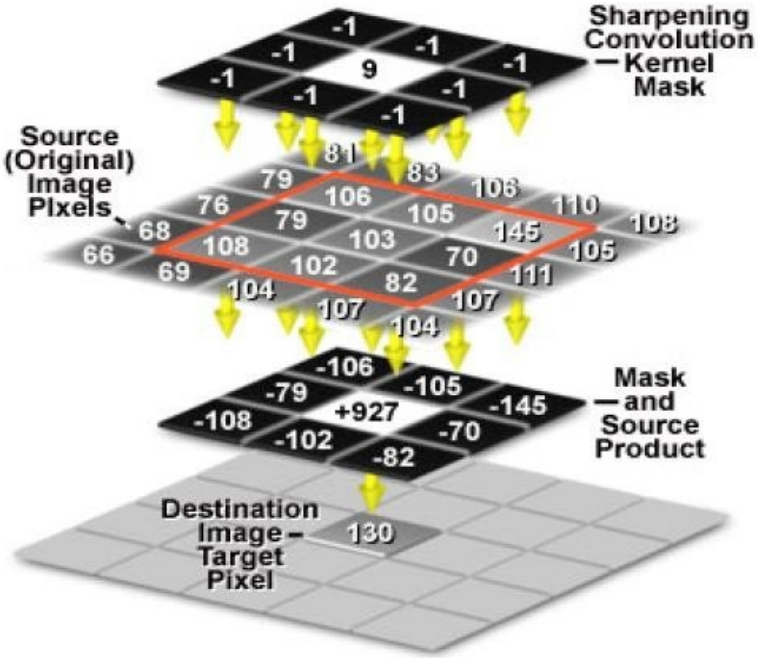
- Exemplo: *w* é uma janela de *n x n = k* pixels. O processo de filtragem para cada pixel na imagem ***g(x,y)*** será dada por:

$$g(x,y) = w_i . f(x,y)$$



- *(a,b,c,d,e,f,g,h,i):* são os valores dos níveis de cinza na vizinhança de *f(x,y)*
- *(w_1 a w_9)*: são os coeficientes da máscara
- O valor do pixel *g(x,y)* é dado por

$$g(x,y) = w_1.a + w_2.b + w_3.c + w_4.d + w_5.e + w_6.f + w_7.g + w_8.h + w_9.i$$



Existem dois conceitos matemáticos importantes e que estão relacionados com a filtragem espacial: **correlação** e **convolução.**

- Correlação
  - Desloca-se a máscara sobre a imagem e calcula- se a soma dos produtos em cada local

- Convolução
  - Mesmo processo que a correlação, exceto que a máscara é antes espelhada (rotacionada em 180º)

- Equações para máscaras de tamanho *m* x *n*
  - Correlação

$$w(x,y) \circ f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

  - Convolução

$$w(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

Espelhamento ou rotação, feito na imagem

$$a = (m-1)/2 \qquad b = (n-1)/2$$

- Observações
  - As equações devem ser avaliadas para todas as posições *x* e *y* da imagem
  - Se a máscara for simétrica, os resultados da convolução e da correlação são os mesmos
    - No geral, em aplicações de processamento de imagens, as máscaras são simétricas sendo correlação e convolução consideradas como a mesma coisa

Desloca, Multiplica, Soma

Imagem

Resultado

máscara

1	0
0	1

1	1	3	3	4
1	1	4	4	3
2	1	3	3	3
1	1	1	4	4

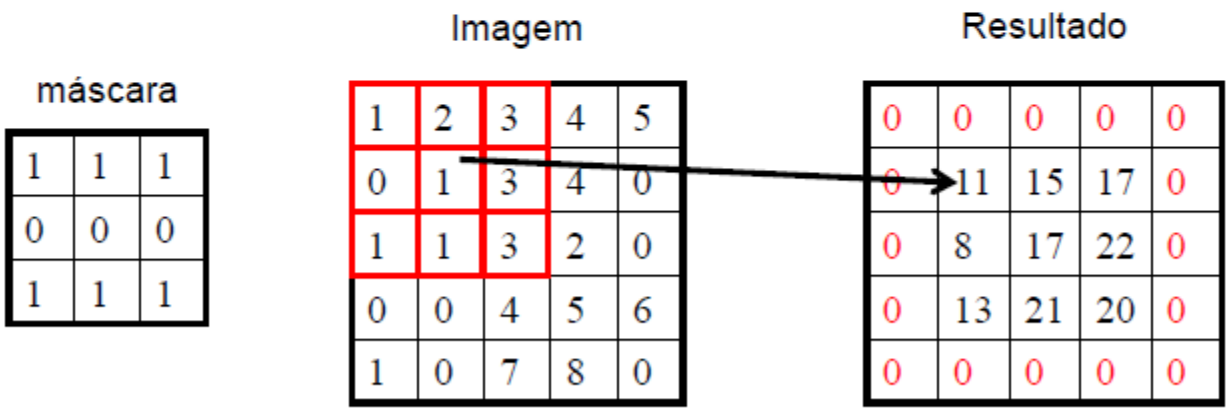
2	5	7	6	*
2	4	7	7	*
3	2	7	7	*
*	*	*	*	*

A imagem resultado é menor do que a imagem original. Os valores marcados com * não podem ser calculados.

- Convenção
  - Nas máscaras de organização par (2 x 2, 4 x 4 , ...) o resultado é colocado sobre o primeiro pixel
  - Nas máscaras de organização ímpar (3 x 3, 5 x 5, ...) o resultado é colocado sobre o pixel de centro
  - A imagem resultado da convolução não necessita obrigatoriamente ser menor que a imagem original.
    - Convolução aperiódica
    - Gabarito truncado
    - Convolução periódica

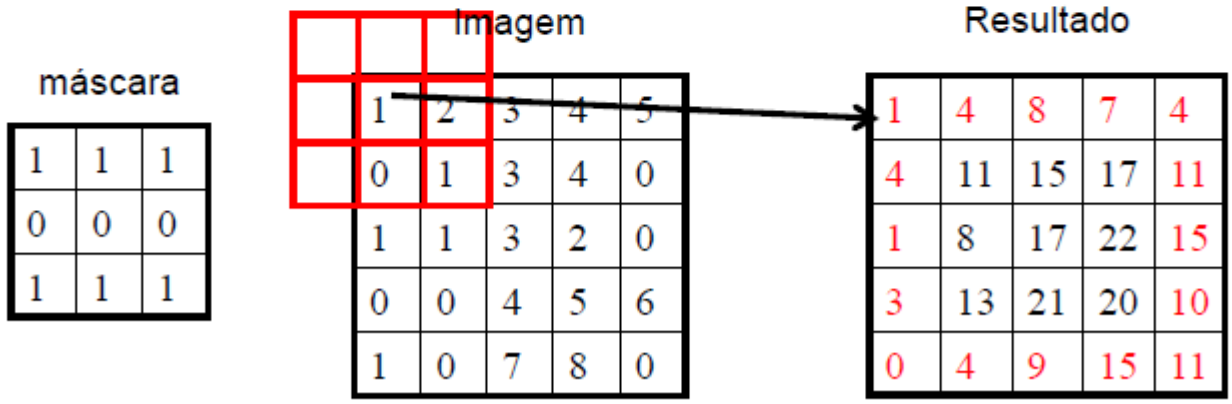
Convolução aperiódica

- O valor 0 é atribuído aos resultados não calculáveis



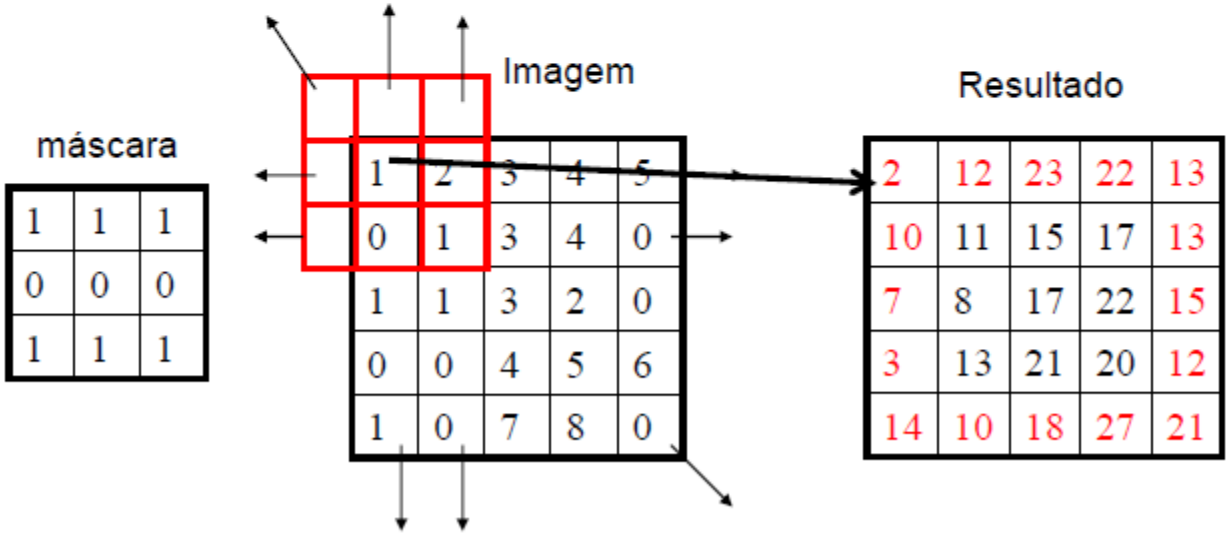
Gabarito truncado

- Centra-se a máscara com o primeiro pixel da imagem atribuindo o valor 0 aos valores inexistentes na imagem



Convolução periódica

- A máscara é deslocada sobre todos os pixels da imagem original como se esta fosse adjacente em suas extremidades



Função “pronta” para convolução:

SciPy.org Sponsored By ENTHOUGHT

Scipy.org | Docs | SciPy v1.0.0 Reference Guide | Multi-dimensional image processing (`scipy.ndimage`)

scipy.ndimage.convolve

scipy.ndimage.**convolve**(*input, weights, output=None, mode='reflect', cval=0.0, origin=0*) [source]

Multidimensional convolution.

The array is convolved with the given kernel.

Parameters:

input : *array_like*
Input array to filter.

weights : *array_like*
Array of weights, same number of dimensions as input

output : *ndarray, optional*
The *output* parameter passes an array in which to store the filter output. Output array should have different name as compared to input array to avoid aliasing errors.

mode : *{'reflect','constant','nearest','mirror', 'wrap'}, optional*
the *mode* parameter determines how the array borders are handled. For 'constant' mode, values beyond borders are set to be *cval*. Default is 'reflect'.

cval : *scalar, optional*
Value to fill past edges of input if *mode* is 'constant'. Default is 0.0

origin : *array_like, optional*
The *origin* parameter controls the placement of the filter, relative to the centre of the current element of the input. Default of 0 is equivalent to `(0,)*input.ndim`.

Returns:

result : *ndarray*
The result of convolution of *input* with *weights*.

Entendendo o “mode”:

```
    mode       |   Ext   |         Input         |   Ext
    -----------+---------+-----------------------+---------
    'mirror'   | 4  3  2 | 1  2  3  4  5  6  7  8 | 7  6  5
    'reflect'  | 3  2  1 | 1  2  3  4  5  6  7  8 | 8  7  6
    'nearest'  | 1  1  1 | 1  2  3  4  5  6  7  8 | 8  8  8
    'constant' | 0  0  0 | 1  2  3  4  5  6  7  8 | 0  0  0
    'wrap'     | 6  7  8 | 1  2  3  4  5  6  7  8 | 1  2  3
```

Exemplo de uso:

```
>>> a = np.array([[1, 2, 0, 0],
...               [5, 3, 0, 4],
...               [0, 0, 0, 7],
...               [9, 3, 0, 0]])
>>> k = np.array([[1,1,1],[1,1,0],[1,0,0]])
>>> from scipy import ndimage
>>> ndimage.convolve(a, k, mode='constant', cval=0.0)
array([[11, 10,  7,  4],
       [10,  3, 11, 11],
       [15, 12, 14,  7],
       [12,  3,  7,  0]])
```

Fonte: https://stackoverflow.com/questions/22669252/how-exactly-does-the-reflect-mode-for-scipys-ndimage-filters-work

- O custo computacional da convolução é alto
  - Em um imagem de tamanho *M x M* e máscara *N x N*, o Número de multiplicações é de M^2N^2
  - Exemplo: imagem de 512 x 512 e máscara de 16 x 16 = 67.108.864 multiplicações.

- Alternativa: domínio da frequência (Fourier)
  - Só é justificável se a máscara for maior do que 32 x 32
  - Custo da Transformada de Fourier

Máscaras de convolução

- O tamanho da máscara e os valores de seus coeficientes definem o tipo de filtragem produzido
- Exemplos
  - Passa Baixa e média espacial (suavização)
  - Filtragem mediana
  - Passa Alta (realce)
  - Passa banda
  - Gradientes (robert, sobel, etc): detectores de borda

Filtros de Suavização

- Também chamados de filtros passa-baixa
  - Utiliza uma máscara que realiza a média da vizinhança.
  - Numa máscara de média, os coeficiente são positivos e a soma deles é igual a 1
  - Quanto maior a máscara maior efeito de borramento
- Exemplos de máscaras

$$\frac{1}{5}\begin{bmatrix}0 & 1 & 0\\ 1 & 1 & 1\\ 0 & 1 & 0\end{bmatrix} \quad \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \quad \frac{1}{32}\begin{bmatrix}1 & 3 & 1\\ 3 & 16 & 3\\ 1 & 3 & 1\end{bmatrix} \quad \frac{1}{8}\begin{bmatrix}0 & 1 & 0\\ 1 & 4 & 1\\ 0 & 1 & 0\end{bmatrix}$$

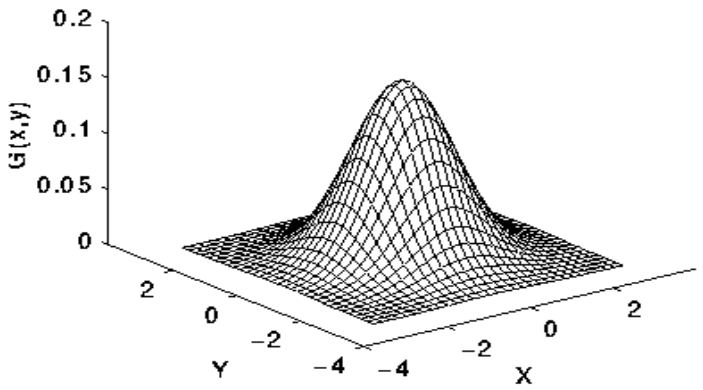
Filtros de Suavização

- São filtros usados para o borramento

- São filtros usados para a redução de ruídos

- Filtro Gaussiano
  - Utiliza a função gaussiana para o cálculo dos coeficientes da máscara

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

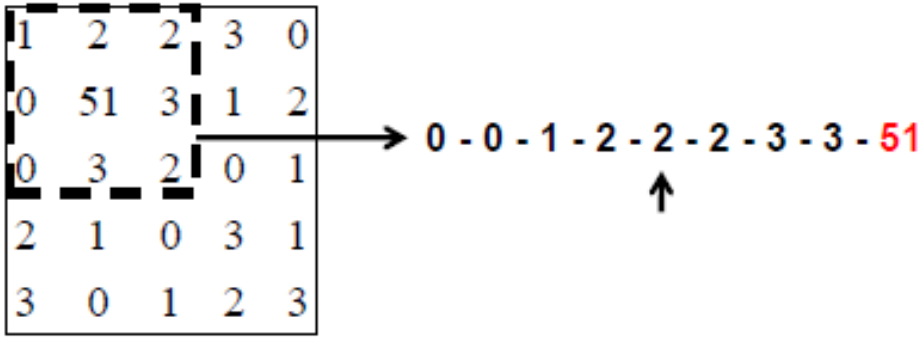


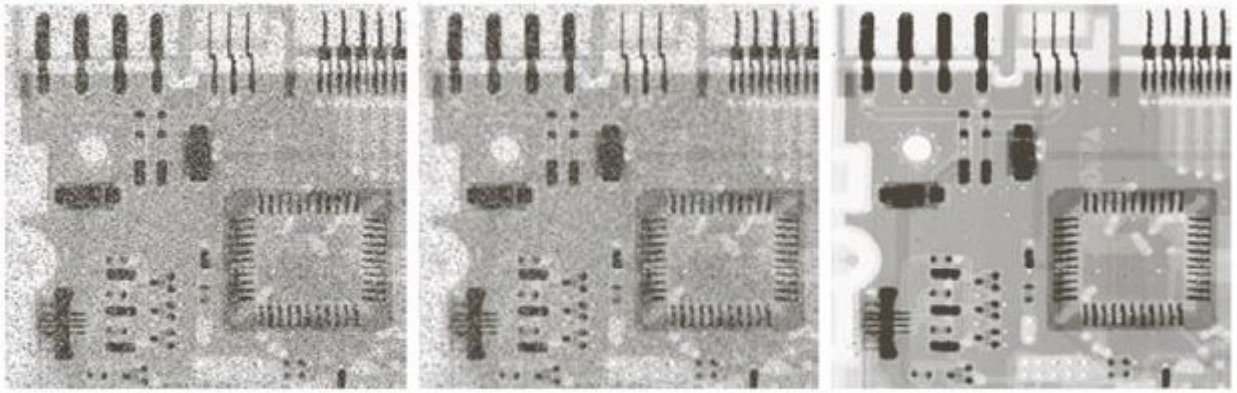
Máscara (sigma = 1)

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

- Filtro de mediana
  - Mediana: valor que ocupa a posição central de um conjunto
  - Trata-se de um filtro não linear: não é feita a convolução de uma máscara
  - A intensidade de cada pixel é substituída pela mediana das intensidades na vizinhança daquele pixel.
    - Ex: o ponto de valor 51 é um ruído:





Original Filtro de média 3x3 Filtro de mediana 3x3

Filtros de Realce

- Também chamados de filtros passa-alta
  - O realce (*sharpening*) tem como objetivo destacar as transições de intensidade na imagem
  - Utiliza um tipo de máscara (normalmente baseada em derivadas) que tende a realçar as diferenças de níveis de cinza na imagem

- Derivadas são proporcionais ao grau de descontinuidade na imagem
  - Enfatizam as regiões de bordas e os ruídos
  - Não enfatizam regiões constantes ou com variações de intensidade suaves
- Filtros
  - Laplaciano
  - *Unsharp masking* e *highboost filtering*
  - Derivativos

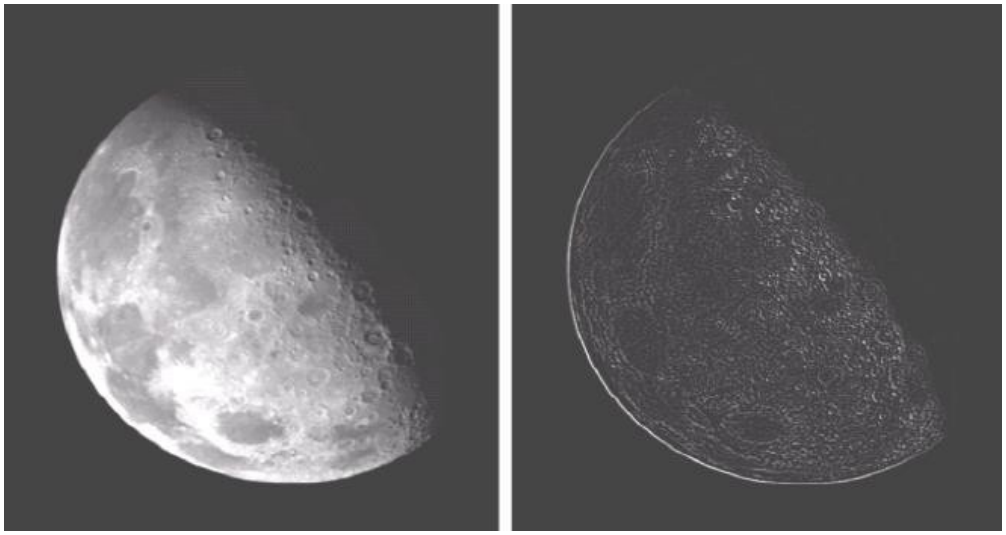
Exemplos de filtros Laplacianos 3x3

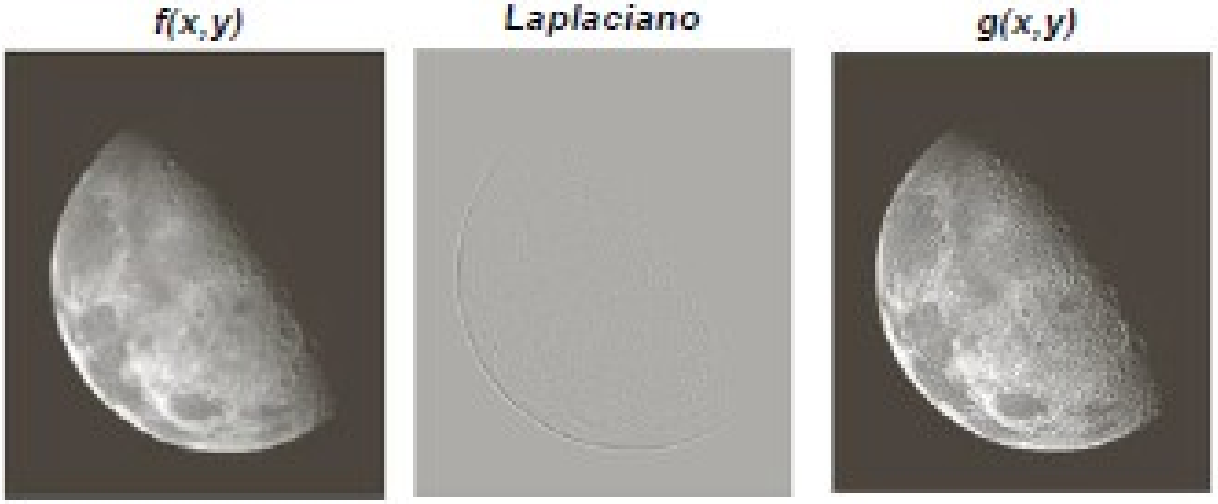
0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1





- Como o filtro Laplaciano é linear, existem máscaras que já combinam as duas operações (realce + reconstrução do fundo da imagem)

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1

Filtros de Realce

- Um processo para aumentar a nitidez das imagens consiste em subtrair uma versão não nítida (*suavizada*) de uma imagem da imagem original

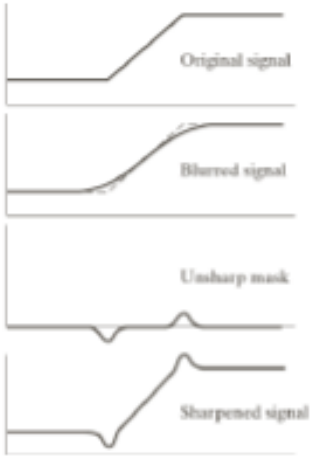
- Passos
  - Borrar a imagem original
  - Subtrair a imagem borrada da original – a diferença resultante é chamada de *máscara*
  - Adicionar a *máscara* à imagem original

- Unsharp masking (máscara de nitidez) e filtragem highboost
  - Seja s(x,y) uma suavização da imagem f(x,y)

$$g_{mask}(x,y) = f(x,y) - s(x,y)$$

$$g(x,y) = f(x,y) + g_{mask}(x,y)$$

- Exemplo unidimensional para entender o processo



Generalizando

- *k = 1 → unsharp masking*
- *K > 1 → highboost filtering (filtragem alto-reforço)*
- *K < 1 → atenua a contribuição da máscara de nitidez*

$$g_{mask}(x,y) = f(x,y) - s(x,y)$$

$$g(x,y) = f(x,y) + k.g_{mask}(x,y)$$

Unsharp masking e filtragem *highboost*

Resultado usando
unsharp mask

Resultado usando filtragem
highboost (k=2)

Links:

http://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.convolve2d.html

http://stackoverflow.com/questions/14765891/image-smoothing-in-python

http://scikit-image.org/docs/dev/api/skimage.filters.html