

Computação Visual

Ciência da Computação

Prof. André Kishimoto
2024

Referências

Este material foi baseado e adaptado a partir das seguintes referências:

- **Processamento digital de imagens, 3ª ed.**
GONZALES, R. C.; WOODS, R. C.
Pearson Prentice Hall, 2010.
- **Introduction to Visual Computing: Core Concepts in Computer Vision, Graphics, and Image Processing**
MAJUDER, A.; GOPI, M.
CRC Press, 2018
- **Fundamentos da Imagem Digital (slides de aula)**
CRUNIVEL, L., 2012.

Interpolação

Interpolação de imagens

- Interpolação é um processo em que uma função **estima valores** que não foram medidos ou amostrados.
- É uma ferramenta usada em tarefas como ampliação, redução, rotação e correções geométricas.
- Neste material: aplicação de interpolação no redimensionamento de imagens (redução e ampliação).
 - Métodos de reamostragem (*resampling*) de imagens.

Interpolação por vizinho mais próximo

- *Nearest neighbor interpolation*
- O pixel mais próximo na imagem original é usado para determinar a cor do pixel na imagem interpolada.
- Técnica simples e rápida, porém resulta em imagens com aparência pixelada ou de baixa qualidade quando há um aumento significativo de escala.
 - Aumentar a escala de uma imagem: cada novo pixel na imagem maior é uma cópia do pixel mais próximo na imagem original.

Interpolação por vizinho mais próximo

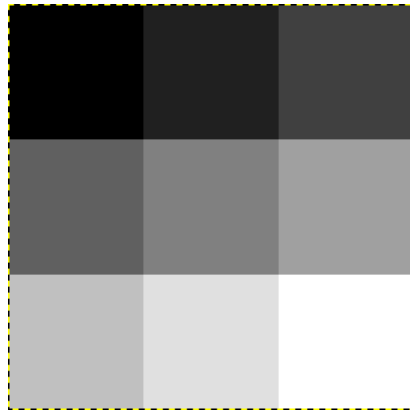


Imagem 3x3 pixels
width = 3
height = 3

(A borda pontilhada não faz parte da imagem)

Interpolação por vizinho mais próximo

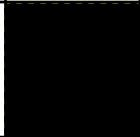
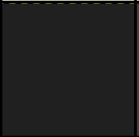







	0	1	2
0			
1			
2			

Imagem 3x3 pixels
width = 3
height = 3

Interpolação por vizinho mais próximo

	0	1	2
0	0	32	64
1	96	128	160
2	192	224	255

Imagem 3x3 pixels
width = 3
height = 3

Interpolação por vizinho mais próximo

	0	1	2
0	0	32	64
1	96	128	160
2	192	224	255

Aplicando uma escala de fator 2, dobramos o tamanho da imagem (3x3 para 6x6).

Interpolação por vizinho mais próximo

	0	1	2
0	0	32	64
1	96	128	160
2	192	224	255

input: 3x3

Quais intensidades devem ser atribuídas em cada pixel que não existe na imagem original (indicado em magenta)?

	0	1	2	3	4	5
0	0		32		64	
1						
2	96		128		160	
3						
4	192		224		255	
5						

output: 6x6

Interpolação por vizinho mais próximo

	0	1	2
0	0	32	64
1	96	128	160
2	192	224	255

input: 3x3

Aumentar a escala de uma imagem: cada novo pixel na imagem maior é uma cópia do pixel mais próximo na imagem original.

	0	1	2	3	4	5
0	0	0	32	32	64	64
1	0	0	32	32	64	64
2	96	96	128	128	160	160
3	96	96	128	128	160	160
4	192	192	224	224	255	255
5	192	192	224	224	255	255

output: 6x6

Interpolação por vizinho mais próximo

	0	1	2
0	0	32	64
1	96	128	160
2	192	224	255

input: 3x3

```
nearest_x = floor(output_col/scale)
nearest_y = floor(output_row/scale)
```

	0	1	2	3	4	5
0	0	0	32	32	64	64
1	0	0	32	32	64	64
2	96	96	128	128	160	160
3	96	96	128	128	160	160
4	192	192	224	224	255	255
5	192	192	224	224	255	255

output: 6x6

Interpolação por vizinho mais próximo

	0	1	2
0	0	32	64
1	96	128	160
2	192	224	255

input: 3x3

O que acontece quando aplicamos uma escala de fator 0.5 na imagem da direita?

	0	1	2	3	4	5
0	0	0	32	32	64	64
1	0	0	32	32	64	64
2	96	96	128	128	160	160
3	96	96	128	128	160	160
4	192	192	224	224	255	255
5	192	192	224	224	255	255

output: 6x6

Interpolação linear

- Um ponto V no segmento de linha V_0V_1 é dado por:

$$V = (1 - \alpha) \times V_0 + \alpha \times V_1$$

(sendo que $0.0 \leq \alpha \leq 1.0$)



Interpolação linear

- Um ponto V no segmento de linha V_1V_2 é dado por:

$$V = (1 - \alpha) \times V_0 + \alpha \times V_1$$

(sendo que $0.0 \leq \alpha \leq 1.0$)

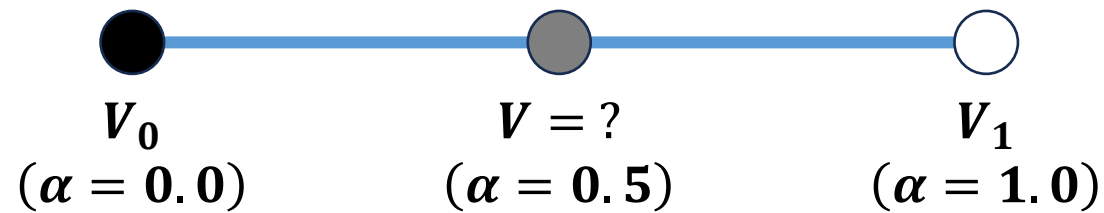


Interpolação linear

- Um ponto V no segmento de linha V_1V_2 é dado por:

$$V = (1 - \alpha) \times V_0 + \alpha \times V_1$$

(sendo que $0.0 \leq \alpha \leq 1.0$)



Interpolação linear

- Um ponto V no segmento de linha V_1V_2 é dado por:

$$V = (1 - \alpha) \times V_0 + \alpha \times V_1$$

(sendo que $0.0 \leq \alpha \leq 1.0$)



Interpolação linear

- Um ponto V no segmento de linha V_1V_2 é dado por:

$$V = (1 - \alpha) \times V_0 + \alpha \times V_1$$

(sendo que $0.0 \leq \alpha \leq 1.0$)

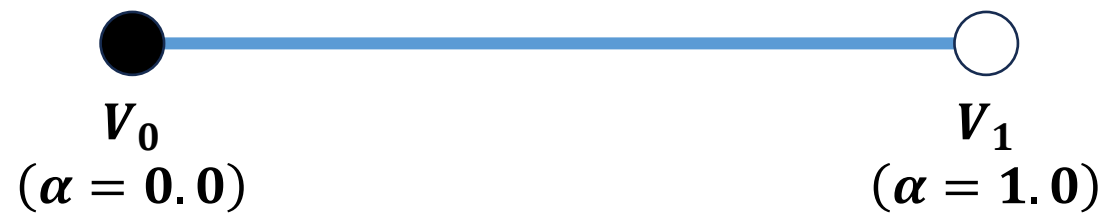


Interpolação linear

- Um ponto V no segmento de linha V_1V_2 é dado por:

$$V = (1 - \alpha) \times V_0 + \alpha \times V_1$$

(sendo que $0.0 \leq \alpha \leq 1.0$)

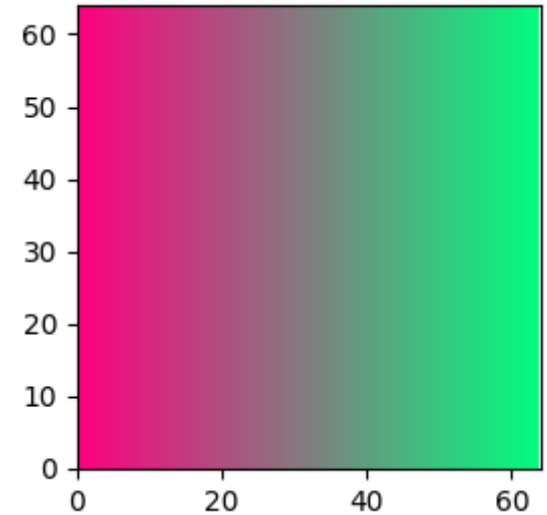


```
float lerp(float v0, float v1, float alpha) {  
    return (1.0 - alpha) * v0 + alpha * v1  
}
```

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def lerp(v0, v1, alpha):
5     return (1.0 - alpha) * v0 + alpha * v1
6
7 def gradient(width, height, r_min, r_max, g_min, g_max, b_min, b_max):
8     img_array = np.zeros((height, width, 3), dtype=np.float32)
9     step = 1 / width
10
11     for row in range(height):
12         for col in range(width):
13             alpha = col * step
14             img_array[row, col, 0] = lerp(r_min, r_max, alpha)
15             img_array[row, col, 1] = lerp(g_min, g_max, alpha)
16             img_array[row, col, 2] = lerp(b_min, b_max, alpha)
17
18     return img_array
19
20 def main():
21     WIDTH = 64
22     HEIGHT = 64
23     img_array = gradient(WIDTH, HEIGHT, 1.0, 0.0, 0.0, 1.0, 0.5, 0.5)
24     img_procedural = np.array(img_array)
25
26     plt.figure(figsize=(5, 3))
27     plt.axis([0, WIDTH, 0, HEIGHT])
28     plt.imshow(img_procedural, cmap=plt.cm.gray)
29     plt.show()
30
31 if __name__ == "__main__":
32     main()

```



Interpolação bilinear

- Essa técnica usa os quatro vizinhos mais próximos para estimar a intensidade de uma determinada posição.
- Usa a média ponderada dos quatro pixels com base na distância do pixel de destino em relação aos pixels vizinhos.

Para a interpolação bilinear* o valor atribuído é obtido utilizando a equação:

$$v(x, y) = ax + by + cxy + d \quad (2.4-6)$$

onde os quatro coeficientes são determinados a partir das quatro equações de quatro incógnitas que podem ser escritas utilizando os quatro vizinhos mais próximos do ponto (x, y) .

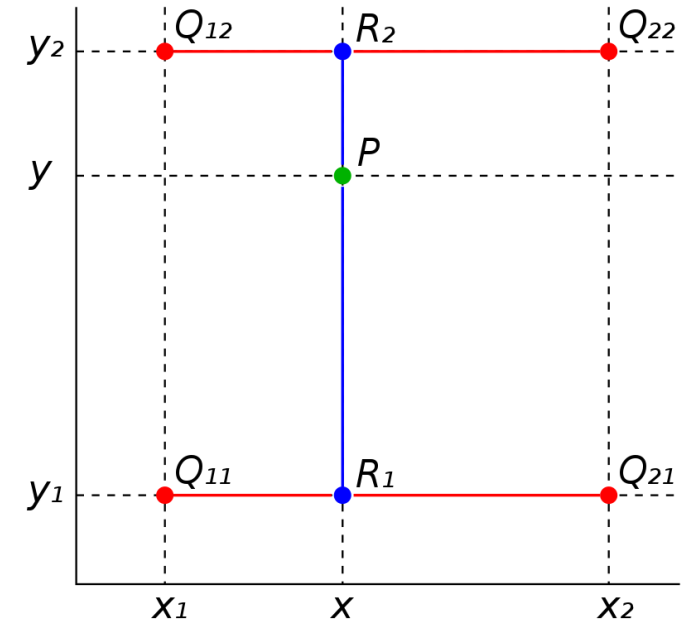
Interpolação bilinear

- Interpolação na horizontal (R_1 e R_2) seguida de interpolação na vertical (P).

$$F(R_1) = (1 - \alpha) \times Q_{11} + \alpha \times Q_{21}$$

$$F(R_2) = (1 - \alpha) \times Q_{12} + \alpha \times Q_{22}$$

$$F(P) = (1 - \beta) \times F(R_1) + \beta \times F(R_2)$$



The four red dots show the data points and the green dot is the point at which we want to interpolate.

Interpolação bilinear

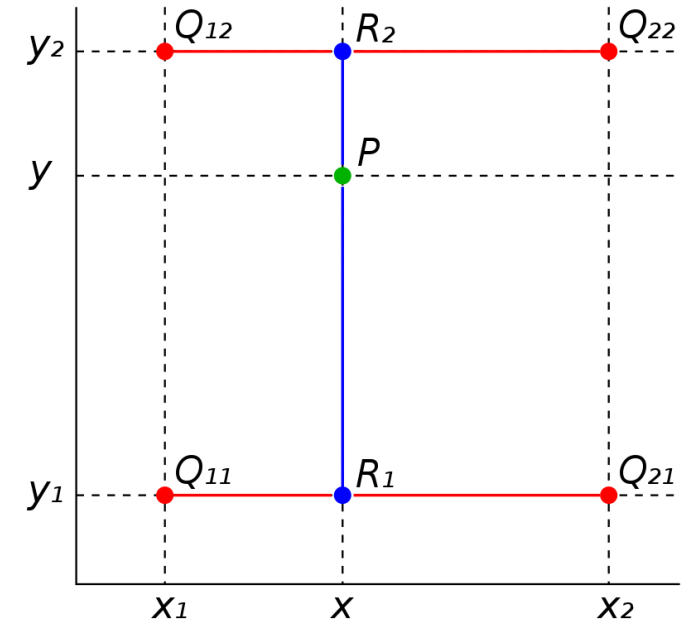
- Interpolação na horizontal (R_1 e R_2) seguida de interpolação na vertical (P).

$$F(R_1) = (1 - \alpha) \times Q_{11} + \alpha \times Q_{21}$$

$$F(R_2) = (1 - \alpha) \times Q_{12} + \alpha \times Q_{22}$$

$$F(P) = (1 - \beta) \times F(R_1) + \beta \times F(R_2)$$

```
lerp(  
    lerp(q11, q21, alpha),  
    lerp(q12, q22, alpha),  
    beta  
)
```



The four red dots show the data points and the green dot is the point at which we want to interpolate.

Filtragem Espacial

Filtragem espacial

- A filtragem espacial é uma operação realizada sob um pixel e sua vizinhança.
- Filtrar: aceitar (passar) ou rejeitar certos componentes de frequência.
 - Aceitar baixas frequências: filtro passa-baixa.
 - Aceitar altas frequências: filtro passa-alta.
- Filtros espaciais também são chamados de:
 - Máscaras
 - Kernels
 - Janelas
 - Templates

Filtragem espacial

- A operação de filtragem **cria um novo pixel** com coordenadas iguais às coordenadas do **centro da vizinhança**.
- O **valor desse pixel** é o **resultado** do processo de filtragem espacial.
- Uma **imagem processada** (filtrada) é gerada à medida que o centro do filtro percorre cada pixel da imagem de entrada.

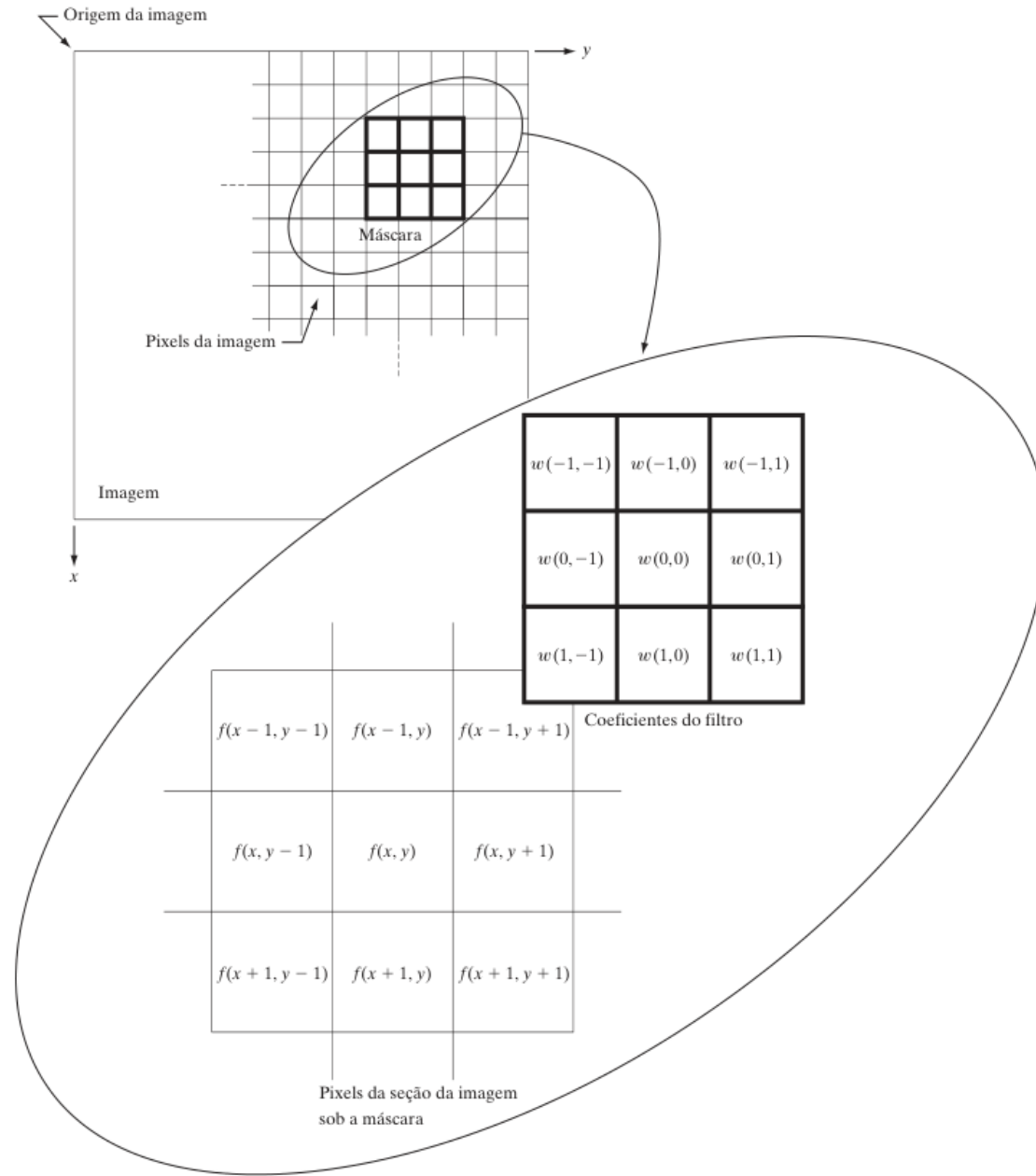
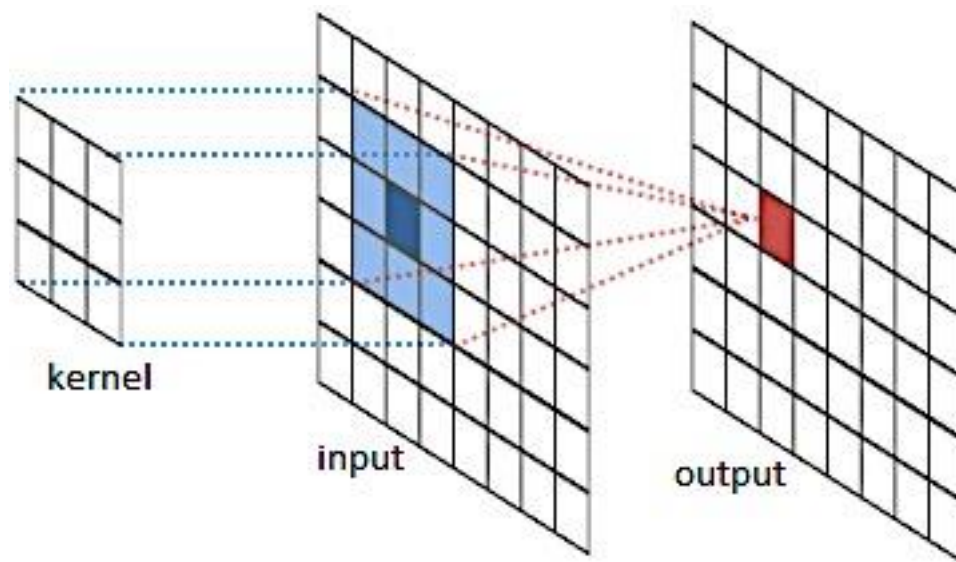


Figura 3.28 O funcionamento da filtragem espacial linear utilizando uma máscara 3×3 . A forma escolhida para expressar as coordenadas dos coeficientes da máscara simplifica a escrita de expressões para a filtragem linear.

Filtragem espacial



Tipos de filtros espaciais

- Os filtros que usaremos a partir de agora são classificados como:
- **Filtro passa-baixa:** reduzem ruídos de uma imagem digital; apresentam o efeito de borramento (blur/desfoque) da imagem processada.
- **Filtro passa-alta:** realçam características presentes em uma imagem (bordas, linhas curvas, manchas, etc.).

Filtros espaciais de suavização

- Os filtros de suavização (passa-baixa) são usados para **borramento** e **redução de ruído**.
- Principais características da suavização:
 - Conexão de pequenas descontinuidades nas regiões com transições abruptas na imagem.
 - Eliminação das altas frequências (bordas) da imagem.
 - A saída desse tipo de filtro linear é a média dos pixels contidos na vizinhança da máscara de filtragem.

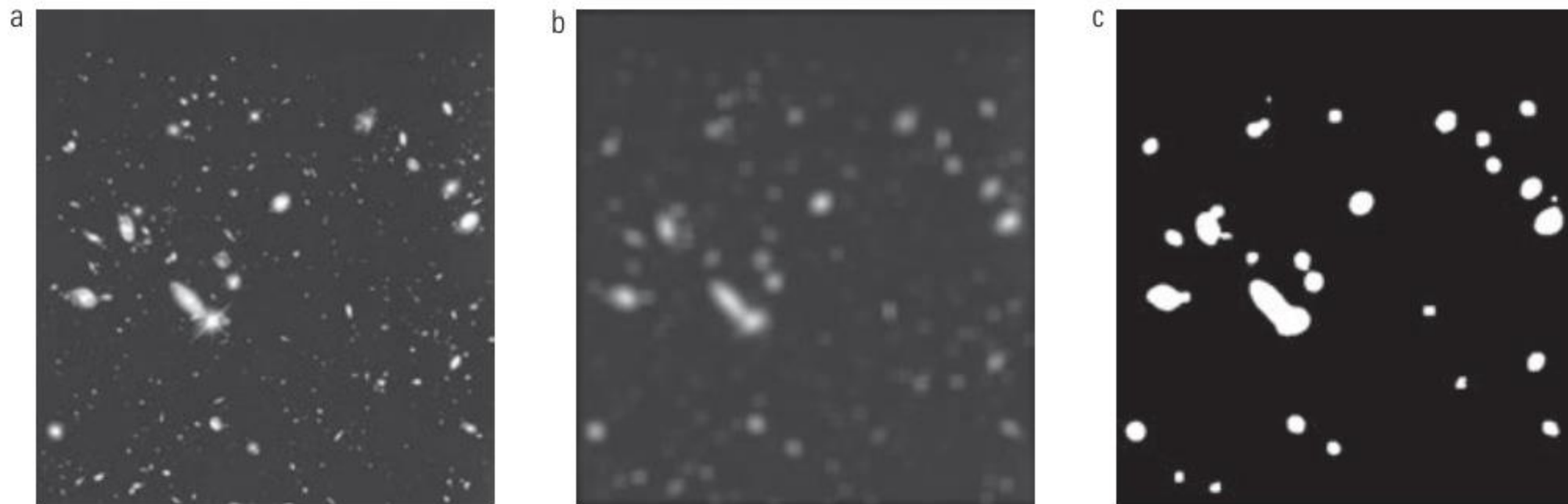


Figura 3.34 (a) Imagem de 528×485 pixels obtida com o telescópio espacial Hubble. (b) Imagem filtrada com um filtro de média de tamanho 15×15 . (c) Resultado da limiarização de (b). (Imagem original: cortesia da Nasa.)

Filtros espaciais de suavização

- Geralmente, os filtros de suavização são **quadrados** e de **tamanho ímpar** (centro de simetria).
- Podemos usar filtros de suavização para:
 - Calcular a **média aritmética** dos pixels filtrados.
 - Calcular a **média ponderada** dos pixels filtros.

Filtros espaciais de suavização

a

1	1	1
1	1	1
1	1	1

$\frac{1}{9} \times$

b

1	2	1
2	4	2
1	2	1

$\frac{1}{16} \times$

Figura 3.32 Duas máscaras 3×3 (de média) para suavização. A constante de multiplicação diante de cada máscara é igual a 1 dividido pela soma dos valores de seus coeficientes, o que é necessário para calcular uma média.

Filtros espaciais de suavização

- Para uma máscara de suavização de tamanho $m \times n$, onde todos os elementos do filtro são iguais a 1, temos que usar uma **constante de normalização** igual a:

$$\frac{1}{m \times n}$$

- Para máscaras de suavização ponderadas, a constante de normalização é formada pela **soma dos pesos** dos elementos do filtro.
 - De preferência, um número inteiro de potência 2.

Filtros espaciais de suavização

- O efeito do processo de suavização **não depende** exclusivamente **dos valores** do filtro.
 - O **tamanho do filtro** também deve ser levado em consideração.
 - Quanto **maior** o filtro, **maior será** o resultado do **borramento** da imagem filtrada.

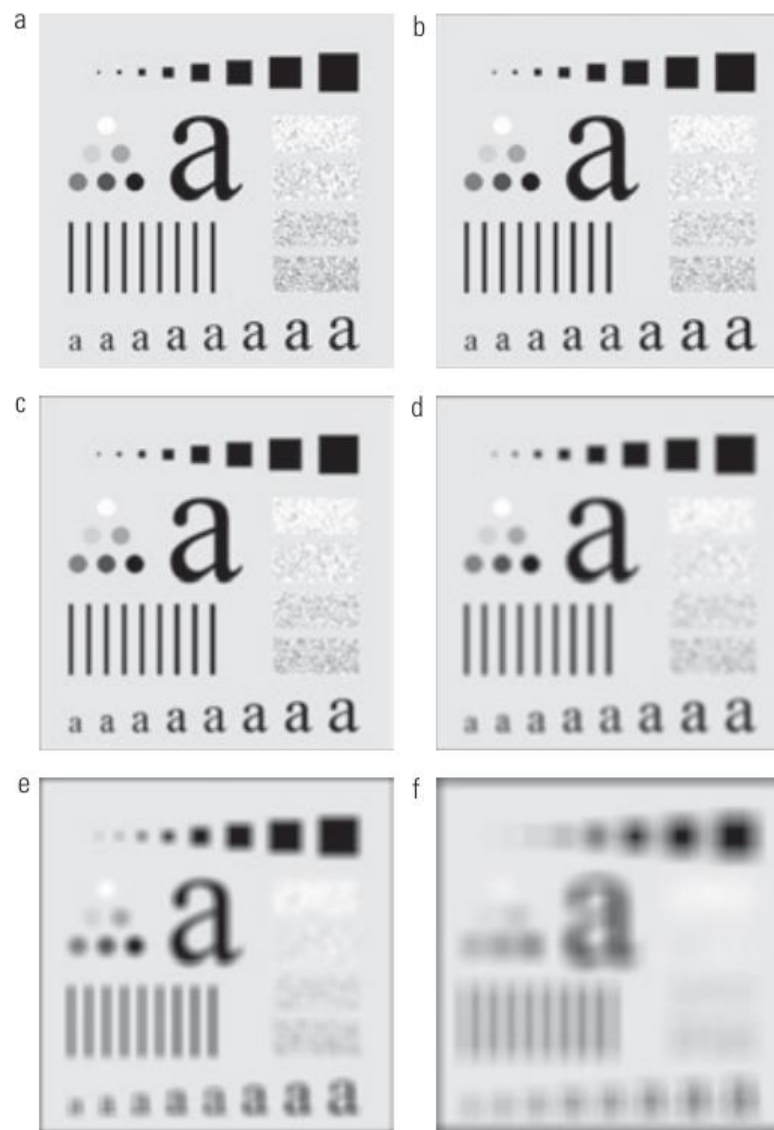


Figura 3.33 (a) Imagem original de 500×500 pixels. (b) a (f) Resultados da suavização com filtros de média, quadrados, de tamanhos $m = 3, 5, 9, 15$ e 35 , respectivamente. Os quadrados pretos no alto das imagens têm tamanhos $3, 5, 9, 15, 25, 35, 45$ e 55 pixels, respectivamente; suas bordas estão distantes 25 pixels umas das outras. O tamanho das

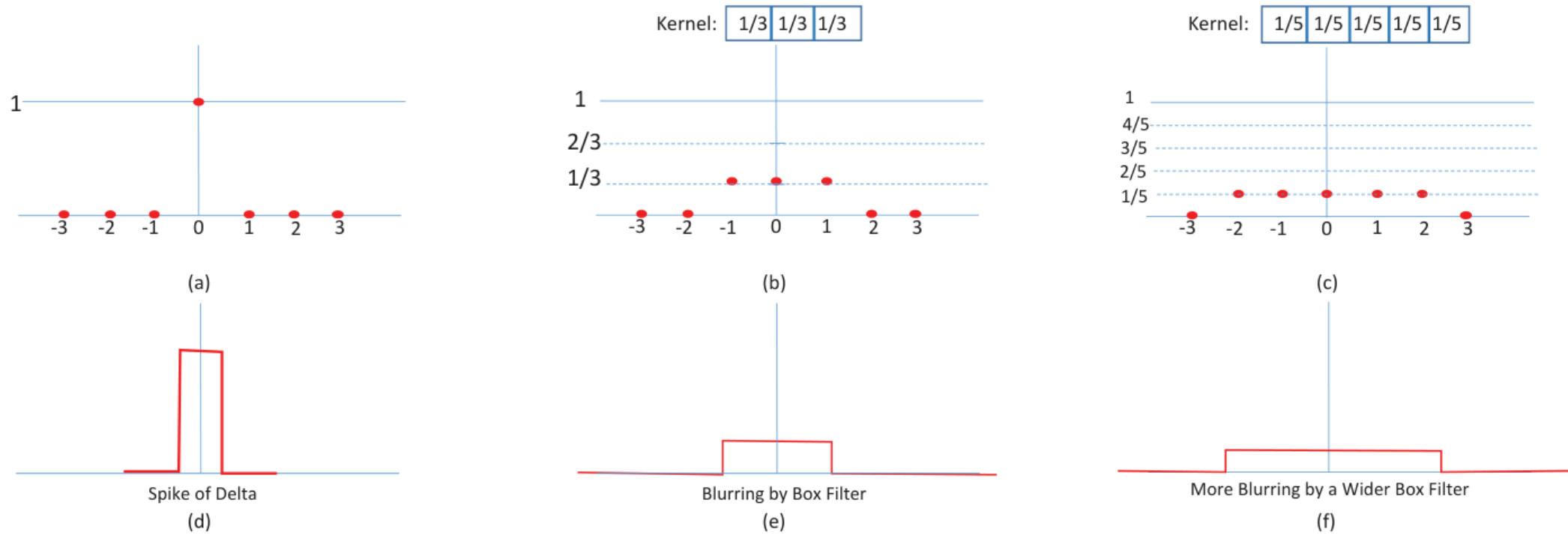


Figure 3.5. This figure shows how to design a blur filter by considering just the simple δ function. The top row shows the concept at work in a digital representation while the bottom row shows the analog counterpart. (a) shows a δ . (b) shows the impulse response of a blurring system, i.e. the fate of δ when passed through a blurring system. Instead of having a spike of 1 at 0, it now has a value of $1/3$ at each of $-1, 0$ and 1 . The width of the filter is also often referred to as the support of the kernel or filter. If the width of the blur is wider, it is said that the support of the filter is now higher.

Filtros espaciais de suavização

Filtros não-lineares de suavização

- Também conhecidos como **filtros de estatística de ordem**.
- São filtros espaciais que se baseiam na **ordenação** dos pixels contidos na região filtrada na imagem.
- Os filtros mais conhecidos nessa categoria são:
 - Filtro de mediana
 - Filtro de moda
 - Filtro de mínimo
 - Filtro de máximo

Filtros espaciais de suavização

Filtro de mediana

- Substitui o valor do pixel central pelo valor da mediana da região filtrada.
- Por exemplo:
 - Dado um filtro mediano de tamanho 3x3, o primeiro passo é ordenar os nove valores de intensidade dos pixels cobertos pelo filtro.
 - O valor da mediana será o valor armazenado na quinta posição dos valores ordenados.
 - Esse valor é substituído na posição (x,y) da região da imagem processada.

Filtros espaciais de suavização

Filtro de mediana

- É particularmente mais eficaz na redução de ruídos do tipo **sal e pimenta** (*salt and pepper*), em comparação ao filtro de média.
 - Aparência do ruído: pontos brancos e pretos sobrepostos em uma imagem.

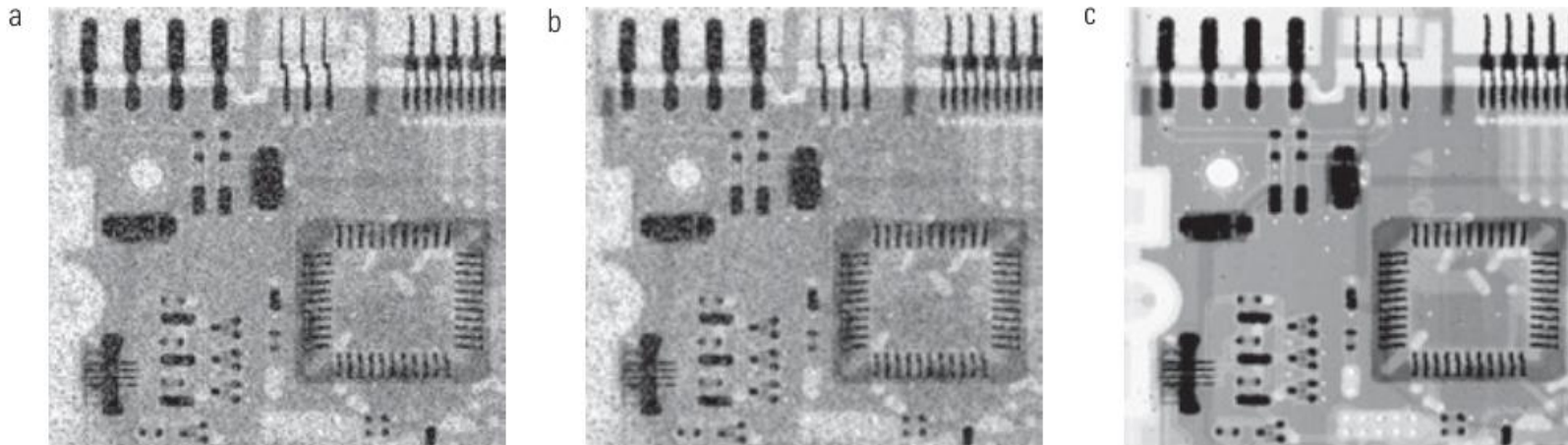


Figura 3.35 (a) Imagem de raios X de uma placa de circuito corrompida pelo ruído sal e pimenta. (b) Redução de ruído com um filtro de média 3×3 . (c) Redução de ruído com um filtro de mediana 3×3 . (Imagem original: cortesia do Sr. Joseph E. Pascente, Lixi, Inc.)



CC BY-SA 4.0 DEED

Atribuição-Compartilhagual 4.0 Internacional

Canonical URL : <https://creativecommons.org/licenses/by-sa/4.0/>

[See the legal code](#)


Você tem o direito de:


Compartilhar — copiar e redistribuir o material em qualquer suporte ou formato para qualquer fim, mesmo que comercial.

Adaptar — remixar, transformar, e criar a partir do material para qualquer fim, mesmo que comercial.

O licenciante não pode revogar estes direitos desde que você respeite os termos da licença.

De acordo com os termos seguintes:

 **Atribuição** — Você deve dar o [crédito apropriado](#), prover um link para a licença e [indicar se mudanças foram feitas](#). Você deve fazê-lo em qualquer circunstância razoável, mas de nenhuma maneira que sugira que o licenciante apoia você ou o seu uso.

 **Compartilhagual** — Se você remixar, transformar, ou criar a partir do material, tem de distribuir as suas contribuições sob a [mesma licença](#) que o original.

Sem restrições adicionais — Você não pode aplicar termos jurídicos ou [medidas de caráter tecnológico](#) que restrinjam legalmente outros de fazerem algo que a licença permita.



CC BY-SA 4.0 DEED

Attribution-ShareAlike 4.0 International

Canonical URL : <https://creativecommons.org/licenses/by-sa/4.0/>

[See the legal code](#)


You are free to:


Share — copy and redistribute the material in any medium or format for any purpose, even commercially.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

 **Attribution** — You must give [appropriate credit](#), provide a link to the license, and [indicate if changes were made](#). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

 **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the [same license](#) as the original.

No additional restrictions — You may not apply legal terms or [technological measures](#) that legally restrict others from doing anything the license permits.

Notices: