

MC920: Introdução ao Processamento de Imagem Digital

Trabalho 3

Luciano Zago - 182835

Universidade Estadual de Campinas — 17 de Outubro de 2019

Introdução

O objetivo do trabalho é obter medidas de objetos presentes em imagens digitais, através da transformação de cores, obtenção de bordas, e extração de propriedades como: área, perímetro, excentricidade e solidez.

1 Especificação do Problema

Utilizaremos imagens de fundo branco com objetos coloridos de entrada, em formato PNG, e retornaremos uma imagem com os objetos em preto, outra somente com o contorno dos objetos, e outra rotulando cada objeto com um número sequencial, para cada imagem de entrada. Além disso, para cada imagem de entrada, será retornado um histograma classificando a área dos objetos dessa imagem, segundo os seguintes critérios:

- objeto pequeno: área < 1500 pixels
- objeto médio: área ≥ 1500 pixels e área < 3000 pixels
- objeto grande: área ≥ 3000

Para cada imagem de entrada, deverá ser retornado o número de regiões (objetos), além de apresentar as seguintes propriedades de cada objeto, seguindo a ordem da imagem rotulada:

- região
- área
- perímetro
- excentricidade
- solidez

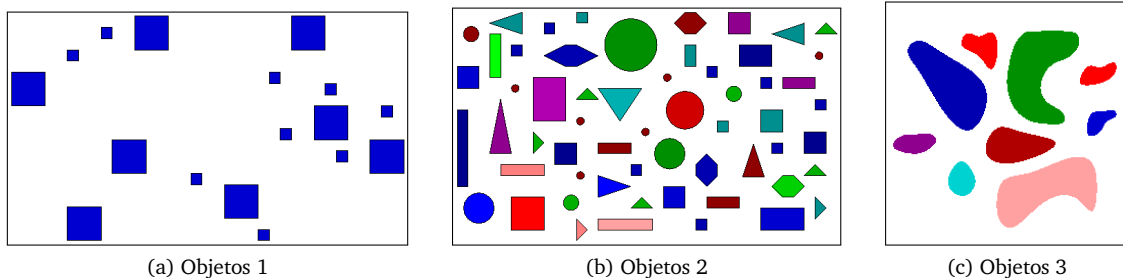


Figura 1: Imagens de entrada

2 Implementação

O programa em Python deve ser usado da seguinte forma:

```
./measurement.py input output-mono output-contour output-regions output-histogram
```

Argumentos:

- input e output-* representam respectivamente o arquivo de entrada e os arquivos de saída (monocromático, contorno, rótulo e histograma de área)

Para rodar o programa para todas as imagens de entrada, utilize o comando `./run.sh > output.txt` que facilita o processo. Ele salva as imagens de saída em `output/`, realiza as medições de propriedade para cada objeto e retorna o tempo de execução, salvando a saída em `output.txt`.

2.1 Transformação de cores

Foi aplicado um threshold para transformar todos os valores menores que 250 em 0.

```
cv2.threshold(img, 250, 255, cv2.THRESH_BINARY)[1]
```

2.2 Contorno

O contorno de cada objeto foi obtido pelas funções do OpenCV:

```
contours, _ = cv2.findContours(img_mono, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
cv2.drawContours(img_contour, contours, -1, 0)
```

Referiremos posteriormente cada contorno de objeto como `c`.

2.3 Área

A área[1] foi calculada através do momento:

```
M = cv2.moments(c)
area = M['m00']
```

2.4 Perímetro

O perímetro[1] foi calculado através do contorno:

```
cv2.arcLength(c, True)
```

2.5 Excentricidade

Foram utilizados os momentos centrais para o cálculo da excentricidade, conforme descrito na referência [3].

Foi calculado a derivada dos momentos centrais da seguinte forma:

```
M = cv2.moments(c)
dv_mu20 = M['mu20'] / M['m00']
dv_mu02 = M['mu02'] / M['m00']
dv_mu11 = M['mu11'] / M['m00']
```

Por fim, a excentricidade foi calculada da seguinte forma:

```
a = dv_mu20 + dv_mu02
b = np.sqrt(4*(dv_mu11**2) + (dv_mu20 - dv_mu02)**2)
np.sqrt(1 - ((a - b) / (a + b)))
```

2.6 Solidez

Foram utilizados diferentes funções do OpenCV, e através do contorno do objeto, foi possível obter a solidez[2]:

```
area = cv2.contourArea(c)
hull = cv2.convexHull(c)
hull_area = cv2.contourArea(hull)
float(area) / float(hull_area)
```

3 Resultados

Os resultados das diferentes imagens de entrada (Figura 1) estão apresentados nas Figuras 3, 4 e 5. Os histogramas das imagens de entrada estão na Figura 2.

A extração de propriedades obtidas para as imagens de entrada Objetos1 e Objetos3 foi a seguinte:

objetos1

número de regiões: 17

região 0:	área: 2448	perímetro: 195.656854	excentricidade: 0.198994	solidez: 1.000000
região 1:	área: 2448	perímetro: 195.656854	excentricidade: 0.198994	solidez: 1.000000
região 2:	área: 304	perímetro: 67.656854	excentricidade: 0.328560	solidez: 1.000000
região 3:	área: 304	perímetro: 67.656854	excentricidade: 0.328560	solidez: 1.000000
região 4:	área: 2399	perímetro: 193.656854	excentricidade: 0.000000	solidez: 1.000000
região 5:	área: 304	perímetro: 67.656854	excentricidade: 0.328560	solidez: 1.000000
região 6:	área: 322	perímetro: 69.656854	excentricidade: 0.000001	solidez: 1.000000
região 7:	área: 2448	perímetro: 195.656854	excentricidade: 0.198994	solidez: 1.000000
região 8:	área: 322	perímetro: 69.656854	excentricidade: 0.000001	solidez: 1.000000
região 9:	área: 322	perímetro: 69.656854	excentricidade: 0.000000	solidez: 1.000000
região 10:	área: 2448	perímetro: 195.656854	excentricidade: 0.198994	solidez: 1.000000
região 11:	área: 2448	perímetro: 195.656854	excentricidade: 0.198994	solidez: 1.000000
região 12:	área: 322	perímetro: 69.656854	excentricidade: 0.000001	solidez: 1.000000
região 13:	área: 304	perímetro: 67.656854	excentricidade: 0.328560	solidez: 1.000000
região 14:	área: 2399	perímetro: 193.656854	excentricidade: 0.000000	solidez: 1.000000
região 15:	área: 2448	perímetro: 195.656854	excentricidade: 0.198994	solidez: 1.000000
região 16:	área: 304	perímetro: 67.656854	excentricidade: 0.328560	solidez: 1.000000

número de regiões pequenas: 9

número de regiões médias: 8

número de regiões grandes: 0

objetos3

número de regiões: 9

região 0:	área: 4107	perímetro: 319.421354	excentricidade: 0.813036	solidez: 0.754963
região 1:	área: 843	perímetro: 125.639609	excentricidade: 0.735146	solidez: 0.904558
região 2:	área: 3690	perímetro: 265.119838	excentricidade: 0.896234	solidez: 0.978264
região 3:	área: 584	perímetro: 104.911687	excentricidade: 0.882918	solidez: 0.913928
região 4:	área: 478	perímetro: 94.426406	excentricidade: 0.850570	solidez: 0.925460
região 5:	área: 1761	perímetro: 179.781745	excentricidade: 0.864278	solidez: 0.971862
região 6:	área: 688	perímetro: 108.669047	excentricidade: 0.882704	solidez: 0.972458
região 7:	área: 4067	perímetro: 311.078208	excentricidade: 0.908225	solidez: 0.780689
região 8:	área: 716	perímetro: 101.982755	excentricidade: 0.610530	solidez: 0.980164

número de regiões pequenas: 5

número de regiões médias: 1

número de regiões grandes: 3

Podemos analisar que os resultados obtidos com a imagem de entrada Objetos3 está similar aos resultados exibidos no enunciado. Dessa forma, entende-se que as diferenças sutis devem-se à escolhas de implementação, que não afetam significativamente o resultado final.

Pode-se aferir, baseado nos resultados, que as imagens que são quadrados, tem solidez igual a 1.

Além disso, foi possível distinguir o tamanho das imagens visualmente e compará-las com o histograma. Dessa forma, entende-se que a área foi calculada de forma coerente.

4 Conclusão

Nesse trabalho foi possível obter medidas de objetos presentes em imagens digitais, através da transformação de cores, obtenção de bordas, e extração de propriedades como: área, perímetro, excentricidade e solidez.

Verificamos assim, que a biblioteca OpenCV fornece diversas funções que auxiliam essa tarefa, como a função `moments`, mas que foi preciso entender como utilizar essas funções para obter as medidas necessárias.

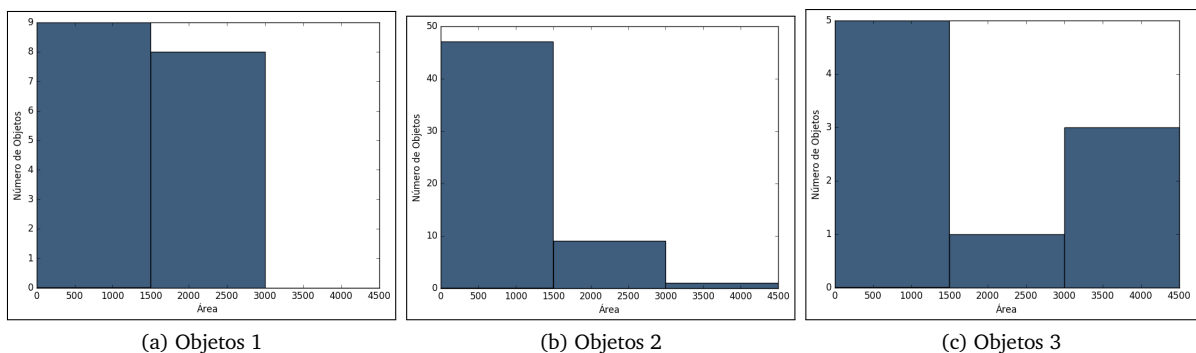
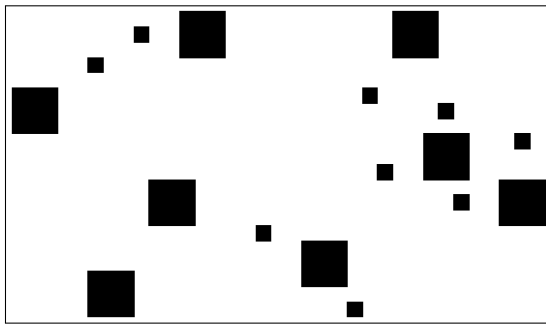


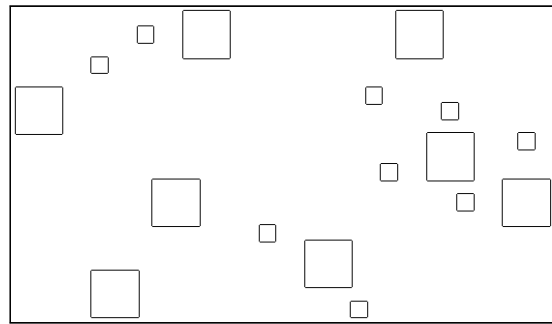
Figura 2: Histograma da área dos objetos das imagens de entrada

Referências

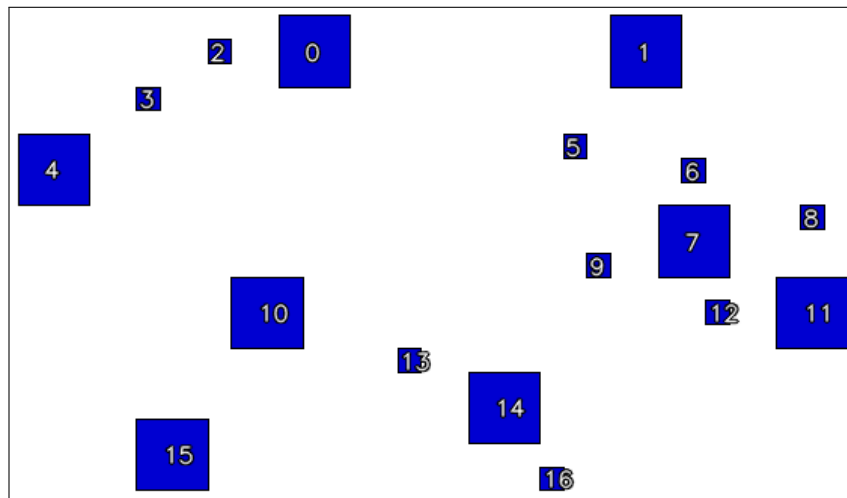
- [1] Opencv: Contour features, 2019. URL https://docs.opencv.org/trunk/dd/d49/tutorial_py_contour_features.html.
- [2] Opencv: Contour properties, 2019. URL https://docs.opencv.org/trunk/d1/d32/tutorial_py_contour_properties.html.
- [3] Wikipédia: Image moment, 2019. URL https://en.wikipedia.org/wiki/Image_moment.



(a) Monocromática

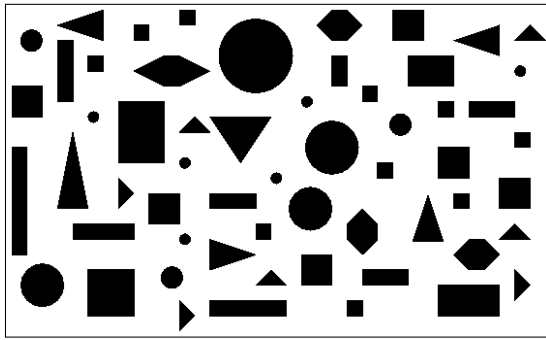


(b) Contorno

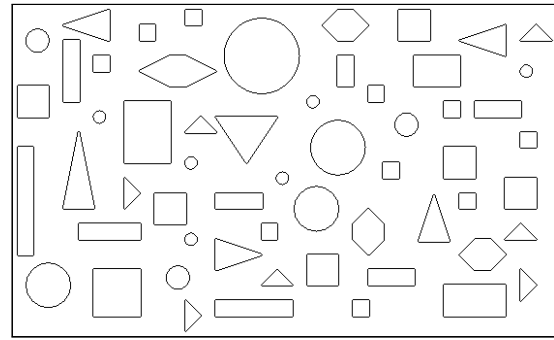


(c) Rótulos

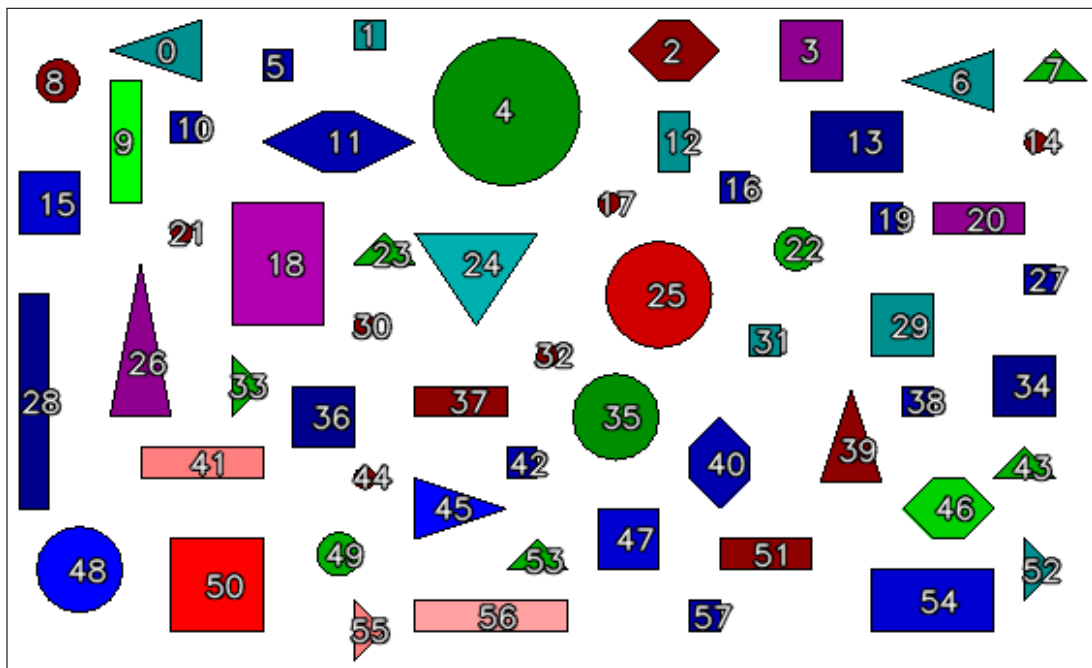
Figura 3: Resultados para Objetos 1



(a) Monocromática



(b) Contorno

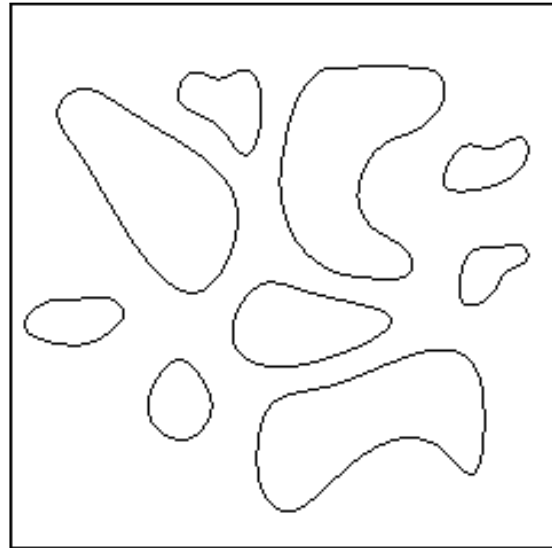


(c) Rótulos

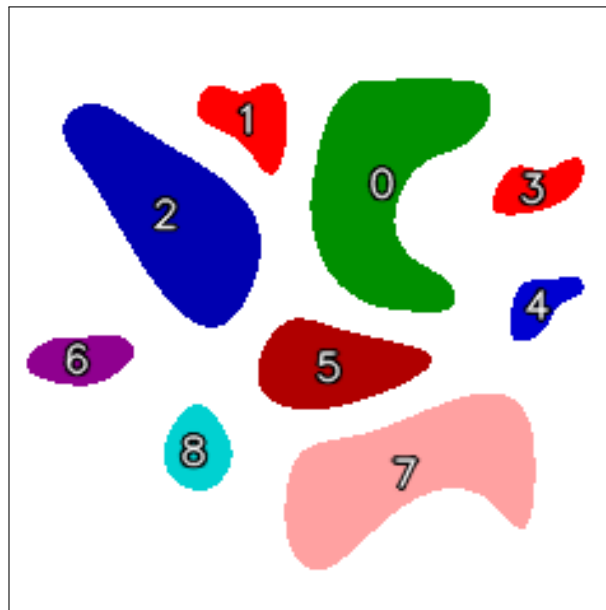
Figura 4: Resultados para Objetos 2



(a) Monocromática



(b) Contorno



(c) Rótulos

Figura 5: Resultados para Objetos 3