Processamento de Imagens

Prof. Jefferson T. Oliva
jeffersonoliva@utfpr.edu.br

Processamento de Imagens
Engenharia de Computação
Departamento Acadêmico de Informática (Dainf)
Universidade Tecnológica Federal do Paraná (UTFPR)
Campus Pato Branco

Sumário

- Vizinhança de um pixel
- Conectividade
- Desenhando e Escrevendo em OpenCV

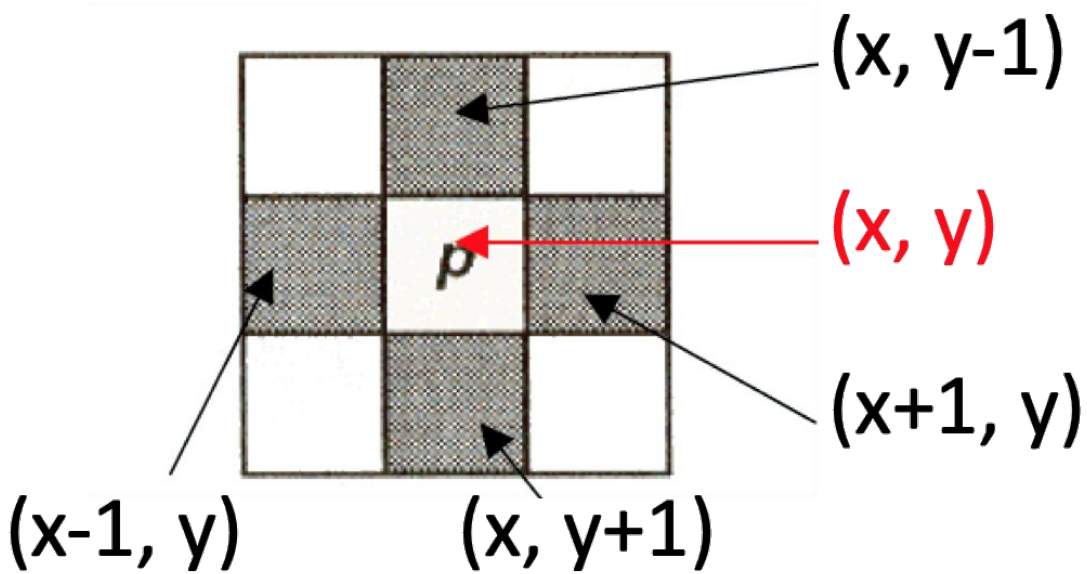
Sumário

Vizinhança de um pixel

- A análise de pixeis vizinhos é comum no processamento de imagens e visão computacional, sendo essencial para diversas tarefas
  - Detecção de bordas e contornos
  - Filtragem
  - Segmentação
  - Detecção de padrões
  - Entre outras

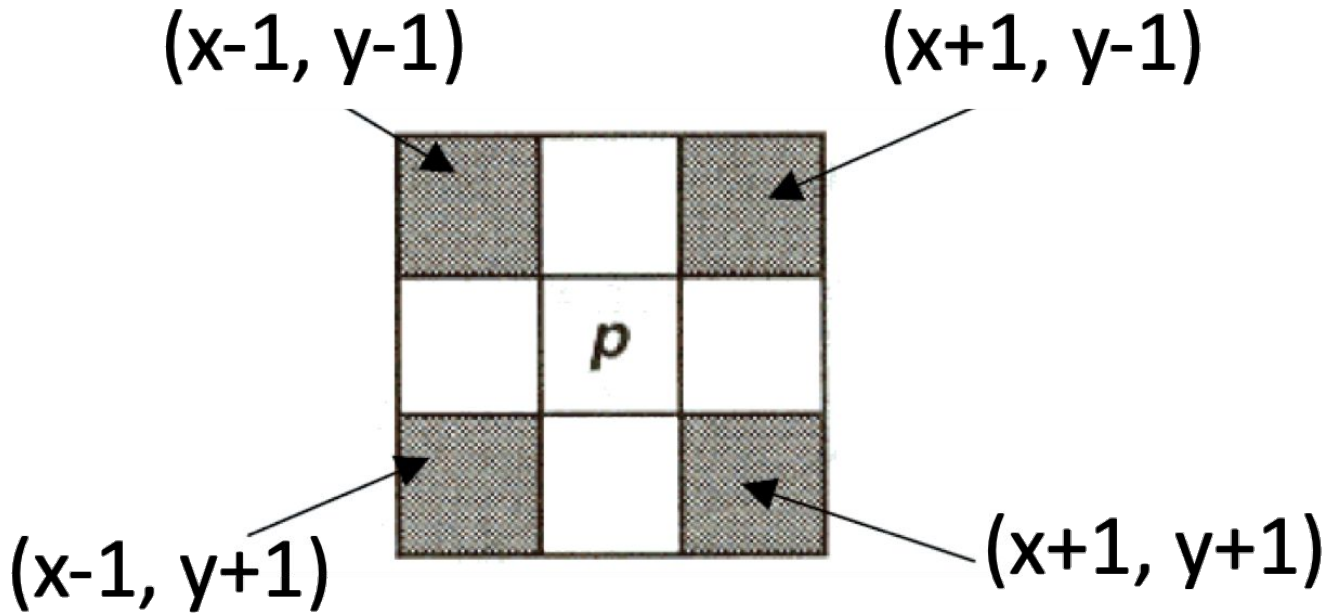
Vizinhança de um pixel

- Vinhança de 4
  - Se p é um pixel de borda, então terá um número menor de bordas (isso é válido para outras formas de vizinhança também)



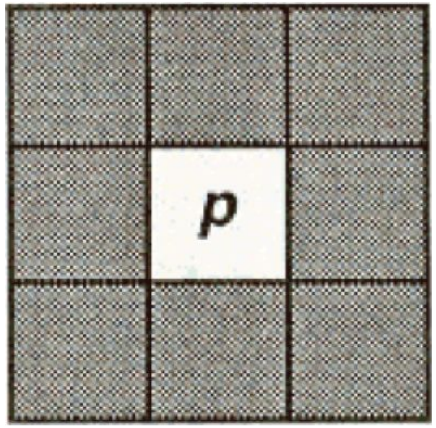
Vizinhança de um pixel

- Vizinhança diagonal



Vizinhança de um pixel

- Vinhança de 8

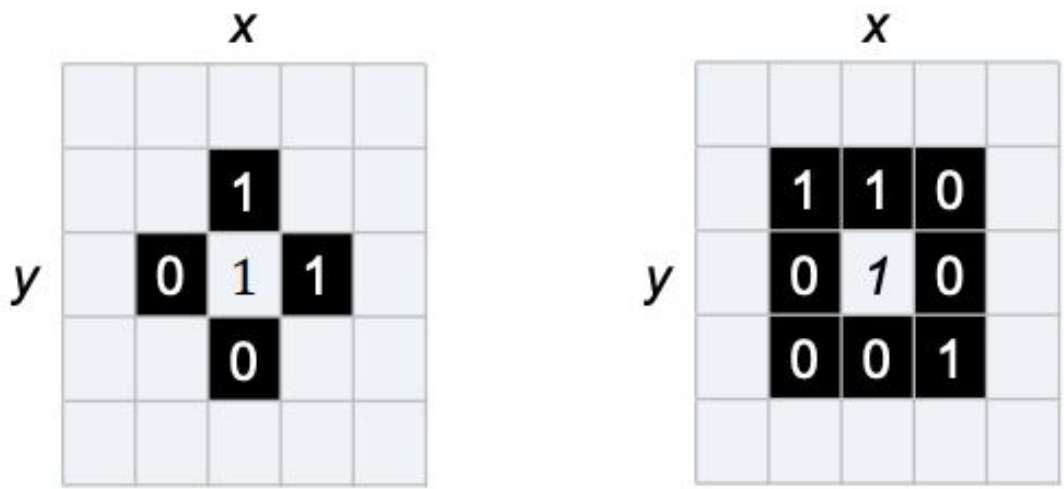


Sumário

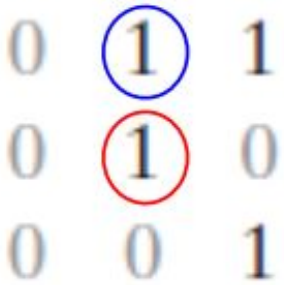
Conectividade

- Conceito usado para estabelecer fronteiras de objetos e regiões em uma imagem

- Dois pixeis são conectados se:
  - São adjacentes
  - Seus níveis de cinza satisfazem a um critério especificado de similaridade

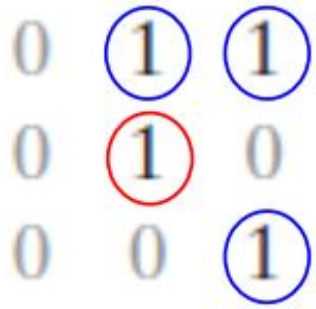


Conectividade



Conectados $N_4(p)$

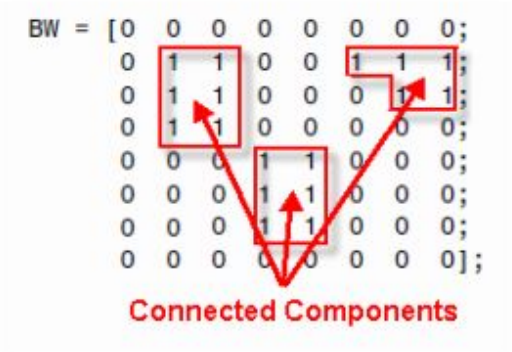
Conectados $N_D(p)$

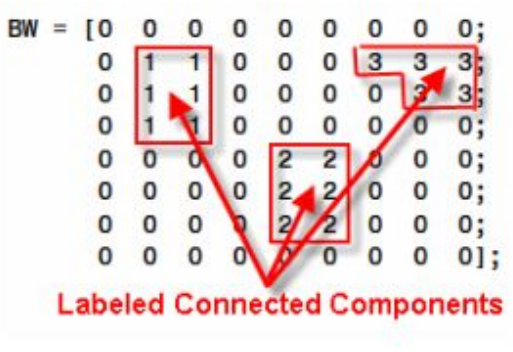


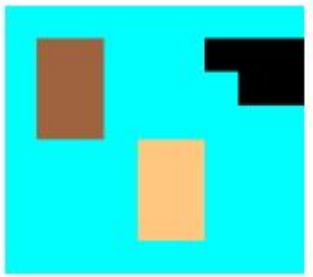
Conectados $N_8(p)$

Conectividade

- Rotulação de componentes conexos
  - Se p e q forem pixeis de um subconjunto S de uma imagem, então p está conectado a q em S se existir um caminho de p a q consistindo inteiramente de pixeis de S



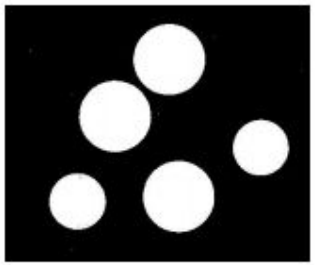




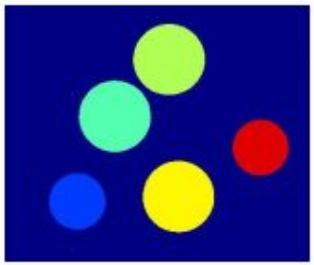
Conectividade

- Rotulação de componentes conexos
  - Exemplo de aplicação: remoção de objetos com área maior que T

Imagem de entrada



Resultado da segmentação



Rotulação dos componentes conexos

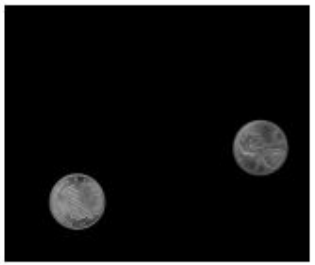


Imagem processada

Desenhando e Escrevendo em OpenCV

Desenhando e Escrevendo em OpenCV

- OpenCV contém diversas funções para o desenho de formas básicas
  - Linhas
  - Círculos
  - Retângulos
- Exemplos de utilidades de desenho e escrita em imagens
  - Reconhecimento facial
  - Detecção de objetos
  - Comparação de algoritmos de processamento de imagens
  - Entre outras

Desenhando e Escrevendo em OpenCV

- Funções openCV para desenho e escrita
  - *circle*: desenha um círculo
  - *ellipse*: desenha uma elipse
  - *line*: desenha uma linha
  - *pollylines*: desenha um polígono
  - *putText*: escreve um texto texto
  - *rectangle*: desenha um retângulo

Desenhando e Escrevendo em OpenCV

- Parâmetros comuns para as funções apresentadas no slide anterior
  - Imagem
  - Cor da forma ou do texto
  - Espessura de linhas ou círculos
  - tipo de linhas
  - Coordenadas na imagem onde a forma é desenhada ou o texto é inserido

Desenhando e Escrevendo em OpenCV

- Abrir imagem e mostrar os pixeis, onde *img1* representa a imagem

```
import cv2
img1 = cv2.imread('lenna.jpeg')
```

```
print(img1)
```

```
[[[162 162 162]
  [162 162 162]
  [161 161 161]
  ...
  [151 151 151]
  [165 165 165]
  [144 144 144]]

 [[162 162 162]
  [161 161 161]
  [161 161 161]
  ...
  [168 168 168]
  [186 186 186]
  [149 149 149]]

 [[161 161 161]
  [161 161 161]
  [160 160 160]
  ...
  [140 140 140]
  [142 142 142]
  [ 90  90  90]]
```

Desenhando e Escrevendo em OpenCV

- Imagem original

- Fragmento de código

```
cv2.imshow("Nome da janela", img1)
```

Desenhando e Escrevendo em OpenCV

- Imprimindo altura, largura e o número de canais da imagem

```
height = img1.shape[0]
width = img1.shape[1]
channels = img1.shape[2]
print('Largura em pixels: {}'.format(width))
print('Altura em pixels: {}'.format(height))
print('Qtde de canais: {}'.format(channels))
```

```
Largura em pixels: 225
Altura em pixels: 225
Qtde de canais: 3
```

- Acessando o valor de um pixel em uma determinada posição

```
px = img1[2, 2]
print('O pixel (2, 2) tem as seguintes cores:')
print(px)
```

```
O pixel (2, 2) tem as seguintes cores:
[160 160 160]
```

- Acessando as cores de um pixel

```
(b, g, r) = img1[6,6]
print('Vermelho:', r, 'Verde:', g, 'Azul:', b)
```

```
Vermelho: 155 Verde: 155 Azul: 155
```

Desenhando e Escrevendo em OpenCV

- Desenhando duas linhas na imagem

- Fragmento de código

```
cv2.line(img1, (100, 0), (100, 224), (0, 255, 0), 3)
cv2.line(img1, (0, 100), (224, 100), (255, 0, 0), 3)
```

Desenhando e Escrevendo em OpenCV

- Desenhando um círculo na imagem

- Fragmento de código

```
cv2.circle(img1, (130, 130), 40, (0, 255, 0), 2)
```

Desenhando e Escrevendo em OpenCV

- Desenhando um retângulo na imagem

- Fragmento de código

```
cv2.rectangle(img1, (100, 100), (160, 170), (0, 0, 255), 2)
```

Desenhando e Escrevendo em OpenCV

- Escrevendo um texto na imagem

- Fragmento de código

```
cv2.putText(img1, 'Lenna', (105, 190), cv2.FONT_HERSHEY_PLAIN, 1.5,
            (255, 255, 0), 2, cv2.LINE_AA)
```

Desenhando e Escrevendo em OpenCV

- Também, podemos combinar várias formas e textos em imagens

Referências I


Backes, A.
Slides de Aula – Modelos de cor.
Universidade Federal de Uberlândia, 2014.


Borges, L. E.
Python para desenvolvedores.
Novatec, 2017.


Bradski, G.
The openCV library.
Dr. Dobb's Journal: Software Tools for the Professional Programmer, v. 25, n. 11, p. 120-123, 2000.


Foley, J. D., Van, F. D., Van Dam, A., Feiner, S. K., Hughes, J. F., e Hughes, J.
Computer Graphics: Principles and Practice.
Addison-Wesley, 1996.

Referências II


Gonzalez, Rafael C., e Richard E. Woods.
Processamento de imagens digitais.
Editora Blucher, 2000.


Oliveira, M. M.
Notas de aula – Fundamentos de Processamento de Imagens.
Porto Alegre, 2010.
Disponível em: `http://www.inf.ufrgs.br/~oliveira/Cursos/INF01046/INF01046_descricao_2010_2.html`.
Acesso em: 10/12/2021.


Prateek, J., Millan, E. D., e Vinivius, G.
OpenCV by Example.
Packt Publishing, 2016.


Villán, A. F.
Mastering OpenCV 4 with Python.
Packt Publishing, 2019.