

Relatório Trabalho 2 – IPI

Gustavo Pereira Chaves
Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
gustavopchaves@hotmail.com

Keywords— *imagem, processamento de imagem, binarização, algoritmos morfológicos, segmentação*

I RESUMO

Os seguintes experimentos foram realizados com o fim de testar a binarização de imagens, buscando encontrar o limiar adequado para cada uma. Além disso, foram realizados testes em algoritmos morfológicos tanto em níveis de cinza quanto em imagens binárias, de forma a aplicar as transformações requeridas em cada um dos problemas.

A respeito dos exercícios, o primeiro propõe uma comparação entre a binarização direta de uma imagem em relação à utilização de algoritmos morfológicos antes desse processo. O segundo pede a eliminação de um objeto de uma imagem binária utilizando também dos algoritmos mencionados. E por fim o terceiro problema pede a segmentação watershed de uma imagem em escala de cinza, aplicando certos processamentos para tornar isso possível.

Os códigos foram todos desenvolvidos em Python, utilizando as bibliotecas OpenCV e Numpy, havendo sido testados em todas as imagens propostas fornecidas.

II INTRODUÇÃO

Este trabalho teve por finalidade a aplicação dos conhecimentos obtidos em sala de aula com relação a binarização de imagens, a aplicação de algoritmos morfológicos tanto em escala de cinza quanto em imagens binárias, e a segmentação de objetos em uma imagem.

Inicialmente, é preciso discutir o processo de binarização de uma imagem, que consiste em tornar todos os pixels desta em preto ou branco. Para tal, define-se o que é chamado limiar, que será o brilho mínimo necessário de um pixel para este ser transformado em branco. Caso isso não ocorra, este torna-se preto.

Em segunda instância necessita-se entender o funcionamento dos algoritmos morfológicos. Estes tratam da transformação de uma imagem baseado nas formas dos objetos contidos nesta. Todos esses algoritmos são baseados em dois processos básicos, a erosão e a dilatação. Estes, por sua vez, possuem funcionamentos diferentes se executados em imagens binárias ou em níveis de cinza. Em imagens binárias, enquanto a erosão promove um afinamento das formas, a dilatação provoca o engrossamento destas. Já em escala de cinza, em geral, a erosão aumenta o tamanho de objetos escuros e reduz o

de objetos claros, enquanto que a dilatação aumenta o tamanho de objetos claros e reduz o tamanho dos escuros.

Por fim, é adequado descrever sobre a segmentação, que nada mais é que a separação da imagem em diferentes regiões utilizando um mesmo critério. Neste caso foi utilizado o watershed, que separa essas regiões para objetos que estão encostando uns nos outros através de marcadores que definem as regiões de transição entre o objeto e o fundo da imagem.

De posse dessas informações, pode-se dar continuidade a exposição do experimento.

III METODOLOGIA

III.A Questão 1

Na questão é solicitada a binarização direta de uma imagem, transformando o fundo em branco e as letras em preto. Logo após é pedido que seja feita uma nova binarização, porém utilizando algoritmos morfológicos em níveis de cinza antes do processo.

Dessa forma, foi criada uma função denominada “binarize” que recebe como argumentos a imagem e o limiar a serem utilizados na binarização. Após vários testes, encontrando-se que para a foto em questão o limiar = 170 era adequado, foi feito então o processo.

Ademais, passando para a segunda parte da questão, verificou-se que a imagem original possui uma quantidade considerável de ruído, que reflete no resultado da binarização direta. De forma a contornar esse problema, era preciso aplicar o algoritmo morfológico Top-Hat, que para imagens em escala de cinza, consiste no realce de objetos claros em regiões escuras. Para tal, inverteu-se a coloração da imagem utilizando a função `bitwise_not` do OpenCV e realizado o algoritmo utilizando como elemento estruturante uma matriz quadrada de lado 15. Verificou-se que o resultado apresentava as cores ainda invertidas, logo foi novamente utilizada a função já mencionada para retornar ao padrão original. Por fim, utilizando o limiar = 220, é obtida a imagem final binarizada.

III.B Questão 2

Nessa questão é pedida a criação de um algoritmo que remova o cookie mordido da direita sem afetar a forma do cookie da esquerda.

Portanto, primeiro foi binarizada a imagem original utilizando um limiar = 100. Para eliminar o cookie mordido,

utilizou-se o algoritmo morfológico Abertura, que para imagens binárias, além de suavizar o contorno, também remove objetos menores que o elemento estruturante, definido como uma elipse de raio 118. Removido o cookie mordido, era preciso retornar o cookie resultante a forma original. Aplicou-se o então o algoritmo morfológico de dilatação nessa imagem resultante, com uma elipse de raio 20 como elemento estruturante, e posteriormente feito um “and” lógico com a imagem binarizada original, o que teve como retorno uma máscara para o cookie não mordido. Novamente feito um “and” lógico entre a máscara e imagem original de entrada, obtém-se o resultado final com somente o objeto da esquerda.

III.C Questão 3

Nessa questão é pedido para realizar a segmentação da imagem de entrada utilizando o algoritmo watershed e realizando pré-processamentos para tornar isso possível.

Inicialmente binarizou-se a imagem usando o limiar = 150. Feito isso, é necessário então preencher os buracos em cada um dos objetos. Para isso, a imagem foi invertida e passada para a função `findContours` do OpenCV, que gera uma matriz correspondente aos contornos de cada objeto na imagem. Foram passados esses contornos para a função `drawContours` do OpenCV, que os preenchendo com a cor branca, eliminou a maior parte dos pontos pretos dentro das células.

Verificando que o resultado apresentava alguns ruídos, foi utilizado duas vezes o algoritmo morfológico Abertura, com elemento estruturante sendo um quadrado de lado 3. Com a imagem resultante, buscou-se encontrar a região de transição entre as células e o fundo da imagem, para finalmente gerar os marcadores para a função `watershed`.

Dessa forma, primeiro encontrou-se a região denominada “background” que engloba o objeto e parte do fundo, bastando realizar a transformação de dilatação três vezes com a imagem anterior, utilizando o mesmo elemento estruturante da remoção de ruído.

Depois foi encontrado o “foreground”, região que certamente está contido dentro de uma célula. Para tal, utilizou-se a função `distanceTransform` do OpenCV, que substitui cada pixel pela sua distância até o fundo da imagem. Aplicando a função `threshold` no resultado dessa operação, filtrou-se somente os picos, que indicam os pontos mais distantes do fundo, ou seja, o centro de cada célula.

Com isso, é possível encontrar a região de transição entre os dois, que foi denominada “border”, sendo necessário somente subtrair o o foreground do background.

Após tudo isso, torna-se possível criar os marcadores. Utilizando a função `connectedComponents` do OpenCV no foreground, cada centro de célula recebe um valor inteiro positivo, enquanto o restante recebe zero, criando a matriz denominada “markers”, que enumera todas as células. No entanto, para aplicar a segmentação, somente a região de borda deve ser zero. Soma-se então o valor “um” a todos os elementos da matriz e posteriormente, utilizando a imagem “background”, define-se zero na matriz `markers` todas as regiões de borda.

Finalizado a criação dos marcadores, bastou utilizar a função `watershed` do OpenCV, passando a imagem de entrada

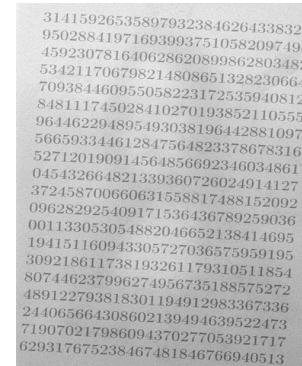
original e a matriz de marcadores, que retorna como resultado os contornos dos objetos encontrados. Por fim, adicionou-se esses contornos a imagem original, obtendo o resultado final.

IV RESULTADOS

Após a descrição da metodologia desenvolvida em cada questão, a seguir serão apresentadas as imagens geradas por cada algoritmo a partir de suas entradas.

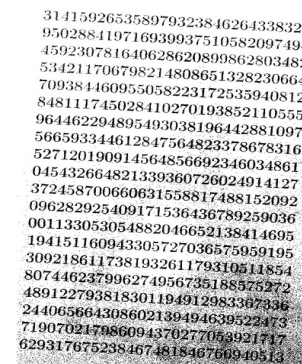
IV.A Questão 1

Foi passada então a seguinte imagem para a Questão 1:



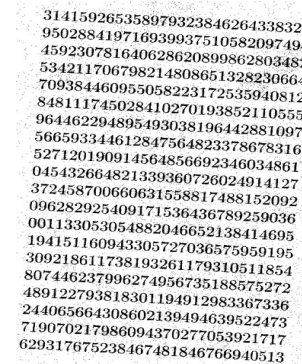
1 Imagem de entrada “*morf_test*”

Binarizando a imagem diretamente obtém-se o seguinte:



2 Imagem binarizada diretamente

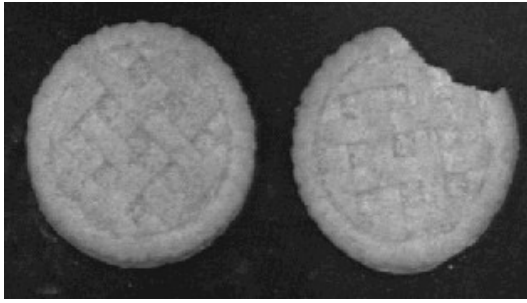
A seguir a imagem após a inversão, utilizando da transformação Top-Hat e uma nova inversão:



3 Imagem final

IV.B Questão 2

Foi passada a seguinte imagem para a Questão 2:



4 Imagem de entrada “cookies”

Aplicando a binarização:



5 Imagem binarizada

Após a transformação morfológica Abertura:



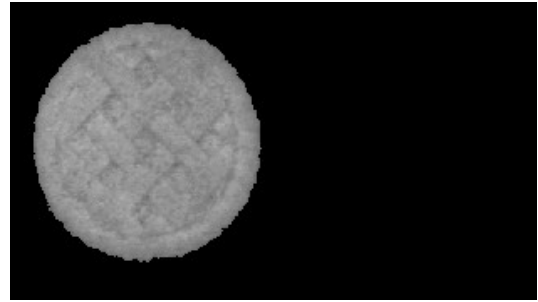
6 Imagem com cookie mordido removido

Dilatando a imagem anterior e utilizando um “and” lógico com a imagem binária original:



7 Imagem da máscara final

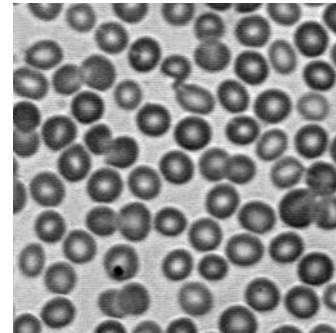
Aplicando a máscara na imagem de entrada:



8 Imagem final

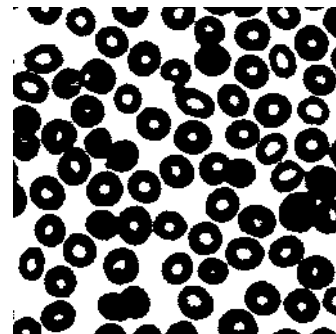
IV.C Questão 3

Foi passada a seguinte imagem para a Questão 2:



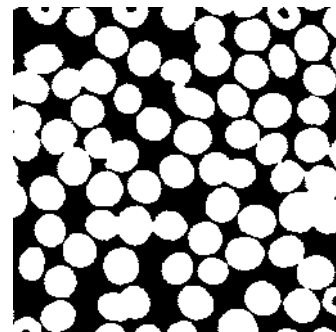
9 Imagem de entrada “img_cells”

Aplicando a binarização:



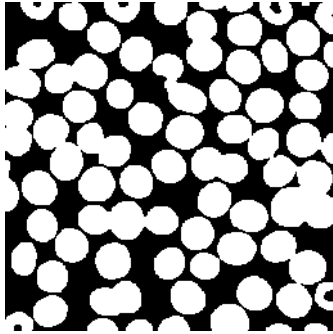
10 Imagem Binarizada

Após a inversão das cores e utilização da função de preenchimento de contornos:



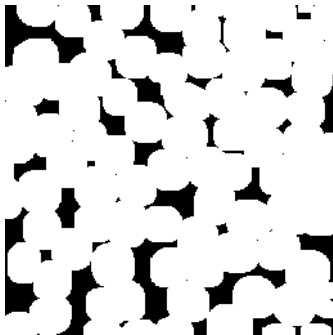
11 Imagem com buracos preenchidos parcialmente

Aplicada a transformação Open:



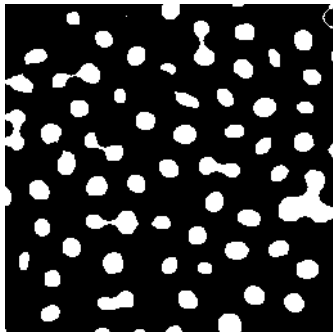
12 Imagem com alguns ruídos removidos

Encontrando a região de fundo:



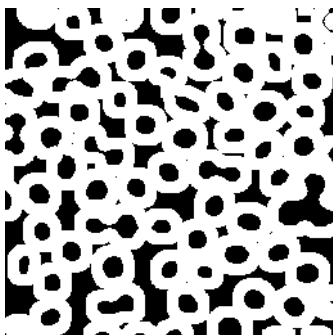
13 Imagem sem ruídos após dilatação

Encontrando a região das células:



14 Imagem sem ruídos utilizando a função de distância

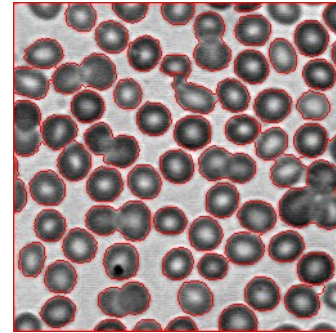
Subtraindo a região de fundo da região das células:



15 Imagem da região de borda

Resultado dos marcadores gerados pela função de segmentação watershed aplicada na imagem anterior:

16 Imagem final com marcadores



V CONCLUSÃO

De forma geral, todos os exercícios apresentaram resultados dentro do esperado e implementaram, mesmo que de forma parcial, o que foi proposto inicialmente em cada enunciado.

Observando o experimento 1, vê-se que o algoritmo morfológico Top-Hat mostrou-se extremamente eficiente, removendo quase completamente o ruído mostrado na binarização direta, além de melhorar ainda mais o contraste dos números da imagem.

Com relação ao experimento 2, fica claro que o algoritmo apresentado mostrou-se adequado. Visualizando o resultado final, o cookie da direita foi totalmente removido, enquanto que o da esquerda manteve-se totalmente, mesmo que com certas distorções nas bordas causadas pela binarização, que utilizando o método adotada era quase inevitável.

Por fim, analisando o experimento 3, é possível perceber que o algoritmo apresentado não funcionou completamente. Em grande parte, o erro deu-se pela função de preenchimento não remover todos os buracos da imagem, resultando em células que não foram reconhecidas. No entanto, para as que foram preenchidas, o resultado chega a ser satisfatório.

Enfim, conclui-se através desses experimentos, portanto, que as transformações morfológicas em uma imagem são de suma importância, visto que além de poderem facilitar certas transformações em imagens utilizando processos de nível médio, podem auxiliar na visualização de uma imagem, removendo elementos indesejados.

REFERÊNCIAS

- 1 Slides do professor Bruno Macchiavello.
- 2 Site de documentação do OpenCV