

Trabalho 2 Introdução ao Processamento de Imagens

Algoritmos morfológicos e segmentação watershed em imagens selecionadas

Guilherme Rodrigues Lodron Pires
Departamento de Ciências da Computação (CiC)
Universidade de Brasília (UnB)
Brasília, Brasil
guilodron7@gmail.com

Abstract—This report has the objective of presenting results from the application of morphological algorithms in order to solve common problems of image processing and their contrast with simpler techniques previously studied. Furthermore, studies the process and results from the watershed segmentation algorithm.

Index Terms—image processing, watershed, morphological, report

I. INTRODUÇÃO

Este relatório tem como objetivo apresentar resultados provenientes da aplicação de algoritmos morfológicos na resolução de problemas comuns em processamento de imagens e seu contraste com técnicas mais básicas anteriormente estudadas. Ademais, estuda o funcionamento e resultados do algoritmo de segmentação watershed.

II. CORREÇÃO DE RUÍDO POR BINARIZAÇÃO DIRETA E POR ALGORITMOS MORFOLÓGICOS

A. Binarização

Para facilitar operações ou retirar informação indesejada de imagens, uma opção simples é a binarização. Nela é estabelecido um valor limite, chamado threshold, a partir do qual para todos os pixels com valor acima desse marca é atribuídos o valor 255 (branco) e aos demais é atribuído o valor 0 (preto). Dessa forma resultando em uma imagem composta apenas por preto e branco.

Inicialmente foi realizada uma binarização na imagem tendo como base o valor retornado pela função 'threshold' da biblioteca OpenCV, em seguida esse valor foi alterado de forma a atender a melhor qualidade subjetiva para realçar a área desejada. Nesse caso o objetivo foi recuperar as letras de uma imagem, que deveriam ficar em preto com um fundo branco para melhor visualização. O valor de threshold escolhido foi o de 170 e o resultado pode ser observado na imagem 1.

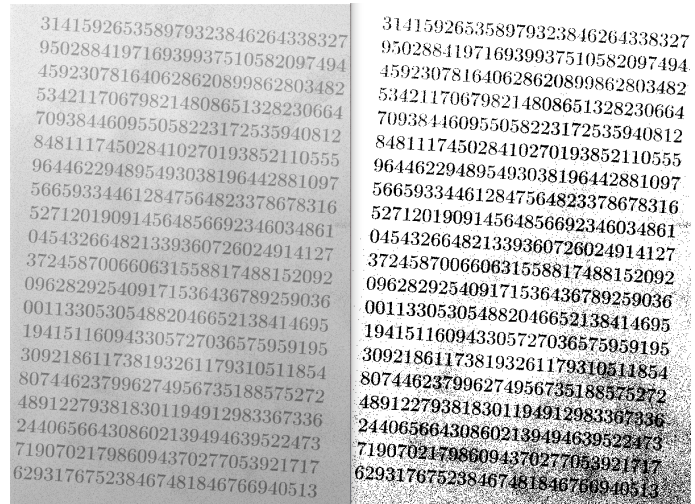


Fig. 1. A esquerda a imagem original e a direita a imagem binarizada com threshold 170

B. Algoritmo morfológico em níveis de cinza

Os algoritmos morfológicos para processamento de imagens utilizam conceitos matemáticos da teoria dos conjuntos para oferecer ferramentas poderosas para problemas de processamento de imagens. Neles são realizadas operações de translação e reflexão sobre um pequeno conjunto ou subimagem, que é denominado elemento estruturante para extrair ou descartar áreas de interesse. Algumas das operações base para esses algoritmos são:

a) *Erosão*: mantém na imagem todos os conjuntos de pixels que têm tamanho suficiente para conter o elemento estruturante utilizado. Com isso o elemento geralmente é afinado ou diminuído. Essa operação é descrita pela seguinte fórmula:

$$A \ominus B = \{z \mid B \subseteq A\} \quad (1)$$

b) *Dilatação*: mantém na imagem todos os deslocamentos de pixels de forma que possuam pelo menos alguma interseção com o elemento estruturante. Ao contrário da erosão

o elemento tende a engrossar ou aumentar de tamanho. A fórmula que descreve essa transformação é:

$$A \oplus B = \{z \mid B \cap A \neq \emptyset\} \quad (2)$$

A partir dessas duas operações básicas é possível derivar a operação de fechamento:

c) *Fechamento*: o fechamento geralmente funde as descontinuidades estreitas e alonga os golfos finos, elimina pequenos buracos e preenche as lacunas em um contorno. Essa operação é feita pela dilatação seguida da erosão.

$$A \bullet B = (A \oplus B) \ominus B \quad (3)$$

As operações citadas acima são feitas em cima de imagens binárias, porém também existem implementações para o seu uso em imagens em escala de cinza.

1) *Algoritmo proposto*: Para resolver o problema de realçar as letras, inicialmente foi utilizada a transformação Bottom-Hat, que pode ser feita em imagens em escala de cinza e tem como objetivo realçar detalhes em lugares claros da imagem. Essa transformação é feita a partir da subtração da operação de fechamento pela imagem original. Os resultados encontrados foram:

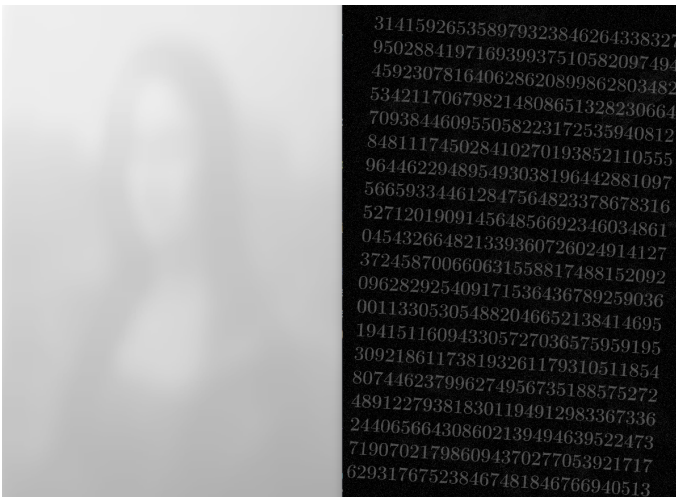


Fig. 2. A esquerda o resultado do fechamento e a direita o resultado do bottom-hat

Para distinguir melhor as letras do fundo foi realizada uma binarização com o threshold de 55 e em seguida a imagem foi invertida para mostrar as letras em preto e o fundo em branco, que podem ser observados na imagem 3.

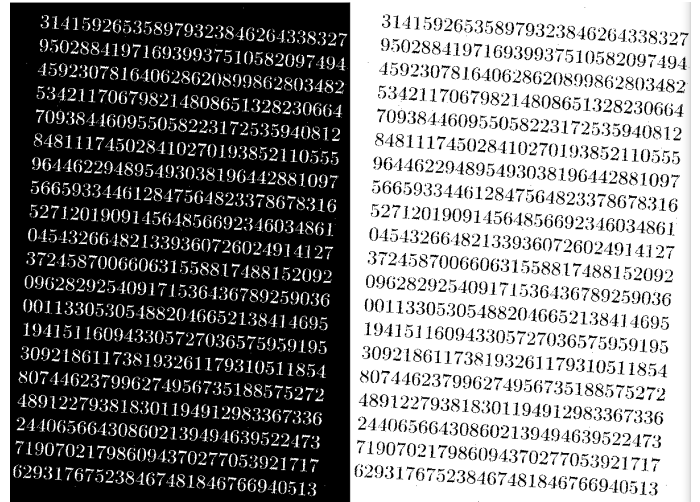


Fig. 3. A esquerda o resultado binarizado e a direita a imagem invertida

Esse processo eliminou o ruído de fundo da imagem, porém algumas letras tiveram a integridade de suas linhas prejudicadas. Para resolver isso foi utilizado uma erosão, a fim de engrossar as partes pretas da imagem, gerando um resultado mais legível. Os resultados obtidos podem ser observados na imagem e contrastados com os resultados obtidos na binarização direta inicial abaixo.

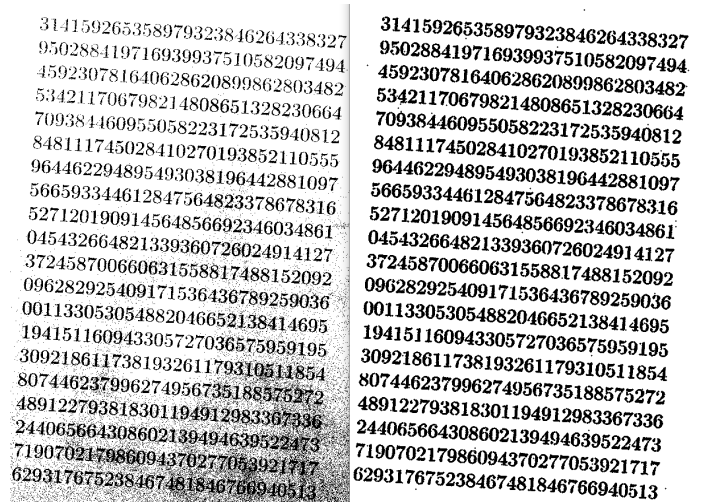


Fig. 4. A esquerda o resultado da binarização direta e a direita o resultado do algoritmo proposto

III. REMOVENDO ÁREAS INDESEJADAS E RECUPERANDO FORMATOS COM EROÇÃO E DILATAÇÃO

Como apresentado na seção anterior, os algoritmos de erosão e dilatação podem servir para diminuir ou aumentar áreas de interesse em uma imagem. Tendo isso em mente, para trabalhar com os algoritmos morfológicos, a imagem de interesse foi binarizada utilizando o threshold automaticamente detectado pela biblioteca OpenCV e os resultados podem ser vistos na imagem 5:

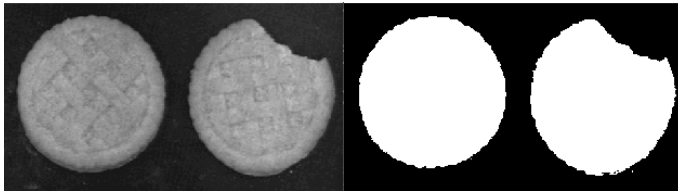


Fig. 5. A esquerda a imagem original e a direita a imagem binarizada

Em seguida, para retirar a forma do biscoito mordido foram realizadas duas iterações de erosão com um elemento estruturante em formato de elipse e de tamanho 65 x 65. Por possuir um recorte no canto superior, o formato do biscoito mordido foi retirado da imagem e o centro do biscoito inteiro foi mantido (primeira imagem da figura 6).

Para recuperar o formato original da figura resultante, foram realizadas duas iterações de dilatação, utilizando o mesmo elemento estruturante e a forma obtida possuiu tamanho e formato semelhante ao desejado. Por fim a área reconstruída foi utilizada como máscara para a imagem original, extraíndo apenas a informação desejada do biscoito inteiro. Os resultados dos passos descritos são representados pela segunda e terceira imagens da figura abaixo.

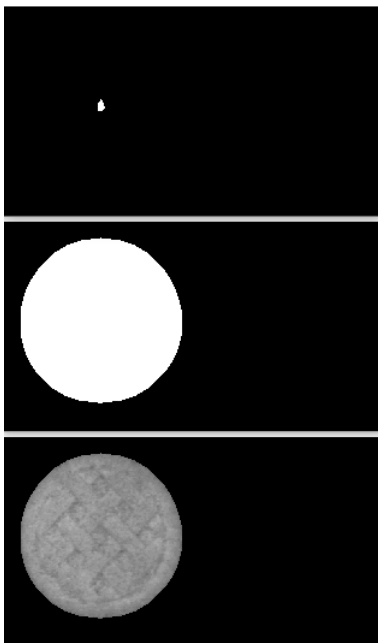


Fig. 6. Em cima o resultado das erosões, ao meio o formato recuperado por dilatação e abaixo a imagem original com a máscara aplicada

IV. DETECÇÃO DE CÉLULAS COM SEGMENTAÇÃO WATERSHED

A segmentação watershed faz uma interpretação topográfica da imagem, a fim de imaginá-la como um conjunto de bacias hidrográficas, que representam as regiões que devem ser marcadas. As áreas onde essas bacias transbordam são entendidos como os limites ou interseção de áreas de interesse. Para exercitar o algoritmo, a imagem proposta foi inicialmente binarizada utilizando o threshold escolhido pela biblioteca OpenCV

gerando uma série de círculos pretos com falhas brancas no centro.

Para preencher essas falhas foi implementado um algoritmo similar ao 'bwareopen' provido pelo matlab, mas não disponível na biblioteca OpenCV. Esse algoritmo retira da imagem todos os componentes conectados que possuem tamanho menor do que um limite mínimo fornecido. Os resultados da binarização e do preenchimento podem ser observados abaixo.

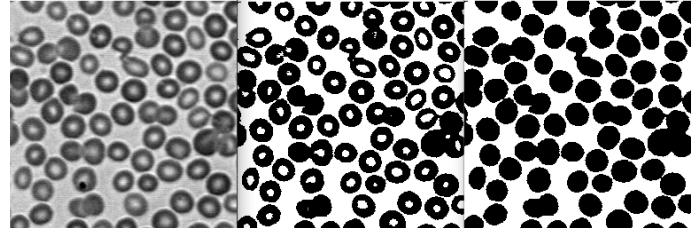


Fig. 7. A esquerda a imagem original, ao centro a imagem binarizada e a direita a imagem com as falhas preenchidas

Em seguida a imagem foi invertida para que os formatos das células ficassem em branco e o fundo em preto, assim permitindo as manipulações posteriores. Também foi utilizada uma operação de abertura para a eliminação de um possível ruído de fundo e uma dilatação para garantir que a região que será entendida como fundo não tem nenhum tipo de ruído. Os passos podem ser visualizados na imagem abaixo.

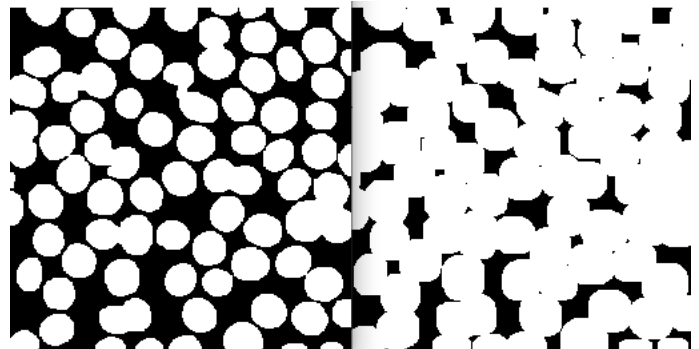


Fig. 8. A esquerda a operação de abertura na imagem invertida e a direita a dilatação para garantir o fundo desejado

Para delimitar áreas de confiança que fazem parte dos segmentos primeiro foi calculada a distância euclidiana de cada pixel em relação ao pixel preto mais próximo utilizando a função 'distanceTransform' do OpenCV e em seguida foi aplicado um threshold com base nos valores encontrados, resultando assim em áreas que certamente deveria estar em primeiro plano.

A partir das áreas encontradas foi realizada a subtração da área de fundo e de primeiro plano para obter áreas que são desconhecidas. Os passos descritos são representados na imagem a seguir.

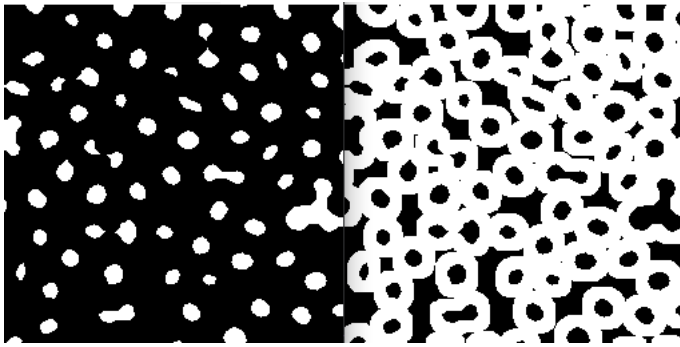


Fig. 9. A esquerda as áreas de primeiro plano e a direita as áreas desconhecidas

Por fim as áreas de primeiro plano foram utilizadas para gerar marcadores e às áreas desconhecidas foi atribuído o valor 0. Com isso foi possível utilizar a função 'watershed' da biblioteca OpenCV passando como argumentos a imagem original e os marcadores para obter a segmentação watershed. Com isso os valores retornados foram coloridos em verde para realçar o resultado obtido na imagem final. O resultado obtido pela função watershed e a imagem final obtida podem ser observados a seguir.

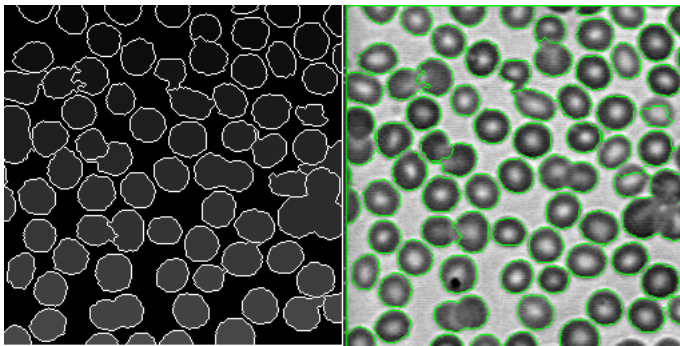


Fig. 10. A esquerda o resultado obtido pela função watershed e a direita a imagem final com a segmentação colorida em verde

V. CONCLUSÃO

No trabalho apresentado foi possível entender e implementar alguns algoritmos importantes de processamento de imagens. As operações morfológicas podem ser uma boa alternativa para resolver problemas inicialmente não triviais, porém devido a sua dependência de parâmetros que se adequem a imagem de interesse, eles podem não ser a melhor opção para tratar um grupo de imagens com aspectos e formatos diferentes. Fato que se tornou evidente nos testes de parâmetros durante a realização das questões.

A segmentação watershed também se provou muito interessante para detectar elementos agrupados em uma imagem de forma eficiente e, a forma como o algoritmo é pensado como uma analogia à bacias hidrográficas foi um fator motivador de estudo.

Os resultados da resolução das questões foram bastante satisfatórios, uma vez que em todos os casos o objetivo

proposto foi alcançado de maneira rápida, direta e que também refletiu em melhoras subjetivas na visualização da informação a ser extraída das imagens.

REFERENCES

- [1] R.C. Gonzalez, R.E. Woods, "Digital Image Processing" 4th edition
- [2] Tutorial Segmentação Watershed, disponível em <https://datasmarts.net/es/como-segmentar-imagenes-con-watershed-en-opencv/>. Acesso em 17/10/2021
- [3] Documentação OpenCV, disponível em <https://opencv.org/>. Acesso em 18/10/2021