

Segunda Atividade Prática de Processamento Digital de Imagens

Francisco Jair de Oliveira Neres¹

¹Departamento de Computação – Universidade Federal do Piauí (UFPI)

Teresina – PI – Brasil

fjair123@gmail.com

1. Introdução

O objetivo dessa atividade consiste em explorar na prática os conceitos básicos aprendidos até o momento na disciplina de Processamento Digital de Imagens. Neste sentido, a atividade consistiu em desenvolver um programa que a partir da imagem ‘aula18_Pratica02.png’, enviada pelo professor e mostrada a seguir, resolvesse os seguintes problemas listados, fornecendo como resposta uma imagem de saída, relacionada com o que se pede:

1. Eliminar todos os pontos pretos;
2. Preencher os buracos dos objetos: azul, amarelo e verde;
3. Encontrar o fecho convexo dos objetos: azul, amarelo e verde;
4. Encontre o esqueleto da imagem de cor verde, azul e amarelo;
5. Utilizando a transformada hit-or-miss, localize cada um dos objetos vermelhos restantes na imagem.

Além disso, também foi proposto que as funções executassem e fornecessem a saída da forma mais automática possível, bem como que cada problema fosse resolvido a partir da imagem original.

2. Desenvolvimento

O código para resolver os problemas propostos pelo professor foi feito utilizando a linguagem Python, com auxílio das bibliotecas OpenCV e NumPy.

2.1. Código

```
import cv2
import numpy as np

def eliminar_pontos Pretos(imagem):
    black_pixels = np.where(
        (imagem[:, :, 0] == 0) &
        (imagem[:, :, 1] == 0) &
        (imagem[:, :, 2] == 0)
    )
    imagem[black_pixels] = [255, 255, 255]
    cv2.imwrite("imagem_sem pontos.png", imagem)
```

```

    return imagem

def preencher_buracos(imagem):
    imagem_complemento = cv2.bitwise_not(imagem)
    contador, mais_alto = cv2.findContours(imagem_complemento,
cv2.RETR_CCOMP, cv2.CHAIN_APPROX_SIMPLE)
    for conta in contador:
        cv2.drawContours(imagem_complemento, [conta], 0, 255, -1)
        imagem_preenchida = cv2.bitwise_not(imagem_complemento)
        cv2.imwrite("imagem_preenchida.png", imagem_preenchida)
    return imagem_preenchida

def main():
    imagem_original = cv2.imread('morfologia.png')
    imagem_sem_pontos = eliminar_pontos Pretos(imagem_original)
    imagem_escala_cinza = cv2.cvtColor(imagem_sem_pontos,
cv2.COLOR_BGR2GRAY)
    imagem_binaria = cv2.threshold(imagem_escala_cinza, 250, 255,
cv2.THRESH_BINARY)[1]
    preencher_buracos(imagem_binaria)
    cv2.waitKey(0)

main()

```

2.2. Comentários sobre o código

Eliminando todos os pontos pretos função principal chama a função `eliminar_pontos_pretos` que é responsável por remover os pontos pretos da imagem, a função recebe como parâmetro a imagem e retorna a imagem sem pontos. A função principal chama a função `preencher_buracos` que é responsável por preencher os buracos das imagens de todas as cores. A função recebe como parâmetro a imagem sem pontos pretos, retornada pela função `eliminar_pontos_pretos`. A função então calcula o inverso da imagem original, que possui os buracos a serem preenchidos. Com o auxílio das funções `findContours` e `drawContours`, da biblioteca OpenCV, são procurados os buracos e preenchidos. Então a função `eliminar_pontos_pretos` retorna a imagem com os objetos de todas as cores preenchidos.

3. Resultados

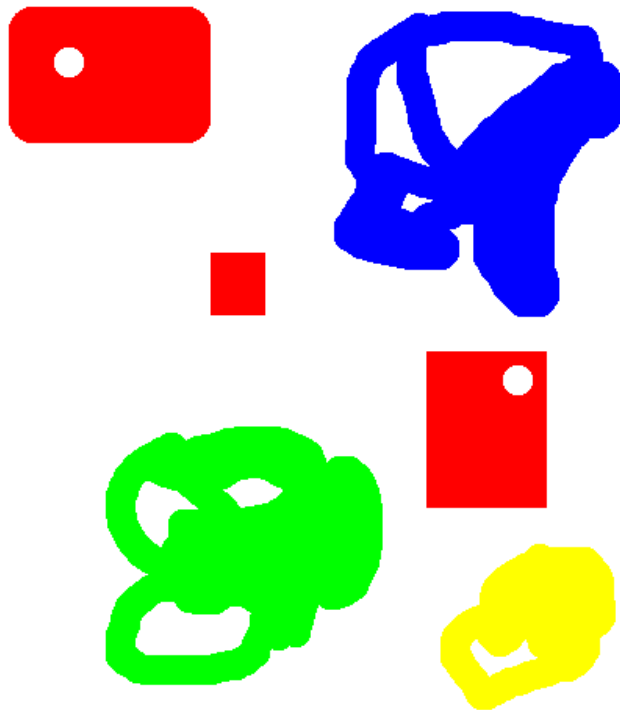


Figura 1. Imagem sem os pontos pretos.

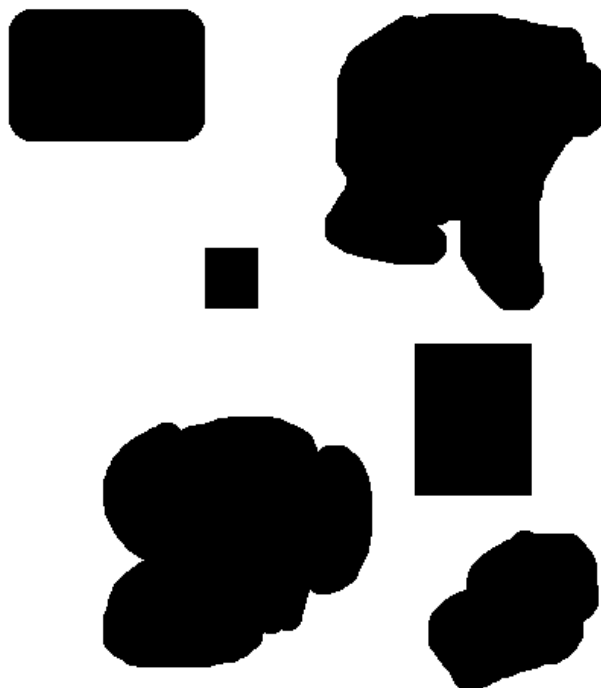


Figura 2. Imagem preenchida.