



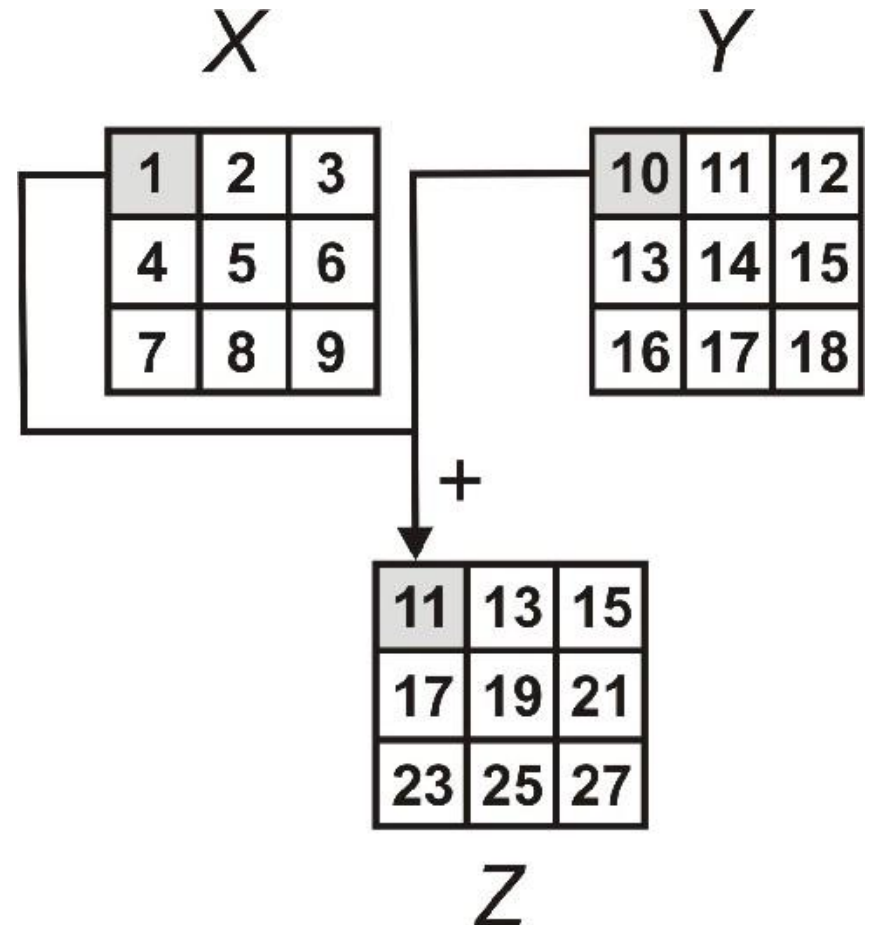
Operações Algébricas e Lógicas

Guillermo Cámara-Chávez

Operações Aritméticas



- São aquelas que produzem uma imagem que é a soma, diferença, produto ou quociente pixel a pixel



Operações Aritméticas



Fig A

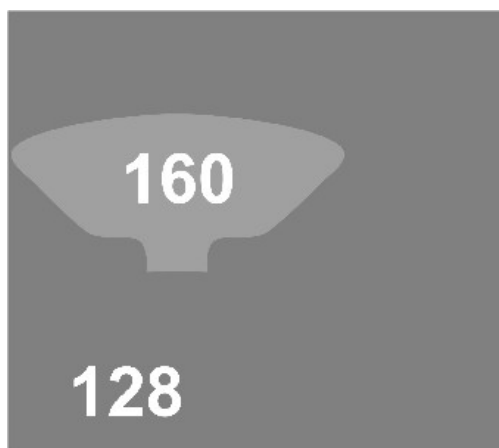
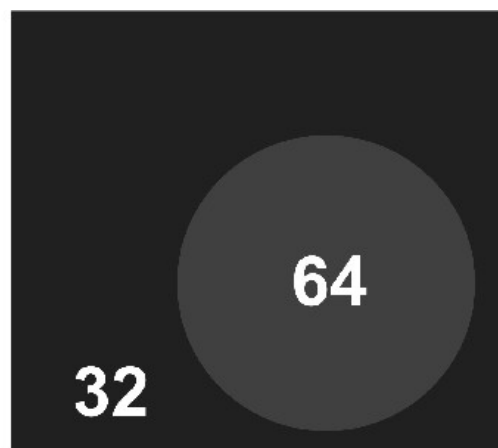
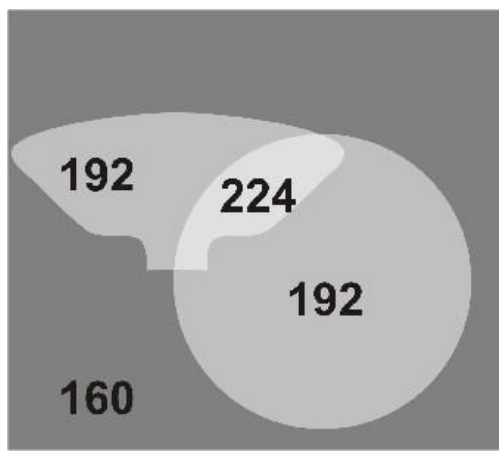


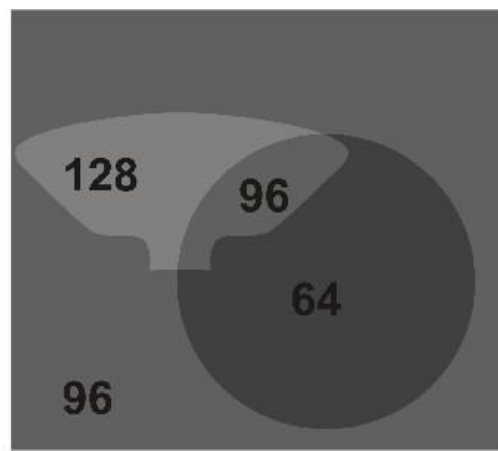
Fig B



$$C = A + B$$



$$C = A - B$$



Operações Aritméticas



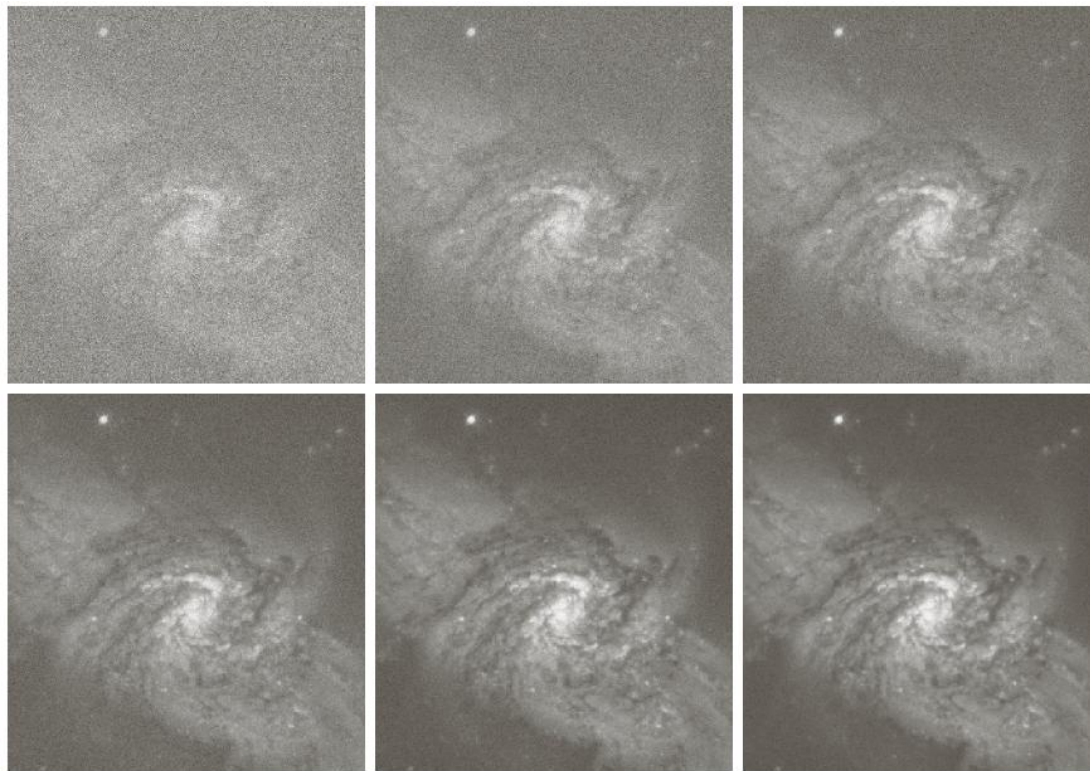
- As quatro operações algébricas de processamento de imagens:
 - $C(x,y) = A(x,y) + B(x,y)$
 - $C(x,y) = A(x,y) - B(x,y)$
 - $C(x,y) = A(x,y) * B(x,y)$
 - $C(x,y) = A(x,y) / B(x,y)$
- $A(x,y)$ e $B(x,y)$ são imagens de entrada e $C(x,y)$ é a imagem resultante

Aplicações das operações Aritméticas



- Adição:
 - Obter a **média** de múltiplas imagens de uma mesma cena
 - Útil para **reduzir os efeitos de ruídos** aleatórios aditivos
 - Pode ser utilizado para **colocar conteúdo** de uma imagem **sobrepondo** outra

Aplicações das operações Aritméticas



a b c
d e f

FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

Aplicações das operações Aritméticas



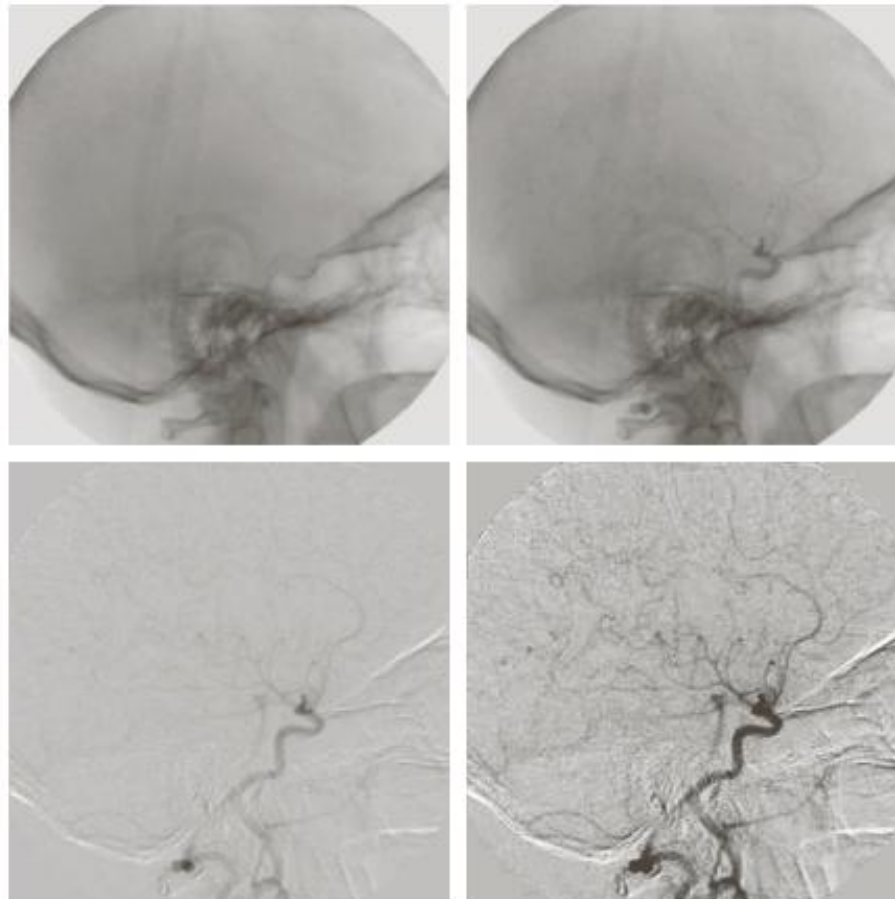
```
function [nimg, figs] = NoiseSum(img, n)
    nimg = zeros(size(img));
    figs = cell(1,n);
    for i = 1 : n
        figs{i} = imnoise(img);
        nimg = nimg + double(figs{i});
    end
    nimg = uint8(nimg / n);
```

Aplicações das operações Aritméticas



- Subtração:
 - Utilizado para **remove** algum padrão indesejável
 - **Detectar mudanças** entre duas imagens da mesma cena.
 - Pode ser utilizada para **calcular o gradiente** (detecção de bordas)

Aplicações das operações Aritméticas



a	b
c	d

FIGURE 2.28
Digital subtraction angiography.
(a) Mask image.
(b) A live image.
(c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

Aplicações das operações Aritméticas



- Multiplicação e divisão:
 - **Corrigir possíveis defeitos** de um digitalizador
 - Multiplicar uma imagem por uma “máscara” pode **esconder certas regiões deixando exposto apenas objetos de interesse**

Aplicações das operações Lógicas

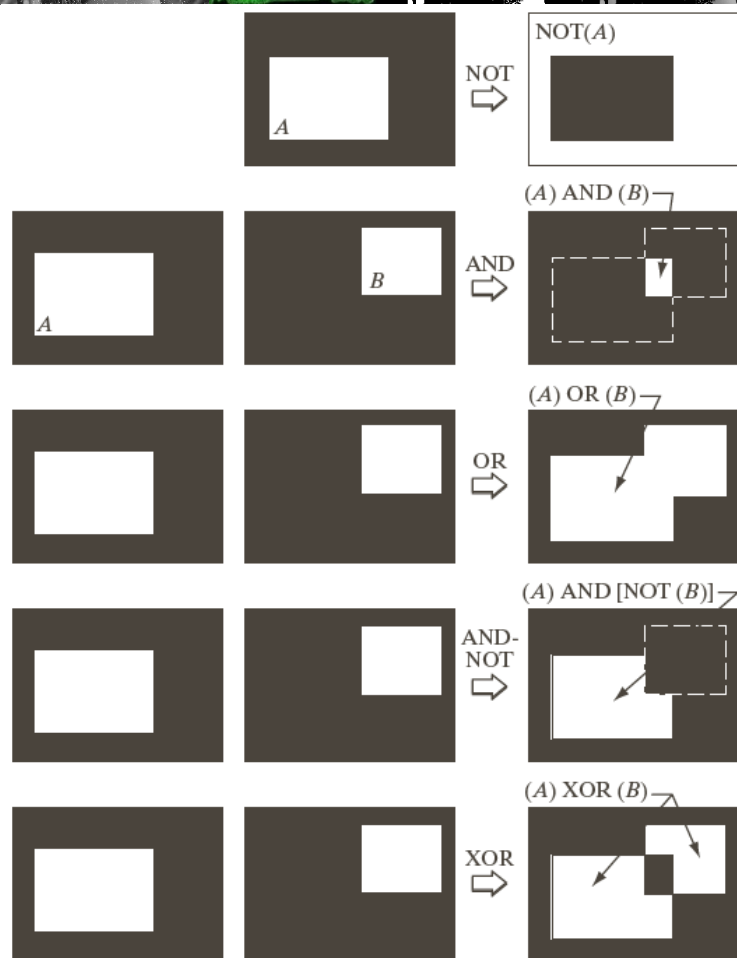


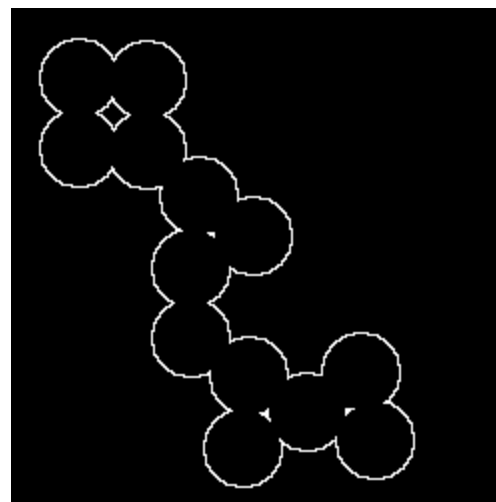
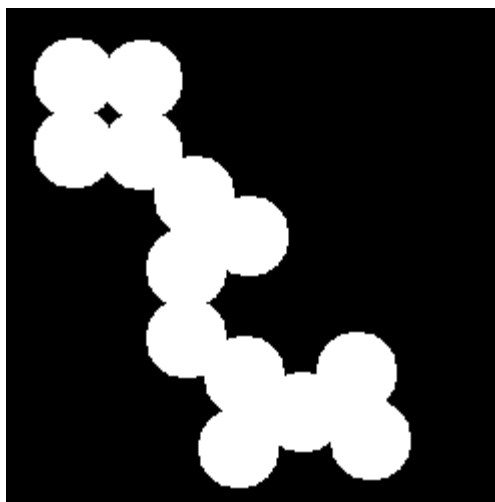
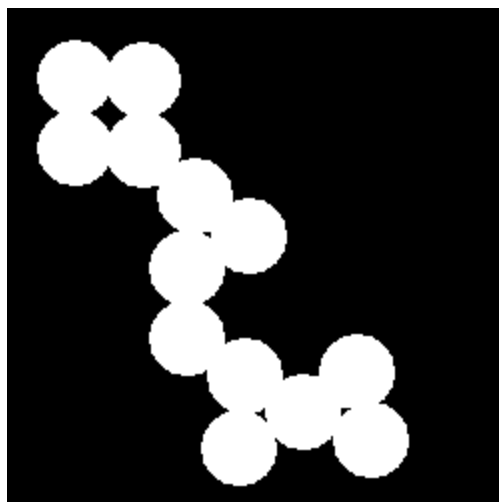
FIGURE 2.33

Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Aplicações das operações Lógicas



```
img = imread('coins.png');  
imgd = imdilate(img, ones(3,3));  
img3 = imgd & ~img;  
imshow(img3)
```

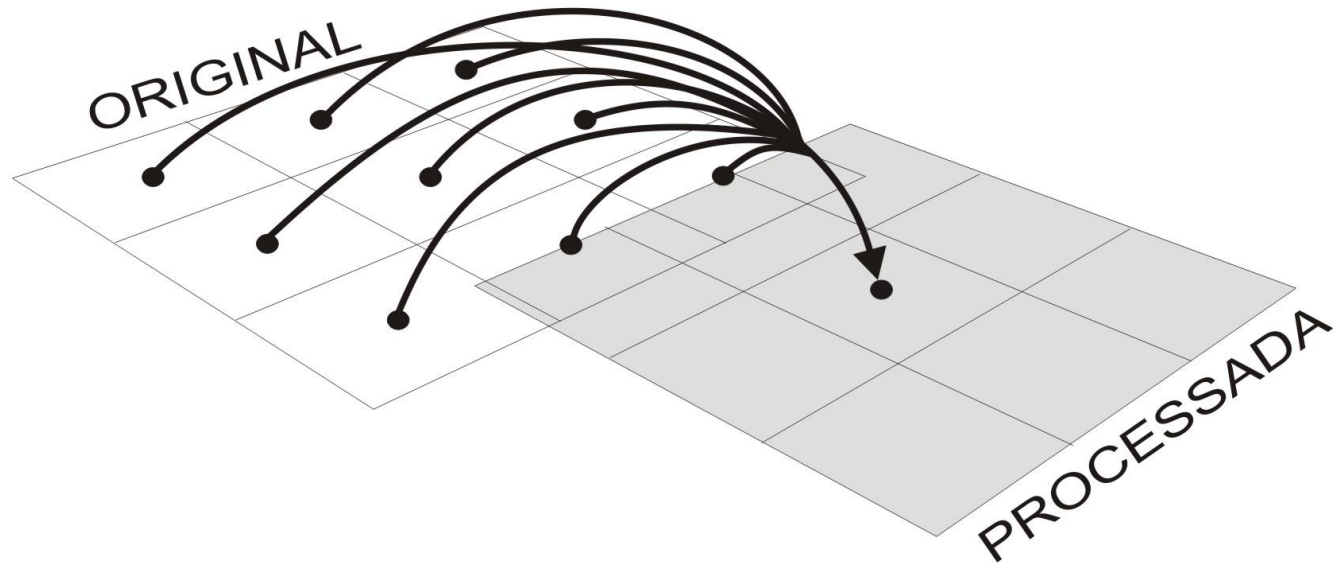


Operações Locais



Um *pixel* da imagem resultante depende de uma **vizinhança** do mesmo *pixel* na imagem original

$$\text{OpL}(f(x_i, y_i), f(x_i-1, y_i-1), f(x_i+1, y_i+1), f(x_i-1, y_i+1), f(x_i+1, y_i-1), \dots)$$



Operações Locais



- Utilizam informação dos **valores dos pontos vizinhos** para modificar o **valor de um ponto**, ou para **verificar a existência** de alguma **propriedade** nesta ponto
- São utilizadas para filtragem espacial e alteração da própria estrutura da imagem.

Operações Locais



- Elas podem:
 - “aguçar” a imagem, acentuando as mudanças de intensidades (através de filtros passa-altas)
 - “suavizar” a imagem, tornando as mudanças de intensidades menos abruptas (através de filtros passa-baixas)

Operações Locais

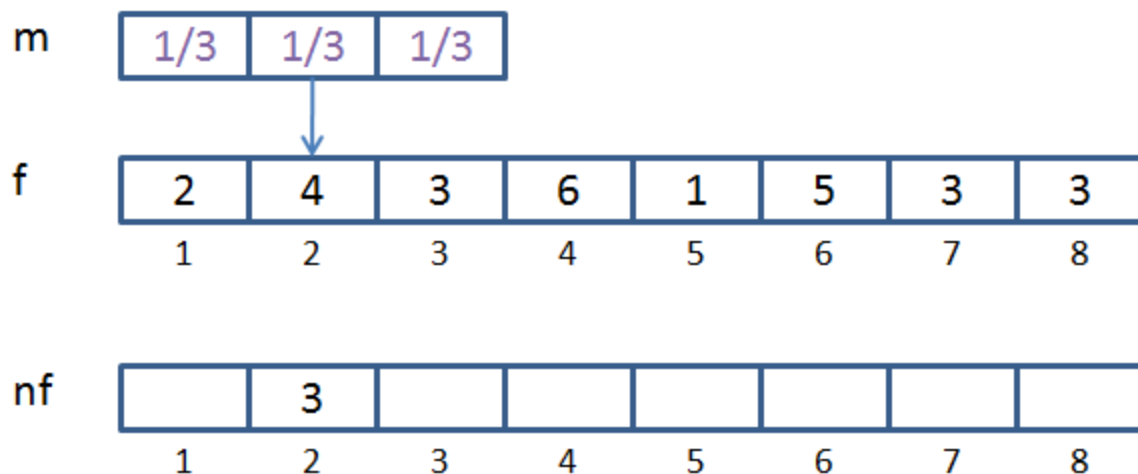


- Elas podem:
 - Procurar formas na imagem através de “padrões de busca” (*match*)
 - Definir bordas na imagem
 - Remover ruído

Operações de Convolução



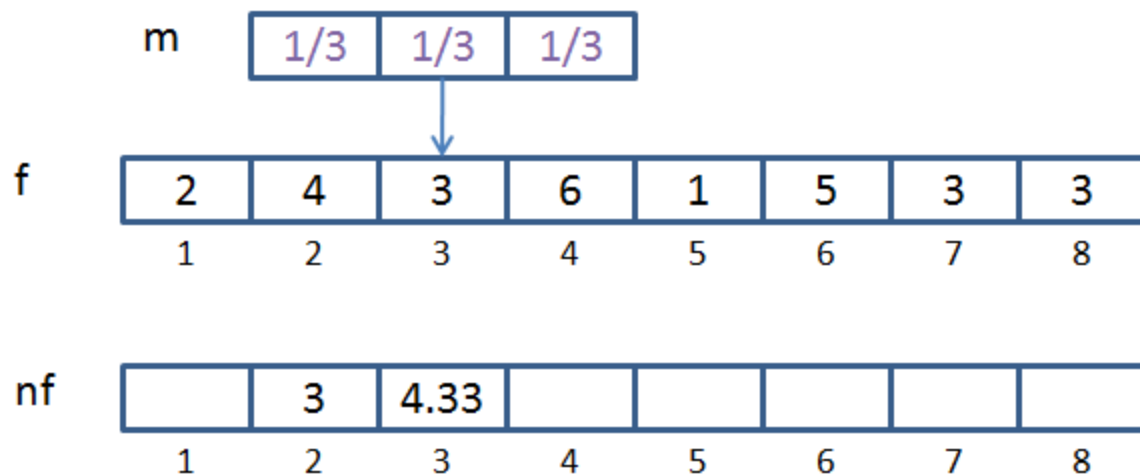
$$nf(2) = \frac{1}{3} * 2 + \frac{1}{3} * 4 + \frac{1}{3} * 3 = 3$$



Operações de Convolução



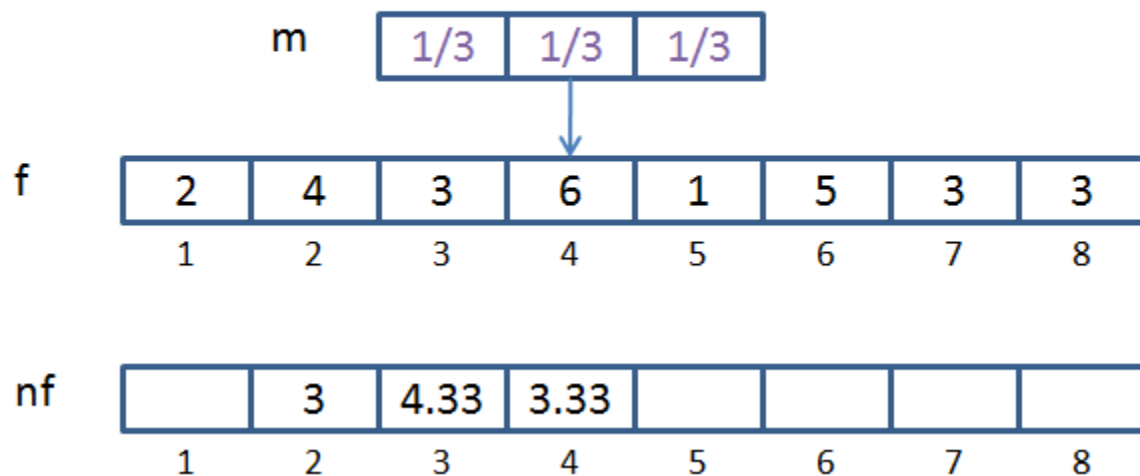
$$nf(3) = \frac{1}{3} \cdot 4 + \frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 6 = 4.33$$



Operações de Convolução



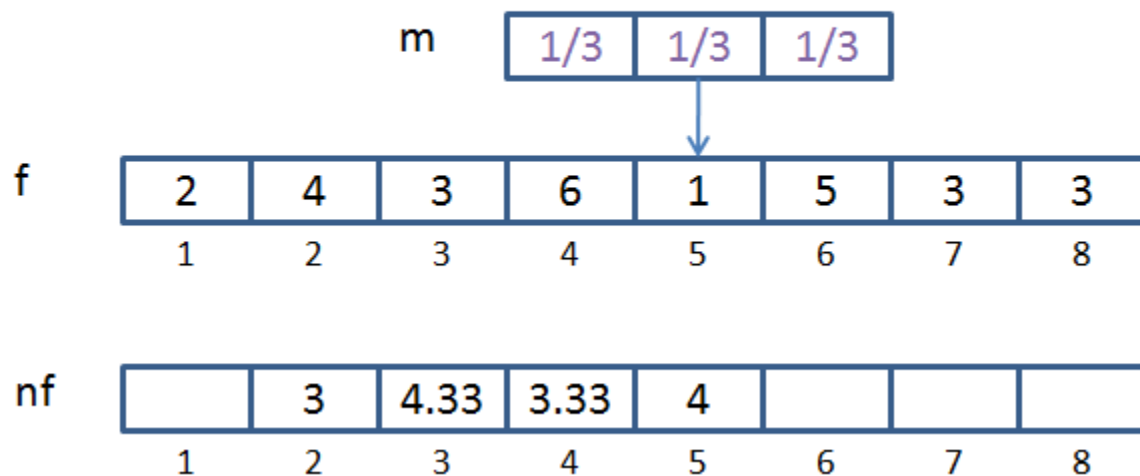
$$nf(4) = \frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 6 + \frac{1}{3} \cdot 1 = 3.33$$



Operações de Convolução



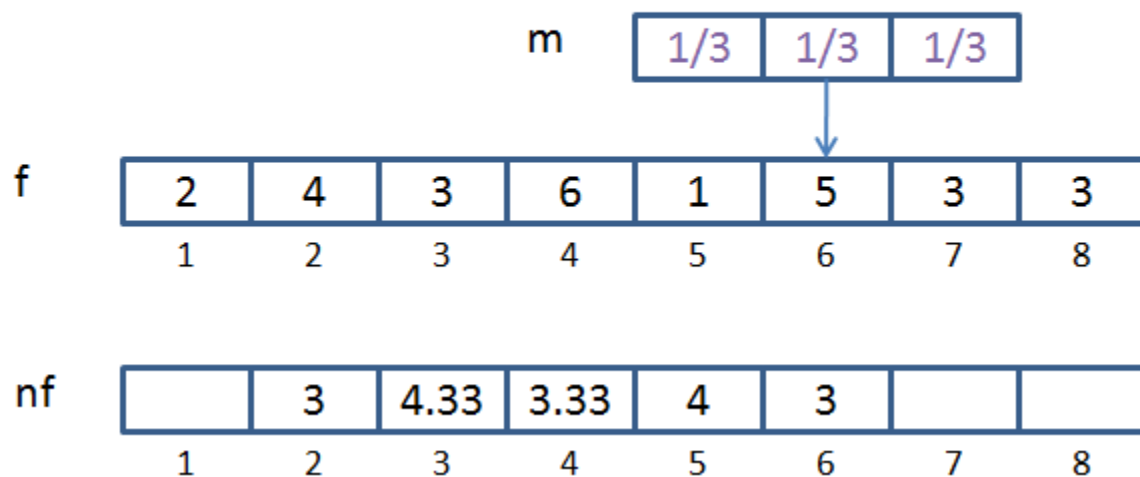
$$nf(4) = \frac{1}{3} \cdot 6 + \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 5 = 4$$



Operações de Convolução



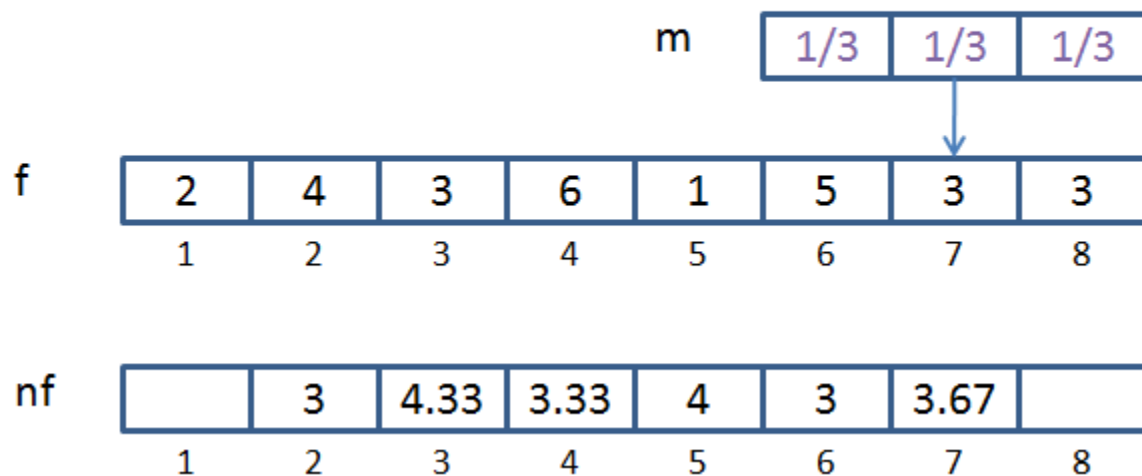
$$nf(4) = 1/3 * 1 + 1/3 * 5 + 1/3 * 3 = 3$$



Operações de Convolução



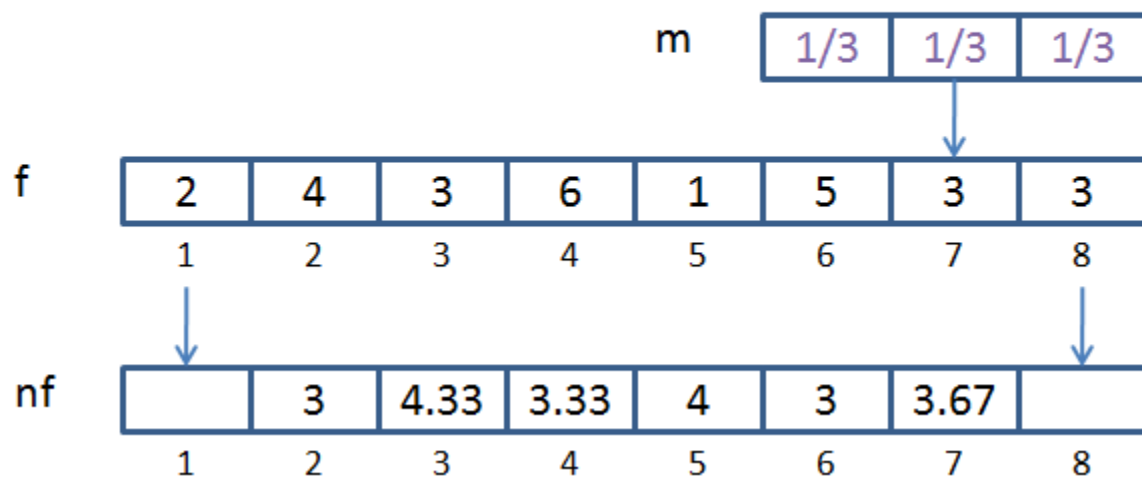
$$nf(4) = \frac{1}{3} * 5 + \frac{1}{3} * 3 + \frac{1}{3} * 3 = 3.67$$



Operações de Convolução



$$nf(4) = \frac{1}{3} \cdot 5 + \frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 3 = 3.67$$



Operações de Convolução



f	2	4	3	6	1	5	3	3
	1	2	3	4	5	6	7	8
	↓							↓
nf	2	3	4.33	3.33	4	3	3.67	3
	1	2	3	4	5	6	7	8

Operações de Convolução



- Seja m a máscara de **correlação**.
- Em geral, seleciona-se uma máscara com um número ímpar de elementos.
- A **correlação** do sinal f com a máscara m pode ser expressa como

$$m \bullet f(x) = \sum_{i=\lfloor -m/2 \rfloor}^{\lfloor m/2 \rfloor} m(i) f(x+i)$$

Operações de Convolução



- A convolução consiste em um processo similar
- A máscara m deve sofrer uma reflexão.
- A convolução de um sinal f por uma máscara m pode ser expressa como

$$m * f(x) = \sum_{i=\lfloor -m/2 \rfloor}^{\lfloor m/2 \rfloor} m(i) f(x-i)$$

Operações de Convolução



- A filtragem linear de uma imagem f de tamanho $M \times N$ com uma máscara m de tamanho $m \times n$ está determinada por:

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

Operações de Convolução



- A saída de um ponto está determinado pela soma dos valores de entrada ao redor do ponto, cada um multiplicado pelo termo correspondente da máscara.
- Para calcular o próxima valor de saída, máscara deve ser deslocada e a operação soma deve ser repetida.

Operações de Convolução



$$X1=m/2$$

$$Y1=n/2$$

Para $x=0$ até $M-1$ faça

Para $y=0$ até $N-1$ faça

soma = 0

Para $i=-x1$ até $x1$ faça

Para $j=-y1$ até $y1$ faça

$$\text{soma} = \text{soma} + m(i,j)*f(x-i,y-j)$$

$$g(x,y) = \text{soma}$$

f : imagem de $M \times N$
 m : máscara de $m \times n$

Operações de Suavização



- São usados para borrar uma imagem (*blurring*) ou para redução de ruído
- O borramento é usado em:
 - operações de pré-processamento.
 - na remoção de pequenos detalhes para extracção de objetos maiores
 - Unir pequenos intervalos em linhas retas ou curvas

Operações de Suavização



- A resposta de um filtro linear espacial “smoothing” é simplesmente a média dos pixels contidos dentro da vizinhança da máscara.
- Este tipo de filtro é conhecido como filtro da média, ou também como filtro passa-baixa

Operações de Suavização



- A técnica é substituir o valor de cada pixel na imagem, pela média dos níveis de cinza da vizinhança definida pela máscara.
- Este processo resulta em uma nova imagem com redução das transições acentuadas entre níveis de cinza.

Operações de Suavização

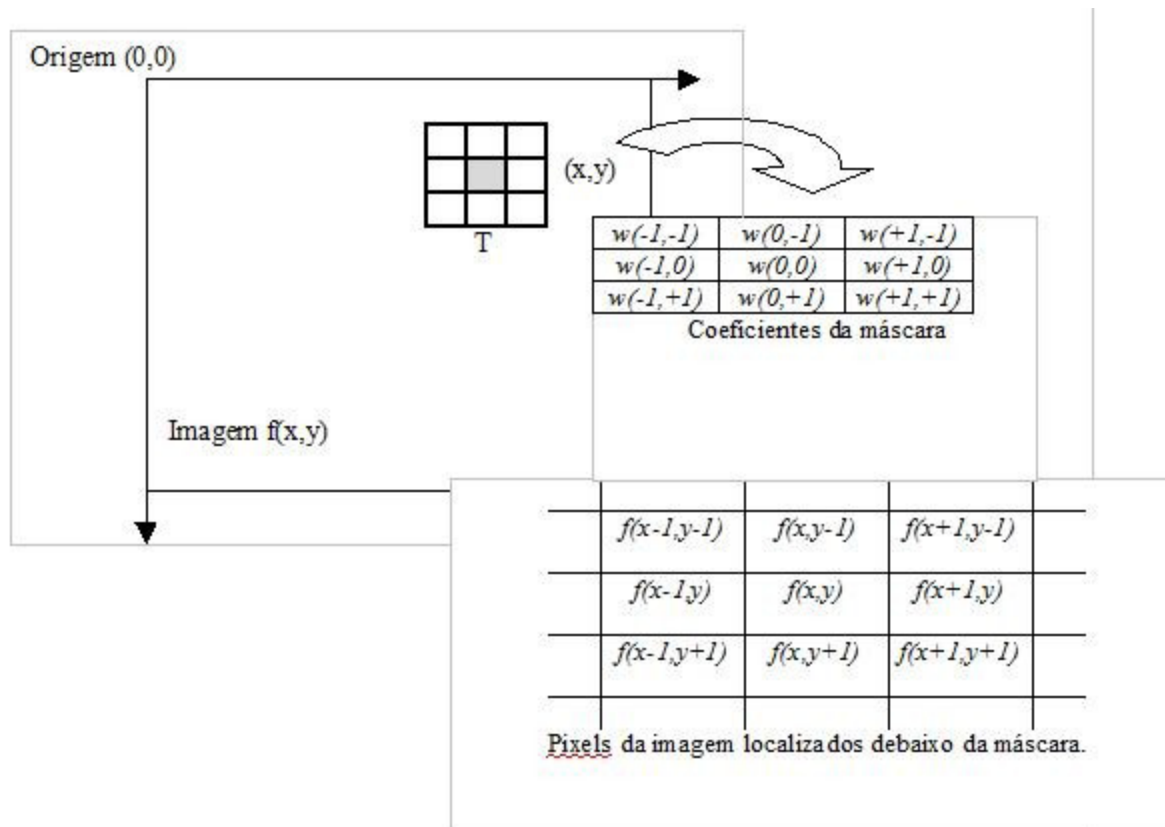


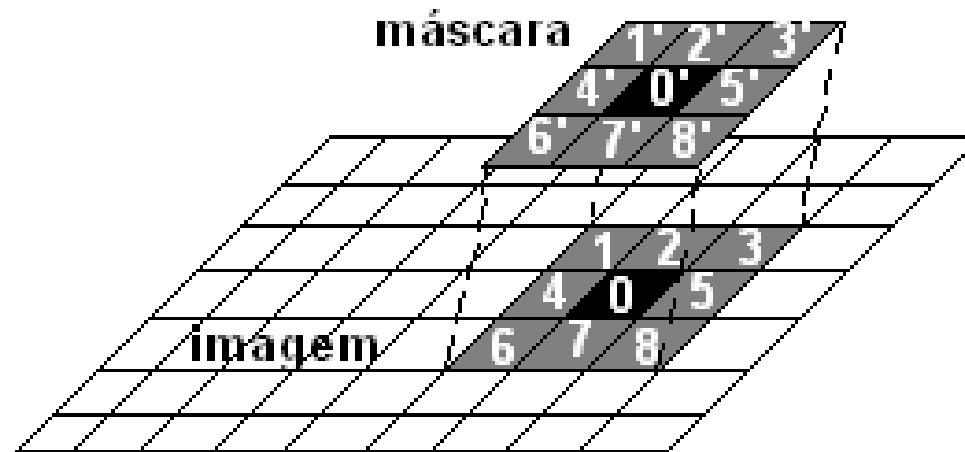
Figura 2: Representação do mecanismo da filtragem espacial.

Operações de Suavização



Na prática (máscara 3x3):

$$0 := 0.0' + 1.1' + 2.2' + 3.3' + 4.4' + 5.5' + 6.6' + 7.7' + 8.8'$$



Algoritmo caro

pode ser reduzido utilizando-se transformada de Fourier

Operações de Suavização



$$M_{3 \times 3} = 1/9^*$$

1	1	1
1	1	1
1	1	1

$$M_{5 \times 5} = 1/25^*$$

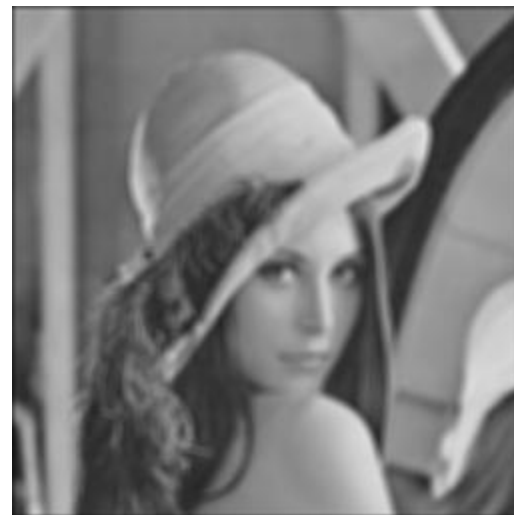
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1



Imagem original

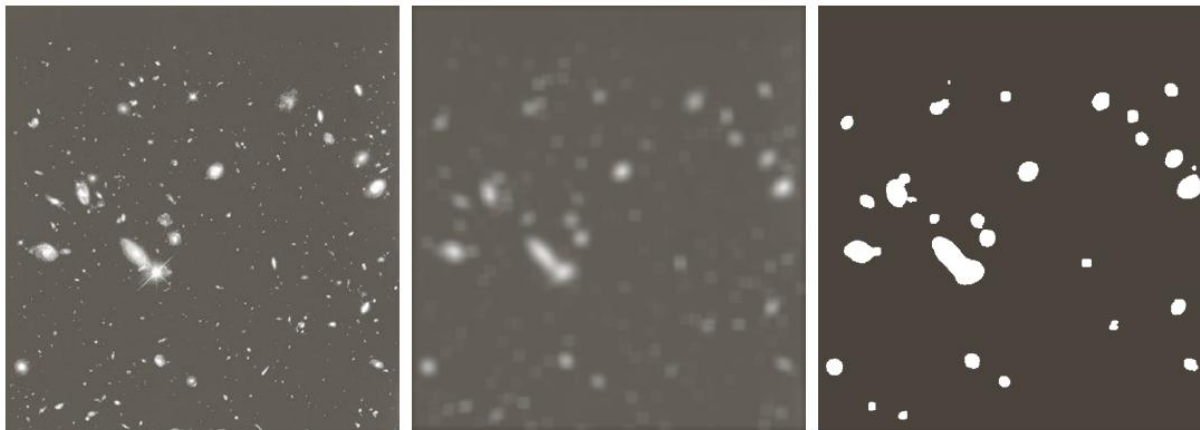


Blurred com masc 3x3



Blurred com masc 5x5

Operações de Suavização



a b c

FIGURE 3.34 (a) Image of size 528×485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15×15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Operações de Suavização



- Tamanhos típicos de máscara de filtragem 3x3, 5x5, 7x7, 9x9 pixels
- A máscara da média é boa para eliminar ruído, mas provoca um borramento da imagem.

Operações de Suavização



- Alguns métodos típicos:
 - Média com pesos: distância ao ponto central
 - Média com os k vizinhos mais próximos
 - Média com pesos espaciais para bordas + linhas
 - Média com vizinhos

Operações de Suavização



- Filtro da média
 - Se o ruído na imagem aparece como **erro aditivo, aleatório e descorrelacionado**, então os pixels afetados podem ser **substituídos por uma média local** para reduzir variações

$$M = \frac{1}{E} \sum_{i=1}^E p(i)$$

Janela $w \times w$ com $E = w^2$ elementos, $p(i)$ = NC's dos pixels

Operações de Suavização



- Exemplo

$$w_{3 \times 3} = \begin{bmatrix} 11 & 50 & 51 \\ 10 & 80 & 52 \\ 10 & 49 & 50 \end{bmatrix}$$

Diagram illustrating a 3x3 neighborhood operation. The value 80 is circled in red, and red arrows point from it to the value 40 and the text "tem pinta de ser um ruído" (looks like noise).

$$M = \frac{1}{9} [11 + 50 + 51 + 10 + 80 + 52 + 10 + 49 + 50]$$
$$= \frac{1}{9} \cdot 363 = 40,33 \Rightarrow P = p(5) = 40$$

Operações de Suavização



- Para contornar o problema de borramento:

$$p = p(5) = \begin{cases} M, & \text{se } |M - p(5)| > T \\ p(i) & \text{caso contrário} \end{cases}$$

Operações de Suavização



$$W_{3 \times 3} = \begin{bmatrix} 11 & 50 & 51 \\ 10 & 52 & 52 \\ 10 & 49 & 50 \end{bmatrix}$$

não parece ser diferente da vizinhança

$$M = 1/9 [11 + 50 + 51 + 10 + 52 + 10 + 49 + 50]$$

$$M = 37$$

Para:

$$T = 20$$

$$|37 - 52| > 20 ?$$

$$p(5) = 52$$

Para:

$$T = 10$$

$$|37 - 52| > 10 ?$$

$$p(5) = 37$$

Operações de Suavização



- Outras máscaras de cálculo da média

$$M_1 = 1/10 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$M_2 = 1/16 \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Operações de Suavização



$$W_{3 \times 3} = \begin{bmatrix} 11 & 50 & 51 \\ 10 & 80 & 52 \\ 10 & 49 & 50 \end{bmatrix}$$

$$M1 = 1/10 [160 + 11 + 50 + 51 + 10 + 52 + 10 + 49 + 50]$$

$$M1 = 44 \Rightarrow p(5) = 44$$

$$M1 = 1/16 [320 + 11 + 100 + 51 + 20 + 104 + 10 + 98 + 50]$$

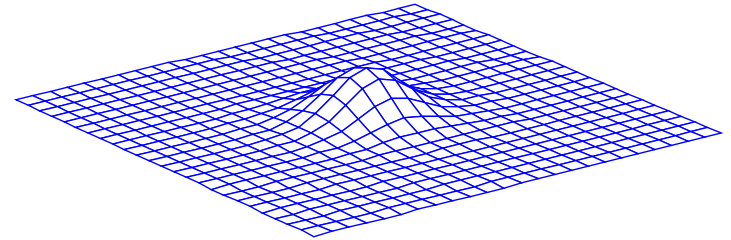
$$M1 = 48 \Rightarrow p(5) = 48$$

Filtro Gaussiano

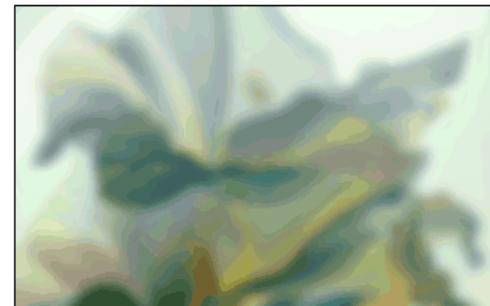
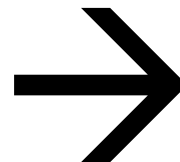
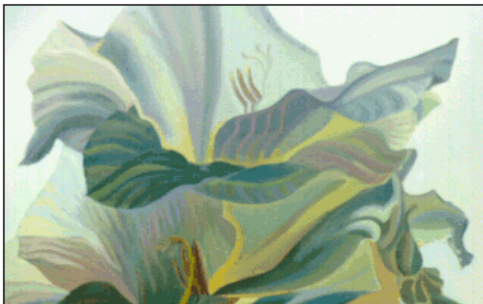


$$G_{\sigma}(x, y) = \frac{1}{2\sigma^2\pi} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

- Faz uma média ponderada com os pixels vizinhos
- Suaviza a imagem



From Computer Desktop Encyclopedia
© 1999 The Computer Language Co., Inc.



From Computer Desktop Encyclopedia
© 1999 The Computer Language Co., Inc.

Operações Gaussiano



fspecial (tipo, tamanho): cria filtros 2-D pré-definidos

- 'average': averaging filter
- 'disk' : circular averaging filter
- 'gaussian ' : Gaussian lowpass filter
- 'laplacian' : filter approximating the 2-D Laplacian operator
- 'log' : Laplacian of Gaussian filter
- 'motion': motion filter
- 'prewitt' : Prewitt horizontal edge-emphasizing filter
- 'sobel' : Sobel horizontal edge-emphasizing filter
- 'unsharp': unsharp contrast enhancement filter

Operações Gaussiano



imfilter(img, mask, opSaída) : convolução entre a imagem *img* e a máscara *mask*, *opSaída* indica o formato de saída, pode ser “*full*” ou “*same*”.

- Também pode ser utilizada *conv2* (convolução em duas dimensões) e *filter2*

Operações Gaussiano



```
img = imread('lenna.png');  
mask = fspecial('gaussian', [5 5]);  
nimg = imfilter(img, mask, 'same');  
Imshow(nimg);
```

Filtro da média com k vizinhos



- Filtro da média com os k vizinhos mais próximos
 - É um híbrido do método de filtragem pela média.
 - Utiliza a diferença absoluta entre o valor de cada elemento da máscara de filtragem e o ponto central para selecionar os k elementos que participarão do cálculo da média.

Filtro da média com k vizinhos



- Consegue-se redução do ruído com preservação de bordas
- $k \uparrow$ redução de ruído \uparrow preservação \downarrow
- $k \downarrow$ redução de ruído \downarrow preservação \uparrow
- Valores típicos de k: 2, 4, 6 e 8 ($w_{3 \times 3}$)
- $k=6 \rightarrow$ melhor realce

Filtro da média com k vizinhos



$$W_{3 \times 3} = \begin{bmatrix} 11 & 50 & 51 \\ 10 & 80 & 52 \\ 10 & 49 & 50 \end{bmatrix} \quad k = 6$$

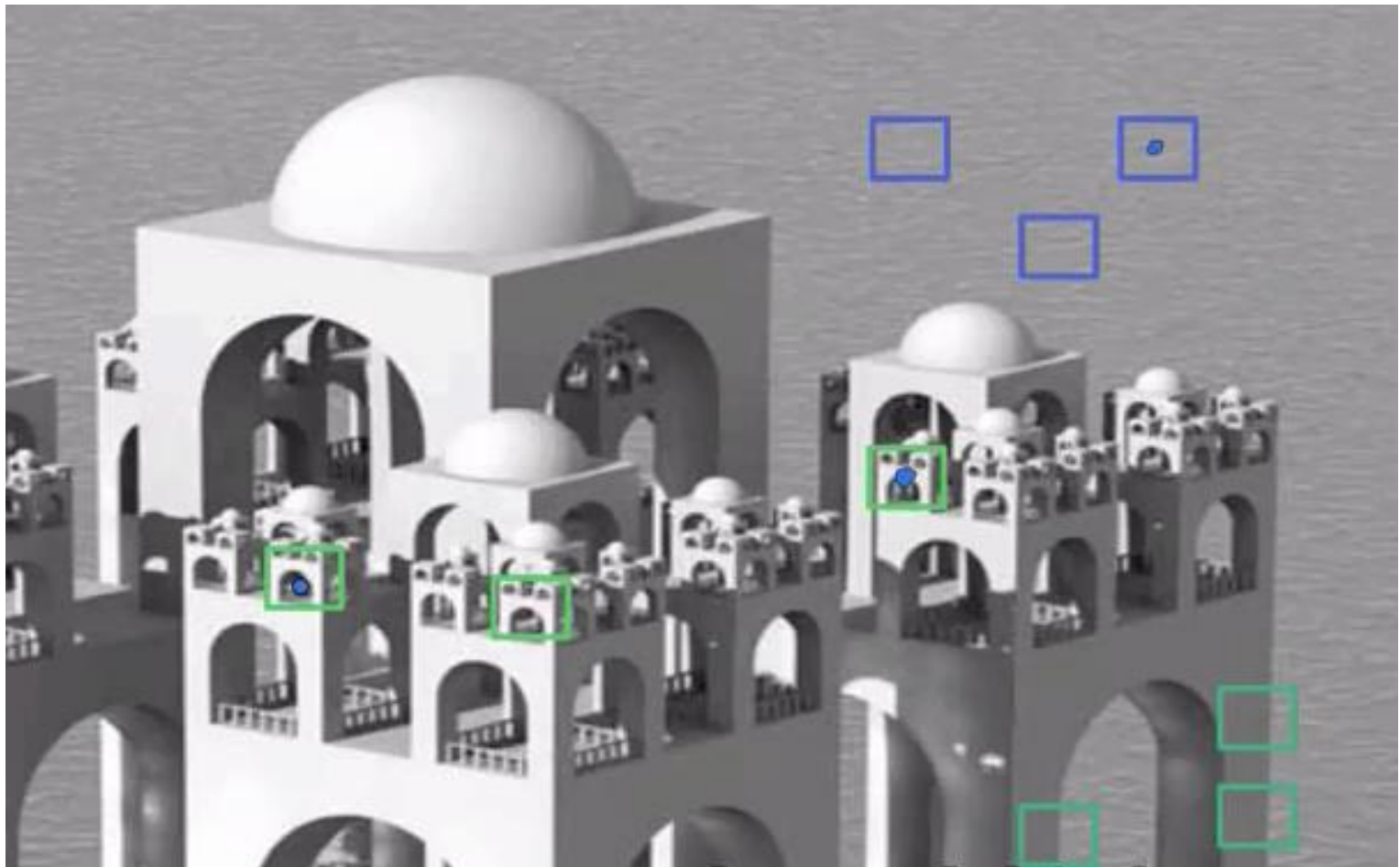
6 vizinhos mais próximos: 52, 51, 50, 50, 49 e 11

$$M1 = 1/7[80 + 52 + 51 + 50 + 50 + 49 + 11]$$

$$M1 = 343/7$$

$$p(5) = 49$$

Non local mean



Exercícios Propostos



1. Implementar o filtro da média com k vizinhos
2. Crie uma função que gere um máscara gaussiana.