

Filtragem 2D

Visão Computacional em Robótica (BLU3040)

Prof. Marcos Matsuo (marcos.matsuo@ufsc.br)

Universidade Federal de Santa Catarina - UFSC

① Operações Espaciais

② Filtragem 2D

- Correlação 2D
- Convolução 2D

③ Aplicações da Filtragem Espacial

- Suavização
- Detecção de Borda
 - Algoritmo Básico

① Operações Espaciais

② Filtragem 2D

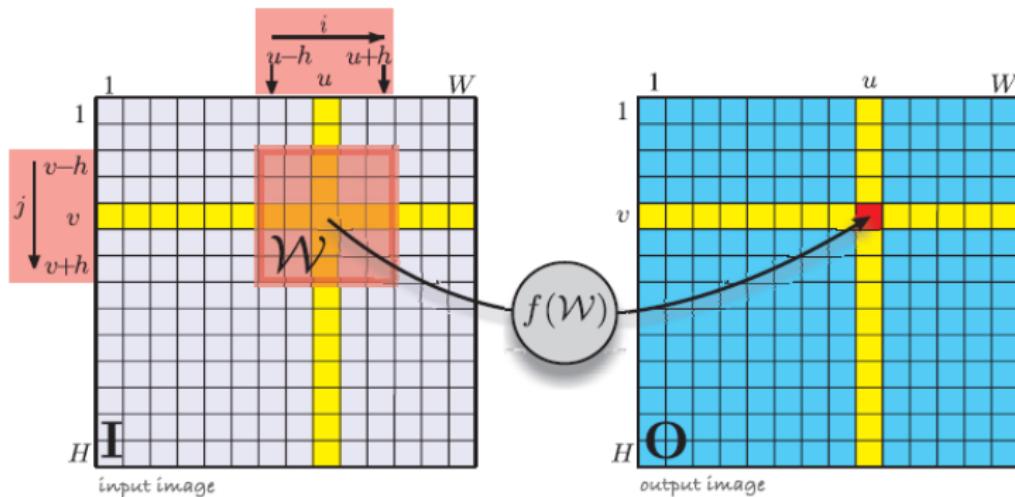
- Correlação 2D
- Convolução 2D

③ Aplicações da Filtragem Espacial

- Suavização
- Detecção de Borda
 - Algoritmo Básico

Operações Espaciais

Cada *pixel* (v, u) da imagem de saída **O** é resultado do processamento de todos os *pixels* em uma dada janela \mathcal{W} da imagem de entrada **I**.



Características da janela \mathcal{W}

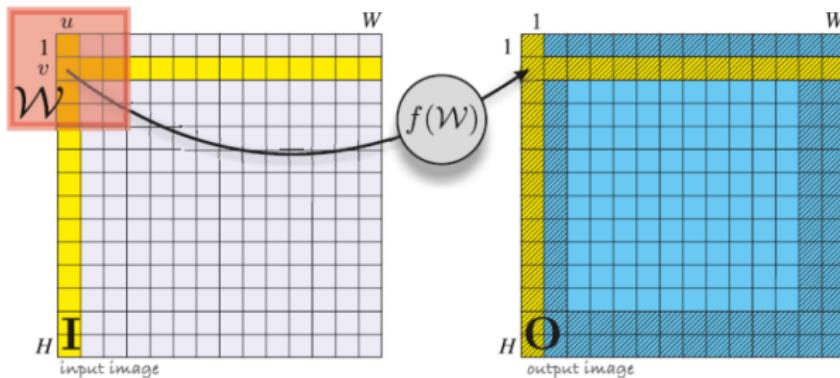
- Quadrada.
- Denota uma região de *pixels*.
- Centrada no *pixel* de entrada (v, u) .
- Distância entre o *pixel* central e a borda da janela é dado por h .
- $h \in \mathbb{Z}^+$ (isto é, h é número inteiro não negativo).
- A dimensão da janela é $w \times w$, onde $w = 2h + 1$ é sempre ímpar.

Operações Espaciais

Problema: Nas regiões de borda da imagem de entrada a janela \mathcal{W} abrange regiões sem *pixels*.

Soluções:

- Não calcular $f(\mathcal{W})$ nas bordas.
- Assumir que a imagem é cercada por *pixels* com valor zero.
- Replicar os *pixels* das bordas "para fora".



① Operações Espaciais

② Filtragem 2D

- Correlação 2D
- Convolução 2D

③ Aplicações da Filtragem Espacial

- Suavização
- Detecção de Borda
 - Algoritmo Básico

Filtragem 2D

Operações Espaciais

Uma operação muito importante em processamento de imagem é a **filtragem 2D**.

Um exemplo de filtragem 2D é a **operação de média**.

The diagram shows a 7x7 input image I with values ranging from 0.05 to 0.81. A 3x3 window is centered at position $(u=4, v=5)$, highlighted by a blue border. The window's center cell is labeled W . An arrow points from this window to a box labeled "Calcula média".

	u														
								\longrightarrow							
								v							
1	0.27	0.12	0.19	0.81	0.75	0.62	0.75								
2	0.31	0.35	0.42	0.38	0.23	0.30	0.12								
3	0.38	0.62	0.19	0.27	0.35	0.31	0.19								
4	0.75	0.38	0.23	0.30	0.81	0.42	0.30								
5	0.35	0.42	0.19	0.75	0.19	0.05	0.62								
6	0.23	0.27	0.62	0.81	0.42	0.23	0.12								
7	0.27	0.30	0.75	0.35	0.75	0.81	0.19								

Imagen de entrada I

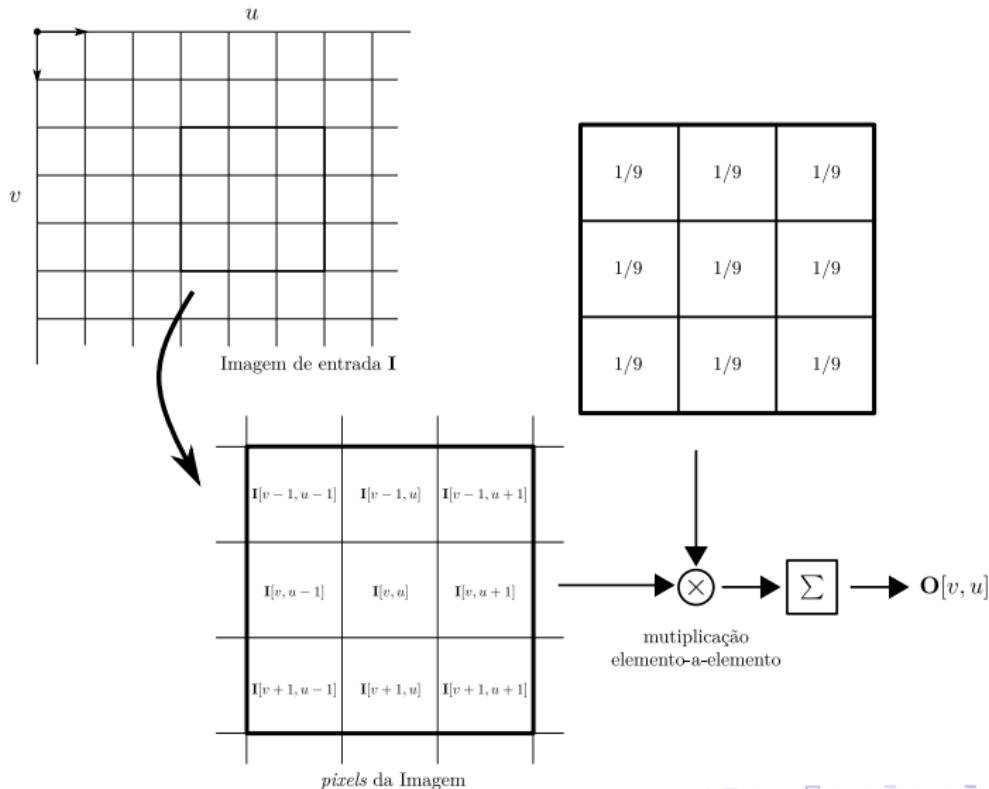
The diagram shows the output image O resulting from the averaging operation. Only the value at position $(u=4, v=5)$ is explicitly shown as 0.35, while all other values are 0.00. An arrow points from the "Calcula média" box to this cell.

	u														
								\longrightarrow							
								v							
1															
2				0.32	0.38	0.40	0.45	0.40							
3			0.40	0.35	0.35										
4															
5															
6															
7															

Imagen de saída O

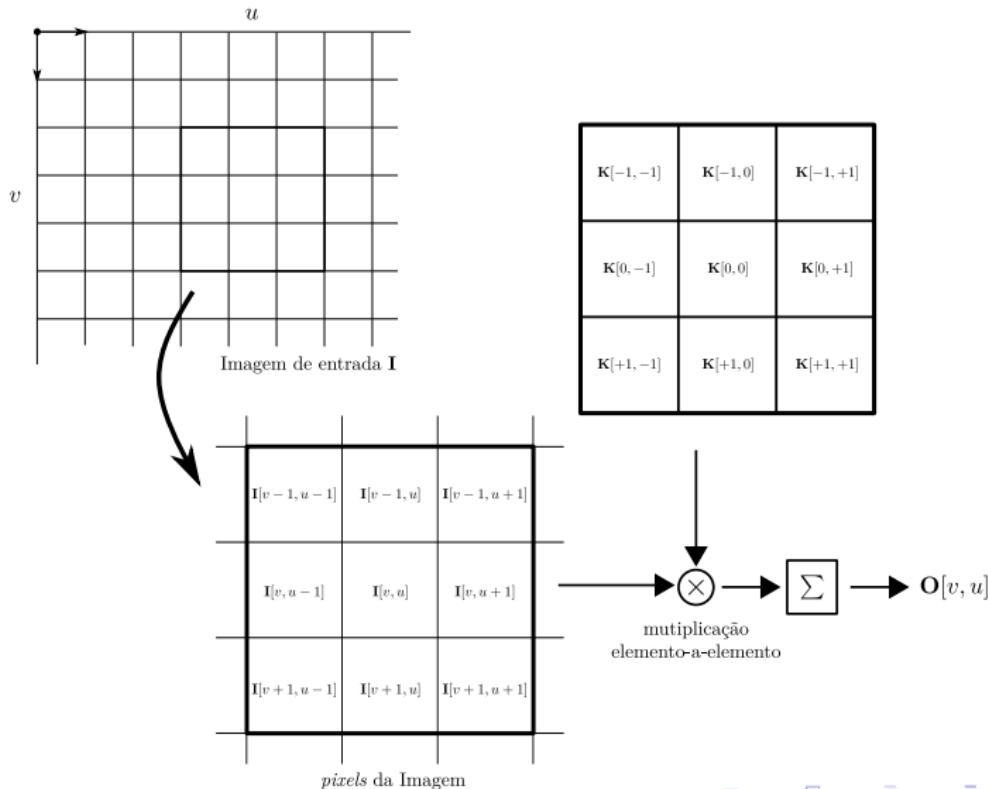
Filtragem 2D

Operações Espaciais



Filtragem 2D

Operações Espaciais



Filtragem 2D

Operações Espaciais

Matematicamente, cada *pixel* (v, u) da imagem de saída \mathbf{O} é computado como

$$\begin{aligned}\mathbf{O}(v, u) &= \sum_{(i,j) \in \mathcal{W}} \mathbf{I}(v + i, u + j) \mathbf{K}(i, j) \\ &= \sum_{j=-1}^{+1} \sum_{i=-1}^{+1} \mathbf{I}(v + i, u + j) \mathbf{K}(i, j)\end{aligned}$$

onde $\mathbf{K}(i, j) = 1/9$ para $i, j \in [-1, +1]$.

Filtragem 2D

Operações Espaciais

Na literatura da área a **filtragem 2D** é apresentada de duas formas distintas.

Filtragem
espacial

$$\left\{ \begin{array}{l} \text{Correlação 2D} \\ \mathbf{O}[v, u] = \sum_{j=-h}^{+h} \sum_{i=-h}^{+h} \mathbf{I}[v + j, u + i] \mathbf{K}[j, i] \end{array} \right.$$
$$\left\{ \begin{array}{l} \text{Convolução 2D} \\ \mathbf{O}[v, u] = \sum_{j=-h}^{+h} \sum_{i=-h}^{+h} \mathbf{I}[v - j, u - i] \mathbf{K}[j, i] \end{array} \right.$$

Filtragem 2D

Operações Espaciais

- No Matlab a operação de **correlação 2D** é realizada através da função `filter2()`.
- Para a operação de **convolução 2D** tem-se a função `conv2()`.
- Os resultados numéricos da correlação e da convolução são iguais se o filtro **K** (também chamado de *kernel*) for simétrico.

① Operações Espaciais

② Filtragem 2D

- Correlação 2D
- Convolução 2D

③ Aplicações da Filtragem Espacial

- Suavização
- Detecção de Borda
 - Algoritmo Básico

Correlação 2D

Filtragem Espacial

- Considere a operação de **correlação 2D**

$$\mathbf{O}(v, u) = \sum_{j=-h}^{+h} \sum_{i=-h}^{+h} \mathbf{I}(v + i, u + j) \mathbf{K}(i, j)$$

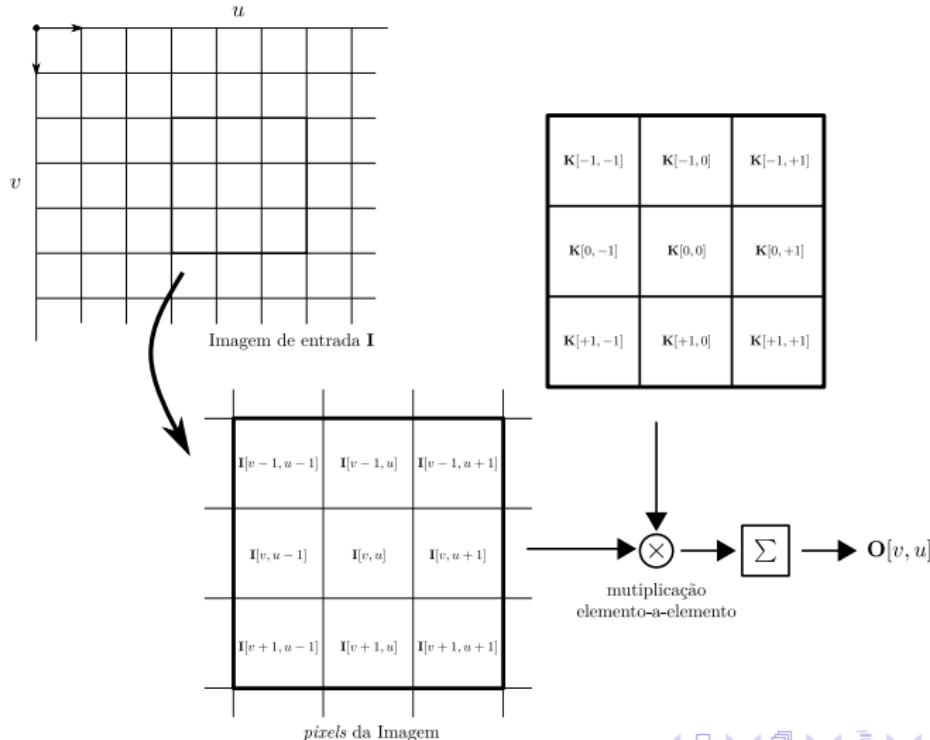
- Para fins de ilustração, assuma $h = 1$ (implicando em uma janela \mathcal{W} de dimensão 3×3). Neste caso, tem-se

$$\mathbf{O}(v, u) = \sum_{j=-1}^{+1} \sum_{i=-1}^{+1} \mathbf{I}(v + i, u + j) \mathbf{K}(i, j)$$

Correlação 2D

Filtragem Espacial

Cálculo do pixel (v, u) da imagem de saída \mathbf{O}



Exemplo 1: Correlação 2D

Filtragem Espacial

Enunciado: Crie no Matlab um imagem de entrada **I** de dimensão 8×8 , com pixels do tipo `uint8` com valores iguais a 100. Na sequência filtre a imagem utilizando um kernel **K** de dimensão 3×3 , com cada elemento do kernel com valor igual a 1. Para fins de comparação, realize a filtragem a função `filter2` de duas formas. Primeiro, considere preenchimento por zero nas regiões de borda e, depois, desconsidere a região de borda.

Exemplo 1: Correlação 2D

Filtragem Espacial

```
1 % imagem de entrada
2 I = 100*ones(8,8, 'uint8');
3
4 % kernel
5 K = ones(3,3);
6
7 % correlação 2D (com preenchimento de zeros nas regiões
8 % de borda)
9 I2 = filter2(K,I, 'same');
10
11 % correlação 2D (com descarte da região de borda)
12 I3 = filter2(K,I, 'valid');
```

① Operações Espaciais

② Filtragem 2D

- Correlação 2D
- Convolução 2D

③ Aplicações da Filtragem Espacial

- Suavização
- Detecção de Borda
 - Algoritmo Básico

Convolução 2D

Filtragem Espacial

- Considere a operação de **convolução 2D**

$$\mathbf{O}(v, u) = \sum_{j=-h}^{+h} \sum_{i=-h}^{+h} \mathbf{I}(v - i, u - j) \mathbf{K}(i, j)$$

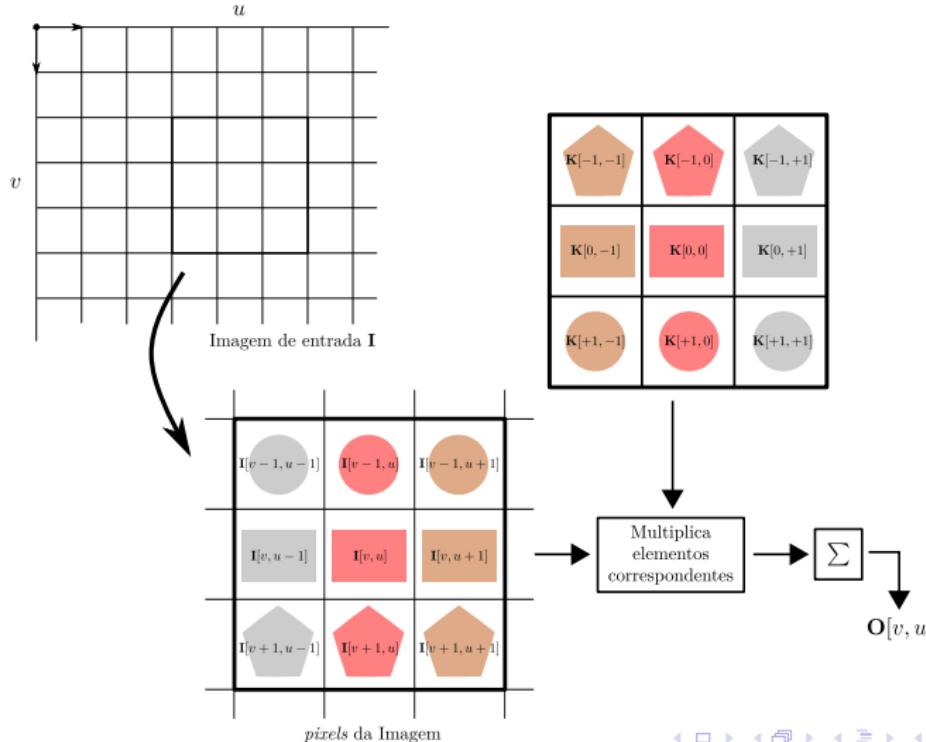
- Para fins de ilustração, assuma $h = 1$ (implicando em uma janela \mathcal{W} de dimensão 3×3). Neste caso, tem-se

$$\mathbf{O}(v, u) = \sum_{j=-1}^{+1} \sum_{i=-1}^{+1} \mathbf{I}(v - i, u - j) \mathbf{K}(i, j)$$

Convolução 2D

Filtragem Espacial

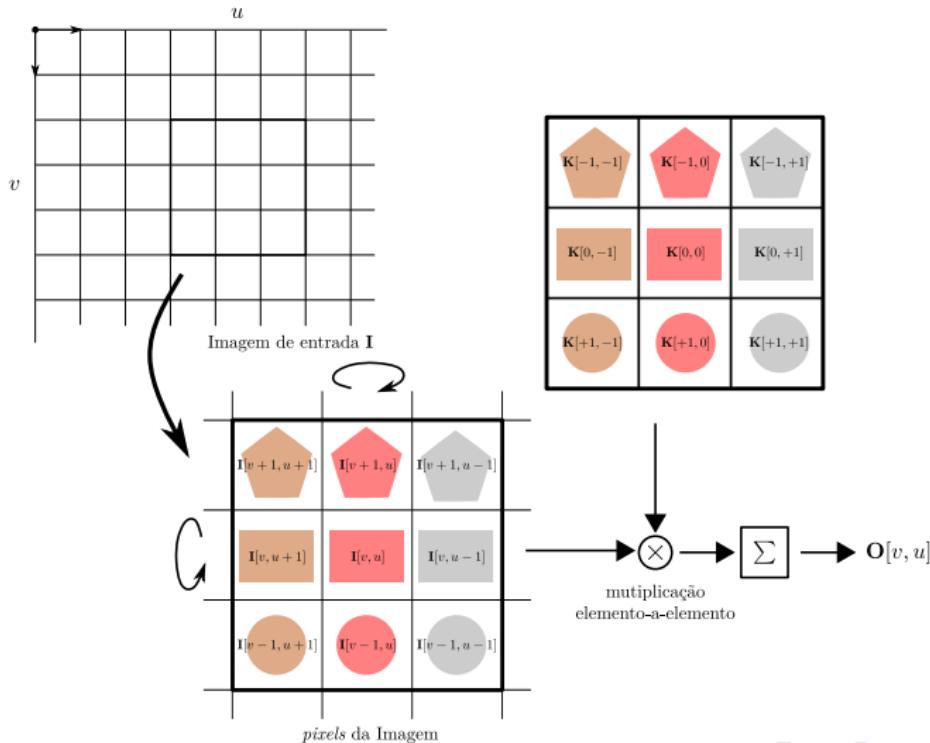
Cálculo do pixel (v, u) da imagem de saída \mathbf{O}



Convolução 2D

Filtragem Espacial

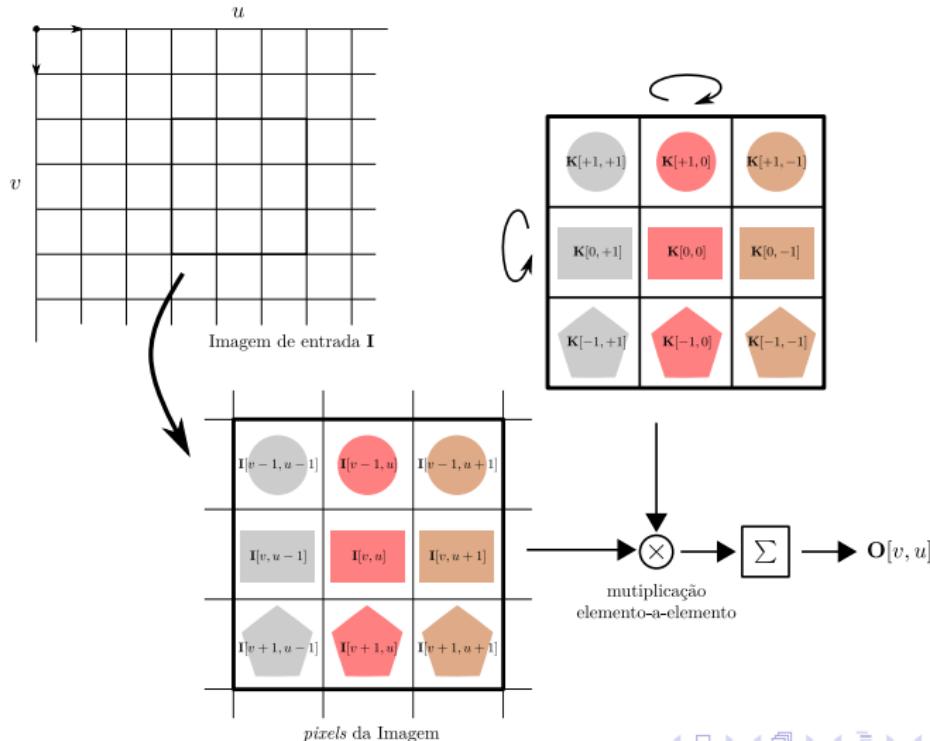
Forma alternativa 1



Convolução 2D

Filtragem Espacial

Forma alternativa 2 (preferencial)



Exemplo 2: Convolução 2D

Filtragem Espacial

Enunciado: Crie no Matlab um imagem de entrada **I** de dimensão 8×8 , com pixels do tipo `uint8` com valores iguais a 100. Na sequência filtre a imagem utilizando um kernel **K** de dimensão 3×3 , com cada elemento do kernel com valor igual a 1. Para fins de comparação, realize a filtragem com a função `conv2` de duas formas. Primeiro, considere preenchimento por zero nas regiões de borda e, depois, desconsidere a região de borda.

Exemplo 2: Convolução 2D

Filtragem Espacial

```
1 % imagem de entrada
2 I = 100*ones(8,8, 'uint8');
3
4 % kernel
5 K = ones(3,3);
6
7 % convolução 2D (com preenchimento de zeros nas regiões
8 % de borda)
9 I2 = conv2(I,K, 'same');
10
11 % convolução 2D (com descarte da região de borda)
12 I3 = conv2(I,K, 'valid');
```

① Operações Espaciais

② Filtragem 2D

- Correlação 2D
- Convolução 2D

③ Aplicações da Filtragem Espacial

- Suavização
- Detecção de Borda
 - Algoritmo Básico

Aplicações

Filtragem Espacial

- Alterando-se os valores dos coeficientes do filtro \mathbf{K} , tem-se diferentes operações de filtragem.
- Exemplos de aplicações envolvendo filtragem:
 1. Suavização/redução de ruídos.
 - Filtro normalizado.
 - Filtro gaussiano.
 2. Detecção de borda.
 - Algoritmo básico.
 - Algoritmo de Canny.

① Operações Espaciais

② Filtragem 2D

- Correlação 2D
- Convolução 2D

③ Aplicações da Filtragem Espacial

- Suavização
- Detecção de Borda
 - Algoritmo Básico

Suavização

Aplicações da Filtragem Espacial

- Os filtros utilizados para suavização são também conhecidos como filtros passa-baixa.
- Reduzem os componentes de alta frequência.
- Os filtros utilizados realizam a média (aritmética ou ponderada) dos *pixels* dentro da janela \mathcal{W} .
- Quanto maior a janela, maior o efeito de suavização.

Suavização

Aplicações da Filtragem Espacial

Filtro Normalizado

$$\mathbf{K}[j, i] = \frac{1}{w^2}$$

$\forall j, i \in [-h, h]$, com $w = 2h + 1$ denotando a largura (ou altura) da janela quadrada \mathcal{W} .

Exemplos

$$\mathbf{K} = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{bmatrix}$$

Suavização

Aplicações da Filtragem Espacial

Filtro normalizado, $w = 3$

Imagen original



Imagen filtrada



Suavização

Aplicações da Filtragem Espacial

Filtro normalizado, $w = 5$

Imagen original

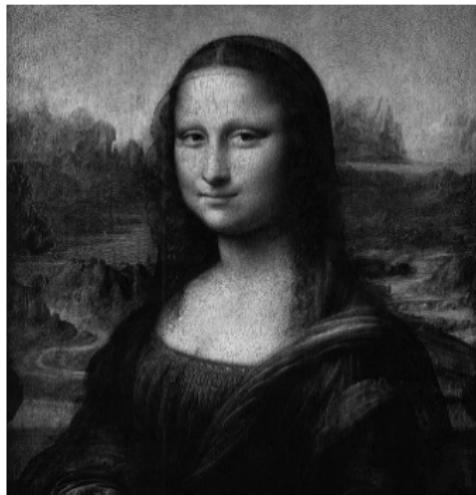


Imagen filtrada



Exemplo 3: filtragem de imagem

Aplicações da Filtragem Espacial

Enunciado: construa um código no Matlab para filtrar a imagem `castle.jpg` (disponível no Moodle e mostrada abaixo). Como kernel utilize um filtro normalizado com dimensão 4×4 .



Exemplo 3: filtragem de imagem

Aplicações da Filtragem Espacial

Solução 1

```
1 I = imread('castle.jpg');
2 figure; imshow(I);
3
4 % kernel normalizado
5 N = 4;
6 K = 1/(N^2) * ones(N);
7
8 % filtragem 2D
9 I2 = filter2(K,I, 'same');
10
11 figure; imagesc(I2);
12 colormap gray;
13 colorbar;
```

Exemplo 3: filtragem de imagem

Aplicações da Filtragem Espacial

Solução 2

```
1 I = imread('castle.jpg');
2 figure; imshow(I);
3
4 % kernel normalizado
5 N = 4;
6 K = 1/(N^2) * ones(N);
7
8 % filtragem 2D
9 I2 = filter2(K,I, 'same');
10
11 % converte valores para uint8
12 I2 = uint8(I2);
13
14 figure; imshow(I2);
```

Exemplo 3: filtragem de imagem

Aplicações da Filtragem Espacial

Solução 3

```
1 I = imread('castle.jpg'); figure; imshow(I);  
2  
3 % kernel normalizado  
4 N = 4;  
5 K = 1/(N^2) * ones(N);  
6  
7 % filtragem 2D  
8 I2 = filter2(K,I, 'same');  
9  
10 % normalizada resultado pelo máximo valor para forçar  
11 % os valores dos pixels entre 0 e 1;  
12 I2 = I2/max(I2(:));  
13  
14 figure; imshow(I2);
```

Suavização

Aplicações da Filtragem Espacial

Filtro Gaussiano

$$\mathbf{K}[j, i] = \frac{1}{2\pi\sigma^2} e^{-\frac{(j^2+i^2)}{2\sigma^2}} \quad (1)$$

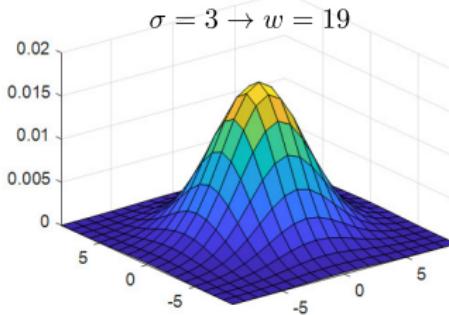
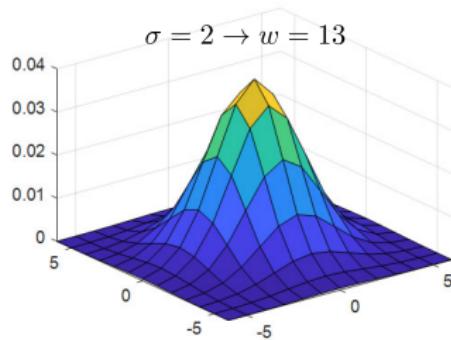
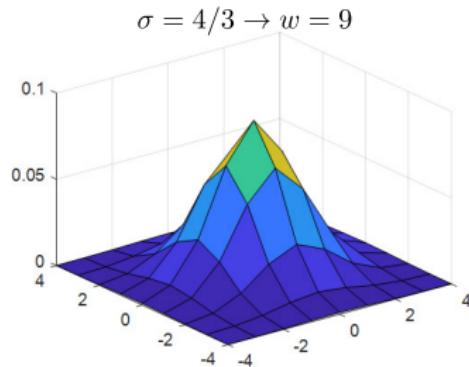
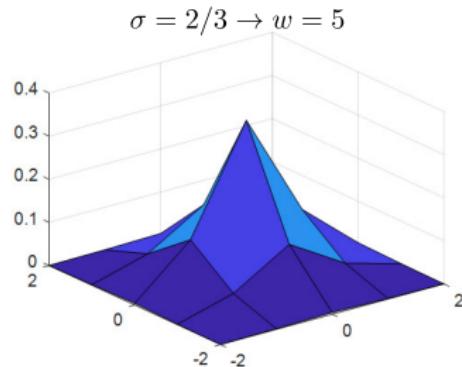
quanto maior σ , menos concentrados em torno do centro estão os coeficientes. Regra geral $h = 3\sigma$, ou seja, $w = 6\sigma + 1$.

Exemplo: $\sigma = 2/3 \rightarrow h = 2, w = 5$

$$\mathbf{K} = \begin{bmatrix} 0 & 0.0013 & 0.0040 & 0.0013 & 0 \\ 0.0013 & 0.0377 & 0.1163 & 0.0377 & 0.0013 \\ 0.0040 & 0.1163 & 0.3581 & 0.1163 & 0.0040 \\ 0.0013 & 0.0377 & 0.1163 & 0.0377 & 0.0013 \\ 0 & 0.0013 & 0.0040 & 0.0013 & 0 \end{bmatrix}$$

Suavização

Aplicações da Filtragem Espacial



Suavização

Aplicações da Filtragem Espacial

Filtro gaussiano, $\sigma = 2/3 \rightarrow w = 5$

Imagen original



Imagen filtrada



Suavização

Aplicações da Filtragem Espacial

Filtro gaussiano, $\sigma = 4/3 \rightarrow w = 9$

Imagen original



Imagen filtrada



Exemplo 4: filtragem de imagem

Aplicações da Filtragem Espacial

Enunciado: construa um código no Matlab para filtrar a imagem `castle.jpg` (disponível no Moodle e mostrada abaixo). Utilize um kernel gaussiano com largura $w = 5$ e $\sigma = (w - 1)/6$.



Exemplo 4: filtragem de imagem

Aplicações da Filtragem Espacial

```
1 I = imread('castle.jpg');
2 figure; imshow(I);
3
4 % kernel gaussiano
5 w = 5;
6 sig = (w-1)/6;
7 K = fspecial('gaussian',w,sig);
8
9 % filtragem 2D
10 I2 = filter2(K,I, 'same');
11 I2 = uint8(I2);
12
13 figure; imshow(I2);
```

① Operações Espaciais

② Filtragem 2D

- Correlação 2D
- Convolução 2D

③ Aplicações da Filtragem Espacial

- Suavização
- Detecção de Borda
 - Algoritmo Básico

Dectecção de Borda

Aplicações da Filtragem Espacial

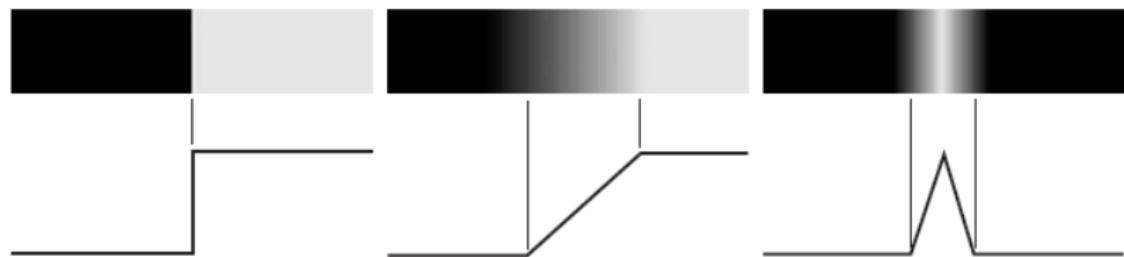
Detecção de borda é uma abordagem utilizada mais frequentemente para segmentar imagens baseadas em mudanças abruptas de intensidade.



Dectecção de Borda

Aplicações da Filtragem Espacial

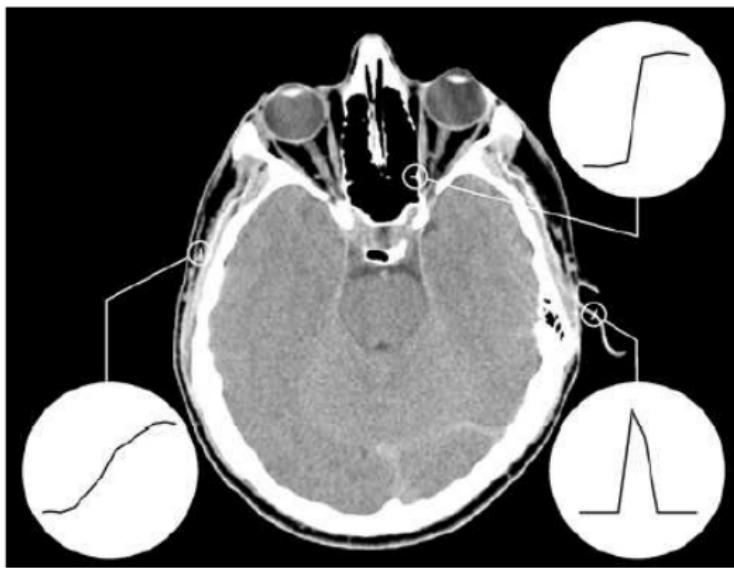
Modelos de bordas são classificadas de acordo com o perfil de variação da intensidade dos *pixels*.



Dectecção de Borda

Aplicações da Filtragem Espacial

É importante destacar que os três modelos de bordas apresentados no *slide* anterior, são de fato observados em imagens práticas.

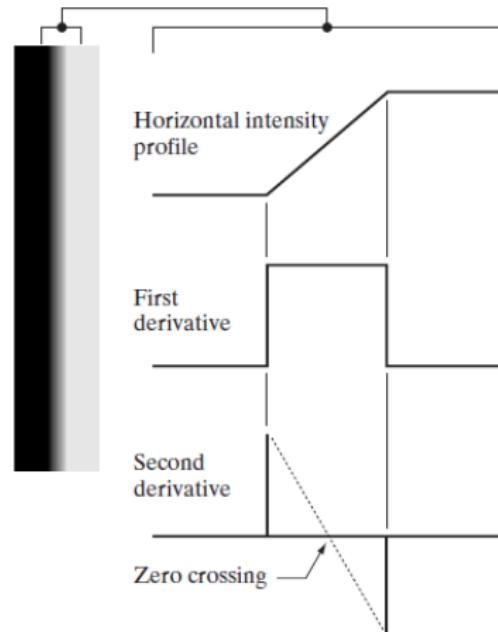


Dectecção de Borda

Aplicações da Filtragem Espacial

Observações:

- Bordas são caracterizadas por variações na intensidade dos *pixels*.
- Assim, o cálculo das derivadas de 1^a e 2^a ordem da imagem pode ser útil para detecção de bordas.



① Operações Espaciais

② Filtragem 2D

- Correlação 2D
- Convolução 2D

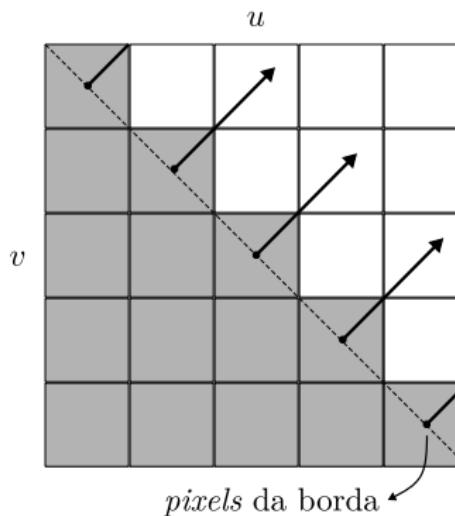
③ Aplicações da Filtragem Espacial

- Suavização
- Detecção de Borda
 - Algoritmo Básico

Algoritmo Básico

Detecção de Borda - Aplicações da Filtragem Espacial

- Baseado no cálculo do gradiente da imagem.
- Para cada *pixel* determina-se um vetor indicando a direção e magnitude da maior variação da intensidade dos *pixels*.

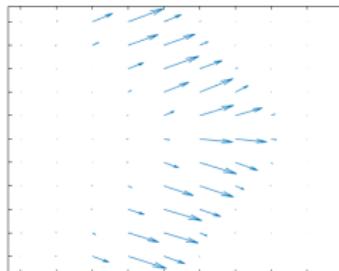
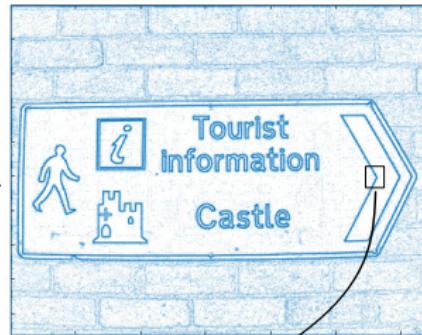


Algoritmo Básico

Deteção de Borda - Aplicações da Filtragem Espacial



→ Gradiente →

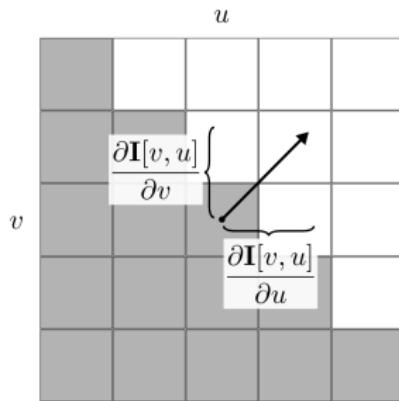


Algoritmo Básico

Deteção de Borda - Aplicações da Filtragem Espacial

Para cada *pixel* (v, u) da imagem \mathbf{I} , o vetor de gradiente é dado por

$$\nabla \mathbf{I}[v, u] = \begin{bmatrix} \frac{\partial \mathbf{I}[v, u]}{\partial u} \\ \frac{\partial \mathbf{I}[v, u]}{\partial v} \end{bmatrix}$$



onde $\partial \mathbf{I}[v, u]/\partial u$ denota a taxa de variação da intensidade do *pixel* no eixo horizontal u e $\partial \mathbf{I}[v, u]/\partial v$ representa a taxa de variação em relação ao eixo vertical v .

Algoritmo Básico

Detecção de Borda - Aplicações da Filtragem Espacial

- Computando-se as derivadas parciais para todos os *pixels* (v, u) da imagem \mathbf{I} , tem-se que $\partial\mathbf{I}/\partial u$ e $\partial\mathbf{I}/\partial v$ passam a ser imagens com a mesma dimensão de \mathbf{I} .
- Ou seja, o gradiente de \mathbf{I} pode ser composto pelas imagens $\partial\mathbf{I}/\partial u$ e $\partial\mathbf{I}/\partial v$.

Algoritmo Básico

Detecção de Borda - Aplicações da Filtragem Espacial

Como calcular $\partial\mathbf{I}/\partial u$ e $\partial\mathbf{I}/\partial v$?

1ª Opção: Para todo (v, u) calcula-se

$$\frac{\partial\mathbf{I}[v, u]}{\partial u} = \mathbf{I}[v, u + 1] - \mathbf{I}[v, u]$$

$$\frac{\partial\mathbf{I}[v, u]}{\partial v} = \mathbf{I}[v + 1, u] - \mathbf{I}[v, u]$$

Algoritmo Básico

Detecção de Borda - Aplicações da Filtragem Espacial

Como calcular $\partial\mathbf{I}/\partial u$ e $\partial\mathbf{I}/\partial v$?

2ª Opção: Para todo (v, u) calcula-se

$$\frac{\partial\mathbf{I}[v, u]}{\partial u} = \frac{1}{2} (\mathbf{I}[v, u + 1] - \mathbf{I}[v, u - 1])$$

$$\frac{\partial\mathbf{I}[v, u]}{\partial v} = \frac{1}{2} (\mathbf{I}[v + 1, u] - \mathbf{I}[v - 1, u])$$

Algoritmo Básico

Deteção de Borda - Aplicações da Filtragem Espacial

Como calcular $\partial\mathbf{I}/\partial u$ e $\partial\mathbf{I}/\partial v$?

3ª Opção:

$$\frac{\partial \mathbf{I}}{\partial u} = \mathbf{I} * \mathbf{K}_u$$

onde $*$ representa a operação de convolução e

$$\mathbf{K}_u = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

é conhecido como filtro de Sobel.

Algoritmo Básico

Deteção de Borda - Aplicações da Filtragem Espacial

Como calcular $\partial\mathbf{I}/\partial u$ e $\partial\mathbf{I}/\partial v$?

3ª Opção:

$$\frac{\partial \mathbf{I}}{\partial v} = \mathbf{I} * \mathbf{K}_v$$

onde

$$\mathbf{K}_v = \mathbf{K}_u^T = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Algoritmo Básico

Deteção de Borda - Aplicações da Filtragem Espacial

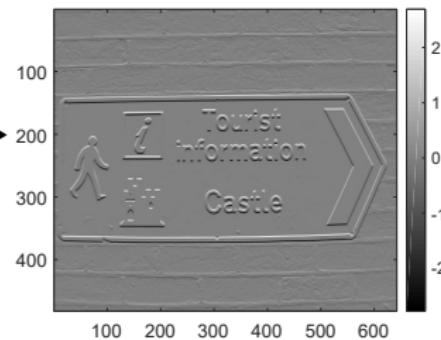
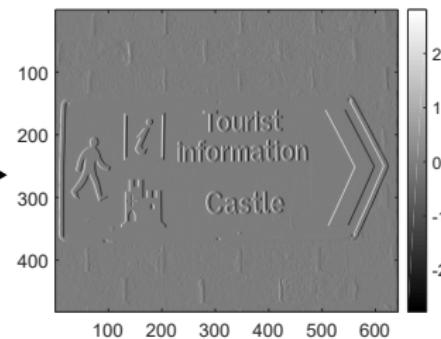


$$\mathbf{K}_u = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Convolução

Convolução

$$\mathbf{K}_v = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$



Algoritmo Básico

Detecção de Borda - Aplicações da Filtragem Espacial

Note que o **gradiente** no *pixel* (v, u) também pode ser representado por da magnitude $\mathbf{M}[v, u]$ e fase $\alpha[v, u]$, definidas como

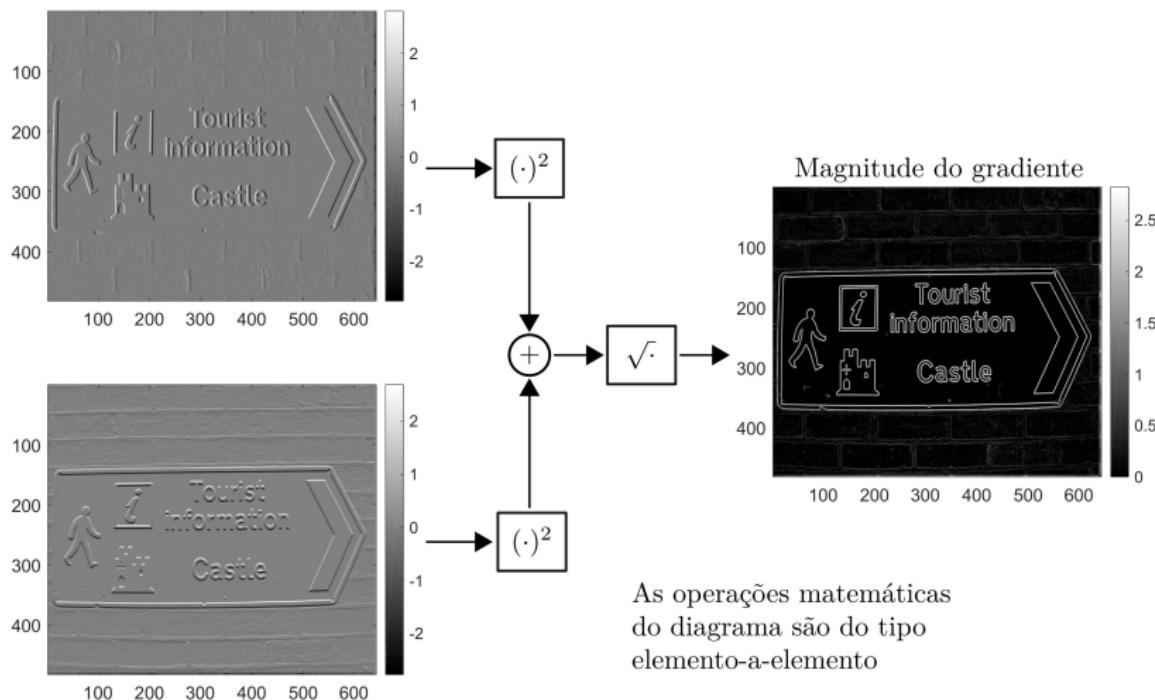
$$\mathbf{M}[v, u] = \sqrt{\left(\frac{\partial \mathbf{I}[v, u]}{\partial u}\right)^2 + \left(\frac{\partial \mathbf{I}[v, u]}{\partial v}\right)^2}$$

e

$$\alpha[v, u] = \text{atan} \left(\frac{\partial \mathbf{I}[v, u]/\partial v}{\partial \mathbf{I}[v, u]/\partial u} \right)$$

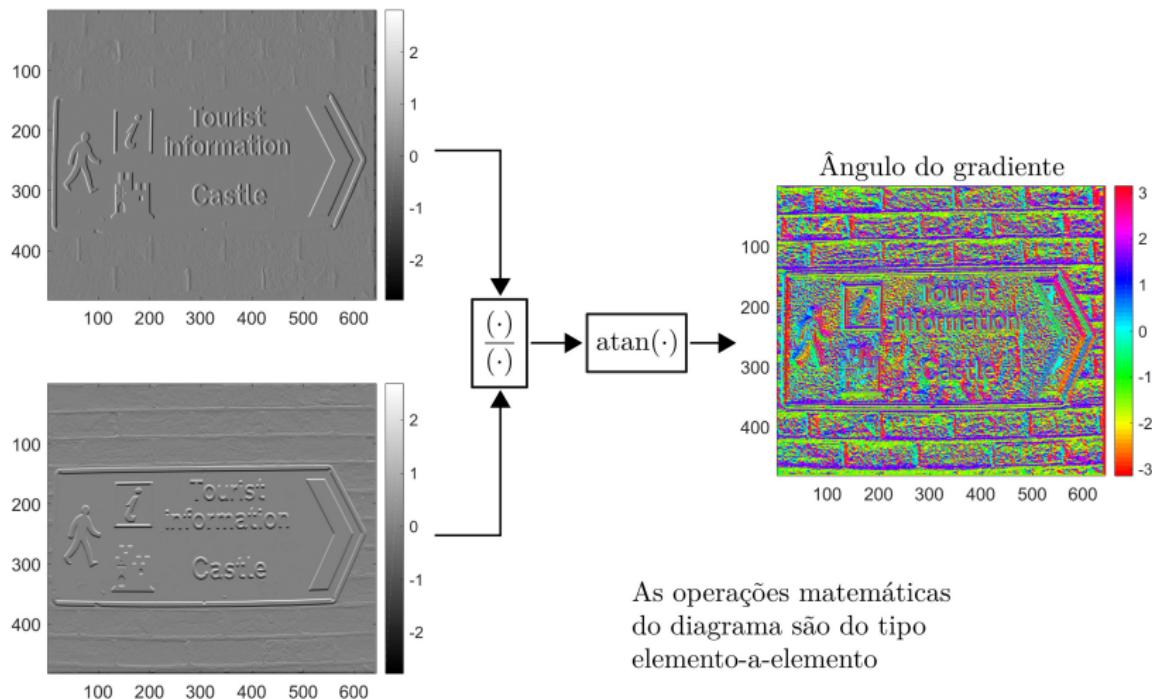
Algoritmo Básico

Deteção de Borda - Aplicações da Filtragem Espacial



Algoritmo Básico

Deteção de Borda - Aplicações da Filtragem Espacial



Algoritmo Básico

Detecção de Borda - Aplicações da Filtragem Espacial

Note que na imagem de magnitude do gradiente (isto é, $\mathbf{M}[v, u]$), os *pixels* de maior valor tendem a corresponder as bordas da imagem.

Assim, uma imagem de borda pode ser obtida a partir da operação de limiarização, ou seja,

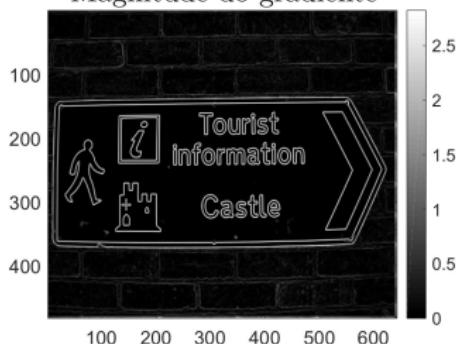
$$\mathbf{I}_b[v, u] = \begin{cases} 1, & \mathbf{M}[v, u] \geq \tau \\ 0, & \text{caso contrário} \end{cases}$$

onde \mathbf{I}_b é uma imagem binária com *pixels* de borda em branco e τ é um limiar de comparação.

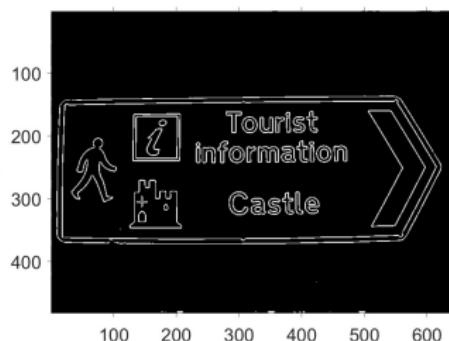
Algoritmo Básico

Deteção de Borda - Aplicações da Filtragem Espacial

Magnitude do gradiente



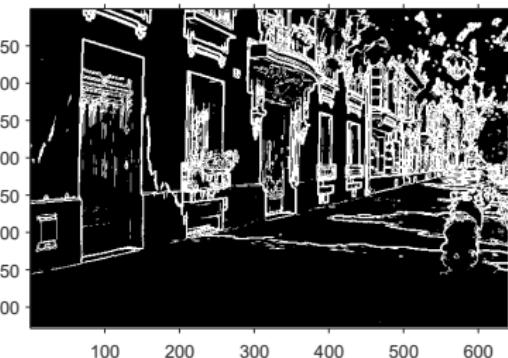
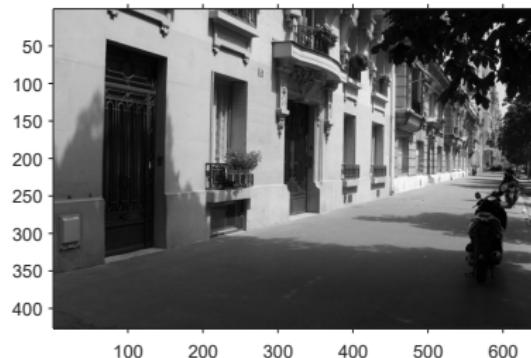
→ Limiarização
 $(\cdot) \geq \tau$ →



Algoritmo Básico

Deteção de Borda - Aplicações da Filtragem Espacial

Se a imagem I tiver muitas variações de textura, o algoritmo básico apresentado pode gerar uma imagem de borda bastante ruidosa.



Algoritmo Básico

Deteção de Borda - Aplicações da Filtragem Espacial

Em casos como o anterior, geralmente, a imagem I é primeiramente filtrada por um [kernel de suavização](#) para então ter suas bordas determinadas.

