SEL 0449 - Processamento Digital de Imagens Médicas

SEL 5895 – Introdução ao Processamento Digital de Imagens

Aula 4
Processamento Espacial – Parte 2

Prof. Dr. Marcelo Andrade da Costa Vieira
mvieira@sc.usp.br

EESC/USP

Processamento Espacial
Parte 2

- Transformações ponto a ponto
  - Histograma
  - Transformações lineares
  - Transformações não-lineares
- Transformações por vizinhança
  - Convolução
  - Filtros lineares
  - Máscara de nitidez

Transformações Por Vizinhança

Operadores Locais (Vizinhança).

Combina a Intensidade de um certo número de píxels (janela), para computar o valor da nova intensidade na Imagem de Saída.

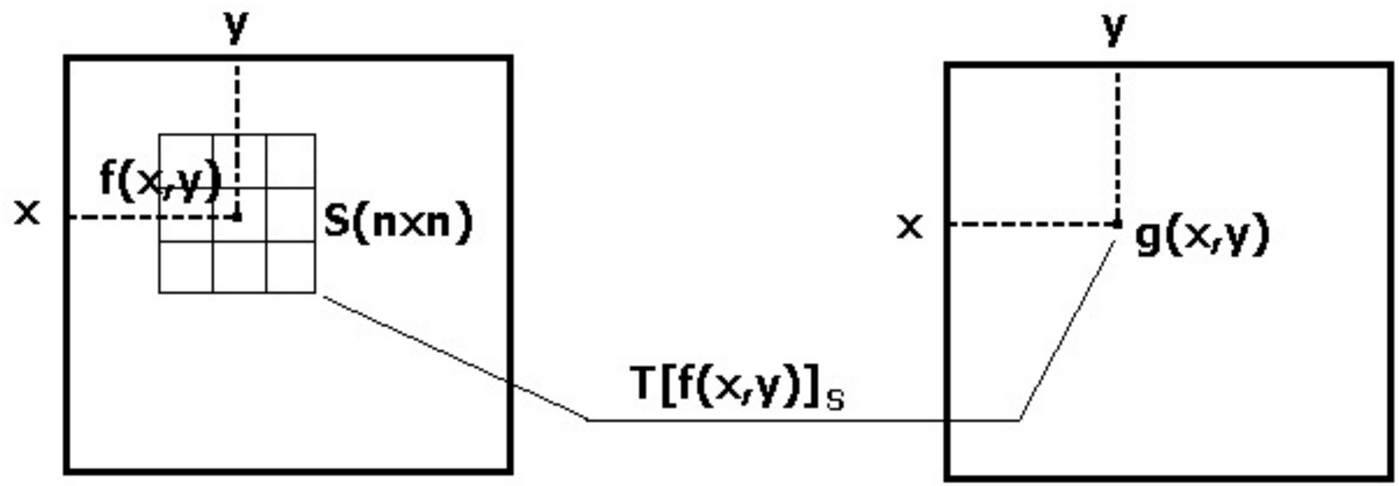


Imagem de Entrada

Imagem de Saída

$T[f(x,y)]_S$ ==> Operação sobre todos os píxels dentro da janela S centrada em f(x,y)

Equalização local de histograma

- Em alguns casos, a equalização global do histograma não é capaz de realçar objetos de interesse em uma imagem
- Isso ocorre quando, estatisticamente, o número de pixels dessas regiões é relativamente pequeno.
- Uma solução é realizar a equalização local do histograma, pixel a pixel, considerando pequenas regiões ao redor do pixel a ser processado.

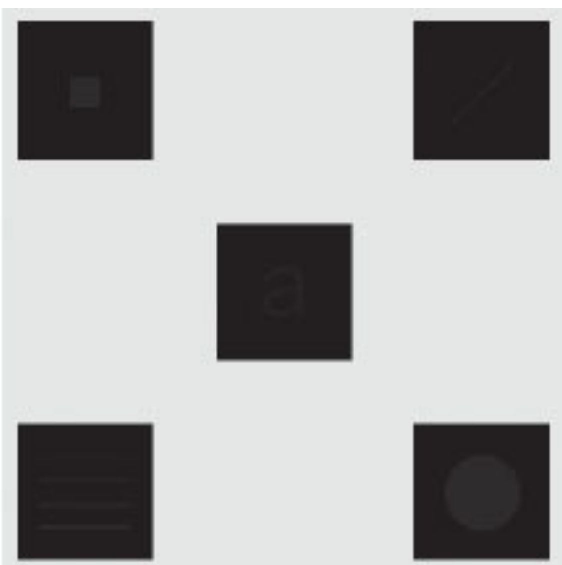
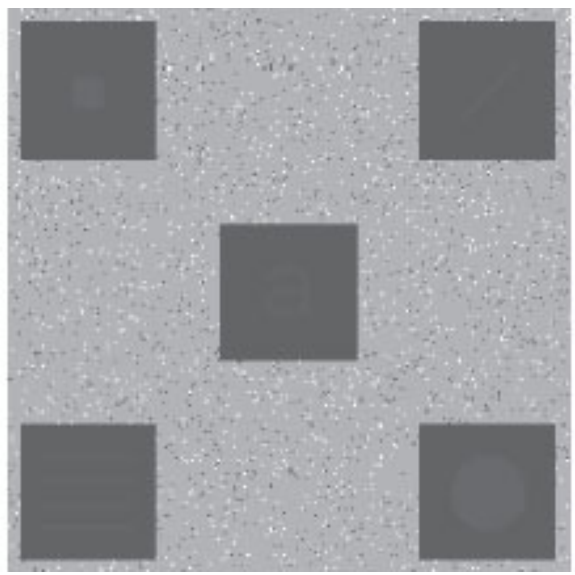
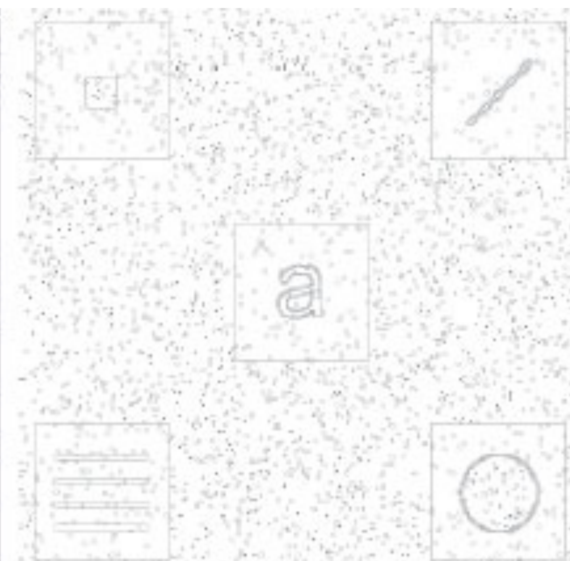


Imagem original



Equalização global



Equalização Local (3x3)

Convolução e Correlação Cruzada

Convolução →

$$f(x) * h(x) = \int_{-\infty}^{+\infty} f(m)h(x-m)\,dm$$

Correlação Cruzada →

$$f(x) \star h(x) = \int_{-\infty}^{+\infty} f(m)h(x+m)\,dm$$

Convolução e Correlação Cruzada
Forma DISCRETA

Convolução →

$$f[x]*h[x]=\sum_{m=-\infty}^{\infty} f[m]h[x-m]$$

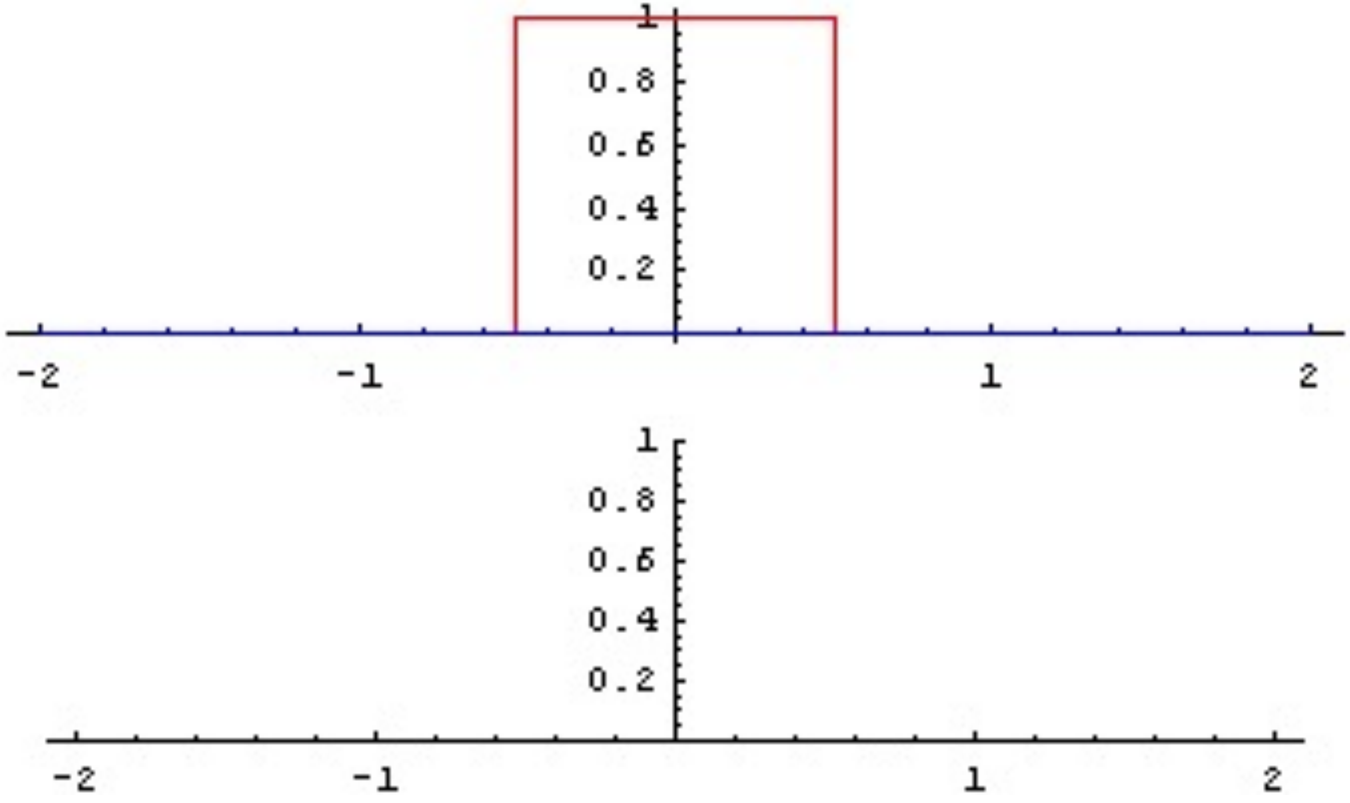
Correlação Cruzada →

$$f[x]\star h[x]=\sum_{m=-\infty}^{\infty} f[m]h[x+m]$$

1D

	Correlation	Convolution	
	Origin f w	Origin f w rotated 180°	
(a)	0 0 0 1 0 0 0 0 1 2 4 2 8	0 0 0 1 0 0 0 0 8 2 4 2 1	(i)
(b)	0 0 0 1 0 0 0 0 1 2 4 2 8 Starting position alignment	0 0 0 1 0 0 0 0 8 2 4 2 1 Starting position alignment	(j)
(c)	Zero padding 0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Starting position	Zero padding 0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Starting position	(k)
(d)	0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Position after 1 shift	0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Position after 1 shift	(l)
(e)	0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Position after 3 shifts	0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Position after 3 shifts	(m)
(f)	0 0 0 0 0 1 0 0 0 0 0 0 1 2 4 2 8 Final position	0 0 0 0 0 1 0 0 0 0 0 0 8 2 4 2 1 Final position	(n)
	Correlation result	**Convolution result**	
(g)	0 8 2 4 2 1 0 0	0 1 2 4 2 8 0 0	(o)
	Extended (full) correlation result	**Extended (full) convolution result**	
(h)	0 0 0 8 2 4 2 1 0 0 0 0	0 0 0 1 2 4 2 8 0 0 0 0	(p)

Convolução e Correlação Cruzada 1D



2D

Origin f

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

w

1	2	3
4	5	6
7	8	9

(a)

Padded f

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(b)

Initial position for w

1	2	3	0	0	0	0
4	5	6	0	0	0	0
7	8	9	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(c)

Correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

(d)

Full correlation result

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	9	8	7	0	0
0	0	6	5	4	0	0
0	0	3	2	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(e)

Rotated w

9	8	7	0	0	0	0
6	5	4	0	0	0	0
3	2	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(f)

Convolution result

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

(g)

Full convolution result

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	2	3	0	0
0	0	4	5	6	0	0
0	0	7	8	9	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

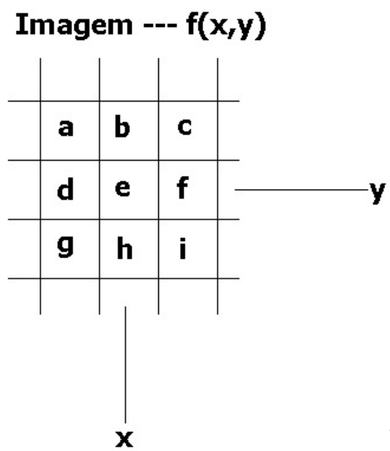
(h)

Propriedades da Convolução e Correlação

Property	Convolution	Correlation
Commutative	$f \star g = g \star f$	—
Associative	$f \star (g \star h) = (f \star g) \star h$	—
Distributive	$f \star (g + h) = (f \star g) + (f \star h)$	$f \mathbin{☆} (g + h) = (f \mathbin{☆} g) + (f \mathbin{☆} h)$

Filtros no domínio do espaço

Exemplo: Janela de 3 x 3



Template
k = 3 x 3 = 9

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$g(x,y) = \sum_{i=1}^{k} w_i . f(x,y)$$

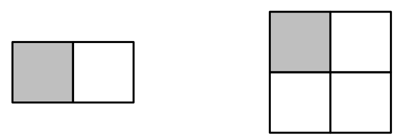
- $(a, b, c, d, e, f, g, h, i)$: valores dos níveis de cinza na vizinhança de $f(x,y) = e$;
- w_1 a w_9: são os "pesos" , ou seja, os valores dos níveis de cinza em cada posição do *Template*.

O valor do pixel $g(x,y)$, na nova imagem, será dado por:

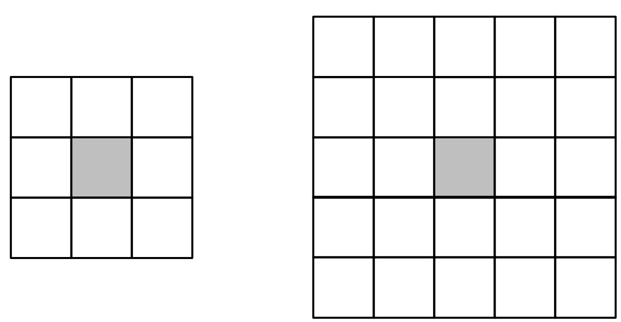
$$g(x,y) = w_1.a + w_2.b + w_3.c + w_4.d + w_5.e + w_6.f + w_7.g + w_8.h + w_9.i$$

Convenção:

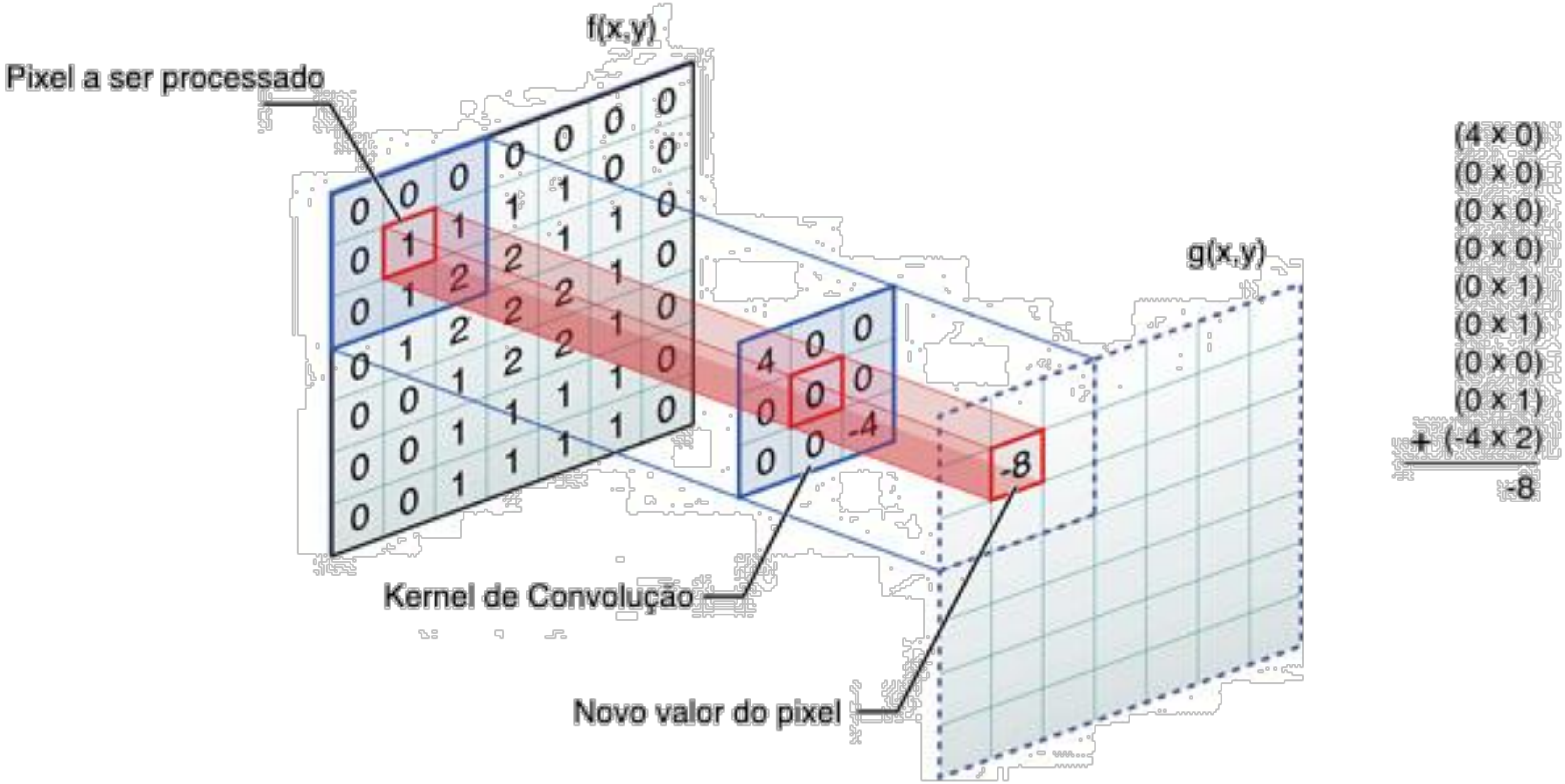
- Máscaras de organização par ($2x2$, $4x4$, ...) o resultado é colocado sobre o **primeiro pixel**.



- Máscaras de organização ímpar ($3x3$, $5x5$, ...) o resultado é colocado sobre o **pixel central**.



Convolução



f(x,y)
Pixel a ser processado
g(x,y)
Kernel de Convolução
Novo valor do pixel
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8

Convolução e Correlação Cruzada:

• No domínio do espaço, a diferença entre a Convolução e a Correlação Cruzada reside apenas no espelhamento do Template a ser utilizado, que deve ser feito na Convolução.

• Como, em geral, os Templates são simétricos, a equação da Correlação Cruzada tem sido empregada com o nome de Convolução na área de Processamento de Imagens.

Convoluir um Template com uma Imagem equivale à operação:

Espelhamento, Desloca, Multiplica e Soma

Exemplo de máscara simétrica, onde a operação de convolução e de correlação são idênticas:

Template

1	0
0	1

T(i,j)

Imagem Original

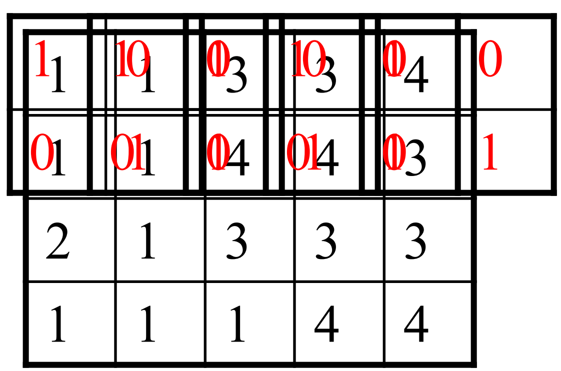
1	1	3	3	4
1	1	4	4	3
2	1	3	3	3
1	1	1	4	4

f(x,y)

Imagem Final

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

T(i,j) * f(x,y)



Os valores marcados com * não podem ser calculados.

Solução para os pixels das bordas:

Podem ser usadas quatro soluções:

1. Preenchimento da imagem com qualquer valor (geralmente zeros) antes do cálculo da imagem final (*padding*) (*X**);
2. Espelhamento dos pixels das bordas (*symmetric**);
3. Replicação dos pixels das bordas (*replicate**);
4. Convolução periódica ou circular (*circular**);

** Função usada pelo Matlab*

Solução para os pixels das bordas:

Para o **OpenCV - Phyton:**

Enumerator	
BORDER_CONSTANT Python: cv.BORDER_CONSTANT	`iiiiii|abcdefgh|iiiiiii` with some specified `i`
BORDER_REPLICATE Python: cv.BORDER_REPLICATE	`aaaaaa|abcdefgh|hhhhhhh`
BORDER_REFLECT Python: cv.BORDER_REFLECT	`fedcba|abcdefgh|hgfedcb`
BORDER_WRAP Python: cv.BORDER_WRAP	`cdefgh|abcdefgh|abcdefg`

Exemplo 1: *Padding* com zeros

Atribuir o valor 0 aos pixels inexistentes fora das bordas da imagem, antes da convolução.

Template

1	1	1
0	0	0
1	1	1

Imagem

1	2	3	4	5
0	1	3	4	0
1	1	3	2	0
0	0	4	5	6
1	0	7	8	0

Resultado

1	4	8	7	4
5	11	15	17	11
1	8	17	22	15
3	13	21	20	10
0	4	9	15	11

Primeiro Ponto ==> (1x0) + (1x0) + (1x0) + (0x0) + (0x1) + (0x2) + (1x0) + (1x0) +(1x1) = 1

Exemplo 2: “Espelhamento” dos pixels das bordas

Refletir os valores das bordas para preencher os pixels inexistentes fora das bordas da imagem, antes da convolução.

Template

1	1	1
0	0	0
1	1	1

Imagem

1	2	3	4	5
0	1	3	4	0
1	1	3	2	0
0	0	4	5	6
1	0	7	8	0

Resultado

5	10	17	19	18
7	11	15	17	16
1	8	17	22	21
5	13	21	20	10
2	12	24	30	25

Primeiro Ponto ==> (1x1) + (1x1) + (1x2) + (0x1) + (0x1) + (0x2) + (1x0) + (1x0) +(1x1) = 5

Exemplo 3: “Replicação” dos pixels das bordas

Copiar os valores da borda mais externa para preencher os pixels inexistentes fora das bordas da imagem, antes da convolução.

Template

1	1	1
0	0	0
1	1	1

Imagem

1	2	3	4	5
0	1	3	4	0
1	1	3	2	0
0	0	4	5	6
1	0	7	8	0

Resultado

5	10	17	19	18
7	11	15	17	16
1	8	17	22	21
5	13	21	20	10
2	12	24	30	25

Primeiro Ponto ==> (1x1) + (1x1) + (1x2) + (0x1) + (0x1) + (0x2) + (1x0) + (1x0) +(1x1) = 5

Exemplo 4: Convolução Periódica ou Circular

Preencher os pixels inexistentes fora das bordas da imagem considerando que ela é circular.

Template

1	1	1
0	0	0
1	1	1

Imagem

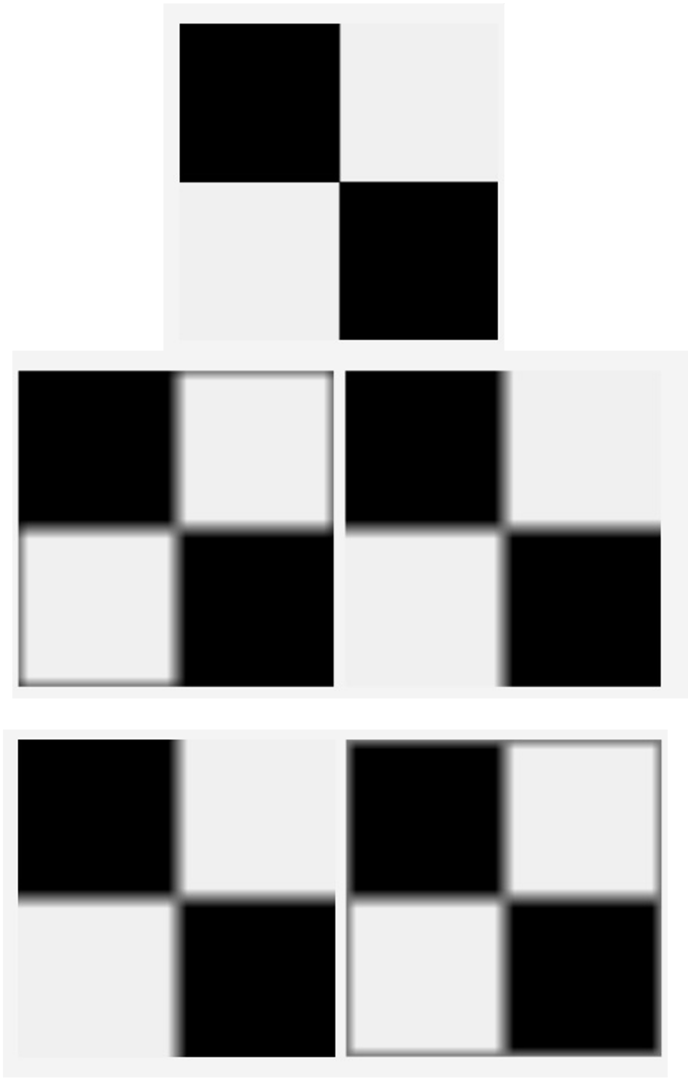
1	2	3	4	5
0	1	3	4	0
1	1	3	2	0
0	0	4	5	6
1	0	7	8	0

Resultado

2	12	23	22	13
10	11	15	17	13
7	8	17	22	15
3	13	21	20	12
14	10	18	27	21

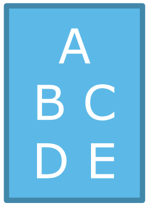
Primeiro Ponto ==> (1x0) + (1x1) + (1x0) + (0x5) + (0x1) + (0x2) + (1x0) + (1x0) +(1x1) = 2

Efeitos nas bordas da imagem



Convolução da imagem original com um filtro da média

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{3\times3}$$



A) Imagem original
B) Padding com zeros
C) Espelhamento
D) Replicação
E) Periódica (circular)

Observações:

- O custo computacional da Convolução espacial é alto.
- Se a Imagem é de tamanho M x M e o Template N x N, o número de multiplicações é de $M^2.N^2$
- Ou seja, se a Imagem é de 512 x 512 e o Template é de 16 x 16 , são necessárias 67.108.864 multiplicações.
- A alternativa é transformar a Imagem e o Template para o domínio da frequência (Fourier) e multiplicar elemento a elemento.

Filtragem Espacial

- Filtros Passa-Baixa

- Filtros Passa-Alta

Altas e baixas frequências em uma imagem

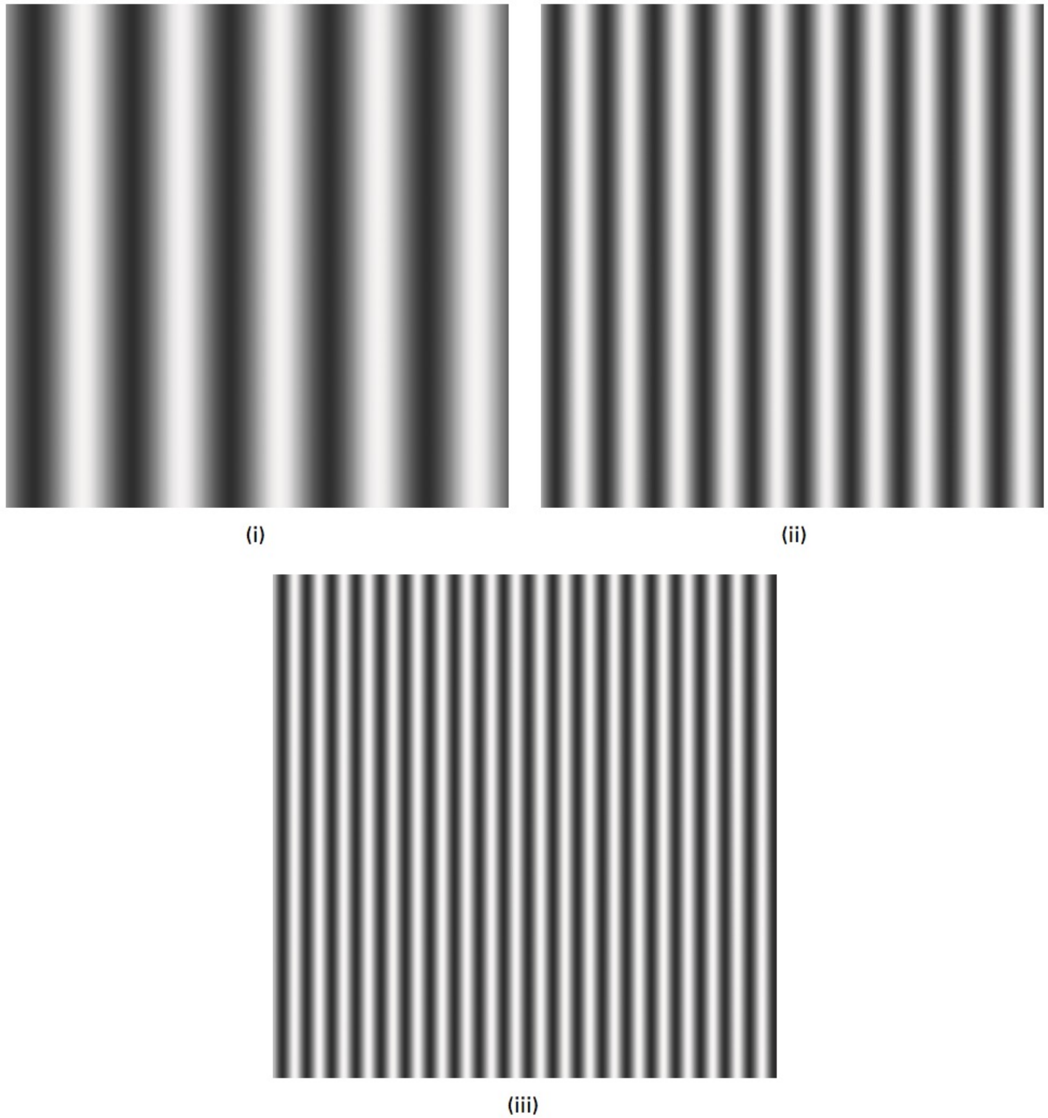
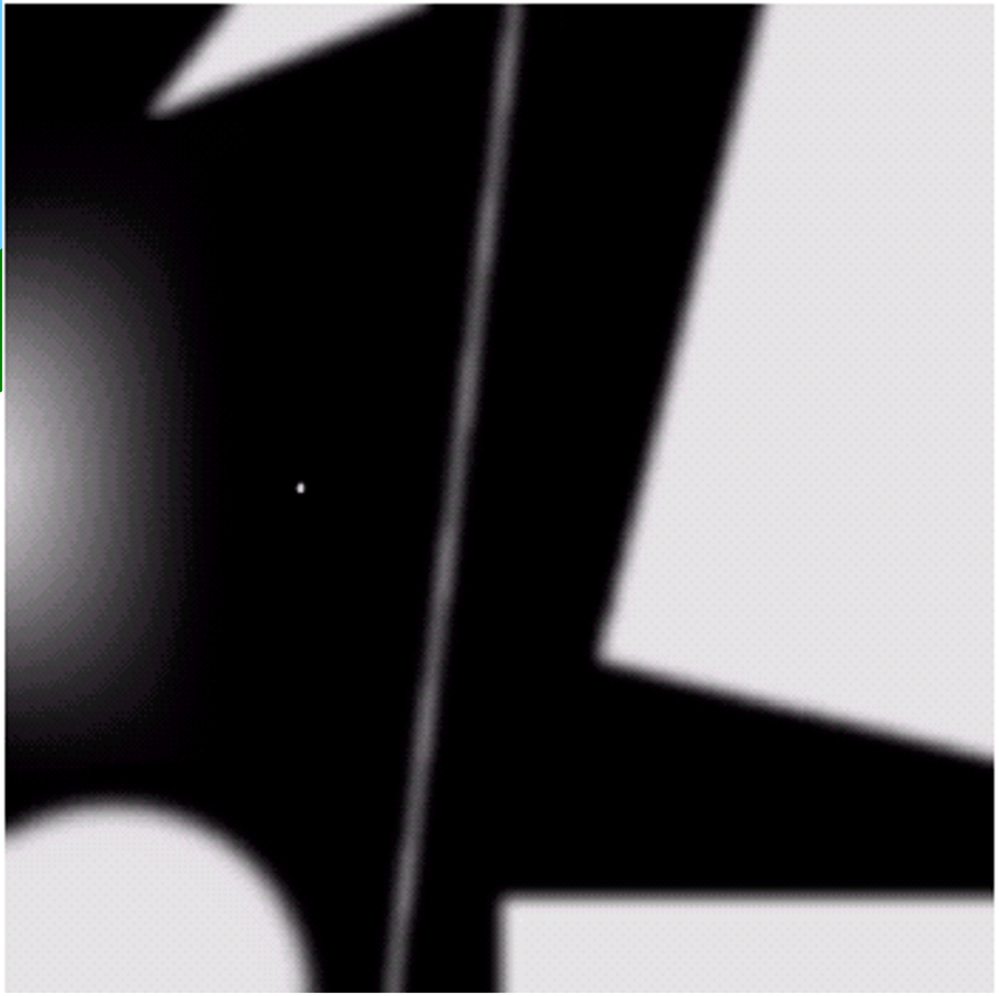
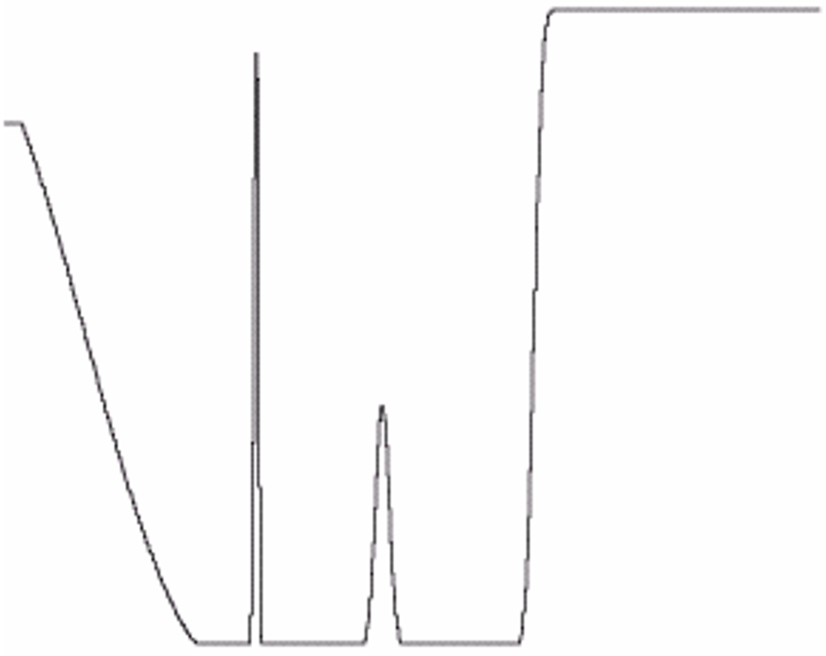


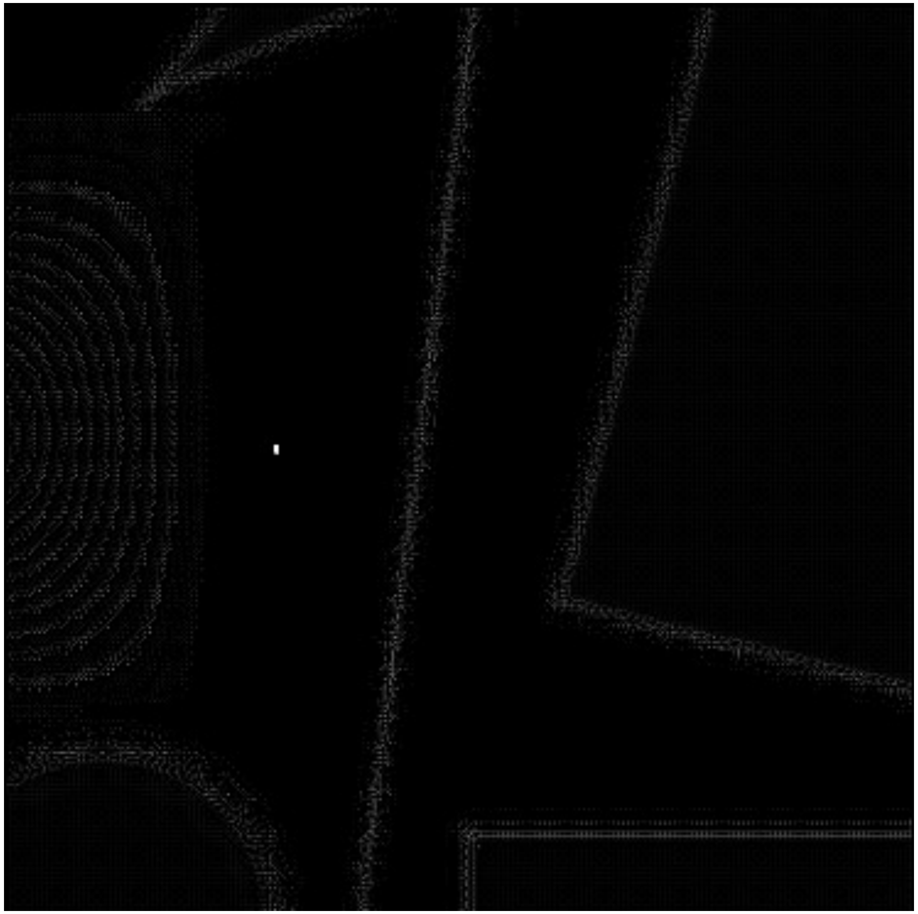
Figure 2.8 Sinusoidal patterns with (i) low, (ii) medium and (iii) high spatial frequency in the horizontal direction.

Perfil radiométrico de uma imagem: altas frequências e baixas frequências





Passa-Baixa



Passa-Alta

Filtragem Espacial: Passa Baixa

- Uma das aplicações da Convolução espacial de uma Imagem com Templates é a Suavização (Smoothing) ou Filtragem Passa Baixa.

- Um filtro espacial Passa-Baixa é implementado através de uma Máscara que realiza a Média da Vizinhança.

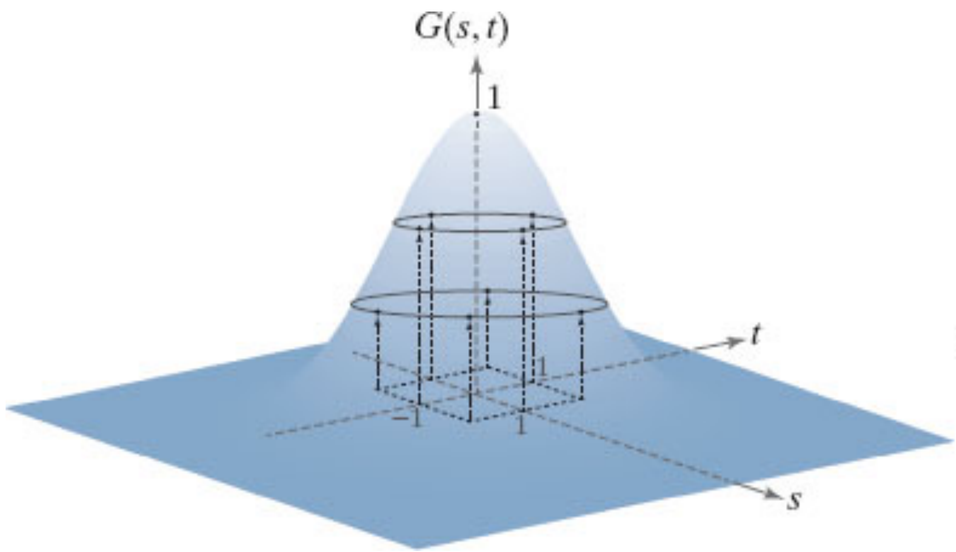
- Um *Kernel* Passa-Baixa é tal que seus pesos são positivos e a soma é igual a 1.

➢ Exemplos de algumas Máscaras de Filtros Passa Baixa:

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

Filtragem Espacial: Passa Baixa

- Filtro Gaussiano (Gaussian Blur)



$$\frac{1}{4.8976}\times\begin{array}{|c|c|c|}\hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline\end{array}$$

Filtragem Espacial: Média da vizinhança

f(x,y)

20	30	24	34	60	80	89	90	12	00
23	24	56	67	88	99	00	00	00	00
12	23	35	65	66	77	88	99	00	00
11	22	99	99	99	99	99	98	88	88
12	12	12	22	22	44	55	65	77	88
11	44	55	76	87	55	66	33	33	33
12	33	44	55	66	77	88	00	00	00

g(x,y)

	25	40							

$$g(0,0) = (20 + 30 + 24 + 23 + 24 + 56 + 12 + 23 + 35) / 9 = 24,77$$

$$g(0,1) = (30 + 24 + 34 + 24 + 56 + 67 + 23 + 35 + 65) / 9 = 39,77$$

Filtragem Espacial: Média da vizinhança

Imagem Original

Vizinhança $3x3$

Vizinhança $5x5$

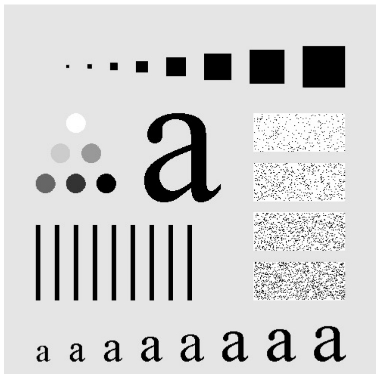
Vizinhança $7x7$

Vizinhança $15x15$

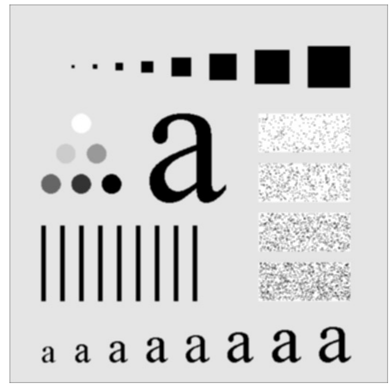
Vizinhança $25x25$

Filtragem Espacial: Média da vizinhança

Imagem Original



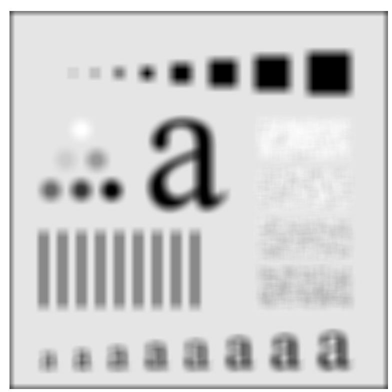
Vizinhança $3x3$



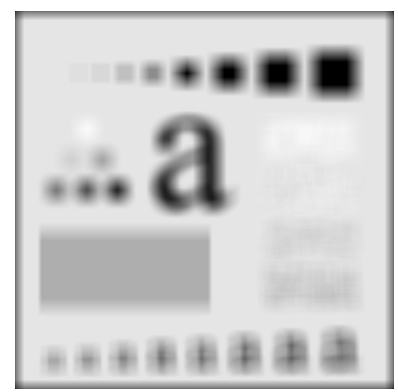
Vizinhança $5x5$

Vizinhança $7x7$

Vizinhança $15x15$



Vizinhança $25x25$

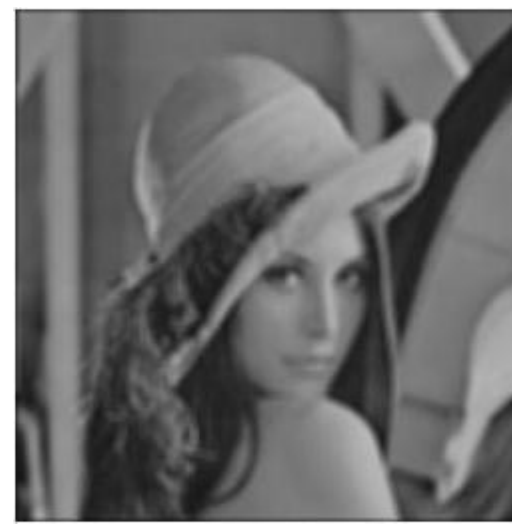


Aplicações de filtros passa-baixa para filtragem de ruído

Filtro Passa Baixa – Média da Vizinhança

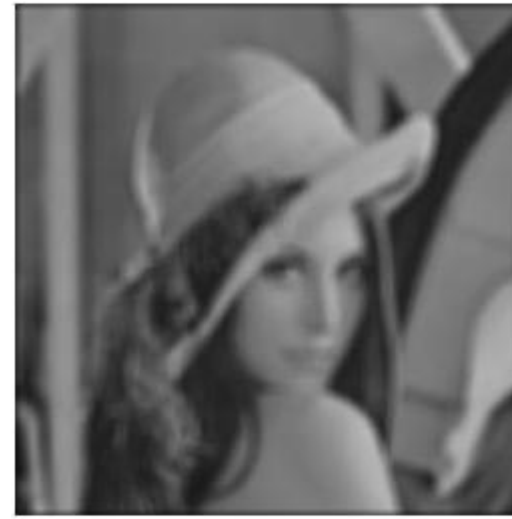
$$* \ \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{3\times3} =$$

$$* \ \frac{1}{25} \times \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}_{5\times5} =$$

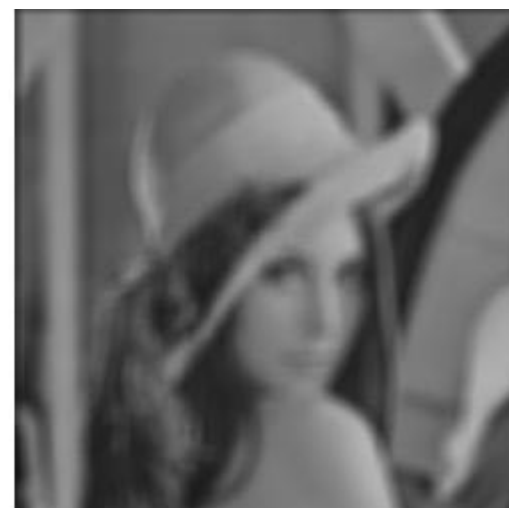


Filtro Passa Baixa – Média da Vizinhança

$$* \frac{1}{49} \times \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}_{7\times 7} =$$

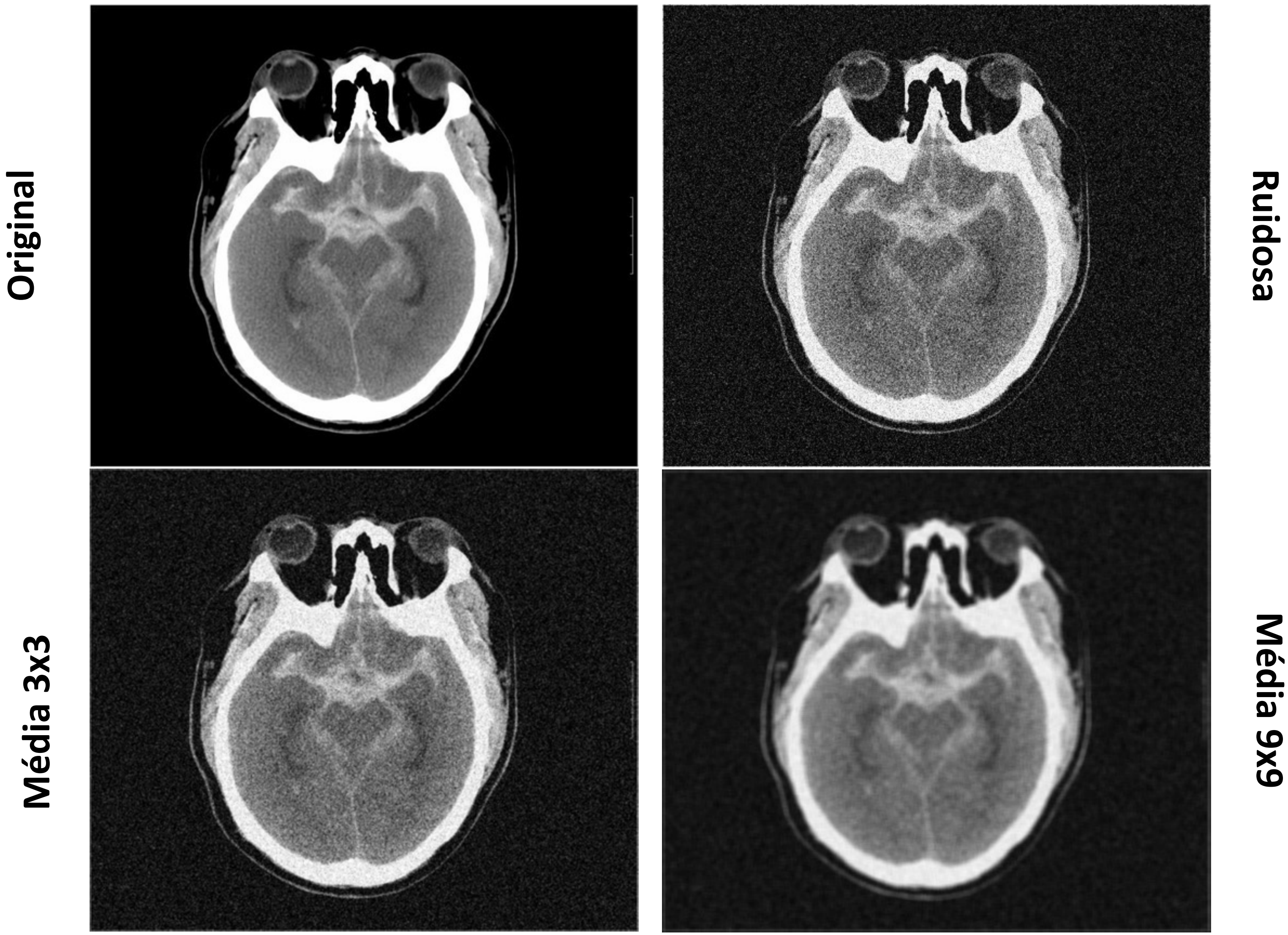


$$* \frac{1}{81} \times \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}_{9\times 9} =$$



Aplicações de filtros passa-baixa para filtragem de ruído

Filtro da média (passa-baixa)



Filtragem Espacial: Passa Alta

- É chamada de filtro de passa-alta porque detecta na imagem os detalhes finos e mudanças abruptas de níveis de cinza na imagem.

- A máscara do filtro passa alta deve ter pesos de valores positivos e negativos de tal forma que a soma seja igual a zero.

Exemplos de máscaras de filtros passa alta:

Normalizado

$$\frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Normalizado

$\frac{1}{5}$

0	-1	0
-1	4	-1
0	-1	0

Operador Laplaciano

Filtro Passa Alta – Detector de Altas Frequências

Normalizado

$$* \; \frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$

$$255 \; -$$

$$=$$

Filtro Passa Alta – Detector de Altas Frequências

Não-normalizado

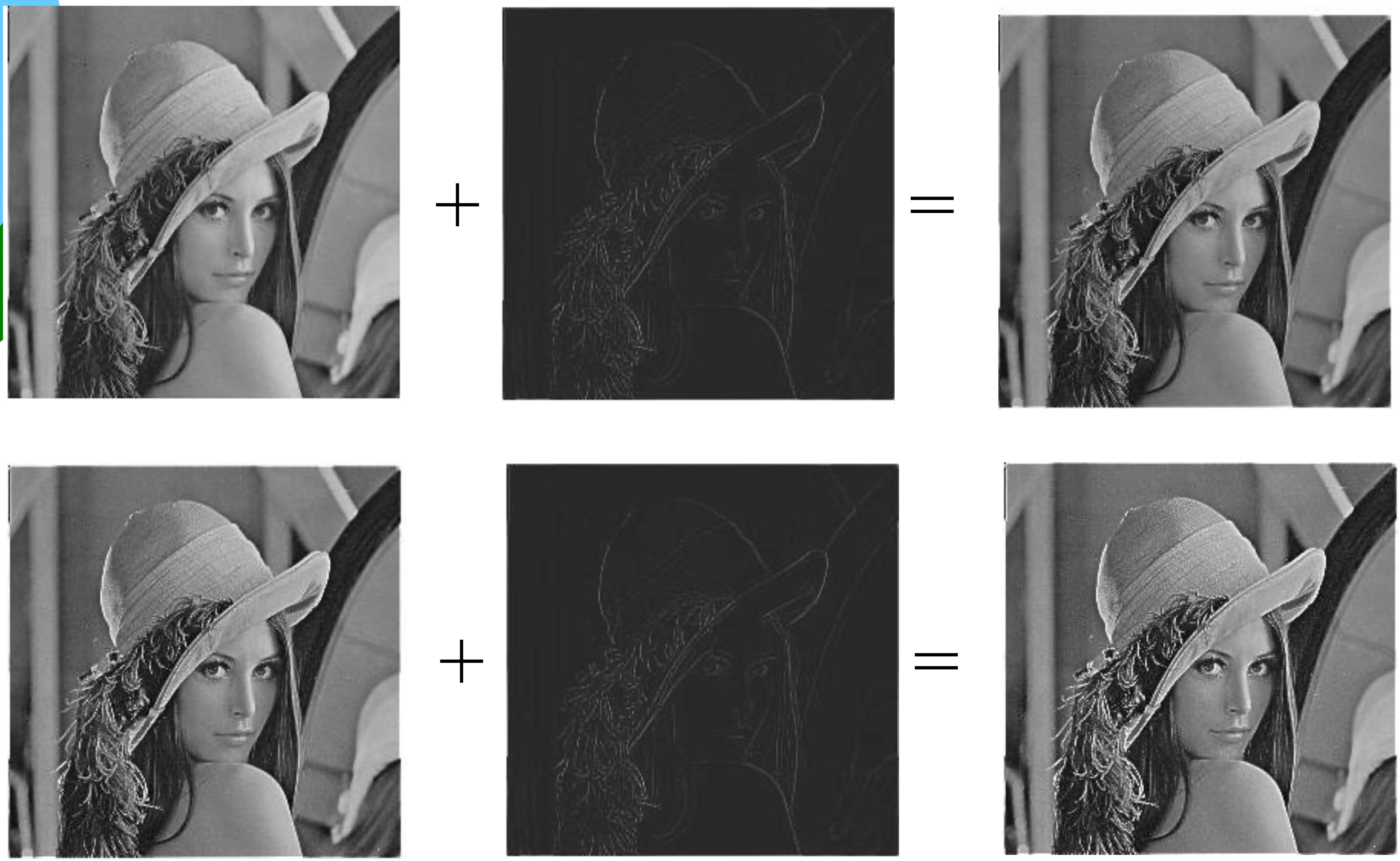
$$* \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$

$$255\ -$$

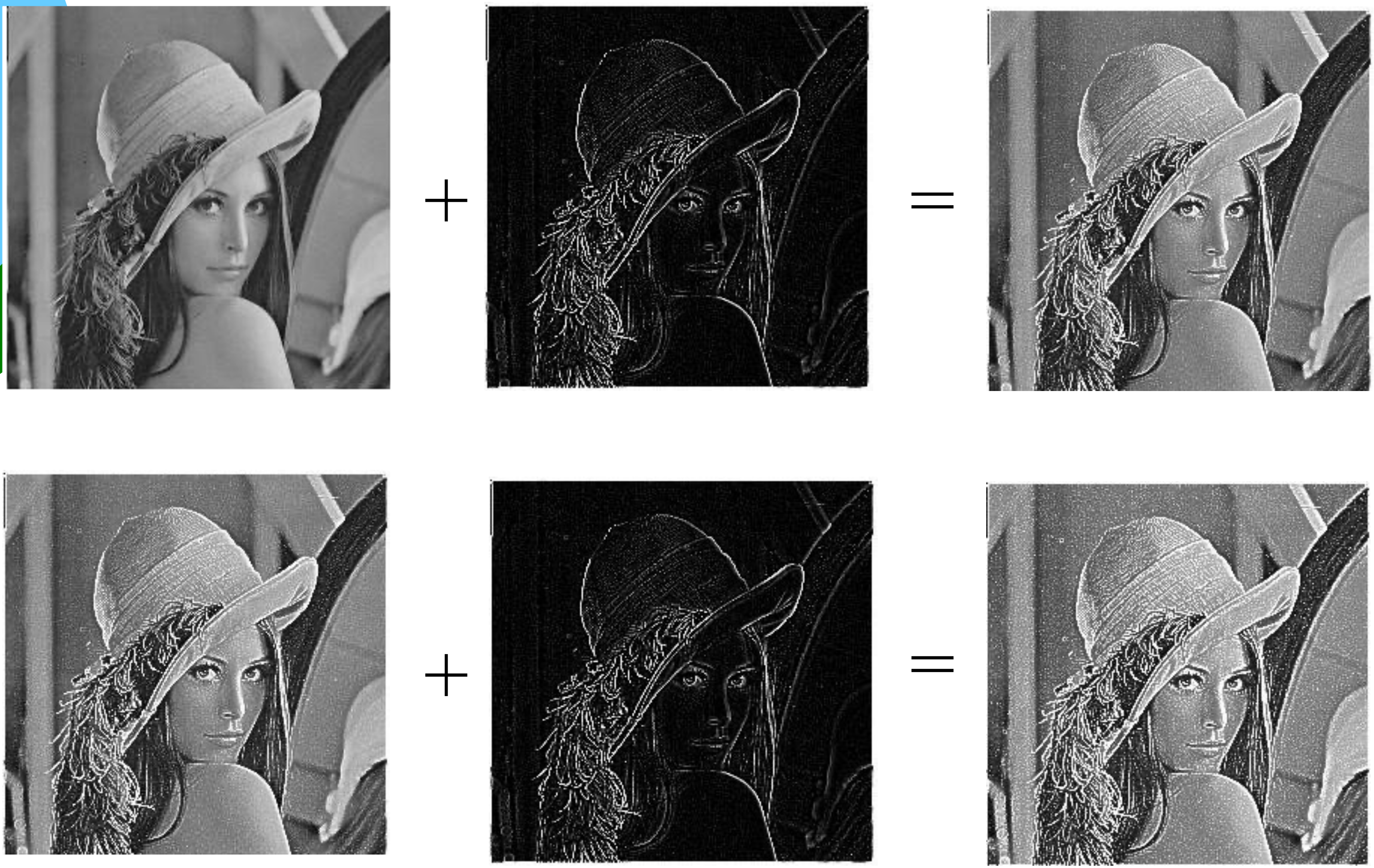
$$=$$

Aplicações de filtros passa-alta para aguçamento (aumento da nitidez) de imagens

Máscara de Nitidez
(*Sharpening*) - realce de altas frequências



Máscara de Nitidez
Não se usa o *Kernel* sem normalização!



Máscara de Nitidez
(*Sharpening*) - realce de altas frequências

Como montar um "template" para a máscara de aguçamento?

* ? =

Máscara de Nitidez
(*Sharpening*) - realce de altas frequências

1) Máscara que detecta apenas as bordas e detalhes (passa-alta)

Normalizado

$$* \ \frac{1}{9}\begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} =$$

Máscara de Nitidez
(*Sharpening*) - realce de altas frequências

2) máscara que gera a mesma imagem após a convolução

*

0	0	0
0	1	0
0	0	0

=

Máscara de Nitidez
(*Sharpening*) - realce de altas frequências

2) Máscara que gera a mesma imagem após a convolução

Normalizado

$$* \ \frac{1}{9}\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 9 & 0 \\ \hline 0 & 0 & 0 \\ \hline\end{array} =$$

Máscara de Nitidez
(*Sharpening*) - realce de altas frequências

Máscara de nitidez =
imagem original + detecção das bordas

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline\end{array} + \frac{1}{9}\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 9 & 0 \\ \hline 0 & 0 & 0 \\ \hline\end{array} = \frac{1}{9}\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 17 & -1 \\ \hline -1 & -1 & -1 \\ \hline\end{array}$$

Máscara de Nitidez (*Sharpening*) - realce de altas frequências

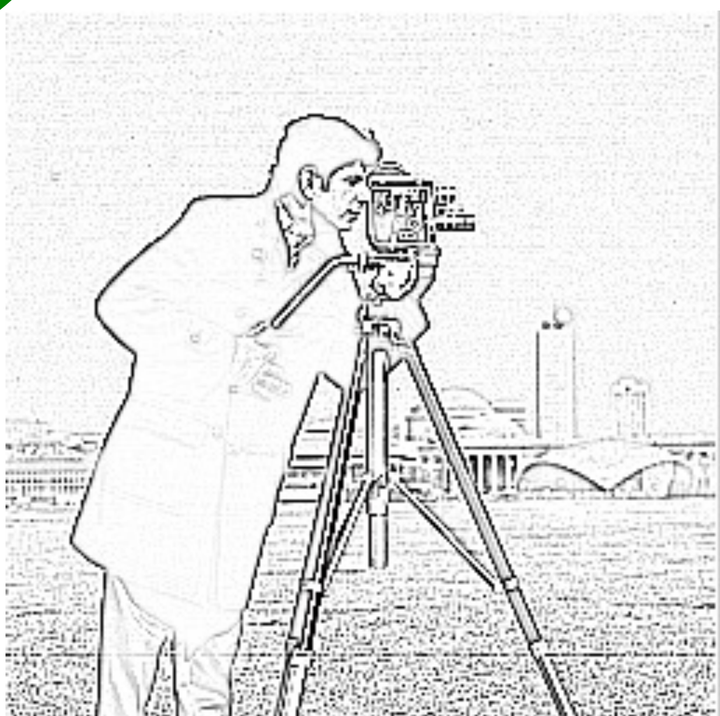
Máscara de Nitidez (normalizado)

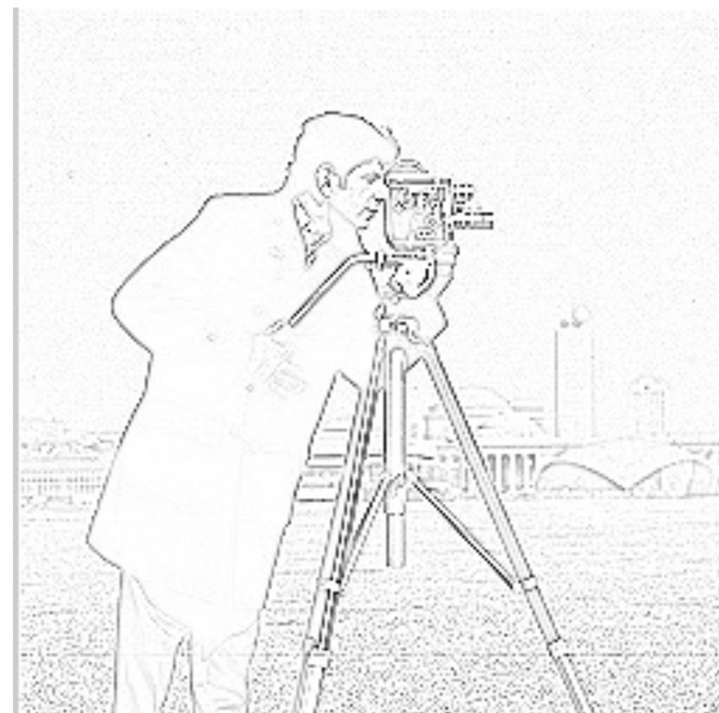
$$* \frac{1}{9} \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 17 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} =$$

Máscaras Isotrópicas

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

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$





http://setosa.io/ev/image-kernels/

FIM