

# IF69D-S11 Proc. Digital de Imagens APNP

## Aula 05

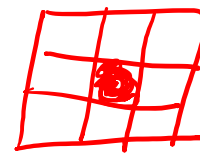
Universidade Tecnológica Federal do Paraná - UTFPR - Curitiba  
Departamento Acadêmico de Eletrônica - DAELN  
Curso de Engenharia Eletrônica  
Prof. Gustavo B. Borba

23.mar.2021



# Processamento baseado em vizinhança (neighborhood processing)

FILTROS



máscara (Mask), Template, Window, Kernel  
Janela  
3x3

IMG  
IN

OPERAÇÃO  
MÉDIA

10	10	10	10	10	10	10
10	10	10	10	10	10	10
30	30	30	30	30	30	30
30	30	30	30	30	30	30
50	50	50	50	50	50	50
50	50	50	50	50	50	50
70	70	70	70	70	70	70

BOX FILTER

Filtro  
da  
média

IMG  
OUT

0	0	0	0	0	0	0
0	17	17				0
0						0
0						0
0						0
0					57	0
0	0	0	0	0	0	0

# Processamento baseado em vizinhança (neighborhood processing)

5 x 5  
média

IN

10	10	10	10	10	10	10
10	10	10	10	10	10	10
30	30	30 <sub>P<sub>1</sub></sub>	30	30	30	30
30	30	30	30	30	30	30
50	50	50	50	50 <sub>P<sub>m</sub></sub>	50	50
50	50	50	50	50	50	50
70	70	70	70	70	70	70

OUT

0	0	0	0	0	-	
0	0	0	0	0	0	
0	0	26 <sub>P</sub>				
0	0					
-				46 <sub>P<sub>m</sub></sub>		
-	-	-	-	-	-	-
-	-	-	-	-	-	-

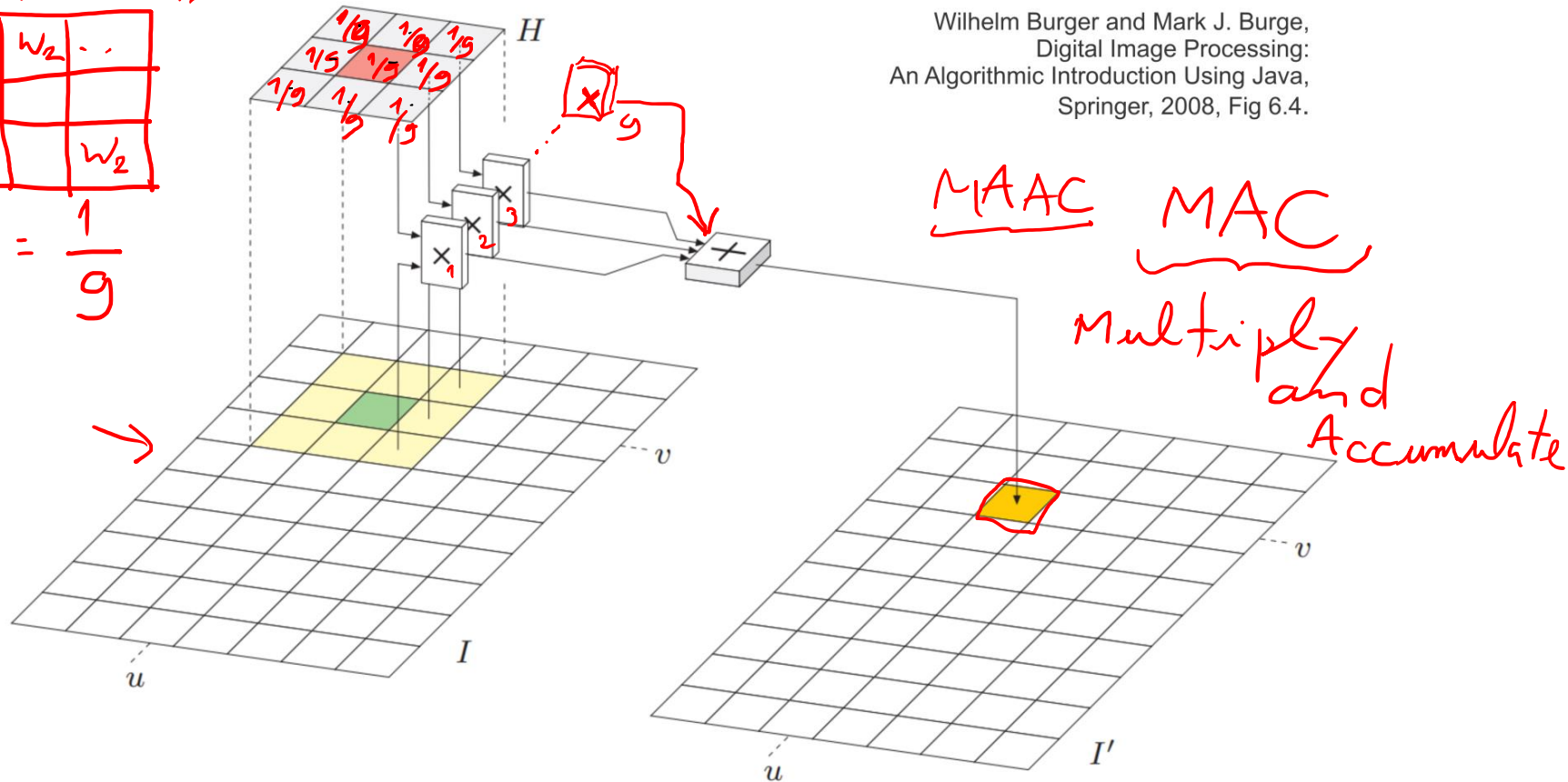
# Convolução

E se considerarmos que a máscara que pode ser imaginada como contendo coeficientes que vão ser multiplicados por cada pixel da imagem e esse resultado é somado? Isso se chama convolução.

BOX FILTER 3x3

$w_1$	$w_2$	...
		$w_2$

$$w_n = \frac{1}{9}$$

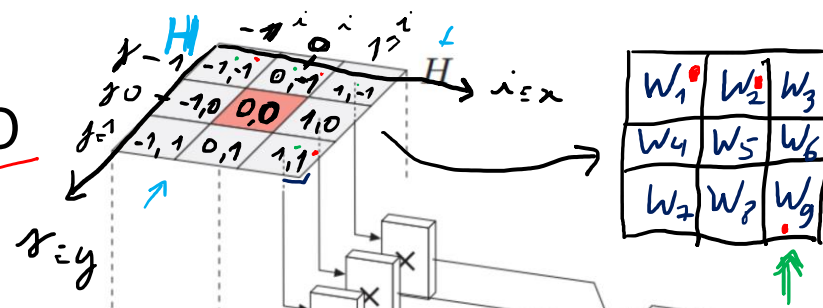


# Convolução

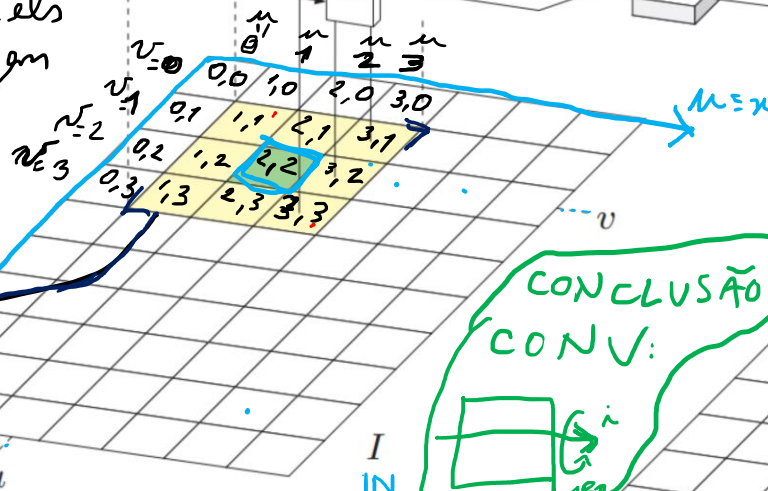
Wilhelm Burger and Mark J. Burge, Digital Image Processing: An Algorithmic Introduction Using Java, Springer, 2008, Fig 6.4.

Apenas para dar nome aos pixels da imagem

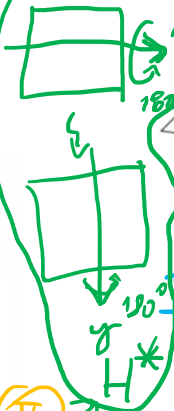
P1	P2	P3
P4	P5	P6
P7	P8	P9



só pra dar nome pros pixels da kernel



CONCLUSÃO:  
CONV:



OUT  
CONV

→  $I' = I * H$  KERNEL (Filtro)

$$I'(u, v) = \sum_{(i, j) \in R_H} I(u-i, v-j) \cdot H(i, j) \quad \text{I}$$

$$= \sum_{(i, j) \in R_H} I(u+i, v+j) \cdot H(-i, -j) \quad \text{II}$$

$$= \sum_{(i, j) \in R_H} I(u+i, v+j) \cdot H^*(i, j) \quad \text{III}$$

R<sub>H</sub>: todos os pares de coord. de um kernel

$$I'(2, 2) = I(2-(-1), 2-(-1)) \cdot H(-1, -1) + I(2-0, 2-(-1)) \cdot H(0, -1) + I(2-1, 2-1) \cdot H(1, 1)$$

P9      P8      P1

w1      w2      w9

# Convolução -> Filtros lineares

✓  
Passa-baixas (atenua componentes de alta-frequência da imagem): box-filter (média), Gaussiano. Usado para suavizar a imagem. Remove detalhes que não são necessários para a aplicação e suaviza ruído. Efeito de blurring (borramento) da imagem. Quanto maior a máscara, maior o efeito.

$\Sigma$  "SOMA" (integral)

→ suavização

$\delta$  (derivada)

✓  
Passa-altas (ênfatisa os componentes de alta-frequência das imagens): Laplaciano. Faz o realce de detalhes da imagem. Operação chamada de realce ou sharpening.

# Filtro da média (box filter) (passa-baixas, suavização)

Lenna  
+  
RUÍDO

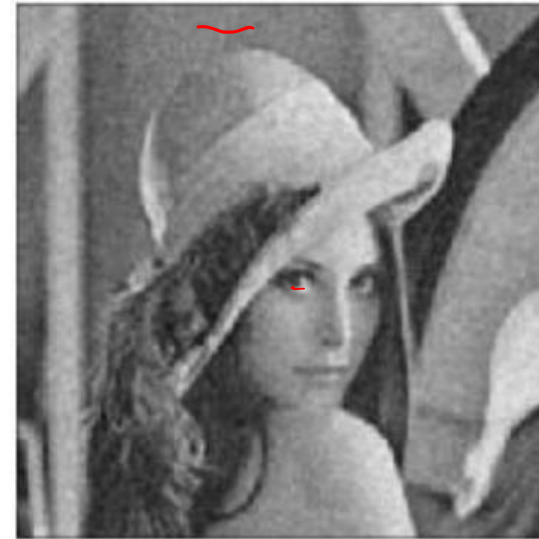
Entrada //



Média 5x5



Média 3x3



Média 7x7



# Queridão

## Filtro Gaussiano (passa-baixas, suavização)

Mais usado que o box filter: isotrópico; dá maior peso para o pixel original (central do kernel); melhor comportamento na frequência, pois os coeficientes decaem até quase zero nas bordas do kernel, não fazendo corte abrupto do sinal.

Parâmetros:

Dimensões  
① De kernel

②  $\sigma$ : desvio padrão

$\sigma^2$ : variância

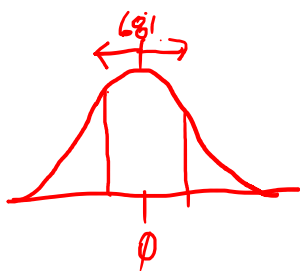
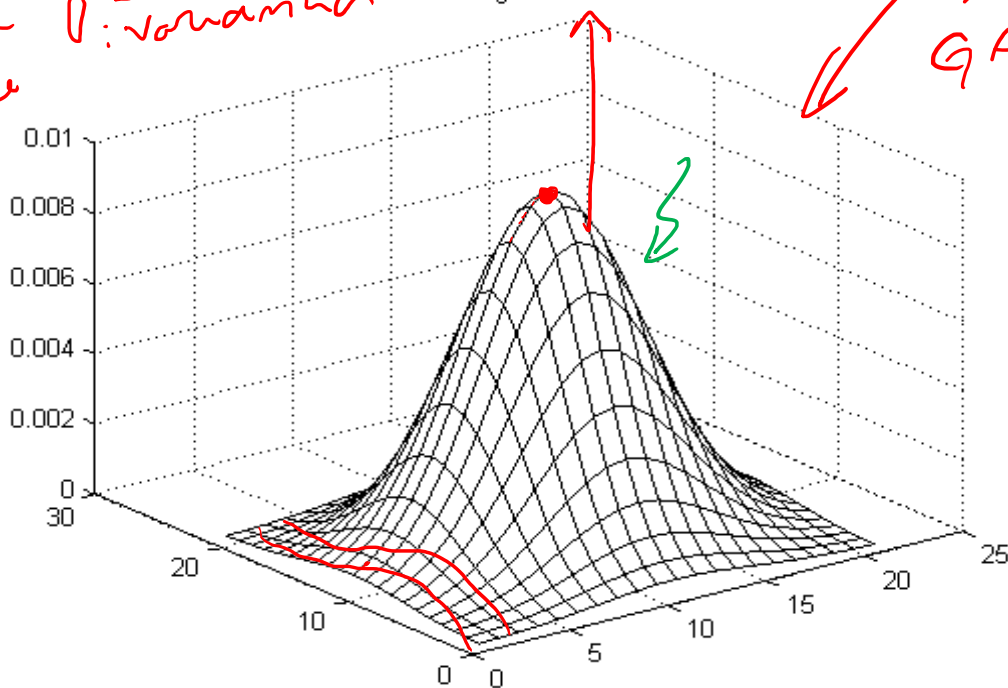
$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2 + y^2}{2\sigma^2}\right)}$$

sigma=4

kernel

$\sum_H = 1$   
para não alterar o nível médio da imagem

Função GAUSSIANA 3D



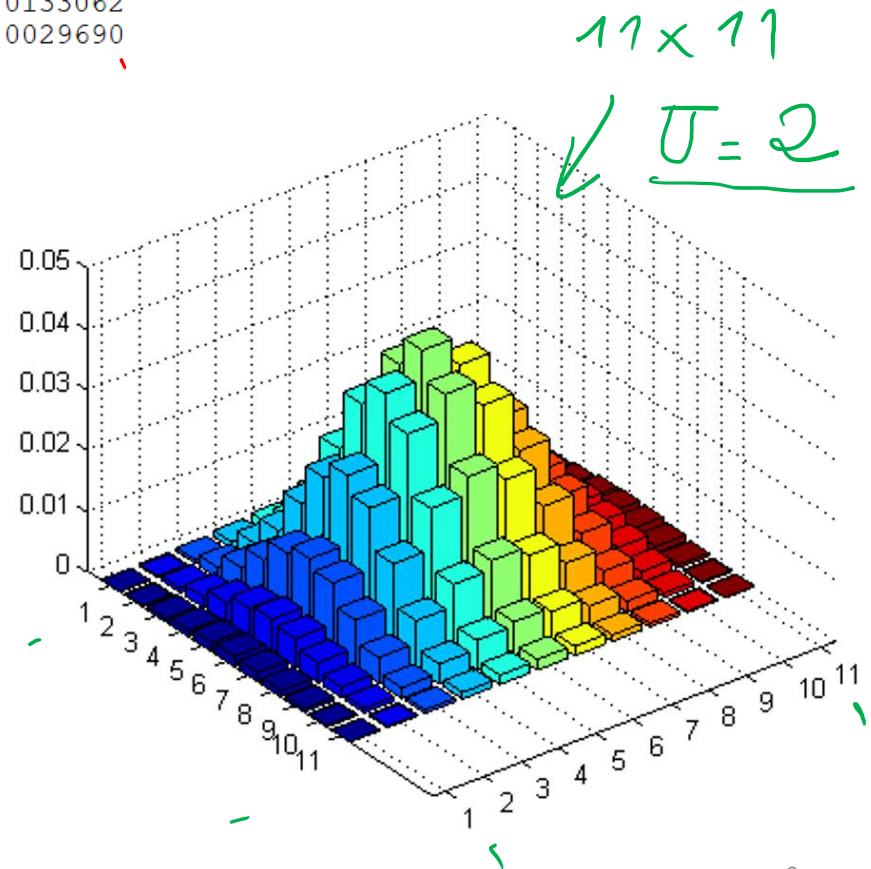
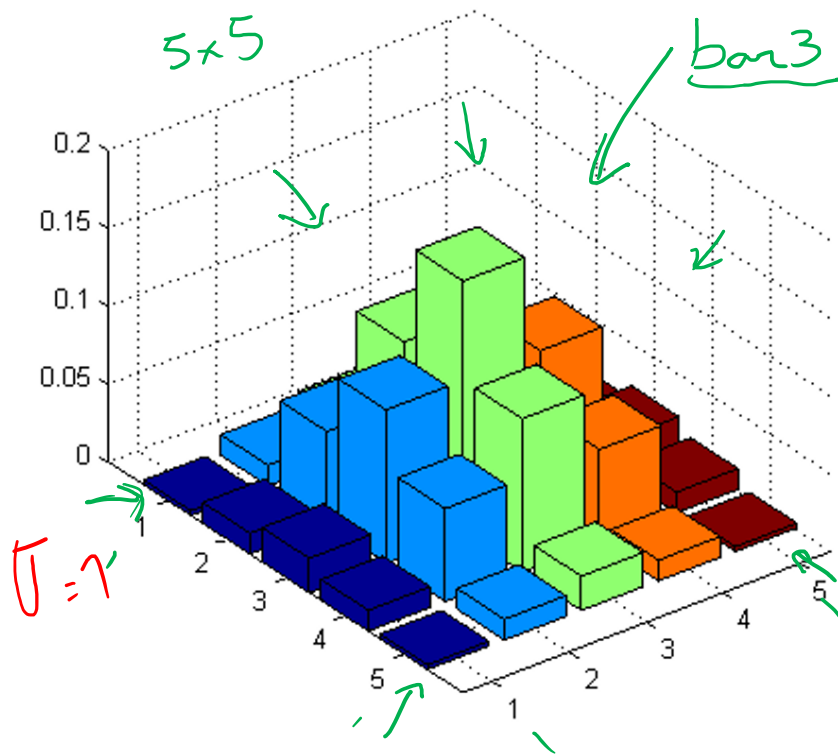


# Filtro Gaussiano (passa-baixas, suavização)

Parâmetros: tamanho do kernel e desvio padrão (sigma).

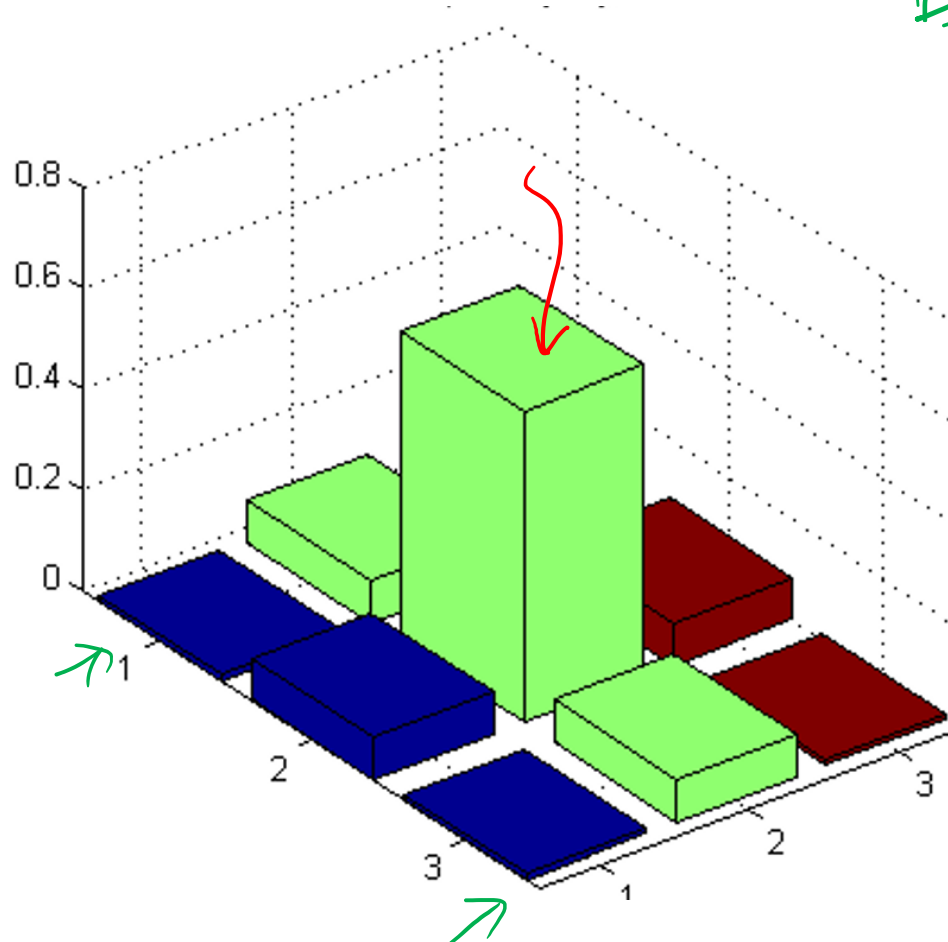
```
>> h = fspecial('gaussian', [5 5], 1)
h =
```

0.0029690	0.0133062	0.0219382	0.0133062	0.0029690
0.0133062	0.0596343	0.0983203	0.0596343	0.0133062
0.0219382	0.0983203	0.1621028	0.0983203	0.0219382
0.0133062	0.0596343	0.0983203	0.0596343	0.0133062
0.0029690	0.0133062	0.0219382	0.0133062	0.0029690



# Filtro Gaussiano (passa-baixas, suavização)

Parâmetros: tamanho do kernel e desvio padrão (sigma).



Default  
Octave  
 $h = f_{\text{special}}$   
(gaussian)

3x3  
 $\sigma = 0,5$



# Filtro Gaussiano (passa-baixas, suavização)

IN



OUT

Gaussiano

$3 \times 3$

$\sigma = 0.5$

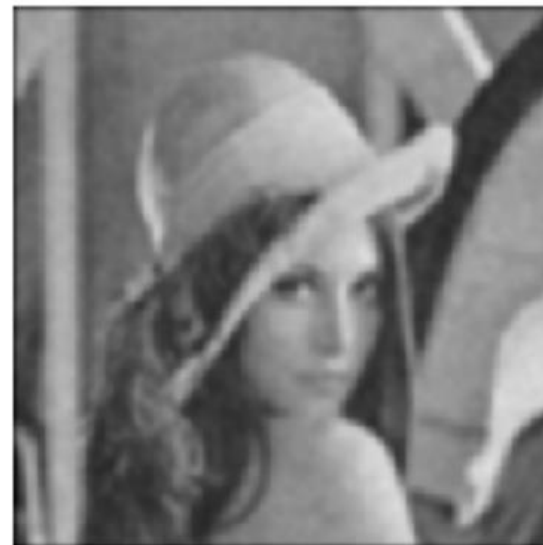


$5 \times 5$   
 $\sigma = 1$



$11 \times 11$

$\sigma = 2$



# Filtro Laplaciano (passa-altas, realce)

Passa-altas (ênfatisa os componentes de alta-frequência da imagem): Laplaciano. Faz o realce de detalhes da imagem. Operação chamada de realce ou sharpening.

aproximação da derivada de segunda ordem (derivada 2ª) de um sinal 2D



$*$  →

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

→

Laplaciano  
é um kernel  
3x3



# Filtro Laplaciano (passa-altas, realce)

IN (original)



CONVOLUÇÃO  
COM LAPLACIANO



SHARPENING

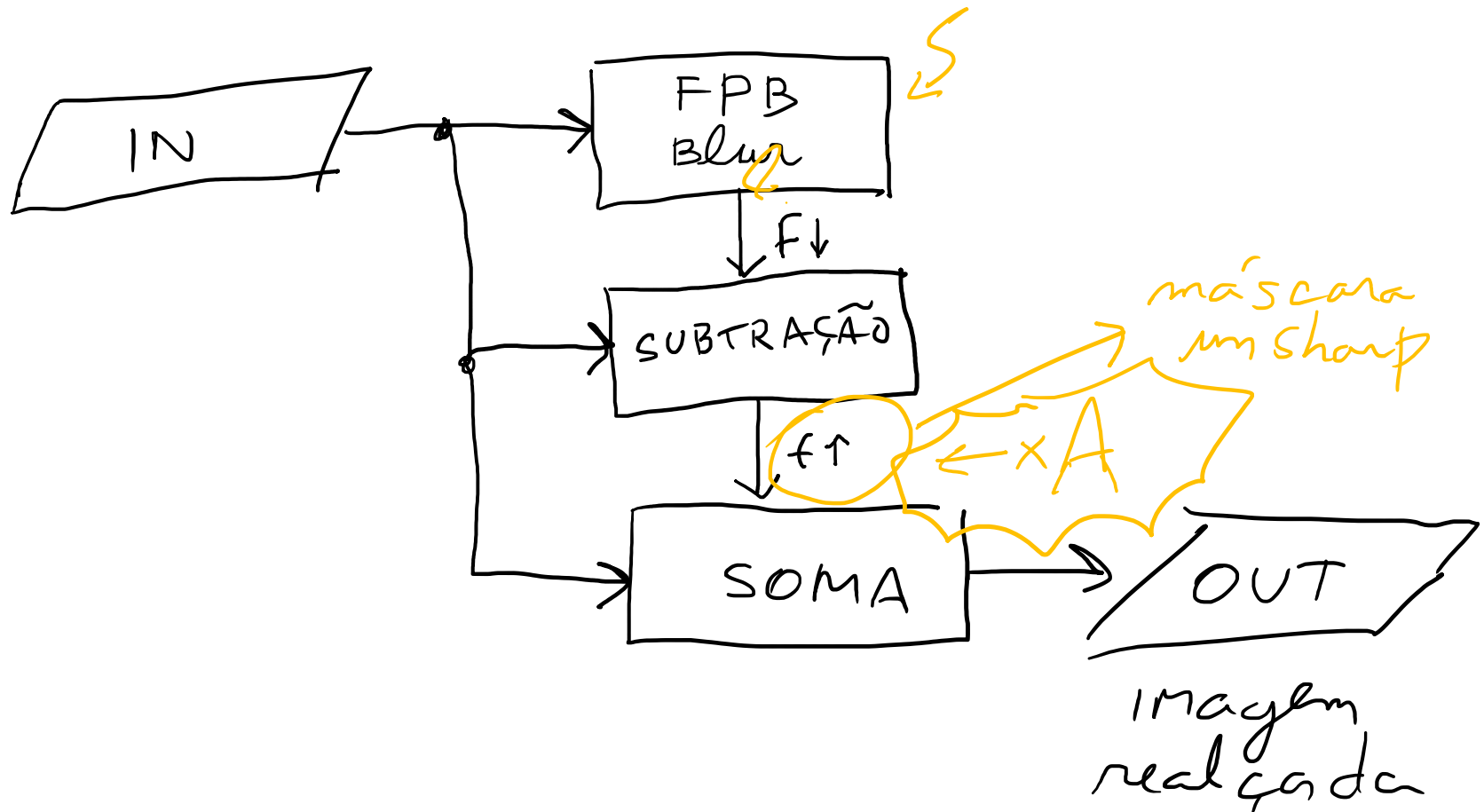


COMBINAR





# Realce (sharpening) método Unsharp Masking



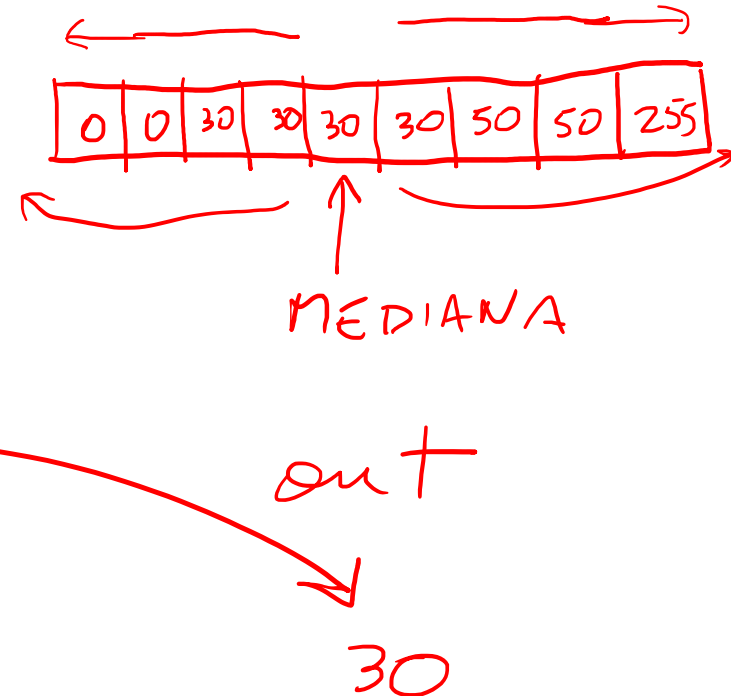
# Filtros não lineares (não usam convolução)

Filtro da mediana: usado para a restauração de imagens com ruído do tipo sal-e-pimenta (salt-and-pepper noise).

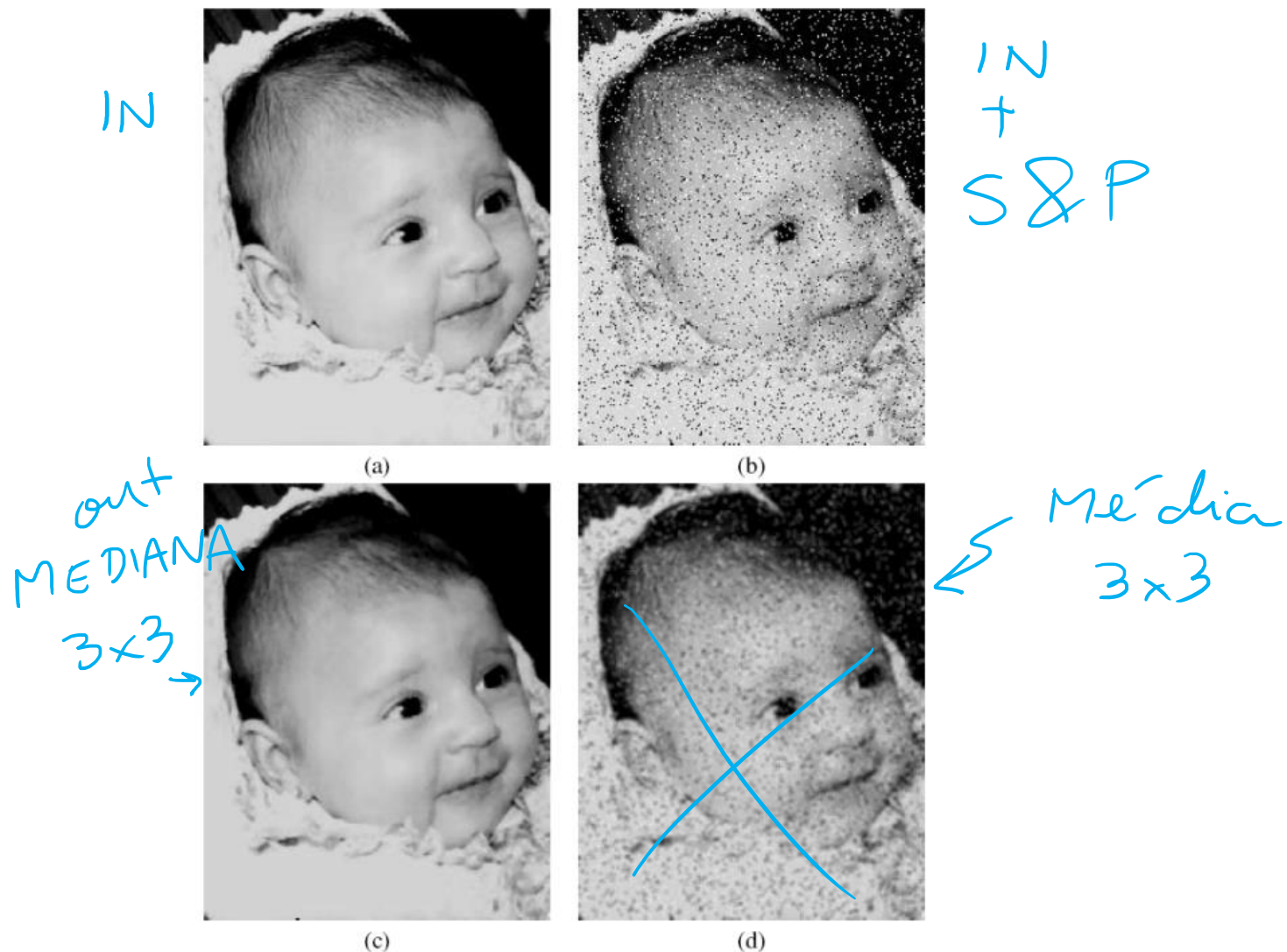
*Filtro da Mediana  
3x3*

10	10	10	10	10	10	10
10	10	10	10	10	10	10
30	30	255	30	30	30	30
30	30	30	0	30	30	30
50	50	0	50	50	50	50
50	50	50	50	50	50	50
70	70	70	70	70	70	70

*OUTLIERS*



# Filtros não lineares (não usam convolução)





# Tratamento das bordas da imagem

Zero-padding

0	0	0	0						
0	0	0	0	0	0	0	0	0	
0	0	10	30	10	70	10	10	50	0
0	0	10	30	10	70	10	10	50	0
		30	30	30	70	30	30	50	
		30	30	30	70	30	30	50	
		50	30	50	70	50	50	50	
		50	30	50	70	50	50	50	
		70	30	70	70	70	70	50	

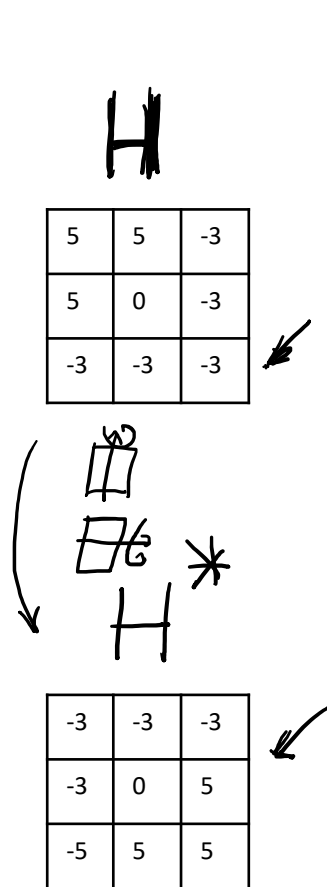

# Tratamento das bordas da imagem

*Symmetric*

30	10	30							
30	10	30							
10	10	30	10	70	10	10	50		
10	10	30	10	70	10	10	50		
30	30	30	70	30	30	50			
30	30	30	70	30	30	50			
50	30	50	70	50	50	50			
50	30	50	70	50	50	50			
70	30	70	70	70	70	50			

*is zero*


# Atenção: convolução vs. correlação



kernel  
↓

$$OUT_1 = \text{imfilter}(H, I_n, 'corr')$$

$H^*$

$$OUT_2 = \text{imfilter}(H, I_n, 'conv')$$