

MC920 – Trabalho 3

Rafael Sartori M. Santos, 186154

17 de outubro de 2019

1 Introdução

O objetivo deste trabalho é separar, rotular, contar objetos desconexos, verificar propriedades, como área, perímetro, centroide, isolar contorno e, por fim, classificá-los entre pequenos, médios e grandes pela área ocupada.

Faremos isso utilizando um programa escrito em *Python* utilizando a biblioteca padrão, *OpenCV*, *NumPy* e *scikit-image* para processar as imagens e *Matplotlib* para histograma da área calculada dos objetos da imagem.

2 Método

O trabalho é facilmente divisível em várias etapas. Para cumprir todos os objetivos, será necessário:

- monocromatizar a imagem de entrada,
- rotular objetos desconexos,
- contá-los,
- extrair propriedades,
- isolar contorno,
- classificá-los quanto à área.

A razão para aplicação de cada um desses passos será explicada junto com sua metodologia mais detalhada. Utilizaremos 3 imagens coloridas fornecidas pelo professor, composta por algumas figuras geométricas bem definidas.

Para a aplicação de fato, no entanto, iremos utilizar funções já implementadas de *OpenCV* e *scikit-image*, já que seus funcionamentos não diferem tanto dos métodos descritos nesta seção.

Algumas das funções requerem o tratamento dos resultados de outra função, já que nem sempre são compatíveis.

2.1 Monocromatização

Como a imagem de entrada é colorida, teremos mais de uma camada de cor e isso tornará difícil a identificação de um objeto ou fundo, as bibliotecas nas funções que utilizaremos recebem apenas uma matriz bidimensional (imagem de camada única). Portanto, qualquer tipo de transformação que produza uma imagem de camada única a partir de várias camadas é suficiente.

Na aplicação, abrimos a imagem em modo de escala de cinza através da *flag* de `imread` em *OpenCV*.

2.2 Rotulação e contagem

Com a imagem monocromática, podemos identificar os objetos através do agrupamento usando `vizinhança-4` ou `vizinhança-8`. Podemos fazer isso seguindo este plano:

- inicializamos uma variável que é o número do objeto que estamos identificando atualmente (começa com zero);
- inicializamos a matriz de “resposta” de mesma dimensão que a entrada com zero (ela guardará o rótulo e quais pontos pertencem a esse objeto);
- percorremos a imagem toda; ao encontrarmos um objeto (ponto desconexo em relação ao que estamos – por exemplo, através de uma cor diferente), fazemos:
 - incrementamos a variável do objeto que estamos identificando, esse é o número do objeto atual;
 - navegaremos dentro dele utilizando a vizinhança selecionada, marcando numa matriz de “resposta” o número do objeto atual;
 - ao não possuir mais pontos que não são fundo alcançáveis pela vizinhança, continuamos percorrendo a imagem.

- terminamos a imagem tendo percorrido todos os pontos e rotulado todos os objetos desconexos entre si.

Dessa forma, automaticamente já contamos os objetos presentes (o valor final da variável do número de objetos que identificamos) e temos uma matriz que possui todos os pontos de cada objeto.

Com essa matriz, será possível contar a área, identificar perímetro, encontrar centroide e contorno de cada objeto.

Essa etapa é realizada pelo *scikit-image* através da função `label`, que identifica e rotula objetos desconexos dada uma vizinhança.

2.3 Medir área

Do resultado do método anterior (2.2), podemos medir a área simplesmente contando os pontos que possuem mesmo valor ao rótulo do objeto. Por exemplo, para o rótulo 4, contamos os pontos cujo valor é 4 na matriz retornada.

Esse algoritmo é implementado pela função `getprops` (utilizando a rotulação anterior) do *scikit-image*, que também produz outras propriedades.

2.4 Isolar contorno e medir perímetro

Do resultado da seção 2.2, podemos isolar cada região individualmente. Começamos com uma matriz de zeros e colocamos os pontos de valor igual ao rótulo da região. Por exemplo, para isolar a região 4, selecionamos os pontos com valor 4 da matriz de rótulos.

Para esse isolamento, utilizamos a função `where` de *NumPy*. Isolar a região facilita a análise e é necessário para algumas funções.

Para o contorno, nessa região que isolamos, podemos encontrar utilizando morfologia matemática ou verificar quais pontos possuem vizinhança com o fundo (de valor zero). Utilizamos a implementação da função `find_contours` de *scikit-image*, que faz uso do algoritmo eficiente *matching squares*.

O perímetro poderia ser facilmente estimado contando os pontos do contorno, porém uma aproximação matemática mais precisa ocorre através de interpolação polinomial nesses pontos, o que é feito pela função `perimeter` também de *scikit-image* que utilizamos.

2.5 Calculando centroide

A centroide da imagem é a posição central (C_x, C_y) do objeto na imagem bidimensional, ou seja, a posição mediana em relação a área. Pode ser calculada por decomposição geométrica em pequenos retângulos (os pontos) através da eq. (1).

$$C_l = \frac{\sum C_{i,l} A_i}{\sum A_i} \quad (1)$$

Onde l é a dimensão em que queremos calcular a centroide, $C_{i,l}$ é a coordenada na dimensão l da i -ésima parte da decomposição do objeto C , de área A_i . Essa decomposição é feita automaticamente pela digitalização da imagem (em pequenos quadrados, os *pixels*), facilitando por ser discreto a expressão matemática de centroide.

No código, esse passo já é realizado na seção 2.3 pelo `getprops`.

2.6 Classificação quanto a área

A classificação é mais simples. Com a área A do objeto calculada, basta verificar em qual categoria se encaixa. São elas:

- pequeno, se $A < 1500$;
- médio, se $A \in [1500, 3000)$;
- grande, se $A \geq 3000$.

Com a classificação, produzimos um histograma horizontal em que, no eixo vertical, temos o tamanho do objeto e, no horizontal, a quantidade de objetos.

O histograma é feito e exportado pela biblioteca *Matplotlib*.

3 Resultados

Executamos essa metodologia em 3 imagens fornecidas pelo professor, obtendo os resultados a seguir.

3.1 Monocromatização

Podemos verificar, lado a lado, na fig. 1 as imagens originais e o resultado da monocromatização.

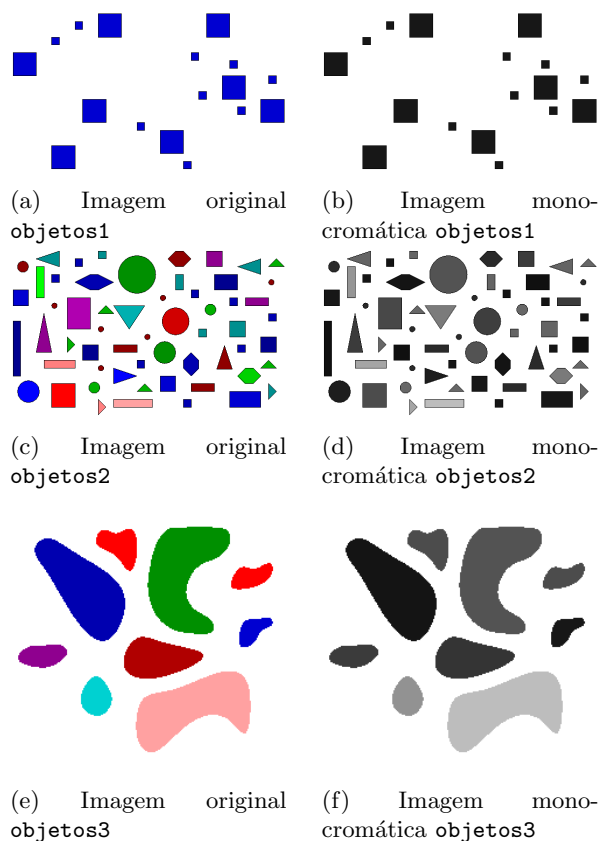


Figura 1: Comparativo do resultado da monocromatização

A monocromatização é um passo simples e os resultados coincidem com o esperado, sem perdas de objetos.

3.2 Rotulação e contagem

A rotulação realizada pelo *scikit-image* é precisa, não levando em consideração nível de cinza, mas sim a conexidade dos pontos, como descrito na metodologia.

Uma limitação do método utilizado, no entanto, é o não descarte do objeto “fundo”. Ao contrário do enunciado do trabalho, nessa implementação, o fundo foi considerado como objeto, dado que existem casos em que seria difícil sua identificação. Por exemplo, estaríamos descartando o objeto errado ao considerarmos algumas soluções de descarte:

- Se escolhessemos o fundo pelo índice zero da lista de objetos identificados e houvesse um objeto no primeiro ponto da imagem (posição (0,0)), ele possuiria índice zero;
- Se escolhessemos o fundo pela maior área e aplicássemos a uma imagem que possui muitos objetos (deixando pouca área ao fundo);

- Se escolhessemos pela cor, necessitaria de intervenção humana relativamente complexa.

Dessa forma, ficamos com essa limitação de considerar o fundo como objeto, mas não temos perdas de objetos errados.

Os resultados serão apresentados posteriormente na seção 3.4, junto com as estatísticas de cada objeto.

3.3 Medição de área e cálculo da centroide

Como descrito no método, para medição da área e cálculo da centroid, temos a solução pronta pela função `getprops` do *scikit-image*. Os resultados coincidem com o apresentado no enunciado e poderão ser verificados na seção 3.4.

3.4 Cálculo do perímetro e isolamento do contorno

Já para o cálculo do perímetro e isolamento do contorno, tivemos que isolar as regiões da imagem, uma a uma, para aplicar as funções da biblioteca.

Salvamos a matriz isolada utilizada para calcular o perímetro e encontrar o contorno: exemplos do isolamento são mostrados na fig. 2. Temos em branco (nesta imagem binária, com valor 1) a região selecionada (objeto) e em preto (0), o fundo. Todas as imagens podem ser verificadas no repositório do trabalho.

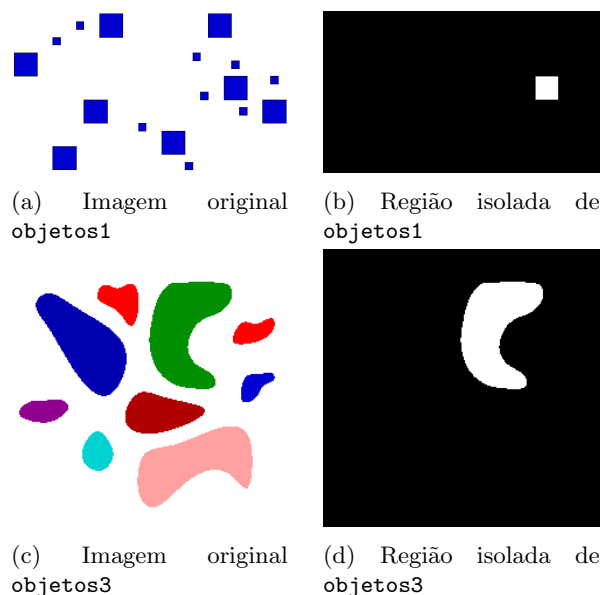


Figura 2: Exemplo de isolamento de regiões

Os resultados numéricos do perímetro podem ser encontrados nas figs. 3 a 5.

Para o contorno, a cada região foi aplicada a função `find_contours` de *scikit-image*, salvamos a matriz binária resultante dessa função (onde 1 é o contorno e 0, o fundo) para verificação.

Para produzir a imagem com os contornos de todas as regiões unificados, somamos todas essas matrizes de contorno (uma para cada região) em uma só matriz acumuladora. Essa matriz também foi exportada como uma imagem, podemos ver para uma região e comparar com o resultado final na fig. 6.

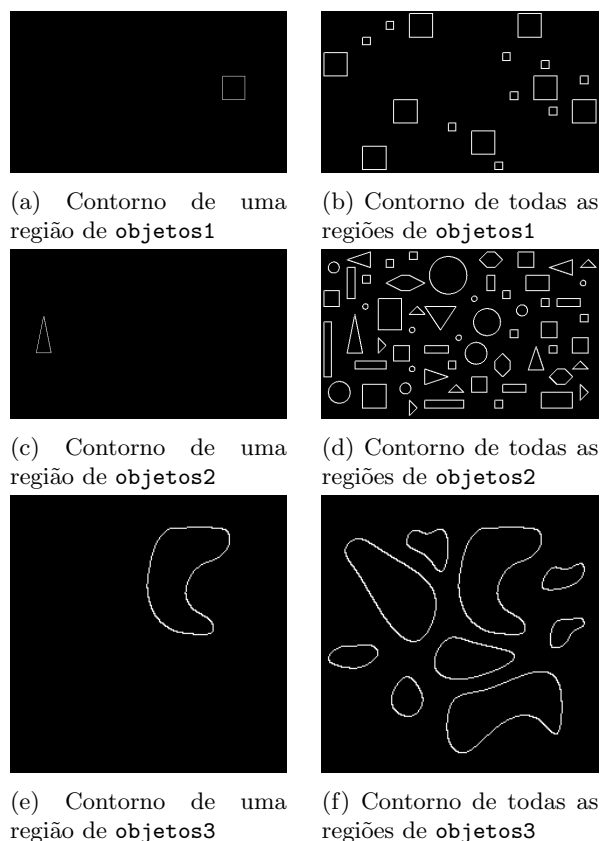
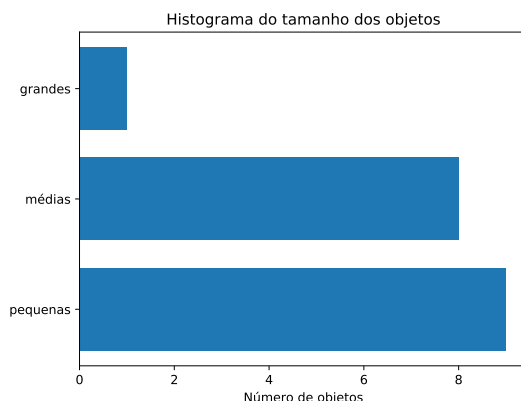


Figura 6: Junção dos contornos das regiões

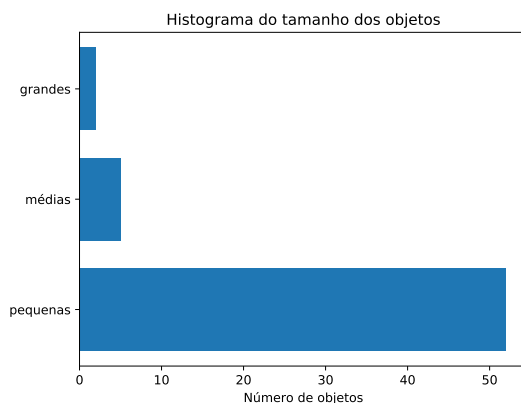
3.5 Classificação quanto área

Como pudemos ver nos resultados numéricos (figs. 3 a 5), as regiões foram corretamente categorizadas quanto à área.

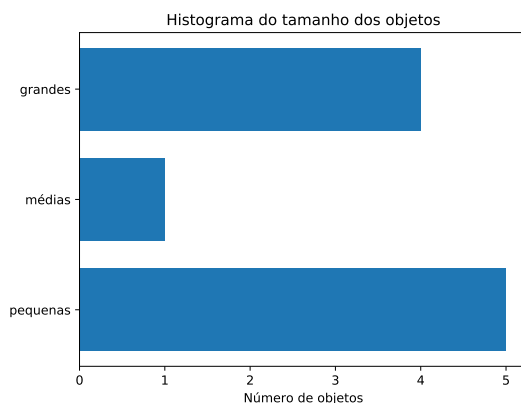
Os histogramas produzidos pelo *Matplotlib* podem ser vistos na fig. 7.



(a) Histograma de objetos1



(b) Histograma de objetos2



(c) Histograma de objetos3

Figura 7: Histograma das imagens

4 Conclusão

O trabalho mostra que, a partir de uma simplificação de uma imagem (que pode ser obtida se as condições podem ser bem controladas), é rápido determinar a área, posição e tamanho de objetos, in-

```

Número de regiões: 18
região 1: área: 162865 centróide: (163.45, 279.92) perímetro: 3954.17
região 2: área: 2162 centróide: (29.00, 202.50) perímetro: 182.00
região 3: área: 2162 centróide: (29.00, 422.50) perímetro: 182.00
região 4: área: 210 centróide: (29.00, 139.50) perímetro: 54.00
região 5: área: 210 centróide: (60.50, 92.00) perímetro: 54.00
região 6: área: 2116 centróide: (107.50, 29.50) perímetro: 180.00
região 7: área: 210 centróide: (92.00, 375.50) perímetro: 54.00
região 8: área: 225 centróide: (108.00, 454.00) perímetro: 56.00
região 9: área: 2162 centróide: (155.00, 454.50) perímetro: 182.00
região 10: área: 225 centróide: (139.00, 533.00) perímetro: 56.00
região 11: área: 225 centróide: (171.00, 391.00) perímetro: 56.00
região 12: área: 2162 centróide: (202.50, 171.00) perímetro: 182.00
região 13: área: 2162 centróide: (202.50, 533.00) perímetro: 182.00
região 14: área: 225 centróide: (202.00, 470.00) perímetro: 56.00
região 15: área: 210 centróide: (234.00, 265.50) perímetro: 54.00
região 16: área: 2116 centróide: (265.50, 328.50) perímetro: 180.00
região 17: área: 2162 centróide: (296.50, 108.00) perímetro: 182.00
região 18: área: 210 centróide: (312.50, 360.00) perímetro: 54.00

número de regiões pequenas: 9
número de regiões médias: 8
número de regiões grandes: 1

```

Figura 3: Informações numéricas de cada região da imagem objetos1

```

Número de regiões: 10
região 1: área: 40387 centróide: (120.47, 117.01) perímetro: 2559.03
região 2: área: 3969 centróide: (68.51, 147.77) perímetro: 313.76
região 3: área: 791 centróide: (43.97, 94.10) perímetro: 119.98
região 4: área: 3584 centróide: (80.95, 63.04) perímetro: 259.46
região 5: área: 540 centróide: (70.46, 206.90) perímetro: 99.25
região 6: área: 438 centróide: (116.38, 208.15) perímetro: 88.77
região 7: área: 1684 centróide: (140.42, 126.64) perímetro: 174.12
região 8: área: 642 centróide: (137.93, 27.62) perímetro: 103.01
região 9: área: 3934 centróide: (182.72, 156.60) perímetro: 305.42
região 10: área: 675 centróide: (173.38, 73.65) perímetro: 96.33

número de regiões pequenas: 5
número de regiões médias: 1
número de regiões grandes: 4

```

Figura 4: Informações numéricas de cada região da imagem objetos3

Número de regiões: 59

região 1: área: 141699 centróide: (172.96, 287.20) perímetro: 8688.86

região 2: área: 690 centróide: (21.50, 84.35) perímetro: 127.18

região 3: área: 210 centróide: (13.50, 187.00) perímetro: 54.00

região 4: área: 930 centróide: (21.50, 344.00) perímetro: 113.20

região 5: área: 930 centróide: (21.50, 415.00) perímetro: 118.00

região 6: área: 4402 centróide: (52.96, 257.51) perímetro: 245.62

região 7: área: 210 centróide: (29.00, 139.50) perímetro: 54.00

região 8: área: 225 centróide: (31.49, 541.00) perímetro: 67.60

região 9: área: 660 centróide: (37.16, 493.66) perímetro: 122.98

região 10: área: 356 centróide: (37.21, 25.87) perímetro: 66.28

região 11: área: 930 centróide: (68.50, 61.00) perímetro: 150.00

região 12: área: 225 centróide: (61.00, 92.00) perímetro: 56.00

região 13: área: 1395 centróide: (68.58, 170.75) perímetro: 171.78

região 14: área: 450 centróide: (68.50, 344.00) perímetro: 86.00

região 15: área: 1380 centróide: (68.50, 438.50) perímetro: 148.00

região 16: área: 82 centróide: (68.67, 530.79) perímetro: 29.31

região 17: área: 930 centróide: (100.00, 21.50) perímetro: 118.00

região 18: área: 210 centróide: (92.00, 375.50) perímetro: 54.00

região 19: área: 82 centróide: (100.23, 310.44) perímetro: 30.73

região 20: área: 2852 centróide: (131.50, 139.50) perímetro: 212.00

região 21: área: 225 centróide: (108.00, 454.00) perímetro: 56.00

região 22: área: 690 centróide: (108.00, 501.50) perímetro: 118.00

região 23: área: 79 centróide: (115.90, 89.78) perímetro: 31.56

região 24: área: 353 centróide: (123.80, 407.00) perímetro: 67.11

região 25: área: 240 centróide: (126.33, 194.50) perímetro: 69.60

região 26: área: 1411 centróide: (131.84, 241.84) perímetro: 174.23

região 27: área: 2248 centróide: (147.45, 336.20) perímetro: 174.27

região 28: área: 210 centróide: (139.50, 533.00) perímetro: 54.00

região 29: área: 1140 centróide: (184.36, 68.50) perímetro: 185.18

região 30: área: 1540 centróide: (202.50, 13.50) perímetro: 244.00

região 31: área: 961 centróide: (163.00, 462.00) perímetro: 120.00

região 32: área: 82 centróide: (163.23, 184.44) perímetro: 30.73

região 33: área: 225 centróide: (171.00, 391.00) perímetro: 56.00

região 34: área: 80 centróide: (178.91, 278.73) perímetro: 31.56

região 35: área: 930 centróide: (194.50, 525.00) perímetro: 118.00

região 36: área: 210 centróide: (194.50, 121.33) perímetro: 64.77

região 37: área: 1457 centróide: (210.43, 313.99) perímetro: 141.88

região 38: área: 930 centróide: (210.50, 163.00) perímetro: 118.00

região 39: área: 658 centróide: (202.50, 234.00) perímetro: 118.00

região 40: área: 210 centróide: (202.50, 470.00) perímetro: 54.00

região 41: área: 690 centróide: (228.35, 435.50) perímetro: 127.18

região 42: área: 945 centróide: (234.25, 367.63) perímetro: 97.64

região 43: área: 930 centróide: (234.00, 100.50) perímetro: 150.00

região 44: área: 210 centróide: (234.00, 265.50) perímetro: 54.00

região 45: área: 225 centróide: (236.49, 525.00) perímetro: 67.60

região 46: área: 81 centróide: (241.93, 184.44) perímetro: 28.73

região 47: área: 690 centróide: (257.83, 225.67) perímetro: 126.18

região 48: área: 946 centróide: (257.64, 485.72) perímetro: 96.23

região 49: área: 900 centróide: (273.50, 320.50) perímetro: 116.00

região 50: área: 1450 centróide: (289.15, 37.04) perímetro: 139.88

região 51: área: 356 centróide: (281.24, 170.92) perímetro: 65.70

região 52: área: 2209 centróide: (297.00, 108.00) perímetro: 184.00

região 53: área: 690 centróide: (281.00, 391.50) perímetro: 118.00

região 54: área: 210 centróide: (283.67, 273.50) perímetro: 64.77

região 55: área: 225 centróide: (289.00, 530.51) perímetro: 67.60

região 56: área: 1922 centróide: (305.00, 477.50) perímetro: 182.00

região 57: área: 1170 centróide: (313.00, 249.50) perímetro: 182.00

região 58: área: 225 centróide: (313.00, 360.00) perímetro: 56.00

região 59: área: 210 centróide: (320.50, 184.33) perímetro: 64.77

número de regiões pequenas: 52

número de regiões médias: 5

número de regiões grandes: 2

Figura 5: Informações numéricas de cada região da imagem objetos2

formações muito requisitadas em automação industrial, por exemplo, para verificação.

Com as informações obtidas, como o contorno, conseguimos comparar com o que era esperado através de funções matemáticas e/ou outros algoritmos estudados na disciplina.

Os resultados foram bastante satisfatórios e pouca correção foi necessária para aplicação das funções já implementadas e testadas pelas bibliotecas.