

CIC0176 IPI - Trabalho 2

Rodrigo Rios

Depto. de Ciência da Computação

Universidade de Brasília

Brasília, Brasil

rodrigorr98@gmail.com

Resumo—Este trabalho consiste na implementação de métodos básicos de processamento de imagens de forma a resolver três casos práticos. No primeiro deles, deseja-se reduzir variações de fundo que são acentuadas quando a imagem é binarizada. Para tanto, utiliza-se um algoritmo morfológico em níveis de cinza. Na segunda etapa, deseja-se remover um elemento de uma imagem, preservando os demais, e então mostrá-la em níveis de cinza. A implementação se deu por meio de outro algoritmo morfológico, dessa vez binário. Por fim, na última etapa, deseja-se implementar um tipo de segmentação capaz de separar células do fundo, tratando, inclusive, aquelas que estejam dispostas de forma contígua. Para tanto, alguns pré-processamentos foram realizados, como preenchimento de buracos e, em seguida, aplicou-se a transformada *watershed*.

I. INTRODUÇÃO

Uma importante ferramenta de processamento de imagens é a morfologia matemática. Em que pese a não versatilidade desse tipo de algoritmo, uma vez que a implementação é extremamente dependente da imagem - ou do conjunto de imagens - de entrada, para definir os elementos estruturantes, por exemplo, esta é uma abordagem capaz de prover bons resultados a partir de códigos bastante simples. Há quatro tipos de operações fundamentais dentro dessa classe: erosão, dilatação, abertura e fechamento, cada qual com a sua notação e caracterização algébrica. A Equação 1 mostra a definição da erosão:

$$A \ominus B = \{z | (B_z) \cap A^c = \emptyset\} \quad (1)$$

A Equação 2 mostra a definição da dilatação:

$$A \oplus B = \{z | (B_z) \cap A^c \neq \emptyset\} \quad (2)$$

A Equação 3 mostra a definição da abertura:

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

A Equação 4 mostra a definição do fechamento:

$$A \bullet B = (A \oplus B) \ominus B \quad (4)$$

Em todos os casos, A representa a imagem de entrada, enquanto que B representa o elemento estruturante. Observa-se das definições que tanto a abertura quanto o fechamento derivam das duas outras operações. Além disso, nota-se que há sempre dependência da imagem de saída em relação ao elemento estruturado, como mencionado. Um caso típico de aplicação da morfologia matemática é a remoção de elementos

indesejados de uma dada imagem de entrada. Por exemplo, a abertura morfológica pode ser utilizada para remover elementos cuja espessura é inferior a um determinado limiar, conforme mostra a Figura 1. Este resultado é possível pois a abertura remove todas as translações de B que não se encaixam em A . Essa operação será utilizada em um contexto semelhante (ver Seção 3 B), conjuntamente com a operação de dilatação.

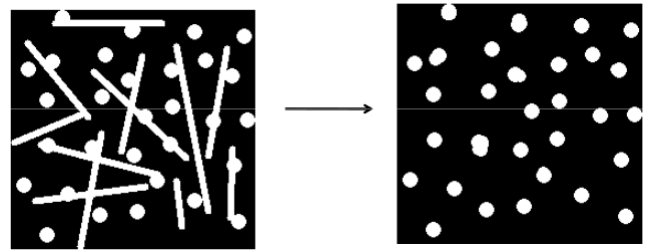


Figura 1. Exemplo de Aplicação da Abertura Morfológica

Uma outra aplicação útil da morfologia matemática é a exclusão de ruídos. Isso pode ser efetivado por meio da aplicação de um gradiente morfológico, tipo de processamento em níveis de cinza capaz de reduzir variações de fundo e tornar, por conseguinte, detalhes mais distinguíveis. A comparação com o gradiente espacial é imediata, uma vez que ambos cumprem objetivos semelhantes. Na Seção 3 A, será mostrado o resultado da aplicação desse processamento morfológico, empregado para corrigir variações no fundo que não puderam ser tratadas com um mera binarização.

Para além da morfologia matemática, um dos casos investigados neste trabalho envolve a segmentação de imagens. Conforme o próprio nome revela, a segmentação envolve dividir uma imagem em regiões ou objetos de interesse. Nesse sentido, ela é relevante para isolar de uma dada imagem de entrada somente os dados que interessem para a aplicação em questão. Um exemplo disso é a extração dos núcleos em uma determinada imagem histológica. Uma importante implementação de segmentação é o *watershed*, que consiste em segmentar por meio do crescimento de regiões. Desse modo, o algoritmo é capaz de separar regiões contíguas de objetos sobrepostos a partir dos mínimos locais da imagem. Os resultados da utilização do *watershed* em um caso prático de segmentação são abordados na Seção 3 C.

II. METODOLOGIA

Na seção 3 A, utilizou-se a função *graythresh* para a binarização e as funções *imdilate* e *imerode*, todas da *toolbox* do *MATLAB*, para a remoção das variações de fundo da imagem entrada. As duas últimas foram utilizadas na implementação, em níveis de cinza, do gradiente morfológico. Por meio de inspeção, foi possível estabelecer os valores dos raios dos discos utilizados como elemento estruturante para a erosão e para a dilatação.

Na Seção 3 B, a imagem de saída é obtida após as operações de abertura, dilatação e aplicação de máscara binária; neste último caso, isto foi feito para que se pudesse obter a saída em níveis de cinza. A abertura morfológica foi realizada pela função *imopen*, enquanto que a dilatação pela *imdilate*. Para a definição dos raios dos discos utilizados como elementos estruturantes, fez-se uma inspeção numérica. Ressalta-se que os resultados obtidos teriam sido significativamente diferentes caso os valores estabelecidos fossem outros: por exemplo, caso o raio da abertura morfológica fosse suficientemente grande, ambos os biscoitos teriam sido apagados.

Na seção 3 C, o tratamento dos espaços desconectados foi feito pela função *bwareaopen*. O preenchimento dos buracos foi feito utilizando-se a função *imfill*. Utilizou-se a função *bwdist* para o cálculo da função de distância. Por fim, para a implementação do *watershed* foi empregada a função homônima. A transformação da imagem de saída para RGB foi empregada por meio da função *label2rgb*. Nesta última função, utilizou-se o mapa de cores "jet", e uma coloração neutra para o preenchimento do fundo (*fillcolor* = 0.7 para os três canais), de modo a não comprometer a visualização da segmentação proposta.

III. RESULTADOS

Esta seção se dedica a mostrar os resultados e discussões correlatas a respeito das operações feitas ao conjunto de imagens

A. Questão 1

Neste tópico, são mostrados os resultados da solução proposta de redução de ruído da imagem original, Figura 2, à esquerda, por meio da utilização de um algoritmo morfológico em níveis de cinza. Além disso, propõe-se uma melhoria ao primeiro resultado obtido, a qual também é mostrada nesta seção.

Utilizando-se uma função de binarização automática (*graythresh*), obteve-se a imagem à direita da Figura 2, que é comparada com a original. Observa-se que esta implementação não foi capaz de eliminar os ruídos presentes na imagem original após a binarização. De fato, o que se percebe é que a operação realizada prejudica o discernimento entre as letras, em especial no canto inferior direito da imagem, onde o ruído se concentra.

Desta feita, aplicou-se o gradiente morfológico em níveis de cinza à imagem original, de modo a tentar reduzir os ruídos da imagem e torná-la mais inteligível após a binarização. Nesse sentido, tendo em conta que o gradiente morfológico consiste

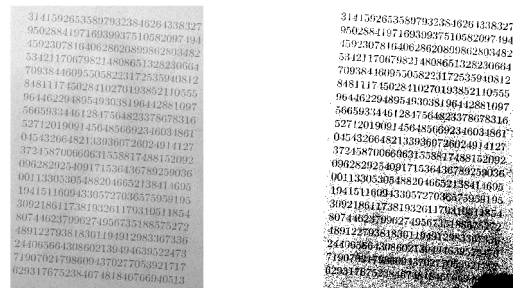


Figura 2. Imagem Original e Imagem Binarizada

em erosões e dilatações, um ponto importante é a definição de qual o raio que será utilizado para realizar cada uma dessas operações. Inicialmente, utilizou-se um raio igual tanto para a erosão quanto para a dilatação. Em seguida, a imagem em que se aplicou o gradiente morfológico foi binarizada e, após isso, invertida. A Figura 3 mostra a imagem obtida pela aplicação do gradiente morfológico, à esquerda, e o resultado da inversão dela, à direita. Nota-se que a operação foi bem sucedida em remover os ruídos que comprometiam a binarização. Com isso, a imagem binária resultante passou a ser mais inteligível do que a obtida pelo primeiro método.

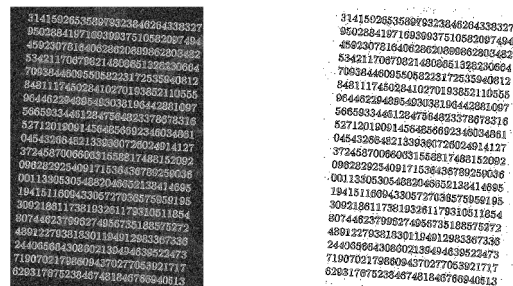


Figura 3. Aplicação do Gradiente Morfológico e Inversão (direita)

Apesar de o resultado ter sido bem sucedido, pode-se investigar para quais valores de raio na erosão e dilatação realizados obtém-se um resultado subjetivamente superior. Em outras palavras, pode-se utilizar valores de raio diferentes de um - e possivelmente diferentes entre si - para cada uma das operações, de modo a se ter letras visualmente mais identificáveis na imagem de saída. Mantendo-se o raio da dilatação em um e aumentando-se gradualmente o valor do raio da erosão, observa-se que as imagens geradas resultam em letras em que somente o contorno é preto, isto é, sem preenchimento. Como isso dificulta a distinção entre as letras, esse método é descartado. Por outro lado, fixando-se o raio da erosão em um e aumentando-se o da dilatação, nota-se que as letras ficam completamente preenchidas, mas o aumento do raio passa a não mais afetar a imagem resultante após certo limite. Assim sendo, o resultado gerado permite

distinguir melhor as letras, tendo em vista que não há mais espaços em branco no interior delas, como havia na Figura 3. Ao se utilizar valores maiores que um tanto para a erosão quanto para a dilatação, os contornos passaram a ficar muito pronunciados, o que também prejudica a identificação dos caracteres, em especial para valores de raio maiores que dois. Portanto, conseguiu-se um resultado superior para o gradiente morfológico utilizando-se raio unitário para a erosão e raio iguala dois para a dilatação. A imagem resultante é mostrada na Figura 4.

314159265358979323846264338327
950288419716939937510582097494
459230781640628620899862803482
534211706798214908651328230664
709384460955058223172535940812
848111745028410270193852110555
964462294895493038190442881097
566593344612847564823378678316
527120190914564856692346034861
045432664821339360726024914127
372458700660631558817488152092
096282925409171536436789259036
001133053054882046652138414695
194151160943305727036575959195
309218611738193261179310511854
807446237996274956735188575272
489122793818301194912983367336
244065664308602139494639522473
719070217986094370277053921717
629317675238467481846766940513

Figura 4. Melhoria do Gradiente Morfológico

B. Questão 2

Nesta seção, são analisados os resultados de algoritmos de morfologia matemática binária propostos para mostrar de forma correta, em níveis de cinza, somente um dos elementos de uma dada imagem de entrada. No caso em questão, a imagem apresenta dois biscoitos, um inteiro e outro mordido, e deseja-se gerar uma imagem de saída em níveis de cinza que contenha somente o biscoito completo. A Figura 5 mostra a imagem original juntamente com a sua binarização, esta obtida por meio da função *graythresh*.

Uma forma de se obter uma imagem que contenha um pedaço do biscoito não mordido, mas excluindo o biscoito mordido da imagem original é utilizar a operação de abertura morfológica. Nesse sentido, pode-se definir três cenários de imagens de saída, a depender do raio do disco utilizado como elemento estruturante. No primeiro caso, o raio não é suficientemente grande e os dois objetos permanecem na imagem. No segundo, o raio é tal que apenas o biscoito maior, não mordido, permanece. No último cenário, o raio é suficientemente grande para apagar ambos os biscoitos da imagem. Desse modo, tendo em conta que o cenário que se

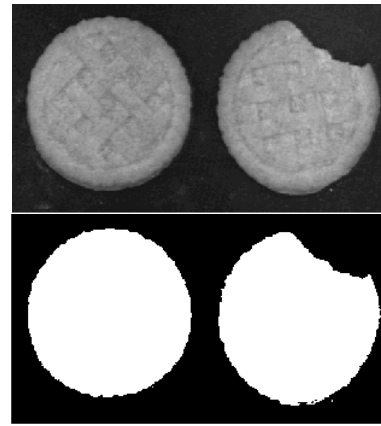


Figura 5. Imagem Original e Binarizada (Abaixo)

deseja é o segundo, é possível se obter o raio do elemento estruturante por inspeção. Assim sendo, se observou que um raio igual a 56 cumpre o requisito estabelecido. A forma original do biscoito, alterada após a primeira operação, pode ser recuperada fazendo uma dilatação em cascata com a abertura feita anteriormente. Com isso, deseja-se obter um objeto maior que o biscoito original. Por inspeção, observa-se que um raio igual a 9 para o disco utilizado como elemento estruturante dessa operação é capaz de gerar esse resultado. Com a imagem resultante da dilatação em seguida da abertura, pode-se fazer a multiplicação elemento a elemento desta pela imagem binarizada (Figura 5), o que equivale a operação lógica *and*, de modo a se recuperar o formato do biscoito. A Figura 6 mostra, de forma compacta, o resultado da dilatação em seguida da abertura, e o resultado do produto desta pela imagem binarizada.

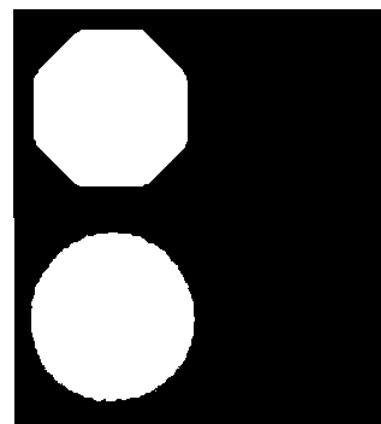


Figura 6. Abertura Seguida de Dilatação e o Produto desta pela Imagem Binarizada (Abaixo)

Com isso, obtém-se uma imagem binária com somente o biscoito não mordido. Portanto, para obter a imagem que desejamos, basta utilizar a última imagem obtida como máscara para a imagem original. Dessa forma, obtemos uma imagem em níveis de cinza contendo somente o biscoito completo. A

Figura 7 mostra a imagem resultante, que foi obtida de forma bem sucedida.

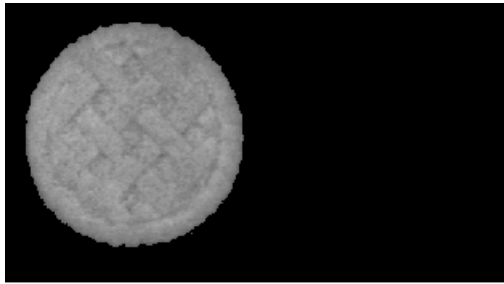


Figura 7. Imagem em Níveis de Cinza com Somente o Biscoito Completo

C. Questão 3

Neste tópico, são analisados os passos intermediários e o resultado final da implementação de uma segmentação *watershed*. Antes da segmentação em si, três passos de pré-processamento são essenciais: a binarização, o tratamento de espaços desconectados e o preenchimento de buracos. Vale destacar que os dois últimos passos não são descartáveis: se não fossem feitos, a segmentação teria identificado espaços vazios como células, ou ainda teria segmentado somente os espaços interiores. A Figura 8 mostra a imagem original e a imagem obtida por essas três operações em sequência. Cabe destacar que a utilizou-se o negativo da imagem para o preenchimento de buracos, de modo que os objetos de interesse fiquem em branco, enquanto que o fundo fique preto, conforme mostra a figura supracitada.

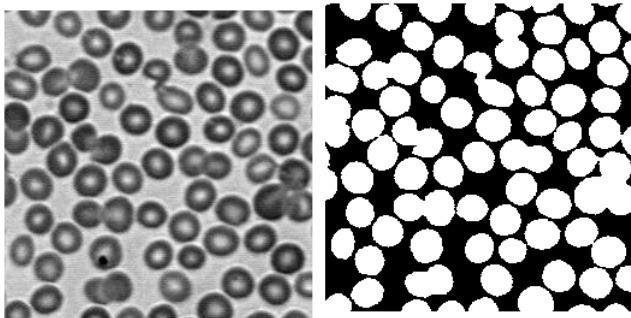


Figura 8. Imagem Original e Imagem Pré-Processada (Direita)

A partir da imagem pré-processada, calculou-se a função de distâncias do complemento dela. O complemento da função de distâncias, por sua vez, será entrada o argumento da transformada *watershed*, de modo que os *pixels* que representem o crescimento sejam pretos e o fundo seja branco. Antes da aplicação da transformada *watershed*, definimos os *pixels* fora do raio de interesse como 0, utilizando a imagem binária como máscara. Este último passo é fundamental para que se evite a hiper-segmentação ao aplicar a transformada *watershed*, o que comprometeria o resultado final. A Figura 9 condensa os resultados: à esquerda, temos o complemento da função de

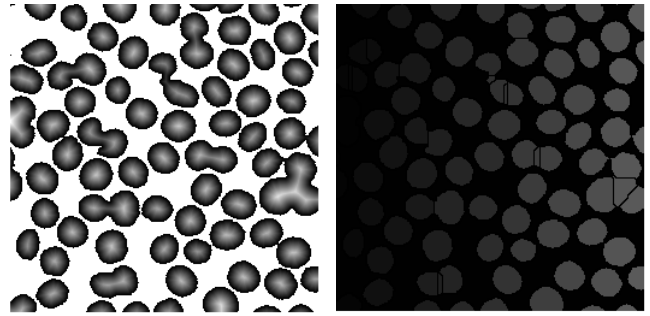


Figura 9. Complemento da Função de Distâncias e Transformada *Watershed* (Direita)

distâncias, e à direita, o resultado da aplicação da transformada *watershed* e da definição do valor dos *pixels* fora do ROI.

A partir da última imagem, conseguimos gerar a saída desejada: uma imagem segmentada e representada em 3 canais de cores (RGB). Isto realizado pela função *label2rgb*. Além disso, escolhem-se um mapa de cores e uma cor de preenchimento do fundo adequados para mostrá-la. A Figura 10 mostra o resultado final obtido. Observa-se que a segmentação proposta foi bem sucedida em alguns casos. No entanto, é possível identificar células que foram repartidas, como se fossem diferentes, mas que, aparentemente, são uma só. Há ainda um aglomerado do que parecem ser três células, mas que foi segmentado como se fossem quatro.

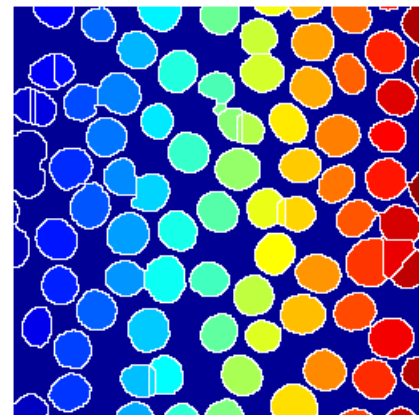


Figura 10. Resultado Final da Transformada *Watershed* em RGB

IV. CONCLUSÕES

Na primeira etapa, foi proposto um processamento morfológico em níveis de cinza tendo em vista a redução das variações do fundo da imagem de entrada, que ficavam bastante pronunciadas quando foi feita a binarização por *graythresh*. O resultado obtido pelo gradiente morfológico foi capaz de gerar uma imagem de saída sem essas interferências, tornando as letras inteligíveis. Além disso, foi possível encontrar, por inspeção, valores para os raios dos elementos estruturantes que melhoraram subjetivamente a saída.

Na segunda etapa, foram utilizadas operações morfológicas binárias para conseguir remover da imagem original, entre os dois biscoitos que havia, somente o mordido. Em última análise, a tarefa consistiu em definir de forma correta os elementos estruturantes utilizados para a abertura e a dilatação morfológicas. Tendo feito isso, recuperou-se, de forma bem sucedida, a imagem original com somente o biscoito inteiro, por meio do emprego de uma máscara binária.

Na terceira etapa, objetivava-se a implementação de um algoritmo de segmentação *watershed*, que fosse capaz de isolar de forma correta as células contidas na imagem de entrada. Para tanto, alguns pré-processamentos foram necessários, quais sejam: binarização, conexão de espaços e preenchimento de buracos. A partir dessas operações, utilizou-se a função de distâncias seguida da função *watershed*. Para que o procedimento ficasse completo, transformou-se a imagem para RGB. Desse modo, conseguiu-se uma implementação correta mesmo para os elementos que se sobrepuseram, salvo em algumas exceções.