

Relatório

Disciplina: Processamento digital de imagens

Discentes:

Diego Takahashi RA: 109889

Pedro H. Landins RA: 103572

Introdução

O objetivo do trabalho é estudar e implementar dois operadores morfológicos para imagens binárias: o fechamento de buracos e a limpeza de bordas. Assim sendo, precisamos entender alguns conceitos necessários.

A morfologia matemática é sobre teoria de conjuntos. Tal fato nos permite usar abordagens poderosas para processamento de imagens, já que objetos encontrados em uma imagem são equivalentes aos conjuntos em morfologia matemática. Um exemplo seria abordar todos os pixels brancos de uma imagem binária como um conjunto. Outro exemplo é representar imagens em níveis de cinza como um conjunto em Z^3 , onde em uma representação (x, y, z) x e y seriam coordenadas de um pixel e z , a intensidade.

Com isso em mente, podemos falar de **preenchimento de buracos**. Um buraco pode ser definido, de acordo com Gonzales, como uma região de fundo rodeada por um contorno de pixels de frente conectados. E para preenchê-los é necessário um algoritmo usando erosão, complemento e interseção de conjuntos que é explicada na implementação.

Já a **limpeza das bordas** pode ser definida como um algoritmo que remove os objetos que tocam as bordas da imagem. Um exemplo da utilização desse algoritmo é sinalizar que objetos parciais, pois não estão completos, estão presentes em um campo de visão.

Implementação

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
from skimage.morphology import reconstruction

def preencherburacos(img):
    seed = img.copy()
    seed[1:-1, 1:-1] = img.max()
    mask = img
    selem = np.ones([3] * seed.ndim, dtype=int)
    prechida = reconstruction(seed, mask, method='erosion',
selem=selem)
    return prechida
```

```
def limpezadebordas(img):
    seed = img.copy()
    seed[1:-1, 1:-1] = img.min()
    mask = img
    selem = np.ones([3] * seed.ndim, dtype=int)
    limpa = reconstruction(seed, mask, method='dilation', selem=selem)
    return img - limpa

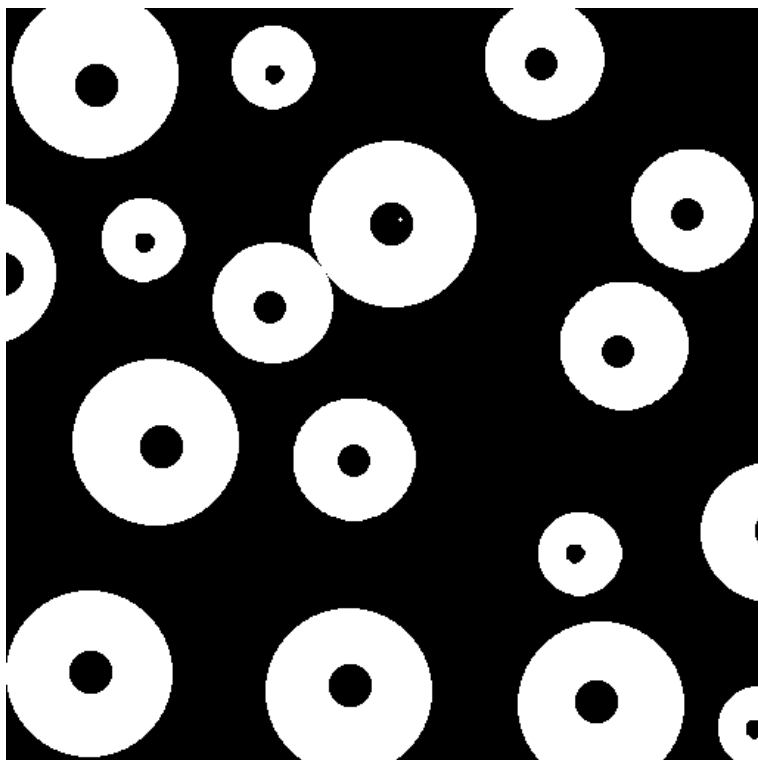
img = cv2.imread('image.tif', 0)
cv2.imshow('original', img)
preenchida = preencherburacos(img)
limpa = limpezadebordas(img)
cv2.imshow('preenchida', preenchida)
cv2.imshow('bordas limpas', limpa)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Na função “**preencherburacos**” tem como entrada uma imagem binária e em seguida é inicializada uma imagem “*seed*” com o valor máximo da imagem original e uma máscara com os valores da imagem original, o selem é nosso elemento estruturante conexo. Utilizando a “*reconstruction*” por erosão, podemos isolar as regiões mais escuras subtraindo-as da imagem reconstruída da imagem original, assim preenchendo os buracos.

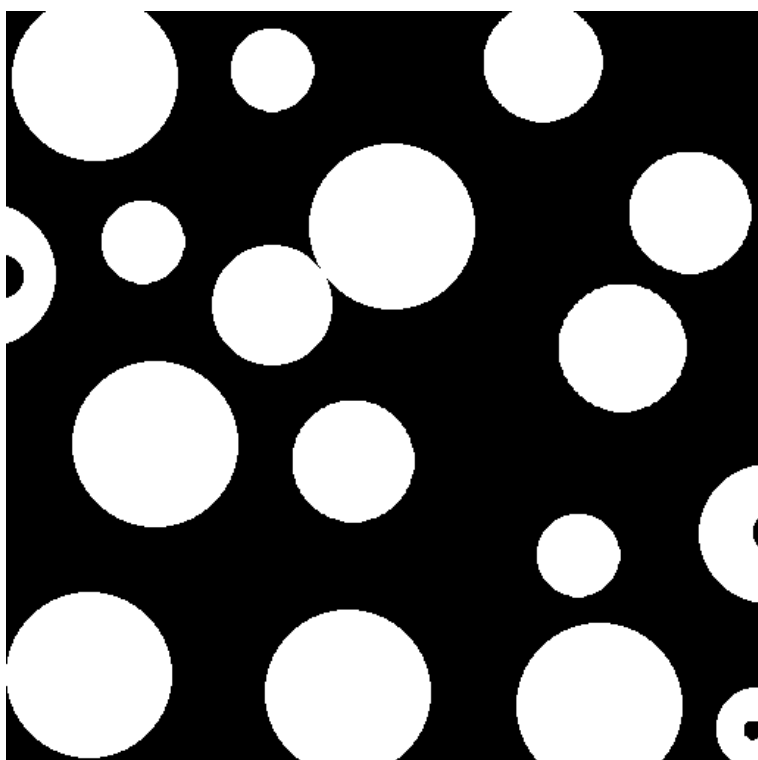
Na função “**limpezadebordas**” funciona de forma similar à função anterior. A diferença é que a imagem “*seed*” recebe o valor mínimo da imagem original e a função “*reconstruction*” utiliza o método de dilatação e ao subtrair o resultado da reconstrução da imagem original gera a imagem com bordas limpas.

Resultados

Abaixo temos a imagem binária utilizada:

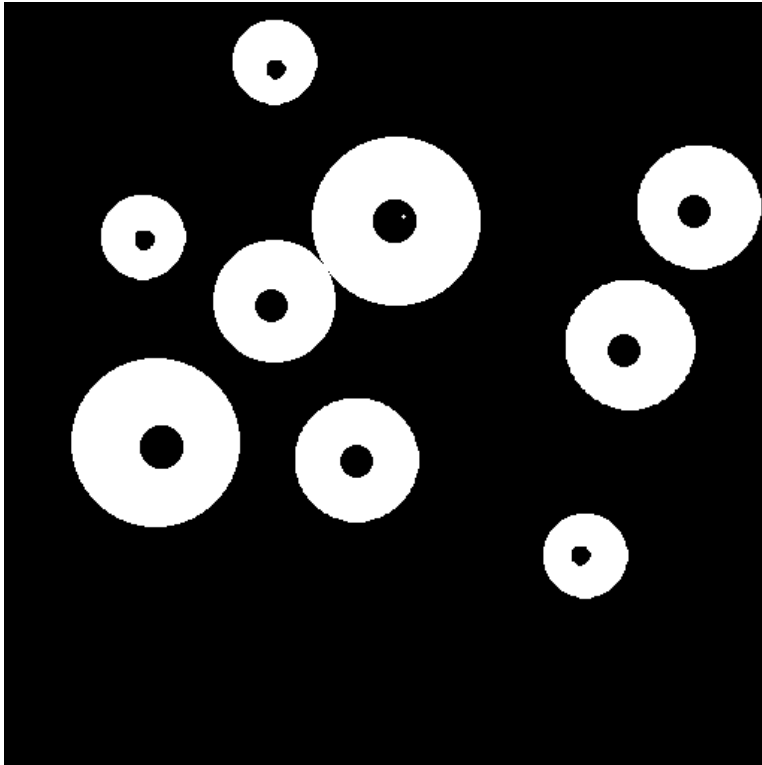


Após a aplicar o fechamento de buracos temos:



Os buracos que tocam as bordas não são fechados.

Após aplicar a limpeza de bordas:



Bibliografias

scikit-image development team. Filtering regional maxima. Disponível em https://scikit-image.org/docs/dev/auto_examples/color_exposure/plot_regional_maxima.html#sphx-glr-auto-examples-color-exposure-plot-regional-maxima-py

scikit-image development team. Filling holes and finding peaks. Disponível em https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_holes_and_peaks.html#sphx-glr-auto-examples-features-detection-plot-holes-and-peaks-py >

GONZALEZ, Rafael C.; WOODS, Richard E. Processamento digital de imagens 3a edição. Pearson 3 de dezembro de 2009.

