

Introdução a Processamento de Imagens

Trabalho 2

João Viktor de Carvalho Mota
Universidade de Brasília
Email: joaoviktor22@hotmail.com

Abstract—Este trabalho é o uso e aprendizado da morfologia matemática para processamento de imagens binárias e em níveis de cinza.

1. Introdução

A morfologia matemática é , baseada na geometria, um estudo usando operadores não-lineares para extrair componentes de uma imagem que sejam úteis para alguma representação. Ela serve para percorrer alguma imagem com um elemento estruturante e marcar e extrair informações de imagens. No estudo da morfologia matemática é usado a teoria dos conjuntos que representam as imagens e seus pixel. Há a imagem binária em que todos os elementos do conjunto são vetores bidimensionais dos pixel pretos da imagem. Há também a imagem em níveis de cinza em que os elementos do conjunto são vetores que além da posição há o nível de cinza do pixel da imagem.

Algumas das operações morfológicas fundamentais são:

-Erosão: São todos os pontos em que B transladado em x cabe em A sendo A uma imagem e B um elemento estrutural. Equação descrita abaixo de Erosão:

$$C = A \ominus B \quad (1)$$

-Dilatação: Erosão e dilatação são operações duais em relação a complementação e reflexão de conjuntos. A dilatação é a união de translações da imagem A por todos os pontos do elemento estrutural B. A equação descrita abaixo de Dilatação:

$$C = A \oplus B \quad (2)$$

Observação: A dilatação expande enquanto que a Erosão reduz uma imagem.

-Abertura: A abertura de uma imagem A por um elemento estruturante B é definida por uma erosão seguida por uma dilatação. A dilatação serve para suavizar o contorno da imagem e eliminar protusões finas. A equação descrita abaixo de Abertura:

$$C = A \circ B \quad (3)$$

Ou também:

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

-Fechamento: O fechamento e a Abertura são operações duais. O fechamento é a dilatação de A por B seguida da erosão por B. O fechamento serve para suavizar o contorno da imagem, eliminar buracos, preencher fendas e fundir as quebras. A equação descrita abaixo de Fechamento:

$$C = A \bullet B \quad (5)$$

Ou também:

$$A \bullet B = (A \oplus B) \ominus B \quad (6)$$

-Transformada Hit-or-Miss: É uma ferramenta básica para detecção de formas em uma imagem. A equação descrita abaixo da transformada:

$$C = A \circledast X \quad (7)$$

Ou também, considerando a dualidade de erosão e dilatação. B₂ é (W-X) sendo X o conjunto de elementos em B associada a imagem e B₂ o conjunto de elementos em B associado ao fundo correspondente.

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2) \quad (8)$$

Há também algumas técnicas usando as operações morfológicas acima como base como: -Extração de fronteiras:

$$\beta(A) = (A \oplus B) - A \quad (9)$$

-Preenchimento de buracos:

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots \quad (10)$$

-Extração de componentes conexos:

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \dots \quad (11)$$

-Afinamento:

$$A \otimes B = A - (A \circledast B) \quad (12)$$

-Espessamento:

$$A \sqcup B = A \cup (A \circledast B) \quad (13)$$

-Esqueleto:

$$A = \bigcup_{k=0}^K [S_k(A) \oplus kB] \quad (14)$$

-Poda: Primeiro faz um afinamento e então pega as extremidades e faz a dilatação, a união entre o afinamento e a dilatação é a poda.

2. Metodologia

Foi usado MatLab para fazer o programa. Foi usada algumas funções do matlab :

- imread() para ler as imagens
- graythresh() para saber um bom nível de threshold para binarizar
- imbinarize() para binarizar imagens
- imshow() para mostrar a imagem
- strel() para criar o elemento estrutural B
- imopen() para abertura
- imclose() para fechamento
- imcomplement() para fazer complemento da imagem
- bwhitmiss() para realizar a transformada hit-or-miss
- imtophat() para filtrar a imagem com top-hat ou imbothat para filtrar com bot-hat
- imerode() para erosão
- imdilate() para dilatação
- rgb2gray() para transformar para bidimensional a imagem
- bwareaopen() para tirar buracos
- bwdist() para calcular a distância
- watershed() para fazer a segmentação watershed

3. Resultados

3.1. Questão 1

A imagem abaixo é a da imagem binarizada diretamente.

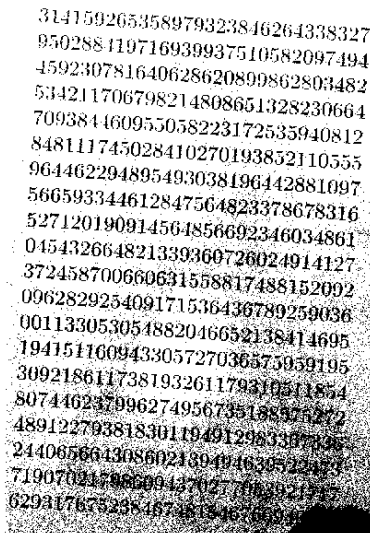


Figure 1. Imagem Após Binarização

Podemos perceber que no canto inferior direito ficou pouco legível com muito pixel preto enquanto que na parte superior esquerda ficou muito claro ficando pouco visível.

A imagem abaixo é a imagem teste após algumas operações morfológicas como imbothar() para filtrar e imerode() para fazer a erosão.

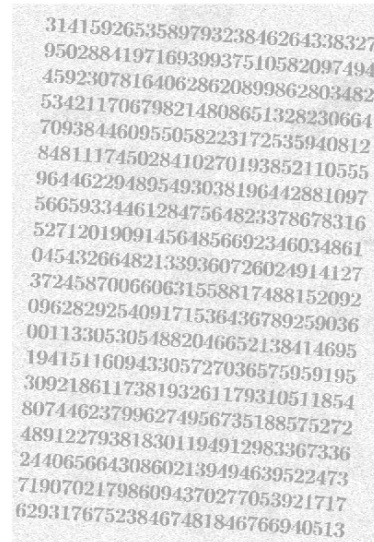


Figure 2. Imagem Após Algoritmo Morfológico

A imagem abaixo é a imagem binarizada após usar o algoritmo morfológico descrito na imagem acima.

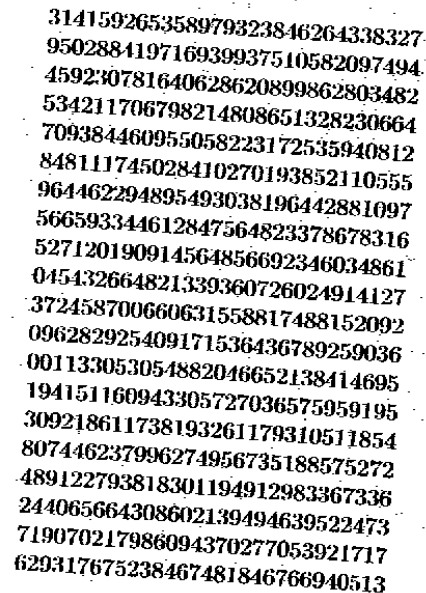


Figure 3. Imagem Final - Binarização Após Algoritmo Morfológico

Podemos perceber que a filtragem conseguiu retirar um pouco do fundo e com a erosão se retirou quase que todo o fundo e o ruído incluso na imagem original. Porém com apenas as operações morfológicas a imagem ficou meio apagada porém com a binarização a imagem ficou muito bem legível.

3.2. Questão 2

Na imagem abaixo foi utilizado erosão para retirar todo o 'cookie mordido' porém praticamente quase todo o 'cookie completo' também foi retirado.

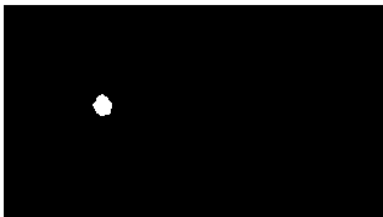


Figure 4. Imagem após retirar 'cookie mordido'

Na imagem abaixo foi utilizado a dilatação para voltar ao tamanho original que foi reduzido pela erosão e por isso foi utilizado o mesmo elemento estrutural em forma de disco('disk').



Figure 5. Imagem de volta a forma original

Podemos perceber que por causa da geometria o 'cookie completo' perdeu suas bordas e ficou com um formato de um octógono por causa da dilatação e da erosão. Colocando a máscara na imagem original para mostrar apenas o 'cookie completo' é a imagem abaixo porém podemos ver que por causa da erosão na primeira parte da questão o biscoito perdeu um pouco da borda.

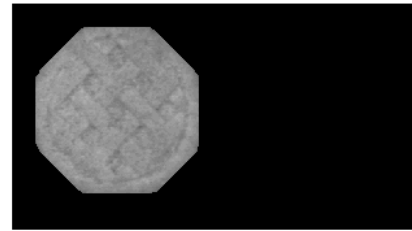


Figure 6. Imagem Final - Mascara Apenas com 'cookie completo'

3.3. Questão 3

A imagem abaixo é a imagem das células binarizadas porém é possível ver alguns buracos dentro das células.

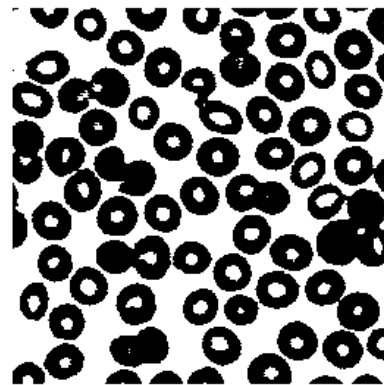


Figure 7. Imagem Após Binarização

Após usar `bwareaopen()` retiramos todos os buracos e acabamos unindo um pouco algumas células por possuírem 'buracos pequenos' entre as células que não queremos para fazer a segmentação.

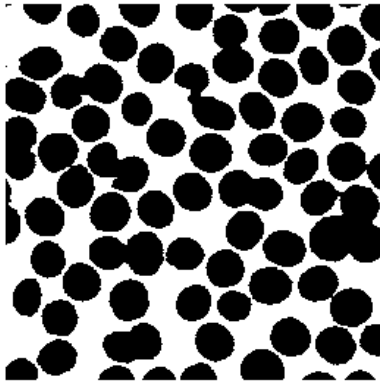


Figure 8. Imagem Após preencher buracos

Foi calculado a função de distância da imagem acima complementada e também utilizada a segmentação watershed para separar em cores aleatórias as áreas de cada célula.

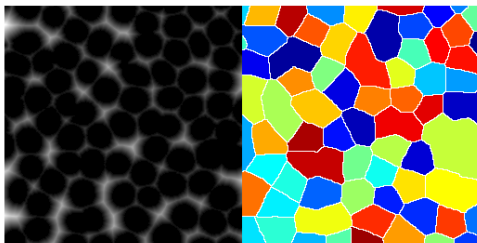


Figure 9. Esquerda - bwdist Direita - Segmentação Watershed

Podemos ver pela segmentação que a grande maioria das células foram segmentadas em áreas diferentes porém teve algumas (no meio da imagem no lado direito) que por estarem muito unidas foi considerada uma célula única mesmo não sendo.

4. Conclusion

Podemos perceber que escolhendo as morfologias e filtragens certas conseguimos retirar características da imagem quando necessário, como foi feita na Questão 1 que retiramos a imagem de fundo. Logo, para alguma situação específica de máscara ou retirar alguma característica específica da imagem as morfologias podem ajudar. Pela Questão 2 percebemos que ao processar a imagem com certas morfologias podemos perder algumas características originais que não seria bom perder, como foi a borda do 'cookie completo' porém com morfologias foi possível retirar um pedaço da imagem ('cookie mordido') apenas por

ser 'menor' (mordido). Na questão 3 podemos perceber que as morfologias podem ajudar em processar a imagem para fazer o necessário que neste caso foi retirar os buracos e também segmentar a imagem mesmo que a segmentação não seja perfeita por causa de certos problemas da própria imagem. Concluímos que a morfologia matemática para processamento de imagens é ótima para retirar características desejadas da imagem porém não é perfeita por causa de certas condições das próprias imagens, podendo ser possível apenas com outras ferramentas como filtragem.

References

- [1] <https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html>
- [2] <https://www.mathworks.com/help/images/marker-controlled-watershed-segmentation.html>
- [3] <http://www.ime.unicamp.br/~valle/PDFfiles/valente10.pdf>
- [4] <https://www.cin.ufpe.br/~tir/ComputacaoCientifica/7.MorfologiaMatematica.pdf>
- [5] https://scikit-image.org/docs/dev/auto_examples/segmentation/plot_watershed.html