

Universidade Estadual de Campinas

MC920 - Introdução ao processamento de
imagem digital

Trabalho 3

Felipe Hideki Matoba
RA: 196767

Sumário

1	Resumo	2
2	Ambiente de execução	2
3	Transformação de cores	3
4	Detecção dos contornos	4
5	Extração das propriedades dos objetos	6
6	Histograma de áreas	9
7	Conclusão	10

1 Resumo

Iremos explorar algumas métricas usadas para descrever objetos em imagens, como área, perímetro, excentricidade, etc. Para obtê-las, é necessário realizar um pré-processamento, convertendo a imagem em tons de cinza ou monocromática e identificando as regiões a serem estudadas.

Dessa vez, boa parte do código utiliza funções fornecidas por bibliotecas externas, então é importante atentar-se à teoria por trás das funções para o entendimento dos valores que serão obtidos.

2 Ambiente de execução

O código foi feito utilizando o Jupyter Notebook, com a inclusão das seguintes bibliotecas:

- skimage, versão 0.17.2
- cv2 (opencv), versão 4.0.1
- numpy, versão 1.19.2
- matplotlib, versão 3.3.2

Para executar o código, basta executar cada célula do notebook *project3.ipynb*, com o cuidado de rodar a importação das bibliotecas pertinentes e as declarações das funções primeiro. A imagem de entrada pode ser escolhida pela variável “img_name” no início de cada célula, desde que esteja dentro da pasta “in”; as imagens de saída são salvas na pasta “out”.

3 Transformação de cores

Primeiramente, serão feitas funções para converter as imagens de RGB para escala de cinza ou monocromáticas, o que é necessário para a aplicação dos algoritmos que serão utilizados posteriormente.

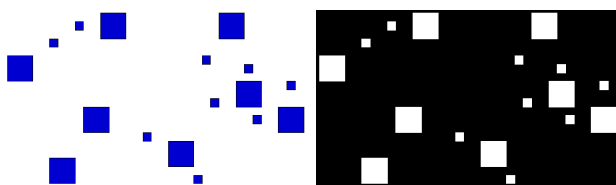
A transformação em escala de cinza utilizada foi a mesma do trabalho 1, dada abaixo, enquanto a monocromática foi feita simplesmente transformando os pixels coloridos em brancos (valor 1) e os brancos em pretos (valor 0), pois queremos o background preto com os objetos brancos. Ambas foram implementadas usando o numpy e tiveram um bom desempenho, convertendo as imagens de exemplo em alguns milésimos de segundo.

$$I = 0.2989R + 0.5870G + 0.1140B$$



(a) Imagem original (b) Tons de cinza

Figura 1: Imagem original e em tons de cinza



(a) Imagem original (b) Monocromática

Figura 2: Imagem original e monocromática

4 Detecção dos contornos

Utilizaremos duas funções diferentes para identificar o contorno os objetos, ambas provenientes da biblioteca OpenCV. A primeira é a função *findContours*, que recebe uma imagem monocromática e a segunda corresponde ao algoritmo de Canny, que por sua vez pode ser feito com uma imagem em tons de cinza. O desempenho obtido pelas funções foi semelhante e seus resultados podem ser vistos a seguir.

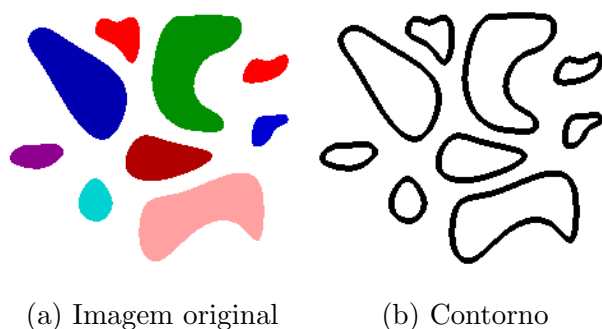


Figura 3: Imagem original e seu contorno obtido pela função *findContour*

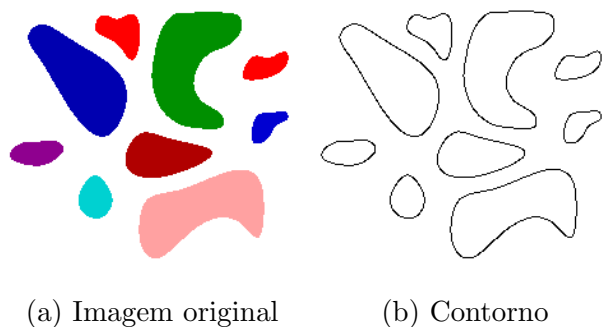


Figura 4: Imagem original e seu contorno obtido pelo método de Canny

Ambos os métodos encontraram o contorno da imagem muito bem, mesmo esta apresentando um formato curvo, irregular. No entanto, o contorno resultante da função *Canny* é mais fino, aparentemente com apenas um pixel de espessura, o que pode ser mais adequado para encontrar algumas propriedades dos objetos, como por exemplo o perímetro.

Vale destacar que a função *Canny* recebe, além da imagem, dois valores de *threshold*, cuja escolha não foi tão crítica devido ao bom comportamento dos objetos da imagem, mas que em problemas mais complicados pode alterar significativamente o resultado do algoritmo. Nesse caso, foram utilizados valores de 100 e 255 para os limiares, sendo que outros valores razoáveis obtiveram o mesmo resultado.

5 Extração das propriedades dos objetos

Para a extração das propriedades desejadas, foi utilizada a função *regionprops_table* da biblioteca *skimage*. Para poder usá-la, antes foi necessário identificar e rotular cada pixel com o objeto ao qual ele pertence. Com esse objetivo, primeiro a imagem foi convertida para monocromática para, em seguida, aplicar a função *connectedComponents*, que realiza a rotulação dos objetos.

Ressalto que a função *connectedComponents*, apesar de ter funcionado muito bem nos exemplos fornecidos, pode não ser suficiente em imagens mais complexas, especialmente em casos de objetos que se tocam. Nesses casos, outras técnicas provavelmente apresentariam um melhor resultado, como por exemplo a segmentação por divisor de águas, mas que exigem um maior trabalho de pré-processamento que se mostraram desnecessários neste trabalho.

Na próxima página temos a imagem com cada região rotulada e suas respectivas propriedades obtidas.



Figura 5: Imagem rotulada

Região	Área	Perímetro	Excentricidade	Solidez
1	3969	313.765	0.81636	0.74774
2	791	119.983	0.74110	0.89886
3	3584	259.463	0.89807	0.97790
4	540	99.255	0.88959	0.91062
5	438	88.770	0.85592	0.91632
6	1684	174.125	0.86817	0.97229
7	642	103.012	0.89024	0.96979
8	3934	305.421	0.91099	0.77426
9	675	96.326	0.62038	0.97685

A área e o perímetro das regiões são autoexplicativas, mas é interessante notar que a medida do perímetro não é um valor exato, então pode-se concluir que seu valor não é obtido simplesmente somando o número de pixels da borda de cada região. Para comparação, o perímetro também foi calculado somando os pixels do contorno de cada região obtido por Canny, o que gerou resultados bem similares, que podem ser verificados no notebook do trabalho.

A excentricidade é uma medida que representa a relação entre os comprimentos dos eixo maior e menor de um objeto, onde valores mais altos indicam um desequilíbrio entre esses valores. Pela documentação da biblioteca, a excentricidade é calculada como a distância entre os pontos de foco de uma elipse com um momento de segunda ordem equivalente, onde valores próximos de zero se aproximam mais de um círculo.

Por fim, temos a solidez do objeto, que é a razão entre sua área e a área do seu fecho convexo. Nesse caso, uma solidez igual a 1 representa um objeto convexo e quanto menor esse valor, mais “buracos” a região possui.

6 Histograma de áreas

Por fim, iremos classificar os objetos de acordo com sua área, de acordo com os seguintes critérios:

- Objetos pequenos: área menor que 1500 pixels;
- Objetos médios: área entre 1500 e 3000 pixels;
- Objetos grandes: área maior que 3000 pixels.

A partir dessa classificação, será gerado um histograma de três colunas, cada uma com altura igual ao número de objetos em cada classe.

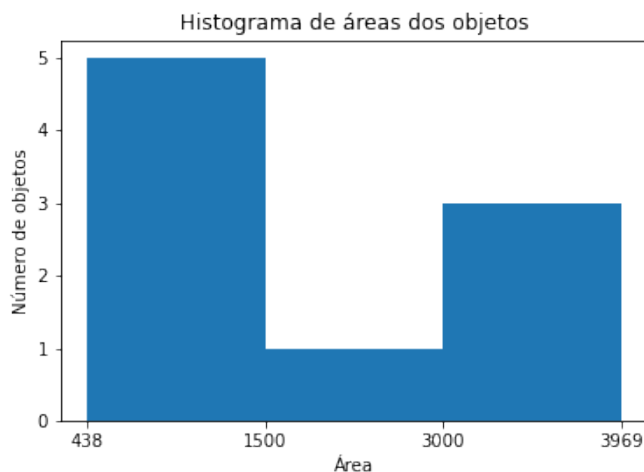


Figura 6: Histograma da Figura 5

A primeira marcação representa a menor área dentre os objetos pequenos (quando há elementos nessa classe). Similarmente, a última marcação representa a maior área dos objetos grandes. A largura das colunas não está em escala.

7 Conclusão

Neste trabalho, ao invés de trabalharmos em maneiras de modificar e filtrar as imagens, exploramos como identificar e extrair propriedades de objetos, o que possui diversas aplicações, como por exemplo na área de aprendizado de máquina para reconhecimento e classificação de imagens.

As características extraídas foram apenas uma parte dos descritores disponíveis, estes que podem ter uma grande importância em aplicações mais específicas, mas que fogem do escopo de estudo para esse primeiro olhar na classificação de objetos. Ainda assim, é essencial ter esse contato inicial e aprender os passos necessários para possamos utilizar as funções e interpretar seus resultados corretamente.