

# Trabalho 2

## Introdução ao Processamento de Imagens

Lucas Santos Lessa  
Departamento de Engenharia Elétrica - ENE.  
Brasília, Brasil  
lucasslessaeng@gmail.com

**Resumo**—Na primeira questão, foi desenvolvido um algoritmo para analisar uma imagem de circuito impresso e identificar a quantidade e diâmetros dos espaços destinados aos componentes eletrônicos. Esse processo foi realizado por meio da aplicação de técnicas de morfologia matemática e outras técnicas de processamento de imagens.

Na segunda questão, foram aplicadas técnicas de binarização e morfologia matemática com o objetivo de converter uma imagem de texto em escala de cinza em uma imagem binária de alta qualidade, apresentando fundo branco e caracteres em preto. Para alcançar esse resultado, foram utilizados filtros para otimizar o processo de binarização da imagem, garantindo uma representação mais precisa dos caracteres.

Na última questão, foi desenvolvido um programa que realiza a segmentação de uma imagem utilizando o algoritmo Watershed. Esse programa emprega técnicas de binarização, como o método de Otsu, morfologia matemática e outras técnicas de extração de regiões de imagens. A imagem utilizada nesse contexto foi a de células em escala de cinza, que foi segmentada com sucesso.

**Index Terms**—erosão, dilatação, abertura, fechamento e watershed.

### I. INTRODUÇÃO

A área de processamento de imagens tem como objetivo desenvolver técnicas e algoritmos para manipular e analisar imagens digitais. Essas técnicas têm uma ampla gama de aplicações em diversas áreas, como medicina, indústria, segurança e muitas outras. Neste relatório, abordaremos o processamento de imagens aplicado à segmentação, que consiste em separar objetos ou regiões de interesse em uma imagem com base em suas características visuais, além da segmentação será abordado técnicas de morfologia matemática que são com uma ferramenta para extrair componentes da imagem e são úteis na descrição da forma da região e até casco convexo (que o menor conjunto que contém todos os pontos  $x$  de uma imagem), [1].

#### A. Morfologia Matemática

Uma das principais técnicas da morfologia matemática, que são as operações de erosão, dilatação, fechamento e abertura. Essas operações têm como objetivo modificar a forma e estrutura dos objetos presentes na imagem.

A erosão é uma operação que reduz o tamanho dos objetos, removendo *pixels* de suas bordas. Essa operação é útil para eliminar ruídos e afinar objetos. Já a dilatação expande os objetos, adicionando *pixels* às suas bordas. Essa operação é útil para preencher lacunas e unir objetos próximos. Tendo

como fundo da imagem preto e o objeto branco, a erosão afina o objeto e a dilatação expande, porém em fundo branco ocorre o contrário.

O fechamento é uma combinação da dilatação seguida pela erosão. Essa operação é utilizada para preencher pequenas lacunas e suavizar as bordas dos objetos. A abertura, por sua vez, é uma combinação da erosão seguida pela dilatação. Essa operação é útil para remover pequenos objetos e separar objetos conectados.

#### B. Watershed

O algoritmo *Watershed* realiza a segmentação da imagem, que é o processo de dividir uma imagem em múltiplas regiões ou objetos, com o objetivo de simplificar e/ou identificar objetos da imagem para facilitar a sua análise. O *Watershed* é uma técnica de segmentação de imagens que se baseia no conceito de bacias hidrográficas. Essa abordagem visualiza a imagem em três dimensões, com as duas coordenadas espaciais representando a posição dos *pixels* e o terceiro eixo representando os níveis de intensidade dos *pixels*. A imagem é tratada como uma paisagem topográfica, onde os mínimos regionais representam os pontos mais baixos e as bacias de captação representam as regiões que drenam para esses mínimos. O algoritmo busca identificar essas bacias de captação e linhas divisórias, que correspondem às fronteiras naturais entre diferentes regiões da imagem, [1].

A utilização do algoritmo de segmentação *Watershed* requer a aplicação de etapas de pré-processamento para obter melhores resultados. Uma dessas etapas é a binarização da imagem, que pode ser realizada com o auxílio de técnicas de morfologia matemática, com o objetivo de separar o objeto de interesse do restante da imagem.

### II. METODOLOGIA

#### A. Questão 1

A metodologia utilizada para resolver a Questão 1 envolveu os seguintes passos. Primeiramente, a imagem foi carregada utilizando a biblioteca *OpenCV* e convertida para escala de cinza. Em seguida, a imagem foi binarizada, onde os *pixels* com valores acima de um limiar de 127 foram definidos como brancos (255) e os demais como pretos (0). A fim de encontrar os contornos dos buracos na imagem binarizada, utilizou-se a função nativa da biblioteca *OpenCV*. Com base nos contornos encontrados, uma máscara preta foi criada

e preenchida com branco nos contornos. A diferença entre a máscara preenchida e a imagem binarizada resultou em uma imagem que destaca os buracos. Em seguida, foram aplicadas operações morfológicas de dilatação e erosão para fechar os buracos e remover ruídos. O processo de fechamento foi repetido duas vezes com diferentes números de iterações para melhorar a segmentação dos buracos. A imagem resultante foi então submetida a uma função nativa da biblioteca *scikit-image*, que rotula as regiões conectadas na imagem. A quantidade de buracos foi determinada contando o número de regiões rotuladas. Por fim, utilizou as propriedades das regiões rotuladas, incluindo o diâmetro equivalente de cada buraco. O número de buracos e seus respectivos diâmetros em *pixels* foram impressos como resultado. A imagem original e as imagens intermediárias resultantes de cada etapa do processamento também foram salvas para fins de análise.

#### B. Questão 2

A metodologia adotada para resolver a Questão 2 envolveu os seguintes passos. Inicialmente, a imagem foi carregada utilizando a biblioteca *OpenCV* e convertida para escala de cinza. Em seguida, um filtro de desfoque mediano foi aplicado à imagem para reduzir o ruído. Para binarizar a imagem, foi utilizado o método de limiar adaptativo com a função nativa do *OpenCV*, com um valor de limiar calculado automaticamente com base na vizinhança local de cada pixel. Posteriormente, operações morfológicas de dilatação e erosão foram aplicadas para fechar regiões conectadas e remover ruídos indesejados. A diferença entre a imagem erodida e a imagem original foi calculada para tentar obter somente o fundo da imagem.

Para melhorar o resultado, um filtro bilateral foi aplicado para reduzir ainda mais o ruído da imagem original. Em seguida, novamente foi realizada a binarização, seguida por uma operação de inversão. Definindo um elemento estruturante de forma elíptica foi criado e, uma operação de abertura foi aplicada para tentar evitar símbolos desconectados ou ruídos. Em seguida, uma erosão adicional foi realizada para afinar os traços dos símbolos. Por fim, a imagem resultante foi submetida a uma operação de fechamento para tentar conectar símbolos adjacentes e aprimorar a forma geral dos símbolos presentes.

#### C. Questão 3

A metodologia adotada para resolver a Questão 3 envolveu os seguintes passos. Primeiramente, a imagem foi binarizada utilizando o método de *Otsu* para que as células aparecessem em preto e o fundo em branco, utilizando métodos similares à Seção II-B.

Em seguida, foram aplicadas operações morfológicas de fechamento para corrigir regiões desconectadas nas células. A diferença entre a imagem original e a imagem resultante do fechamento foi calculada para remover o fundo e manter somente as células.

Após isso, foi realizado um preenchimento de buracos nas células, invertendo a imagem binarizada e encontrando os contornos das células. Os contornos foram preenchidos

para obter uma imagem sem buracos. As células nas bordas não foram totalmente preenchidas, então foi realizada uma correção adicional. Utilizando o conceito de componentes conectados, as células foram rotuladas e somente a maior região, correspondente às células, foi mantida na imagem.

Por fim, a segmentação watershed foi aplicada à imagem. Marcadores foram criados com base nos componentes conectados da imagem de transformação de distância e o algoritmo watershed foi aplicado. As regiões segmentadas foram coloridas com cores aleatórias e a imagem resultante foi gerada.

### III. RESULTADOS

#### A. Questão 1

Nesta primeira etapa foi realizado a binarização da imagem original. Esta etapa não foi um problema porque a imagem já estava praticamente binária.



Figura 1. Imagem original.



Figura 2. Imagem binarizada.

A próxima etapa é foi realizado uma máscara contendo os contornos da imagem binarizada.



Figura 3. Imagem com os contornos em branco, (Máscara).



Figura 4. Binarização da Máscara.

Optou-se por fazer o fechamento por etapas individuais, então foi realizado a dilatação e depois a erosão. Segue imagens:

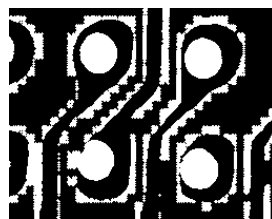


Figura 5. Dilatação



Figura 6. Erosão

Utilizou-se a operação de fechamento novamente, porém dilatou-se um vez somente e depois realizou-se 6 erosões.

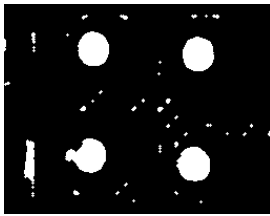


Figura 7. Dilatação da última imagem



Figura 8. Erosão da imagem dilatada, (seis erosões).

Neste último parte, equilibrou-se a quantidade de vezes que a imagem foi dilatada com a mesma quantidade que erodida. Com isso, é necessário que a imagem seja dilatada seis vezes. Segue imagem original com a imagem final.

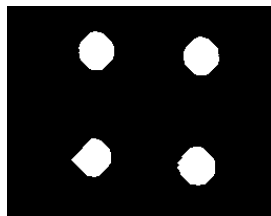


Figura 9. Final Image

Segue imagem original com a imagem final afim de se comparar.



Figura 10. Imagem original.

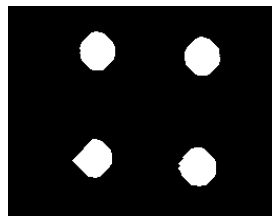


Figura 11. Imagem final.

No algoritmo, foi identificado 4 buracos e com seus respectivos diâmetros. Segue diâmetro de cada buraco:

- Buraco 1 - Diâmetro: 34.6689 pixels.
- Buraco 2 - Diâmetro: 34.6873 pixels.
- Buraco 3 - Diâmetro: 34.0576 pixels.
- Buraco 4 - Diâmetro: 35.3598 pixels.

### B. Questão 2

Nesta questão foi realizado o filtro passa-baixas para borrar a imagem e eliminar uma parte do ruído, e depois disso binarizou-se a imagem borrada. Segue imagem original e a imagem borrada.

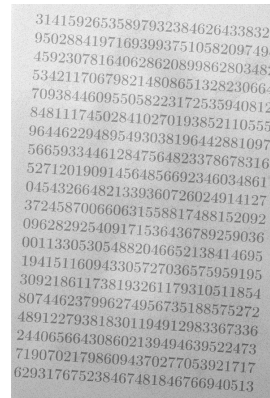


Figura 12. Image Original

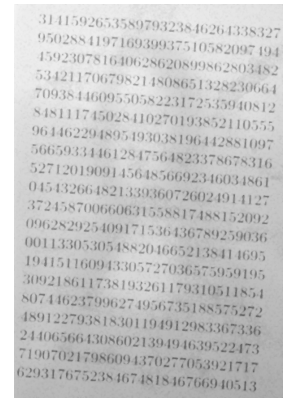


Figura 13. Image Borrada

Depois de binarizar realizou a operação morfológica de fechamento para conectar os pontos brancos dos caracteres.



Figura 14. Imagem binarizada

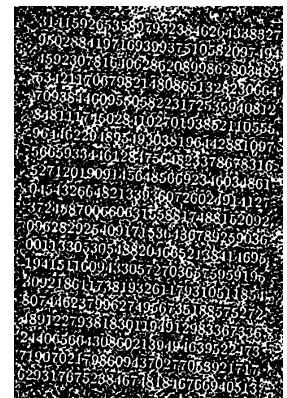


Figura 15. Fechamento

Subtraindo o fechamento pela imagem original, obtêm-se uma imagem sem tanto ruído e mais nítida. Porém ainda se tem ruído e portanto usou-se um filtro passa-baixas bilateral. Segue imagens:



Figura 16. Fechamento - Imagem Original

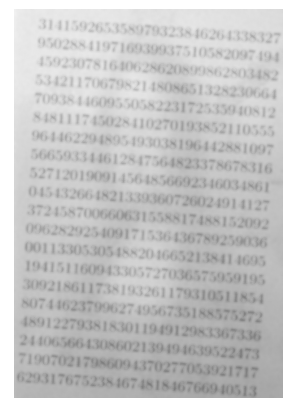


Figura 17. Imagem filtrada

Binarizando a imagem obtida na filtragem e fazendo a abertura para eliminar mais alguns ruídos, temos os seguintes

resultados.

```
314159265358979323846264338327
950288419716939937510582097494
459230781640628620899862803482
534211706798214808651328230664
709384460955058223172535940812
848111745028410270193852110555
964462294895493038196442881097
566593344612847564823378678316
527120190914564856692346034861
045432664821339360726024914127
372458700660631558817488152092
096282925409171536436789259036
001133053054882016652138414695
194151160943305727036575959195
309218611738193261179310511854
807446237996274956735188575272
489122793818301194912983367336
244065664308602139494639522473
719070217086094370277053921717
629317675238467481846766940513
```

Figura 18. Imagem binária filtrada.

```
314159265358979323846264338327
950288419716939937510582097494
459230781640628620899862803482
534211706798214808651328230664
709384460955058223172535940812
848111745028410270193852110555
964462294895493038196442881097
566593344612847564823378678316
527120190914564856692346034861
045432664821339360726024914127
372458700660631558817488152092
096282925409171536436789259036
001133053054882016652138414695
194151160943305727036575959195
309218611738193261179310511854
807446237996274956735188575272
489122793818301194912983367336
244065664308602139494639522473
719070217086094370277053921717
629317675238467481846766940513
```

Figura 19. Abertura

Nessa parte é realizado um ajuste fino e para isso realizou-se uma erosão para ampliar os caracteres.

```
314159265358979323846264338327
950288419716939937510582097494
459230781640628620899862803482
534211706798214808651328230664
709384460955058223172535940812
848111745028410270193852110555
964462294895493038196442881097
566593344612847564823378678316
527120190914564856692346034861
045432664821339360726024914127
372458700660631558817488152092
096282925409171536436789259036
001133053054882016652138414695
194151160943305727036575959195
309218611738193261179310511854
807446237996274956735188575272
489122793818301194912983367336
244065664308602139494639522473
719070217086094370277053921717
629317675238467481846766940513
```

Figura 20. Imagem erodida.

```
314159265358979323846264338327
950288419716939937510582097494
459230781640628620899862803482
534211706798214808651328230664
709384460955058223172535940812
848111745028410270193852110555
964462294895493038196442881097
566593344612847564823378678316
527120190914564856692346034861
045432664821339360726024914127
372458700660631558817488152092
096282925409171536436789259036
001133053054882016652138414695
194151160943305727036575959195
309218611738193261179310511854
807446237996274956735188575272
489122793818301194912983367336
244065664308602139494639522473
719070217086094370277053921717
629317675238467481846766940513
```

Figura 21. Imagem Final

Segue comparação da imagem final com a imagem original:

```
314159265358979323846264338327
950288419716939937510582097494
459230781640628620899862803482
534211706798214808651328230664
709384460955058223172535940812
848111745028410270193852110555
964462294895493038196442881097
566593344612847564823378678316
527120190914564856692346034861
045432664821339360726024914127
372458700660631558817488152092
096282925409171536436789259036
001133053054882016652138414695
194151160943305727036575959195
309218611738193261179310511854
807446237996274956735188575272
489122793818301194912983367336
244065664308602139494639522473
719070217086094370277053921717
629317675238467481846766940513
```

Figura 22. Imagem original

```
314159265358979323846264338327
950288419716939937510582097494
459230781640628620899862803482
534211706798214808651328230664
709384460955058223172535940812
848111745028410270193852110555
964462294895493038196442881097
566593344612847564823378678316
527120190914564856692346034861
045432664821339360726024914127
372458700660631558817488152092
096282925409171536436789259036
001133053054882016652138414695
194151160943305727036575959195
309218611738193261179310511854
807446237996274956735188575272
489122793818301194912983367336
244065664308602139494639522473
719070217086094370277053921717
629317675238467481846766940513
```

Figura 23. Imagem final

### C. Questão 3

Antes de binarizar a imagem, foi realizado uma operação de fechamento e este foi subtraído da imagem original. Segue imagens:

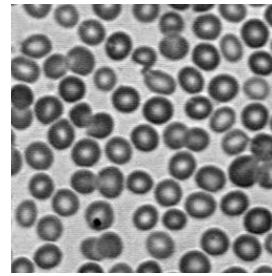


Figura 24. Imagem Original.

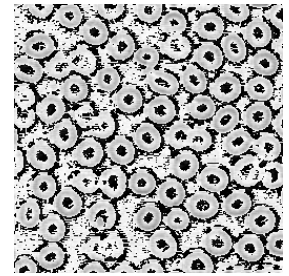


Figura 25. Imagem original - Fechamento

Com isso, binarizou essa diferença usando o algoritmo de Otsu. Além disso, as células precisaram ser preenchidas com preto, portanto realizou operações de contorno e preenchimento por regiões. Segue resultado dessas operações:

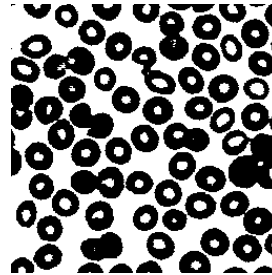


Figura 26. Threshold Image

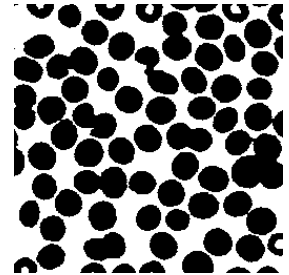


Figura 27. Opening Image

Pode-se perceber que as células que estavam nas bordas não foram preenchidas. Para preenchê-las foi necessário usar técnicas de contorno para detectar essas células. Para melhorar o resultado realizou a operação de fechamento para afinar as conexões nas células. O resultado é mostrado a seguir

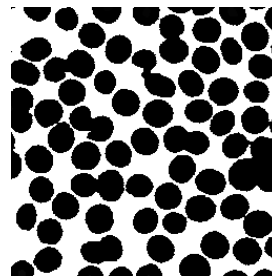


Figura 28. Imagem com todas as células preenchidas.

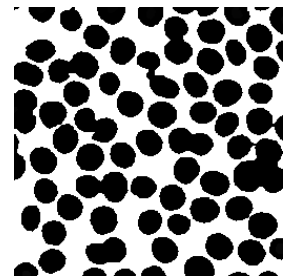


Figura 29. Fechamento

Para realizar o algoritmo *watershed*, utilizou-se as técnicas de detecção de regiões, que é um dos parâmetros para efetuar o *watershed*. Segue resultado do *watershed* e da detecção das células.

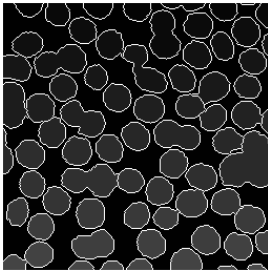


Figura 30. Detecção das células

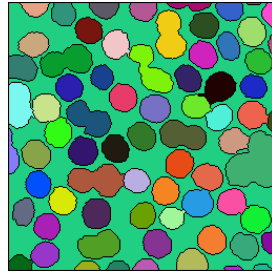


Figura 31. Segmentação watershed

Segue imagem original mais imagem segmentada:

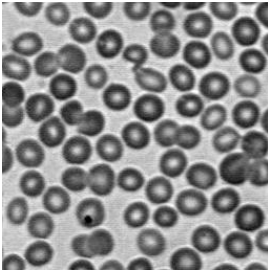


Figura 32. Imagem Original

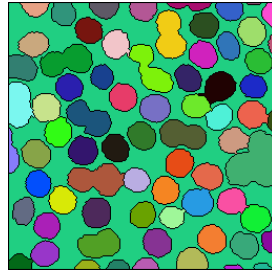


Figura 33. Segmentação watershed

#### IV. CONCLUSÃO

Na primeira questão, foram realizados vários procedimentos que resultaram em uma imagem final sem ruído e melhor definição dos espaços destinados aos componentes eletrônicos. No entanto, um buraco na imagem original apresentou uma ponta indesejada (localizado no canto inferior direito), não atingindo um resultado completamente satisfatório. No entanto, como os buracos têm características semelhantes, pode-se realizar uma média entre eles para obter um resultado mais coerente.

Na segunda questão, adotou-se uma abordagem que incluiu o uso de um filtro passa-baixas em duas etapas para reduzir o ruído. O resultado final apresentou ainda algum ruído residual, além de algumas letras que ficaram parcialmente cortadas. No entanto, ainda é possível distinguir cada caractere de forma geral.

Na terceira questão, realizou-se a segmentação da imagem utilizando o algoritmo de *watershed*. Os resultados obtidos demonstraram que a segmentação por *watershed* foi eficaz na separação das células e na identificação de suas regiões individuais. No entanto, em alguns casos em que as células estavam próximas umas das outras, houve identificação incorreta de uma única célula. Apesar disso, essa abordagem permitiu uma segmentação próxima ao esperado, evitando a sobreposição de células e preservando suas características distintivas.

#### REFERÊNCIAS

- [1] R. C. Gonzalez and R. E. Woods, *Processamento de imagens digitais*. Editora Blucher, 2000.