CIÊNCIAS DA COMPUTAÇÃO

Processamento de Imagens
e
Visão Computacional

Prof. César C. Xavier

ROTEIRO

- Filtros - Relevância
- Filtros em PI e VC
- Tipos de Filtros:
  - Filtro de Média
  - Filtro Gaussiano
  - Filtro de Mediana
  - Filtro Bilateral

Processamento de Imagens e Visão Computacional

Relevância

- Melhorar a qualidade das imagens.
- Facilitar a extração de informações pertinentes.
- Aumentar a eficiência e precisão dos algoritmos.
- Adaptar as imagens para necessidades específicas das aplicações.

Processamento de Imagens e Visão Computacional

Filtros em PI e VC

- Ruído em imagens binárias
  - Visam corrigir, suavizar ou realçar a imagem ou determinadas regiões.
  - Se dá pela aplicação de matrizes, denominadas máscaras ou núcleos, que atual modificando os valores dos pixels da imagem.

10	14	5	7	12	4
8	13	3	1	8	5
12	9	11	14	2	16
4	5	16	9	11	15
6	18	7	13	12	10
11	3	10	9	18	5

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

		2			

Prof. César C. Xavier

Filtros em PI e VC

- Ruído em imagens binárias
  - Operação de aplicar a multiplicação da matriz sobre os pontos da imagem é denominado de convolução.

13	3	1	*	0	-1	0	=	0	-3	0
9	11	14		-1	4	-1		-9	44	-14
5	16	9		0	-1	0		0	-16	0

Processamento de Imagens e Visão Computacional

Filtros em PI e VC

- Ruído em imagens binárias
  - Três iterações consecutivas:

0	-1	0				
-1	4	-1	5	7	12	4
0	-1	0	3	1	8	5
	12	9	11	14	2	16
	4	5	16	9	11	15
	6	18	7	13	12	10
	11	3	10	9	18	5

0	-1	0				
-1	4	-1	7	12	4	
0	-1	0	1	8	5	
12	9	11	14	2	16	
4	5	16	9	11	15	
6	18	7	13	12	10	
11	3	10	9	18	5	

	0	-1	0			
18	-1	4	-1	12	4	
8	0	-1	0	8	5	
12	9	11	14	2	16	
4	5	16	9	11	15	
6	18	7	13	12	10	
11	3	10	9	18	5	

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro de Média
  - Linear e passa-baixas (suavização de imagens)
  - Substitui pelo valor médio da vizinhança
  - Maior a máscara, maior número de vizinhos, maior o efeito de suavização
  - Aplicações:
    - Redução de Ruído Aleatório: Eficaz para atenuar ruídos aleatórios uniformemente distribuídos.
    - Pré-processamento de Imagens: Utilizado antes de operações como segmentação ou detecção de bordas para melhorar a performance desses algoritmos.
    - Suavização Geral: Aplicado em situações onde uma redução geral de detalhes é desejada, como em escalas reduzidas de imagens.

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro de Média
  - Faz uso da biblioteca *blur* da openCV.

Prof. César C. Xavier

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro de Média
  - Função: cv2.blur(*src, dst, ksize, anchor, borderType*)
  - Parâmetros:
    - src - imagem fonte (input).
    - dst - imagem destino (output).
    - ksize - tamanho do núcleo.
    - anchor - ponto de referência (*cv2.Point(-1,-1)* está no centro do núcleo).
    - borderType – para os pixels da borda especifica como o núcleo será aplicado. Valores comuns incluem cv2.BORDER_CONSTANT, cv2.BORDER_REFLECT, etc.

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

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro de Média

```
import cv2
imgOriginal = cv2.imread("moedas.jpg")
imgTratada = cv2.blur(imgOriginal, (5,5))
cv2.imshow("Original", imgOriginal)
cv2.imshow("Tratada", imgTratada)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Processamento de Imagens e Visão Computacional

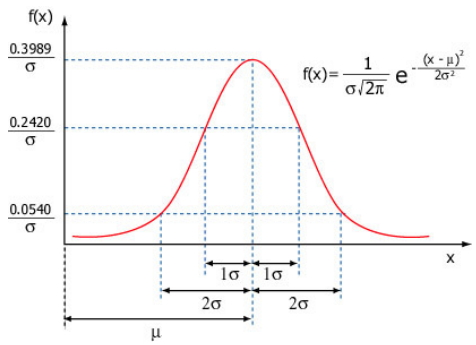
Aplicação de Filtros

- Filtro Gaussiano
  - Linear e passa-baixas (suavização de imagens)
  - Aplica uma função gaussiana sobre a vizinhança do pixel, dando mais peso aos pixels próximos ao centro.
  - Muito bom para imagens com ruído gaussiano
  - Aplicações:
    - Aplicações Particulares:
    - Redução de Ruído Gaussiano: Ideal para imagens afetadas por ruído que segue distribuição gaussiana.
    - Pré-processamento em Detecção de Bordas: Frequentemente usado antes de operadores de detecção de bordas (como Canny) para reduzir ruído sem perder informações importantes.

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro Gaussiano
  - Linear e passa-baixas (suavização de imagens)
  - Muito bom para imagens com ruído gaussiano
  - Faz uso da biblioteca *GaussianBlur* da openCV.



Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro Gaussiano
  - Faz uso da biblioteca *GaussianBlur* da openCV.

Processamento de Imagens e Visão Computacional

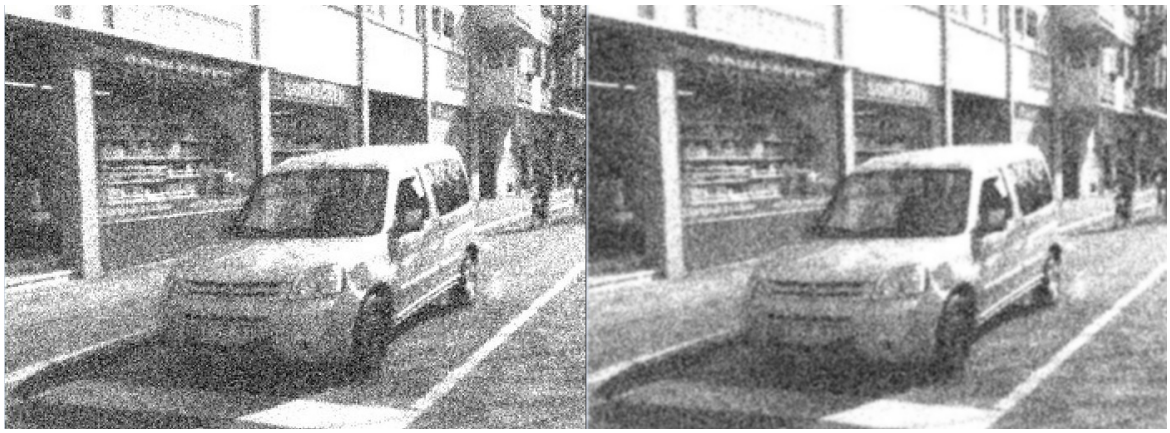
Aplicação de Filtros

- Filtro Guassiano
  - Função: cv2.GaussianBlur(src, ksize, sigmaX[, dst[, sigmaY[, borderType]]])
  - Parâmetros:
    - src: Imagem de entrada.
    - ksize: Tamanho do kernel, deve ser uma tupla de números ímpares positivos, por exemplo, (5, 5). Se ksize for (0, 0), os valores de sigma serão usados para calcular o tamanho do kernel.
    - sigmaX: Desvio padrão no eixo X (horizontal) da distribuição Gaussiana.
    - dst (opcional): Imagem de saída.
    - sigmaY (opcional): Desvio padrão no eixo Y (vertical). Se não especificado, assume o mesmo valor de sigmaX.
    - borderType (opcional): Tipo de tratamento das bordas, semelhante ao parâmetro em cv2.blur.

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro Gaussiano
  - Faz uso da biblioteca *GaussianBlur* da openCV.



Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro Gaussiano
  - Faz uso da biblioteca *GaussianBlur* da openCV.

Prof. César C. Xavier

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro Gaussiano

```
import cv2
imgOriginal = cv2.imread("moedas.jpg")
imgTratada = cv2.GaussianBlur(imgOriginal, (5,5), 0)
cv2.imshow("Original", imgOriginal)
cv2.imshow("Tratada", imgTratada)
cv2.waitKey(0)
cv2.destroyAllWindows()
```


Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro Mediana
  - Não Linear e passa-baixas (suavização de imagens)
  - Substitui o valor do pixel central pela mediana dos valores dos pixels na vizinhança, preservando bordas mais eficazmente.
  - Muito bom para imagens do tipo “sal e pimenta”
  - Preserva mais detalhes de alta frequência
  - Aplicações:
    - Eliminação de Ruído Sal e Pimenta: Amplamente utilizado em imagens afetadas por esse tipo de ruído impulsivo.
    - Processamento de Imagens Médicas: Preserva detalhes críticos em imagens como tomografias e ressonâncias magnéticas.
    - Melhoria de Imagens Digitais: Aplicado em fotografias digitais para limpar imperfeições sem perder qualidade.

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro Mediana
  - Faz uso da biblioteca *MedianBlur* da openCV.

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro de Mediana
  - Filtro de Mediana:
    - Função: cv2.medianBlur(src, ksize)
  - Parâmetros:
    - src: Imagem de entrada.
    - ksize: Tamanho do kernel. Deve ser um número ímpar maior que 1, por exemplo, 3, 5, 7.

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro Mediana


Processamento de Imagens e Visão Computacional

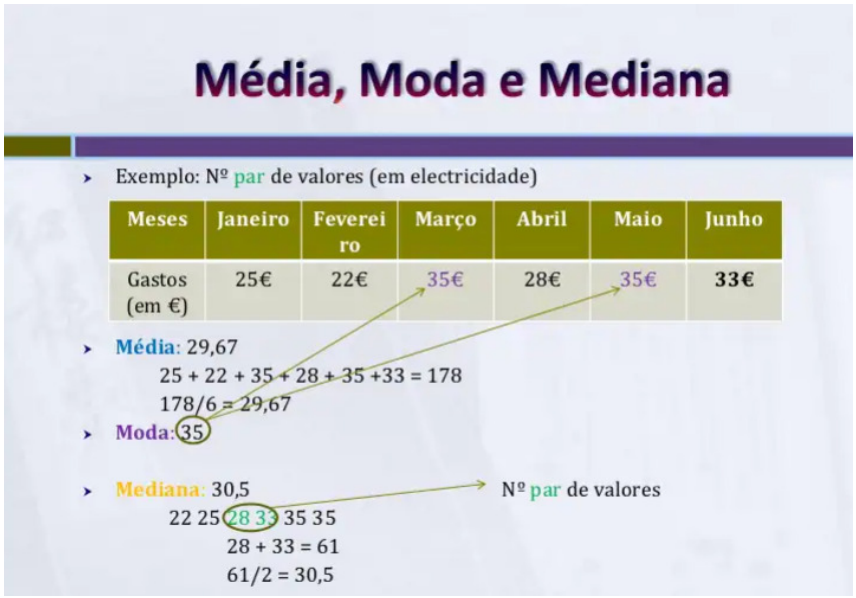
Aplicação de Filtros

- Filtro Mediana

```
import cv2
imgOriginal = cv2.imread("moedas.jpg")
imgTratada = cv2.medianBlur(imgOriginal, 5)
cv2.imshow("Original", imgOriginal)
cv2.imshow("Filtro Mediana", imgTratada)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Processamento de Imagens e Visão Computacional

Relembrando...



Processamento de Imagens e Visão Computacional

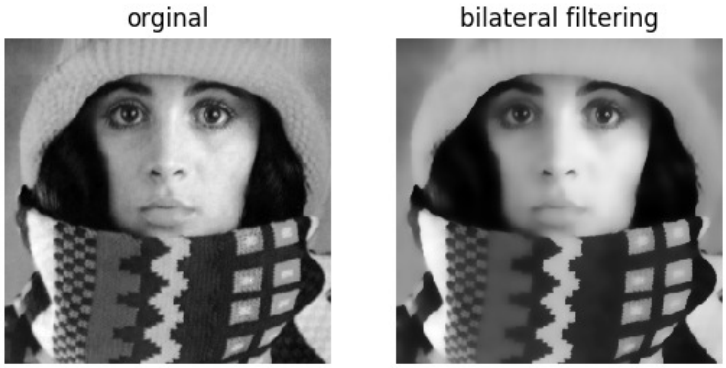
Aplicação de Filtros

- Filtro Bilateral
  - Utiliza o método cv2.bilateralFilter()

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro Bilateral
  - Utiliza o método cv2.bilateralFilter()



https://staff.fnwi.uva.nl/r.vandenboomgaard/IPCV20162017/_images/truibilateral.png

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro Bilateral
  - Função: cv2.bilateralFilter(src, d, sigmaColor, sigmaSpace[, dst[, borderType]])
  - Parâmetros:
    - src: Imagem de entrada.
    - d: Diâmetro do pixel da vizinhança usada durante o filtro. Se d for negativo ou zero, ele é calculado a partir de sigmaSpace.
    - sigmaColor: Desvio padrão no espaço de cor. Valores maiores significam que cores mais diferentes dentro da vizinhança serão misturadas, resultando em uma suavização maior das cores.
    - sigmaSpace: Desvio padrão no espaço de coordenadas (espaço). Valores maiores significam que pixels mais distantes exercerão influência uns sobre os outros, desde que suas cores estejam dentro da faixa de sigmaColor.
    - dst (opcional): Imagem de saída.
    - borderType (opcional): Tipo de tratamento das bordas, como nos outros filtros.

Processamento de Imagens e Visão Computacional

Aplicação de Filtros

- Filtro Mediana

```
import cv2
imgOriginal  =  cv2.imread("moedas.jpeg")
imgTratada  =  cv2.bilateralFilter(imgOriginal,9,  75, 75)
 cv2.imshow("Original",imgOriginal)
cv2.imshow("Tratada", imgTratada)
cv2.waitKey(0)
cv2.destroyAllWindows()
```