Processamento e Análise de Imagens

Processamento no Domínio Espacial II

Felipe Augusto Lima Reis

PUC Minas

Fundamentos da Filtragem Espacial

Filtragem Espacial

- **Filtragem espacial** é um procedimento utilizado em uma gama de atividades de processamento de imagens, principalmente para melhoramento de imagens;
- A expressão “filtragem” é oriunda do domínio de frequência e refere-se a permitir a passagem, modificar ou rejeitar frequências específicas em uma imagem;
    - Filtragem espacial modifica uma imagem pela substituição do valor de cada pixel por uma função que avalia os pixels e sua vizinhança.
- Operações de filtragem espacial podem ser divididas em:
    - **Lineares**: operações executadas nos pixels das imagens são lineares;
    - **Não-lineares**: utilizam operações não lineares para filtragem de imagens;

Filtragem linear espacial

- Um filtro linear espacial executa a operação de soma de produtos entre uma imagem *f* e um **filtro de kernel**, *w*.
  - O **kernel** é um array cujo tamanho define uma vizinhança de uma operação e no qual os coeficientes determinam a natureza do filtro;
  - Kernels também são chamados de **máscaras** (*masks*), templates ou *windows*.
  - São utilizados na literatura, sem distrinções, os termos “filtro de kernel” ou somente “kernel”.

Filtro de kernel

- Na filtragem linear espacial, em qualquer ponto *(x, y)*, a resposta, *g(x, y)*, do filtro é a soma de produtos dos coeficientes do kernel da imagem englobado pelo kernel

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \dots + w(0,0)f(x,y) + \dots + w(1,1)f(x+1,y+1)$$

- As coordenadas *x* e *y* variam, e o centro do kernel move-se pixel a pixel, gerando a imagem filtrada *g* no processo;
- Kernels podem ter tamanhos variados, porém, em geral são quadrados e com tamanhos ímpares (ex.: 3×3, 5×5, 7×7, ...).
  - Para kernels de tamanhos $m \times n$, assume-se que $m=2a+1$ e $n=2b+1$, onde a e b são valores inteiros não negativos.

Filtro de kernel

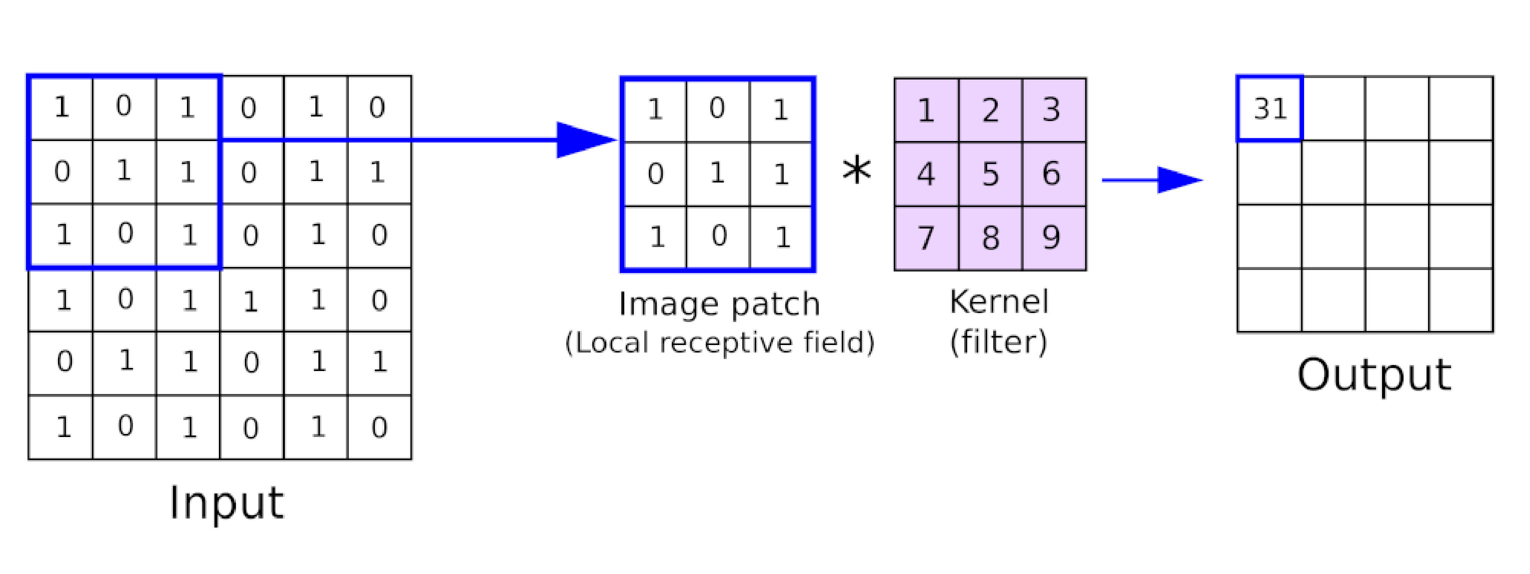
- A **filtragem linear espacial** de uma imagem de tamanho $M \times N$, com tamanho de kernel de $m \times n$ é dada pela expressão:

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

  - onde x e y variam de modo com que o kernel visite cada pixel de f uma vez.
- Para computar os kernels nos cantos das imagens, é necessário a adição de valores 0’s nas bordas
  - Esses 0’s adicionados às bordas das imagens recebem no nome de **pad**;
  - A expressão **padding** é adotada também para esta operação, sendo que *padding* recebe um valor, correspondente aos números de 0’s externos adicionados à imagem.

Filtro de kernel

- O funcionamento de um filtro de kernel pode ser visto na figura abaixo
  - Observe que a imagem de saída possui tamanho menor que a imagem de entrada, uma vez que não foi utilizado padding na operação.



Fonte: Ahn Reynolds (2019).

Correlação e Convolução

Correlação e Convolução

- **Correlação espacial** consiste em mover o centro de um kernel sobre uma imagem e computar a soma dos produtos em cada um dos pontos.

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

  - onde *x* e *y* variam de modo com que o kernel visite cada pixel de f uma única vez.
- **Convolução espacial** contém o mesmo mecanismo da correlação, exceto que o kernel é rotacionado 180º (veremos os motivos adiante)
  - Quando os valores do kernel são simétricos em relação ao centro, correlação e convolução produzem os mesmos resultados.

Correlação e Convolução

- Considerando um kernel de uma dimensão (ex.: 1×5), a equação de correlação é definida por:

$$g(x) = \sum_{s=-a}^{a} w(s) f(x+s)$$

- Para valores nos cantos das imagens, é adotado uma função de padding
  - Se o kernel possui tamanho *1×m*, são necessários *(m-1)/2* zeros de cada lado da imagem para que a correlação / convolução seja aplicada.

Correlação e Convolução

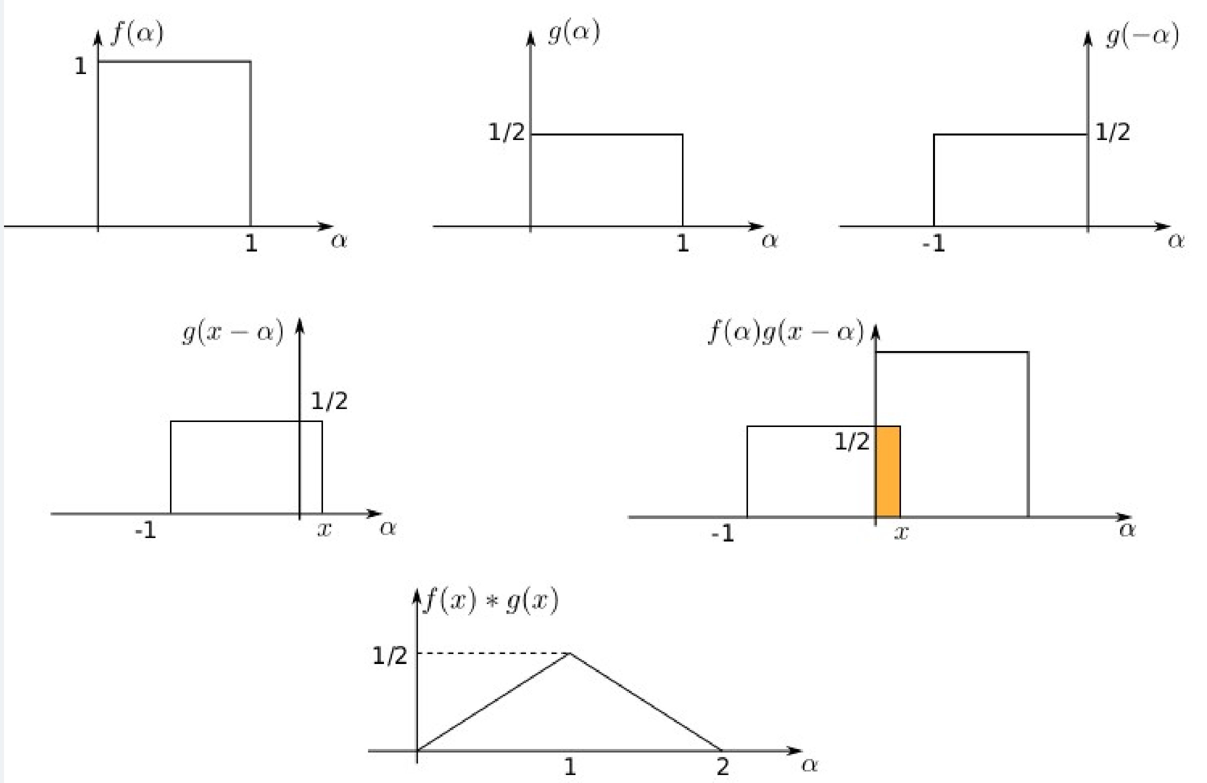
- Os exemplos vistos até aqui correspondem a operações de correlação e convolução discretas
  - A operação, entretanto, pode ser aplicada matematicamente em espaços contínuos
- Uma convolução bidimensional contínua, para uma variável α, é definida como:

$$f(x,y) * g(x,y) = \iint_{-\infty}^{\infty} f(\alpha,\beta) g(x-\alpha, y-\beta) \, d\alpha \, d\beta$$

- Para o caso 1-D, a convolução é definida como:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\alpha) g(x-\alpha) \, d\alpha$$

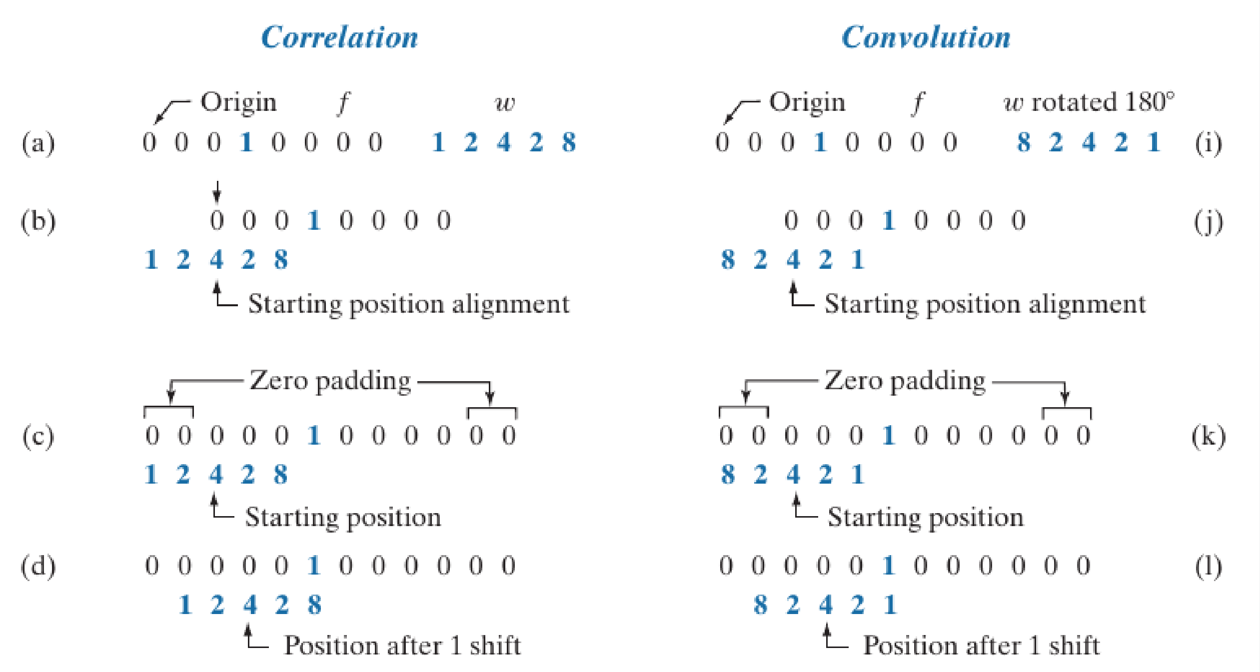
Correlação e Convolução



Fonte: Agostinho Brito Jr. (2018)

Correlação e Convolução

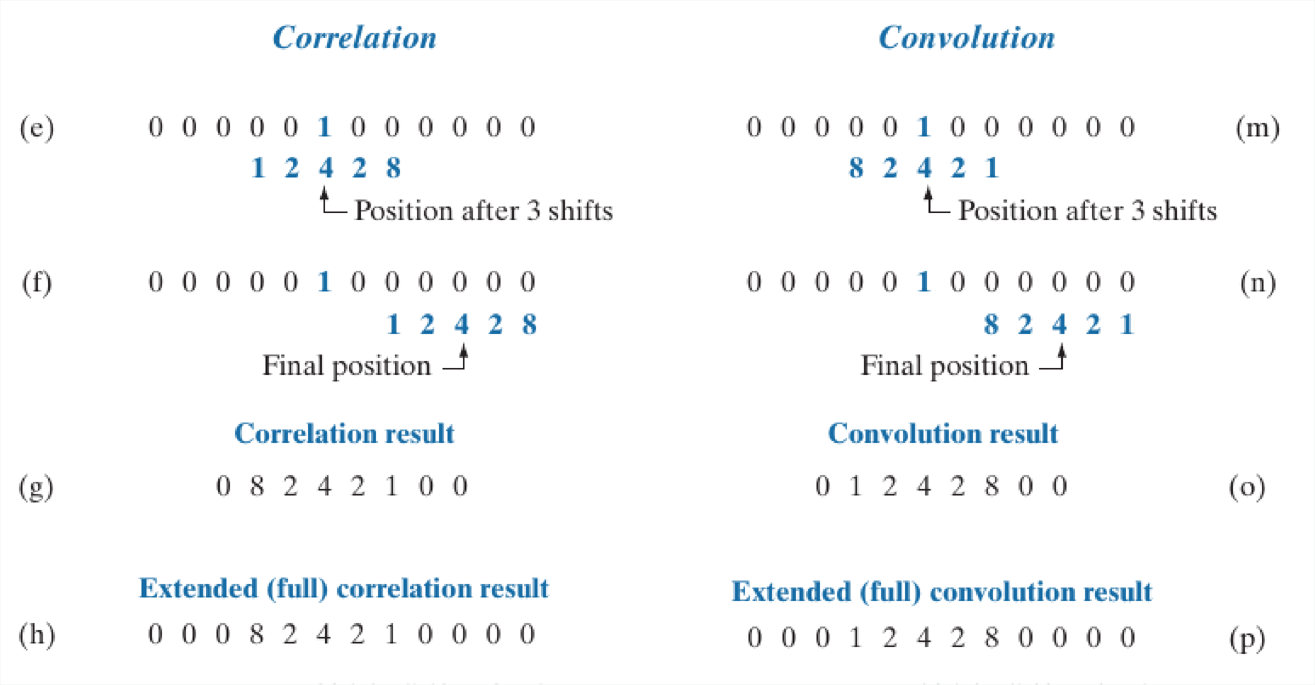
- O funcionamento de um filtro de kernel 1-D pode ser visto na figura abaixo:



Fonte: Gonzalez & Woods (2018).

Correlação e Convolução

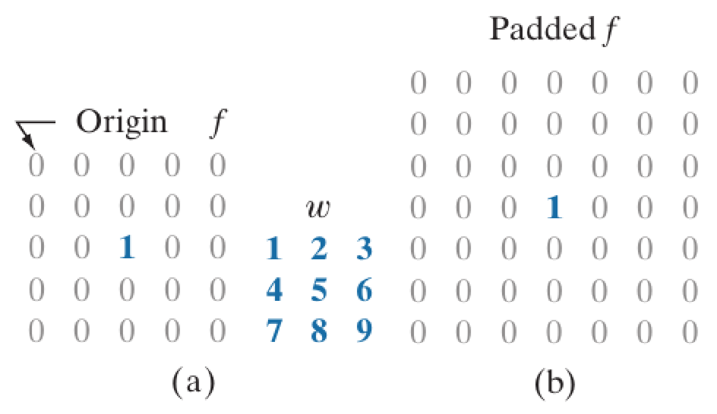
- O funcionamento de um filtro de kernel 1-D pode ser visto na figura abaixo (cont.):



Fonte: Gonzalez & Woods (2018).

Correlação e Convolução

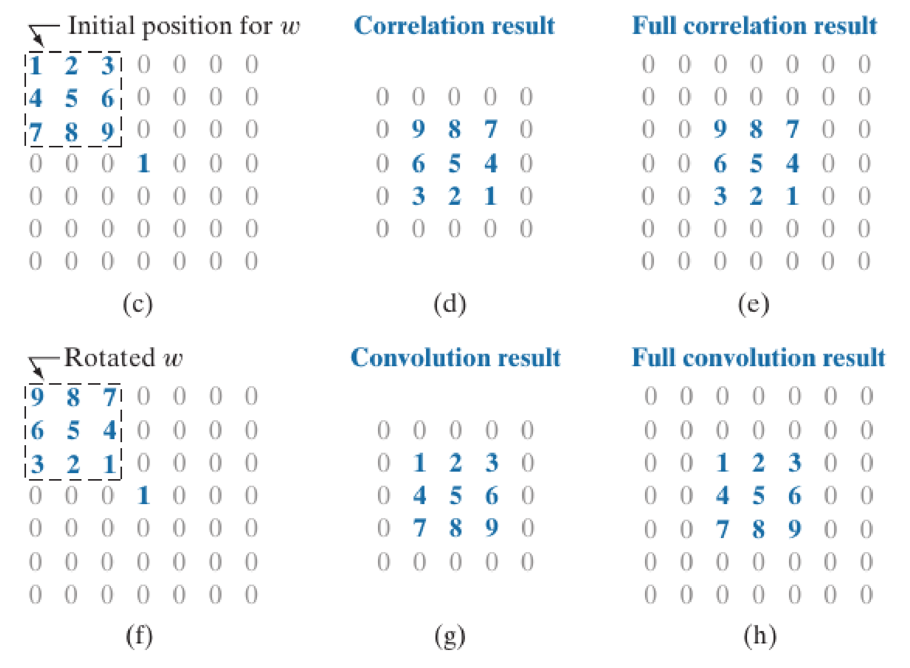
- Para imagens em duas dimensões, podemos visualizar os efeitos da aplicação das operações de correlação e convolução
  - Para isso, consideremos as imagem original abaixo e a respectiva imagem após aplicação do filtro de pad.



Fonte: Gonzalez & Woods (2018).

Correlação e Convolução

- Os resultados das operações de correlação e da convolução podem ser vistos na imagem abaixo:



Fonte: Gonzalez & Woods (2018).

Correlação e Convolução

- De forma genérica, podemos reescrever a equação de uma correlação como:

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

- De forma similar, a equação da convolução pode ser reescrita como:

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

Correlação e Convolução - Propriedades

- As seguintes propriedades podem ser aplicadas às operações de correlação e convolução:

Property	**Convolution**	**Correlation**
Commutative	$f \star g = g \star f$	—
Associative	$f \star (g \star h) = (f \star g) \star h$	—
Distributive	$f \star (g + h) = (f \star g) + (f \star h)$	$f \star (g + h) = (f \star g) + (f \star h)$

Fonte: Gonzalez & Woods (2018).

Correlação e Convolução

- Na prática, correlação e convolução são tratadas de maneira única, como uma mesma operação na qual um kernel desliza sobre uma imagem
  - Frequentemente, a palavra mais utilizada é convolução, mesmo que a operação seja efetivamente uma correlação;
  - Quando se trata do kernel, a expressão mais utilizada é kernel de convolução (*convolution kernel*), mesmo que a operação seja de correlação.
  - Redes neurais convolucionais ajudaram a popularizar ainda mais a expressão convolução, em detrimento da palavra correlação.

Filtro Suavizante (Filtro Passa Baixo)

Filtros Suavizantes

- **Suavização** (*smoothing*) ou **média** (*averaging*) é o filtro espacial usado para reduzir as transições abruptas de intensidade
  - Uma vez que o ruído aleatório constitui em transições abruptas na intensidade, a suavização pode ser utilizada também para redução de ruídos;
  - Filtros de suavização podem ser combinadas com outras técnicas, para melhoramento de imagens (ex.: processamento de histograma).

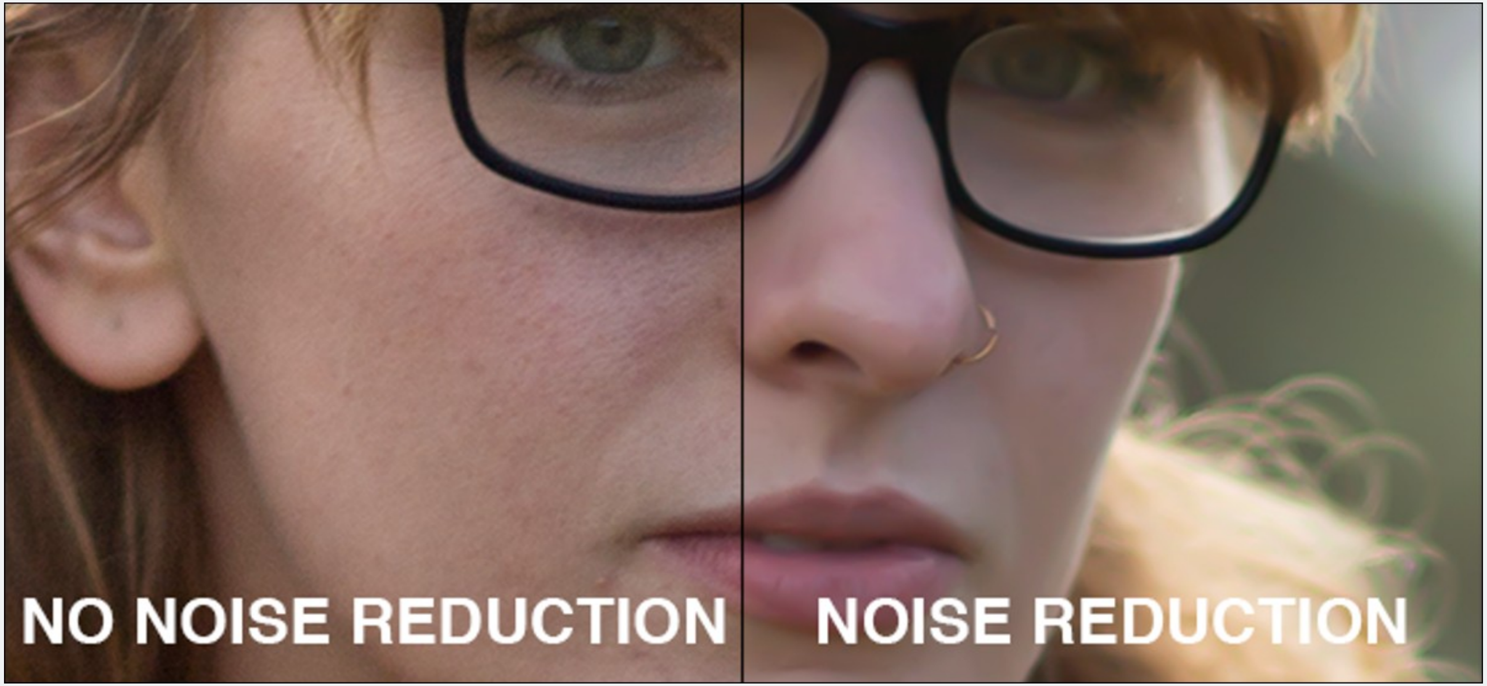
Filtros Suavizantes

Além da redução de ruídos, a **suavização** também pode ser usado para:

- Reduzir o serralhamento (ou *aliasing*), correspondente ao fenômeno que pode fazer que sinais fiquem indistinguiveis após a amostragem;
- Remover informações irrelevantes (pequenos pontos em relação ao *kernel*);
- Remover falsos contornos, oriundos do número insuficiente de níveis de intensidade;

Filtros Suavizantes

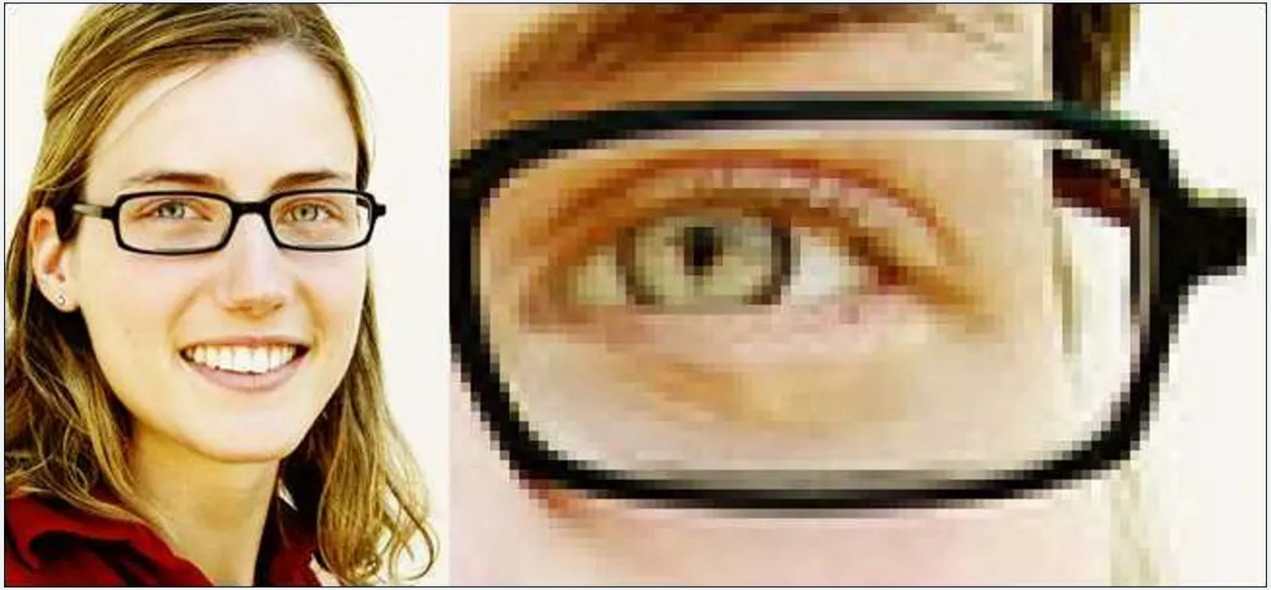
- A imagem abaixo contém um exemplo de redução de ruídos usando suavização.



Fonte: Guinness (2018).

Filtros Suavizantes

- A imagem abaixo contém um exemplo de aliasing (serralhamento), que pode ser melhorado com técnicas de suavização.



Fonte: Goodnight (2011).

Box Filter Kernels

- O **kernel de filtro passa-baixo** (*lowpass filter*) mais simples é o ***box kernel***
  - Nesse filtro todos os coeficientes possuem o mesmo valor (tipicamente 1);
- Um “filtro de caixa”(box kernel) de tamanho $m \times n$ é um array de 1’s, cujo valor é multiplicado por uma constante cujo valor corresponde a 1 dividido pela soma dos elementos no kernel.

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

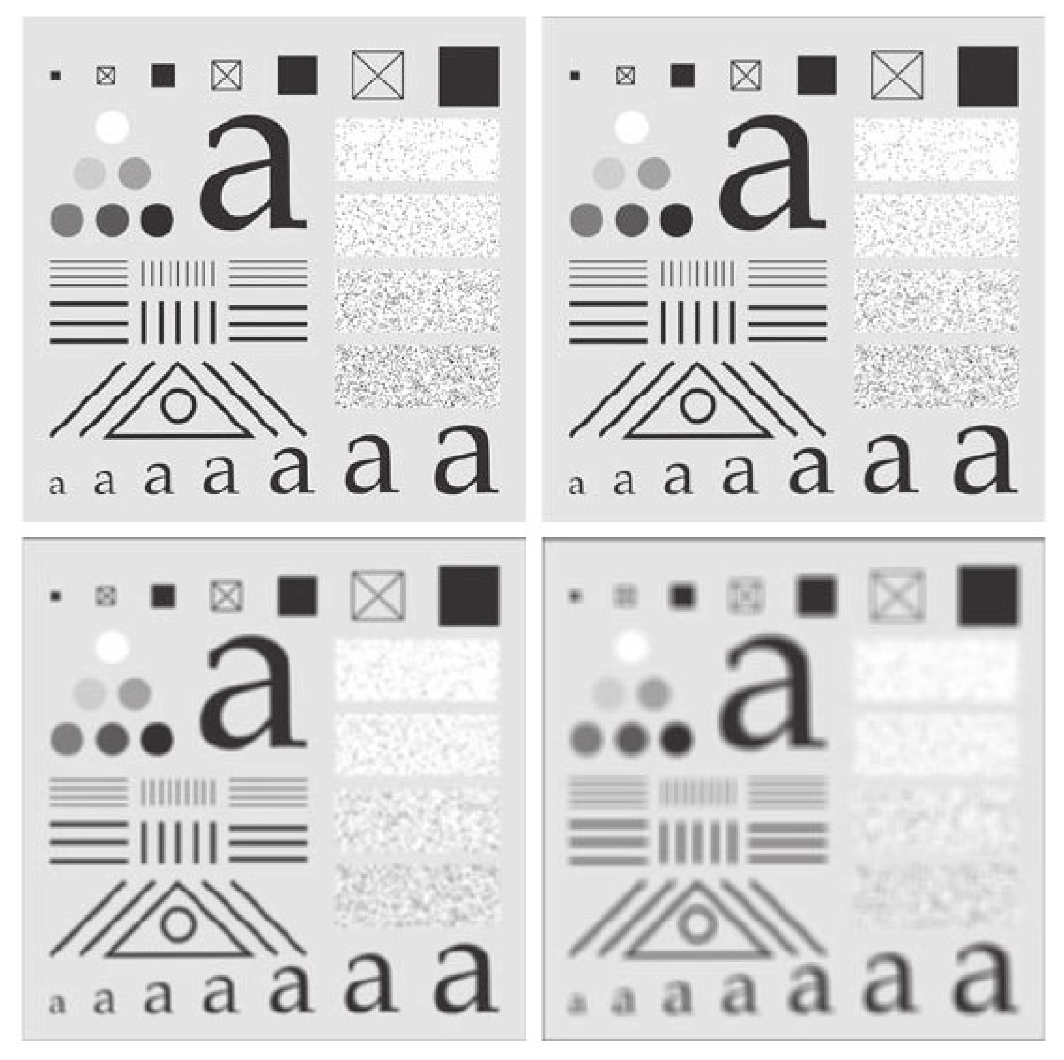
Fonte: TheAILearner (2019).

Box Filter Kernels

- A normalização aplicada a todos os kernels passa baixo possui dois propósitos:
  1. Obter o valor médio de uma área de intensidade constante igual à intensidade da imagem filtrada;
  2. Ao normalizar o kernel, previne-se a introdução de um viés durante a filtragem – a soma dos pixels na imagem original e na imagem filtrada serão os mesmos.

Box Filters Kernels

- A imagem ao lado contém um padrão de testes, de tamanho 1024×1024 e a aplicação de filtros do tipo box kernel com tamanhos 3×3, 11×11 e 21×21, respectivamente
  - Observa-se que quanto maior o filtro, mais a imagem parece borrada;
  - Podemos observar o surgimento de bordas pretas na imagem devido ao pad.



Fonte: Gonzalez & Woods (2018).

Box Filter Kernels

- Devido a sua simplicidade e limitação, os *box filters* são adequados apenas para experimentações rápidas
  - Esses filtros tendem a borrar as imagens após o processamento
    - O borrado ocorre principalmente de forma perpendicular, o que impacta bastante em componentes geométricos.

Gaussian Filter Kernels

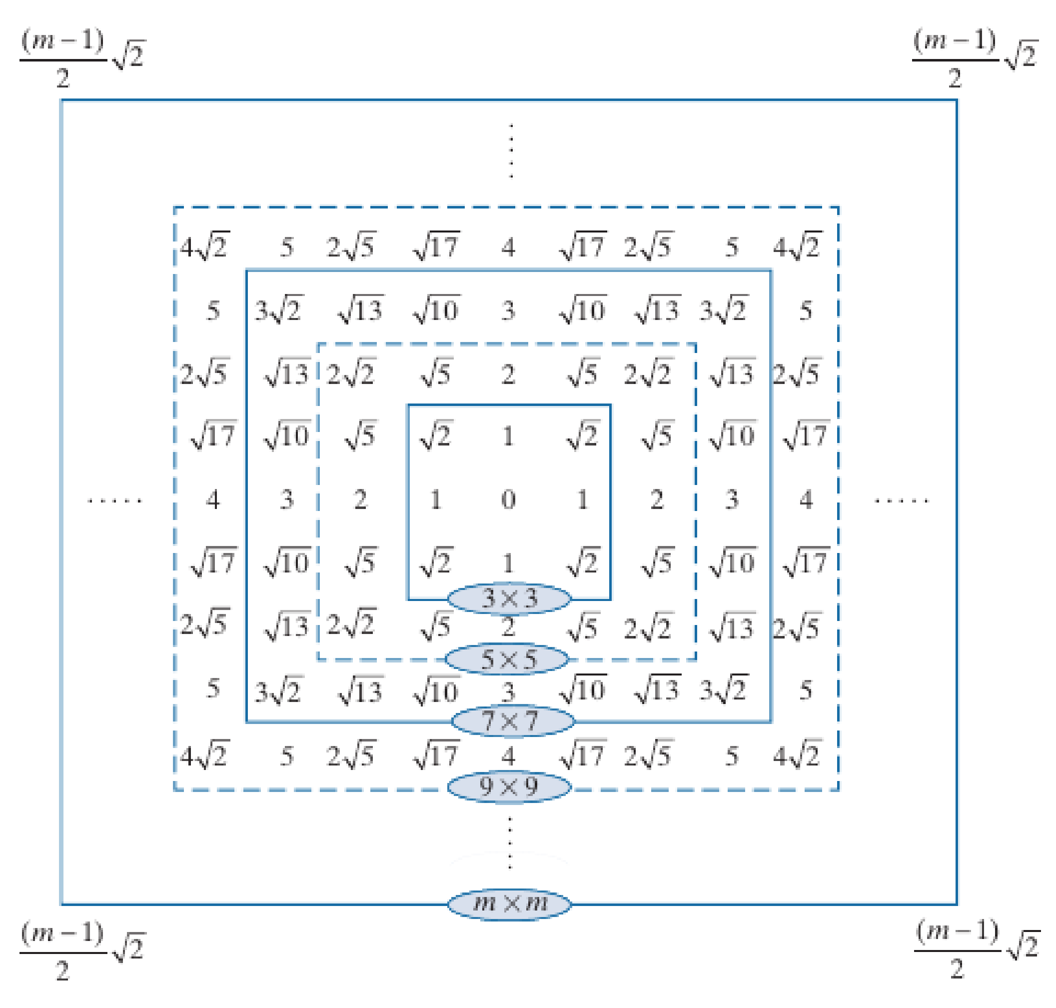
- **Filtros gaussianos** são aqueles que utilizam uma função gaussiana para definição dos valores
  - São utilizadas funções gaussianas em forma de sino;
  - O filtro é denominado frequentemente apenas de Gaussiano;
- Os filtros modificam o sinal por uma convolução com uma função gaussiana
- Os kernels gaussianos possuem a seguinte forma:

$$w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$$

  - onde, K corresponde à altura da curva, t corresponde ao valor esperado (centro da curva) e σ^2 é a variância.

Gaussian Filter Kernels

- A imagem ao lado mostra os valores para kernels gaussianos;
- Os valores do filtro podem ser normalizados
    - Tal abordagem possui o mesmo objetivo dos *box kernels*.



Fonte: Gonzalez & Woods (2018).

Gaussian Filter Kernels

- Filtros gaussianos devem ser muito maiores que filtros box kernels para produzir o mesmo efeito de borramento (caso este seja o objetivo).
    - A figura central possui kernel de tamanho 21×21 (com desvio padrão σ=3.5);
    - A figura à direita possui kernel de tamanho 43×43 (com desvio padrão σ=7).

Fonte: Gonzalez & Woods (2018).

Gaussian Filter Kernels

- Filtros gaussianos possuem ganho pequeno quando o kernel é maior que $|6\sigma|\times|6\sigma|$
  - Na imagem abaixo, podemos ver a aplicação de filtros gaussianos de tamanho 43×43 e 85×85, com $\sigma=7$;
  - A última imagem contém a diferença entre os filtros.

Fonte: Gonzalez & Woods (2018).

Gaussian Filter Kernels

- Filtros gaussianos podem ser utilizados para remoção de sombras em imagens
  - Um tabuleiro de xadrez de 2048×2048 (com retângulos de 128×128) passou por um filtro gaussiano de 512×512, gerando um padrão de sombra, subtraído da imagem original, gerando uma imagem com sombra reduzida.

Fonte: Gonzalez & Woods (2018).

Median Filter Kernels (Mediana)

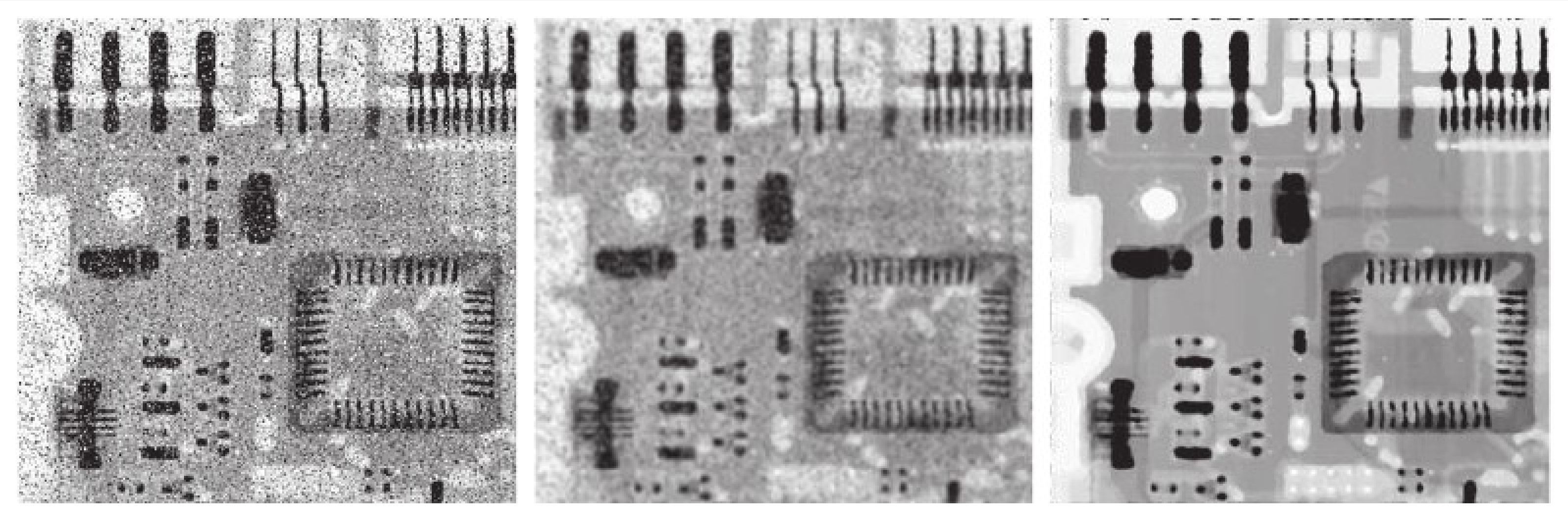
- O **filtro de mediana** (*median filter*) substitui o valor do pixel central pela mediana de valores de intensidade na vizinhança de cada pixel;
- Filtros de mediana são capazes de reduzir muitos tipos de ruídos aleatórios, borrando muito menos as imagens que os filtros lineares de mesmo tamanho;
  - Esses filtros são particularmente efetivos para ruídos do tipo salt-and-pepper (pontos pretos e brancos superimpostos em uma imagem);
- A mediana ξ é o conjunto de valores no qual metade dos pixels são maiores que ξ e a outra metade é menor que ξ;
  - Para cálculo da mediana, é necessário ordenar os pixels da vizinhança e, posteriormente, definir seu valor;

Median Filter Kernels (Mediana)

- Suponha um conjunto de valores de uma vizinhança 5×5:
  - Valores originais: (10, 20, 20, 20, 15, 20, 20, 25, 100);
  - Valores ordenados: (10, 15, 20, 20, 20, 20, 20, 25, 100);
  - Mediana: 20
- A mediana corresponde 50º percentil de uma imagem;
  - O 100º percentil corresponde aos pontos mais claros (*max-filter*), enquanto o 0º percentil corresponde aos pontos mais escuros (*min-filter*).

Median Filter Kernels (Mediana)

- A imagem original corresponde uma placa de circuito eletrônico com ruído do tipo salt-and-pepper
  - As demais imagens contém redução de ruído usando filtro gaussiano 19×19 (σ=3) e uma redução usando filtro de mediana 7×7.



Fonte: Gonzalez & Woods (2018).

Filtros Aguçantes – Filtros Passa Alto

Filtros Aguçantes

- **Filtros aguçantes** acentuam transições em intensidade
  - São utilizados para realçar detalhes finos presentes em uma imagem, ou realçar detalhes borrados por algum processo degradatório;
  - Enquanto filtros suavizantes podem ser associados com operações de integração (em intervalos discretos, somatórios), os filtros aguçantes podem ser associadas as operações de derivação;
  - A diferenciação realça as bordas e outras discontinuidades (como os ruídos) e reduz a enfase em áreas com baixa variação de intensidade;
- Os filtros aguçantes são associados aos **filtros de alta frequência**
  - O nome deriva da capacidade de permitir que detalhes finos (de alta frequência) possam passar pelo filtro enquanto baixas frequências são atenuadas ou removidas.

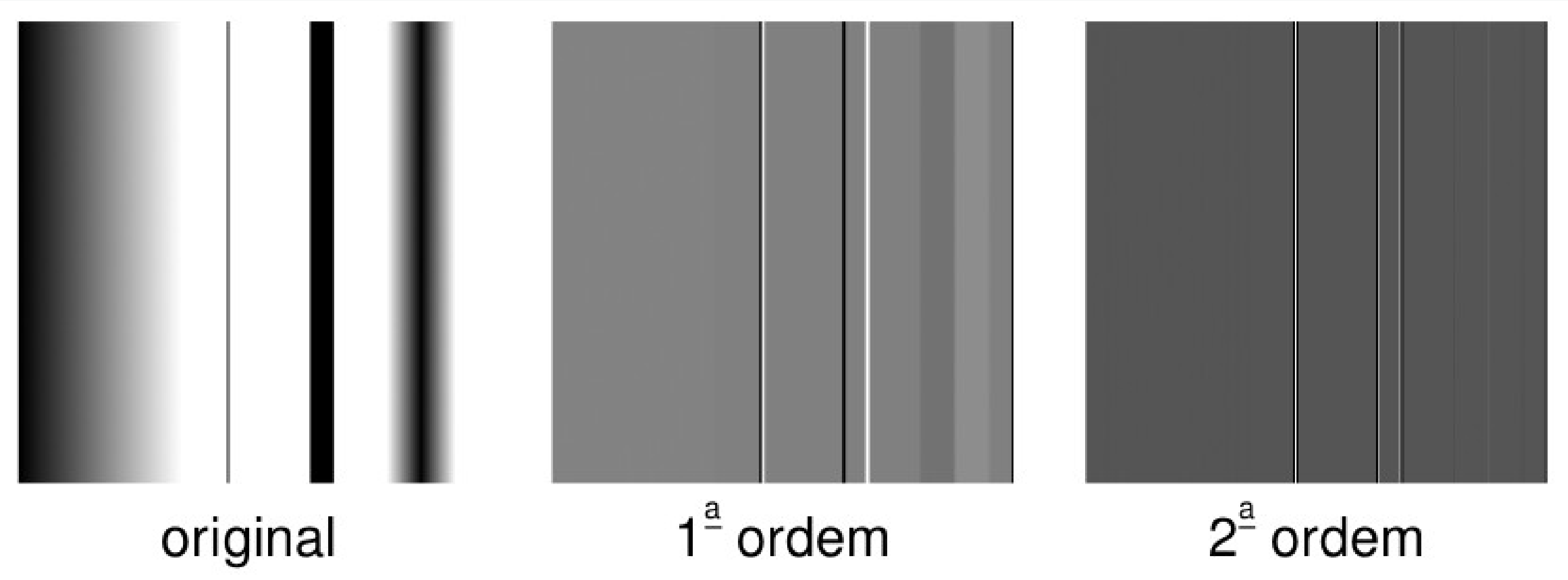
Filtros Aguçantes

- Nos filtros aguçantes, são usadas, em geral, as derivadas de 1ª e 2ª ordem:
  - As derivadas devem ser aplicadas nos eixos *x* e *y*;
  - Para consistência das notações, serão utilizadas apenas derivadas parciais na direção de x.

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Filtros Aguçantes

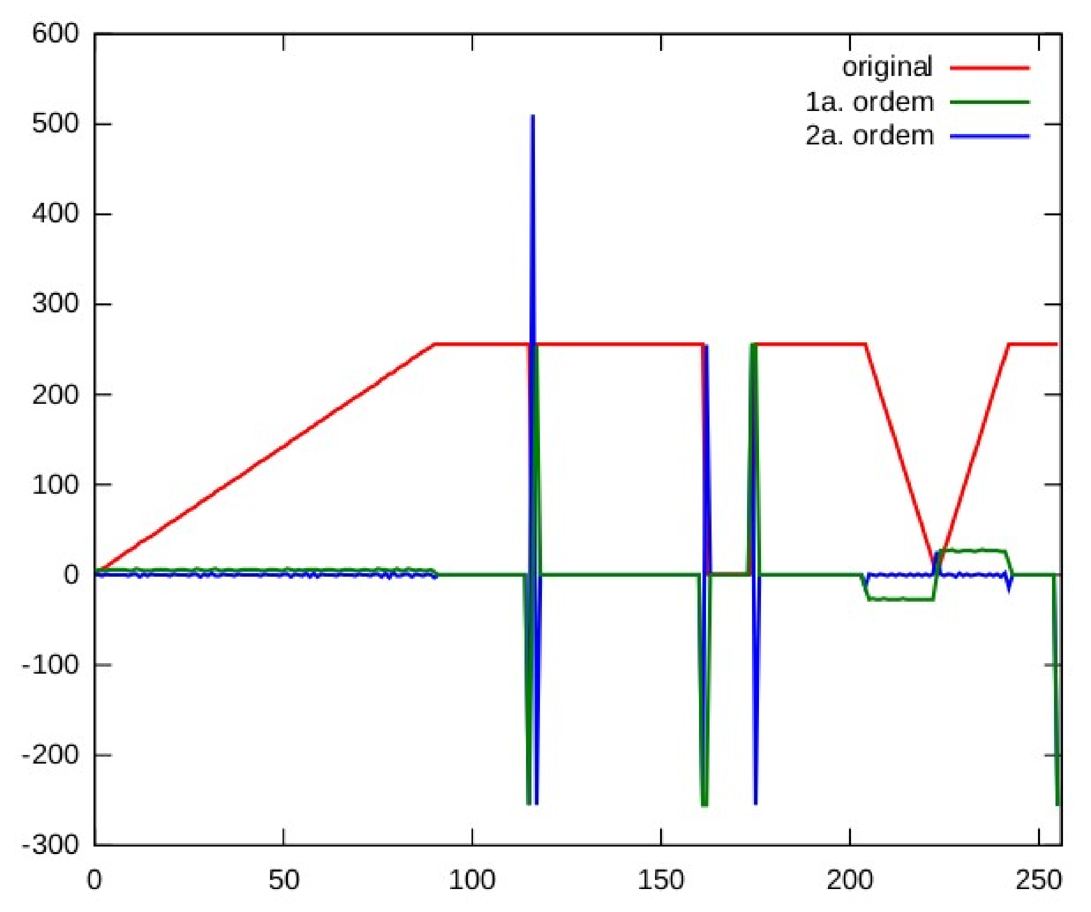
- Os resultados da aplicação das derivadas de 1ª e 2ª ordem podem ser vistas na imagem abaixo:



Fonte: Agostinho Brito Jr. (2018).

Filtros Aguçantes

- **Filtros de 1ª ordem**: possuem arestas mais largas, com resposta constante para rampas
  - São utilizadas para operações de melhoramento.
- **Filtros de 2ª ordem**: possuem arestas mais finas, com melhor resposta para ruídos,
  - Exibem resposta apenas nas transições, com alterações mais agressivas.



Fonte: Agostinho Brito Jr. (2018).

Filtro Laplaciano

- O **Filtro Laplaciano** utiliza um kernel baseado na derivada segunda
  - É denominado frequentemente apenas de **Laplaciano**;
  - Pode ser considerado um filtro isotrópico (invariante à rotação, assim como o filtro gaussiano);
- O filtro laplaciano tende a destacar transições abruptas de intensidade em imagens e remover a enfâse em regiões de baixa variação de intensidade
  - As imagens tendem a produzir linhas de bordas acinzentadas;
  - Descontinuidades tendem a ser superimpostas em um fundo mais escurecido;
  - O fundo pode ser recuperado com a adição do laplaciano à imagem original.

Filtro Laplaciano

- O filtro Laplaciano pode ser definido matematicamente como:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- O filtro Laplaciano pode ser aplicado às direções *x* e *y* com as seguintes equações:

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= f(x+1) + f(x-1) - 2f(x) \\ \frac{\partial^2 f}{\partial y^2} &= f(y+1) + f(y-1) - 2f(y) \end{aligned}$$

- Somando as componentes em ambas as direções, temos:

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Filtro Laplaciano

- O filtro Laplaciano pode ser facilmente implementado pelas seguintes máscaras:

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

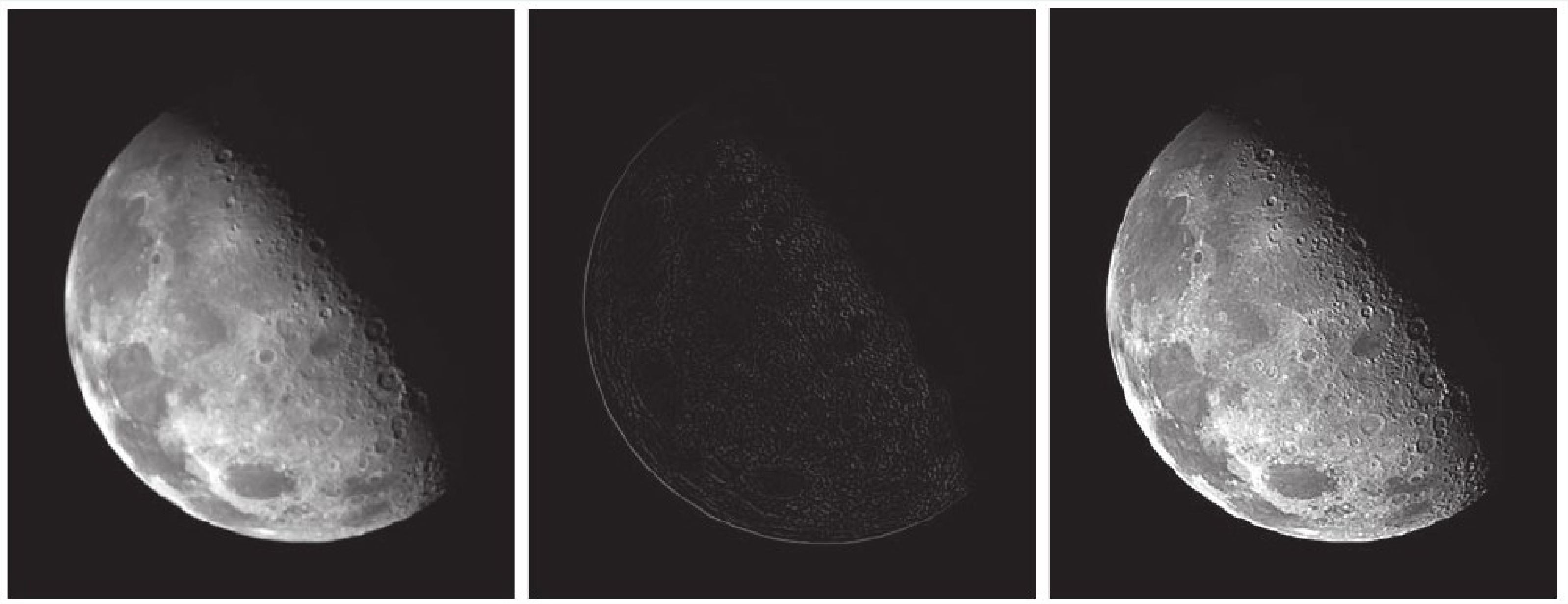
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Fonte: Gonzalez & Woods (2018).

Filtro Laplaciano

- A imagem abaixo contém o pólo norte da lua e a aplicação dos filtros laplacianos 1 e 2 do slide anterior, a ela aplicados.



Fonte: Gonzalez & Woods (2018).

Filtro Gradiente

- O **Filtro Gradiente** utiliza um kernel baseado na derivada primeira
    - O gradiente aponta na direção de maior crescimento (ou decrescimento) de uma função matemática em um ponto *(x, y)*;
    - Pode ser considerado um filtro isotrópico (invariante à rotação, assim como o filtro gaussiano);
    - As derivadas primeiras são implementadas com base na magnitude do gradiente;
- O filtro é utilizado para inspeção industrial, em passos de pré-processamento, uma vez que é menos sensível a ruídos que filtros de segunda ordem (Brito Jr., 2018).

Filtro Gradiente

- O gradiente utiliza um kernel baseado na derivada primeira
  - O gradiente de uma imagem *f* nas coordenadas *(x, y)* é definido como um vetor coluna bidimensional, dado por:

$$\nabla f \equiv \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

  - A magnitude (tamanho) do gradiente do vetor gradiente é definida por:

$$M(x, y) = \|\nabla f\| = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

    - onde o valor em *(x, y)* é a taxa de mudança de direção no gradiente

Filtro Gradiente

- Os filtros de gradiente podem ser implementados por máscaras;
  - Dentre as máscaras, destacam-se os operadores de **cross-gradientes de Roberts** (filtro 2×2) e os **operadores Sobel** (filtro 3×3).

−1	0
0	1

0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Fonte: Adaptado de Gonzalez & Woods (2018).

Filtro Gradiente

- A aplicação dos operadores do filtro de Sobel e do operador de gradiente pode ser visto na imagem abaixo:

Fonte: Agostinho Brito Jr. (2018).

Filtro Highboost

- O **Filtro Highboost**, usado desde a década de 1930, busca subtrair uma versão suavizada da imagem original para realçar bordas e aumento da nitidez em imagens
- O processo consiste nas seguintes etapas:
  1. Borrar (desfocar) a imagem original;
  2. Subtrair a imagem borrada do original (a diferença é chamada de máscara);
  3. Adicionar a máscara à imagem original, produzindo uma nova imagem.
- O filtro pode ser implementado com o uso de máscaras.

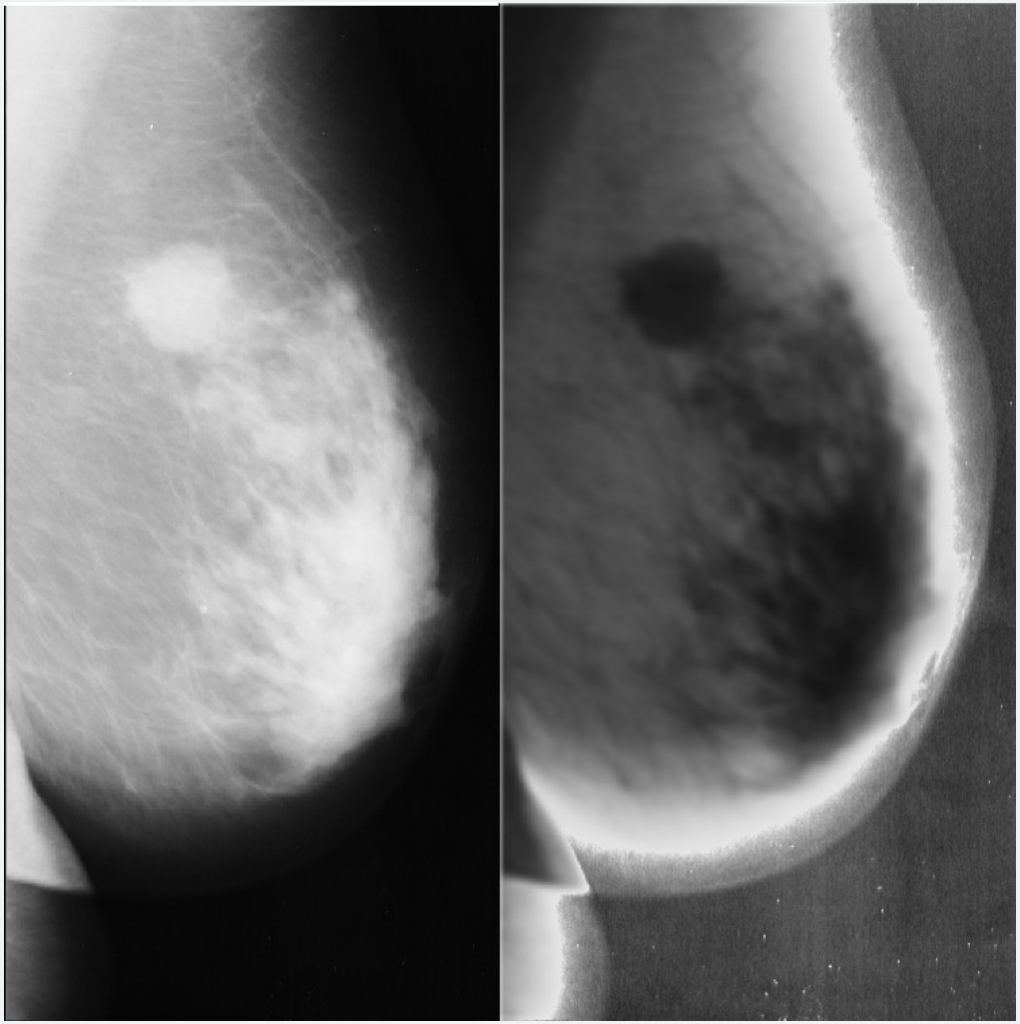
0	-1	0
-1	A+4	-1
0	-1	0

-1	-1	-1
-1	A+8	-1
-1	-1	-1

Fonte: Agostinho Brito Jr. (2018).

Filtro Highboost

- A aplicação do filtro highboost pode ser visto na imagem ao lado
  - O filtro foi aplicado para melhoramento de microcalcificações em imagens de mamografia utilizando raio-x, com objetivo de detecção do câncer de mama.
  - A imagem teve ainda as cores dos pixels invertidas, para melhor visualização.



Fonte: Ankur Jaiswal (2023).

Referências

Referências

- Rafael C. Gonzalez, Richard E. Woods. **Digital Image Processing – 4th Edition.** 2018. Pearson. ISBN: 978-9353062989.
- Agostinho Brito Jr. **Processamento digital de imagens – Slides de Aula.** 2018.
- Anh H. Reynolds. Convolutional Neural Networks (CNNs). 2019. Disponível em: https://anhreynolds.com/blogs/cnn.html
- Harry Guinness. **What Is Noise Reduction in Digital Images?**. 2018. Disponível em: https://www.howtogeek.com/368550/what-is-noise-reduction-in-digital-images/

Referências

- Eric Z. Goodnight. **What Is Anti-Aliasing, and How Does It Affect My Photos and Images?**. 2011. Disponível em: https://www.howtogeek.com/73704/what-is-anti-aliasing-and-how-does-it-affect-my-photos-and-images/
- TheAILearner. **Smoothing filters.** 2019. Disponível em: https://theailearner.com/tag/box-filter/
- Ankur Kumar Jaiswal (MATLAB). **High Frequency Boost Filtering**. 2023. Disponível em: https://www.mathworks.com/matlabcentral/fileexchange/102384-high-frequency-boost-filtering