

Departamento da Área de Informática

Curso: Bacharelado em Engenharia da Computação **Semestre:** 9

Curso: Bacharelado em Engenharia de Controle e Automação **Semestre:** Optativa

Disciplina: Processamento Digital de Imagens.

Professor: Esp. Giuliano Robledo Zucoloto Moreira.

Cuiabá-MT, 28 de julho de 2022.

NOTA EXPLICATIVA

Tópico: Segmentação de imagens

Está fora do escopo desta nota esgotar conteúdos da ementa. Nela são apresentados os passos iniciais necessários ao caminho do estudo da disciplina de processamento digital de imagens (PDI). A nota muitas vezes pode ser objetiva pois considera-se que o estudante tenha conhecimentos prévios de programação dada a evolução no curso até à disciplina de PDI. O *software* utilizado no desenvolvimento dos exemplos foi o *GNU Octave* (modo gráfico) versão 6.1.0. A maioria das figuras é exibida em tamanho grande dada a necessidade do processo visual para a comparação e compreensão das explicações e dos resultados apresentados. Os resultados das operações de segmentação são apresentados como imagens para fins de ilustrar as saídas do processo e facilitar o entendimento do leitor, contudo a operação de segmentação pode retornar apenas informações, como uma contagem de objetos.

1 Segmentação de imagens

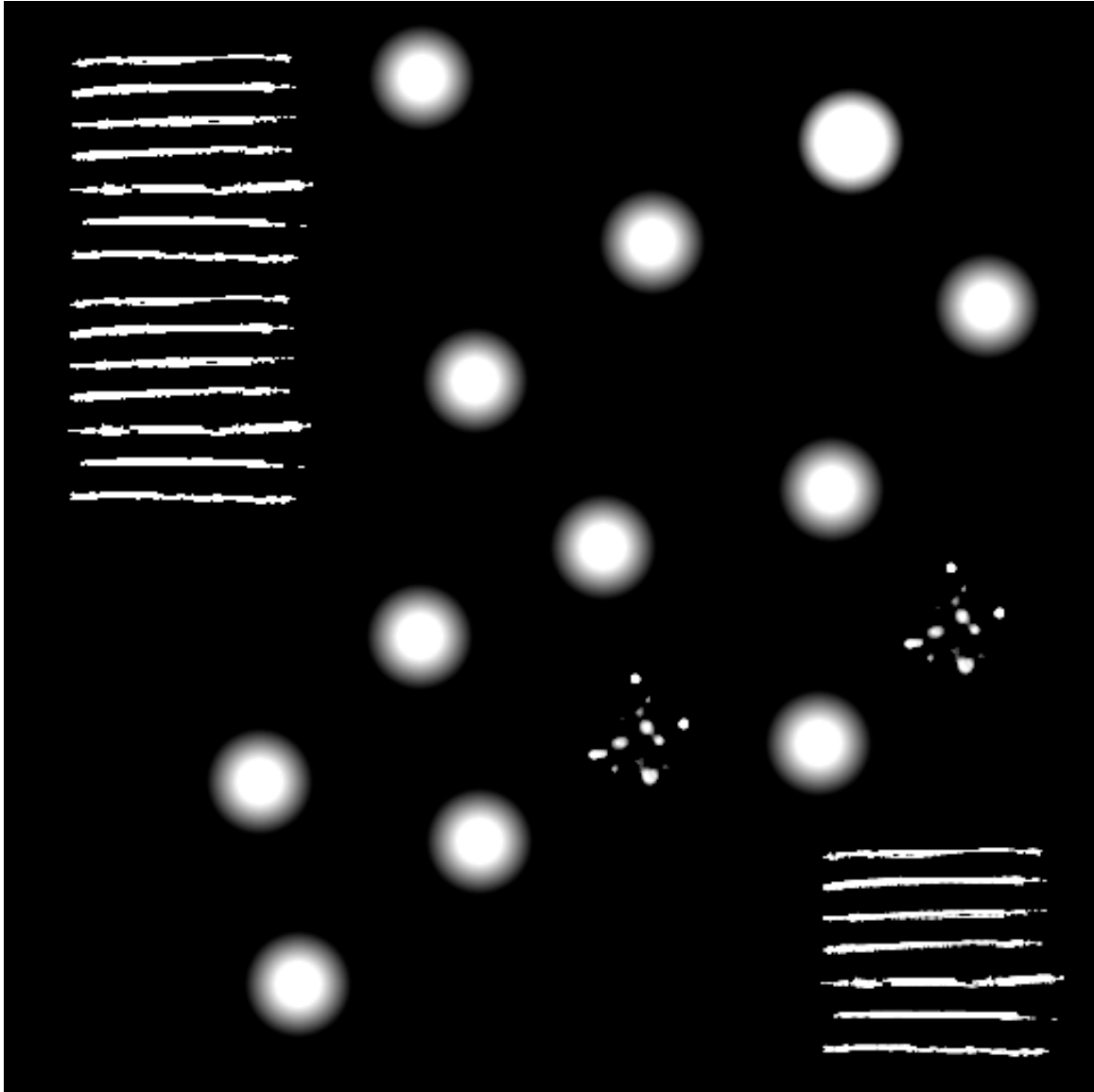
Segmentação de imagens é uma técnica que “subdivide uma imagem em regiões ou objetos que a compõem” [1, p. 454], o que é muito útil para a extração de atributos de imagem (informações). O processo de subdivisão pode ser sucessivo e é mantido até que o objetivo seja alcançado, ou não, dada a complexidade do problema e a solução adotada.

Por se tratar de uma das técnicas mais difíceis de se obter êxito no processamento digital de imagens (PDI) [1, p. 454] encontrar uma solução para uma inspeção automatizada de imagens exige ampla base de conhecimentos prévios, tanto do domínio do problema, quanto das técnicas de PDI e na maioria das vezes técnicas de outras áreas correlatas como geometria analítica, cálculo, e matemática discreta. Dependendo do domínio do problema, além destes, podem ser exigidos conhecimentos da área da física e de outras técnicas de engenharia, principalmente engenharia elétrica.

Esta nota trás um exemplo introdutório de segmentação de imagens. É sintetizada uma imagem populada com objetos de três formatos distintos, e para cada formato de objeto é gerada uma imagem de saída populada apenas com a região que contém a localização dos objetos representados pelo formato mapeado na imagem. O código-fonte do *script* utilizado para realizar a segmentação, apresentação e arquivamento das imagens das saídas utilizadas nas explicações desta nota está transcrito no Quadro 1.

A Figura 1 apresenta a imagem sintética contendo diversos objetos distribuídos aleatoriamente, com formatos distintos predominantemente constituídos de barras horizontais, circunferências e pulverização. É possível observar que as barras horizontais tem um padrão irregular com descontinuidades, os círculos tem uma borda suavizada e as pulverizações diversos tamanhos de fragmentos.

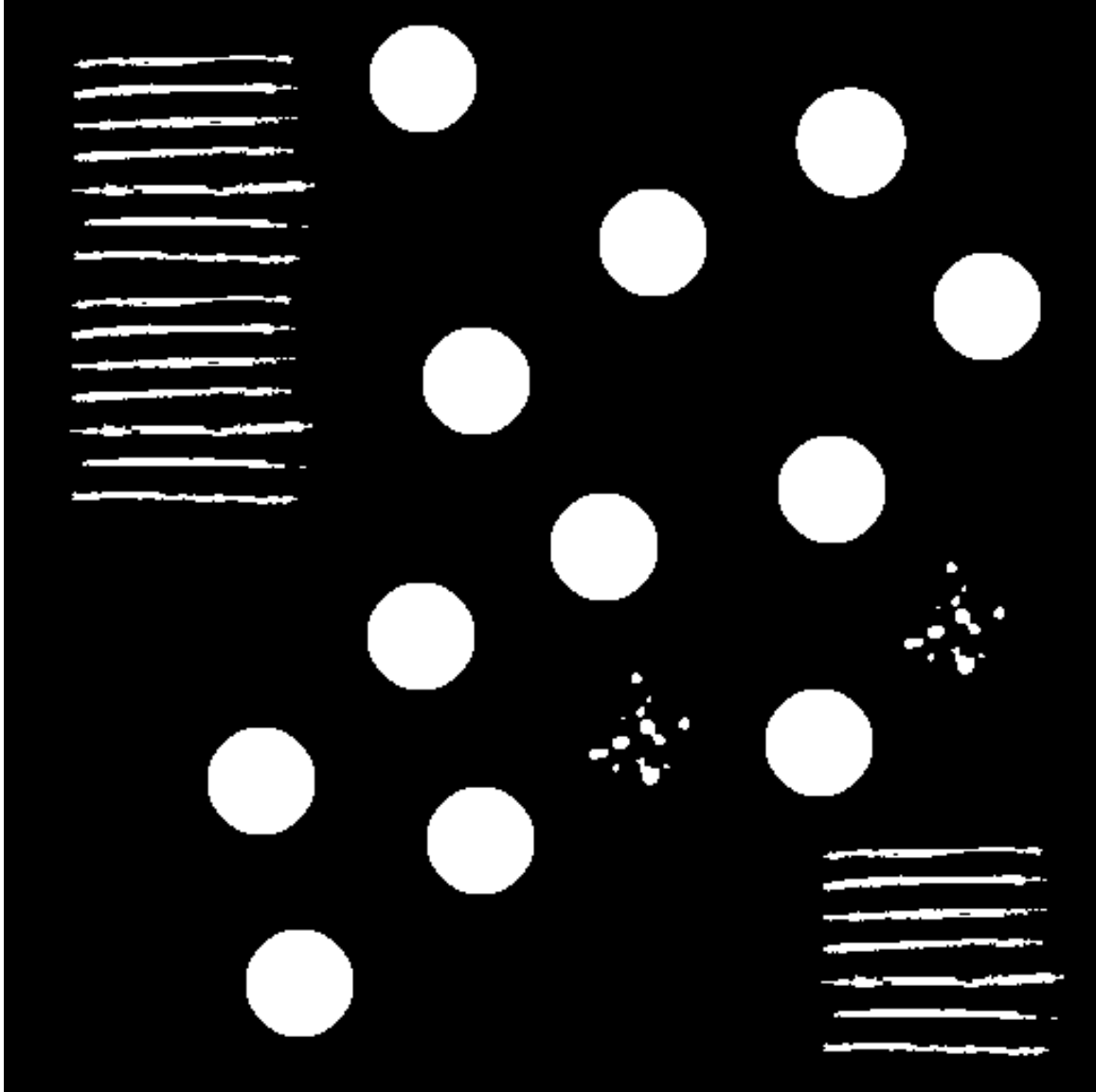
Figura 1: Imagem sintética



Fonte: Elaborado pelo Autor.

A Figura 2 é uma representação binária da imagem obtida por meio da função *im2bw* do pacote *image* do software *GNU Octave*. A principal diferença perceptível entre a imagem sintética original e esta imagem está nas bordas das circunferências, agora em formato de *degrau unitário*.

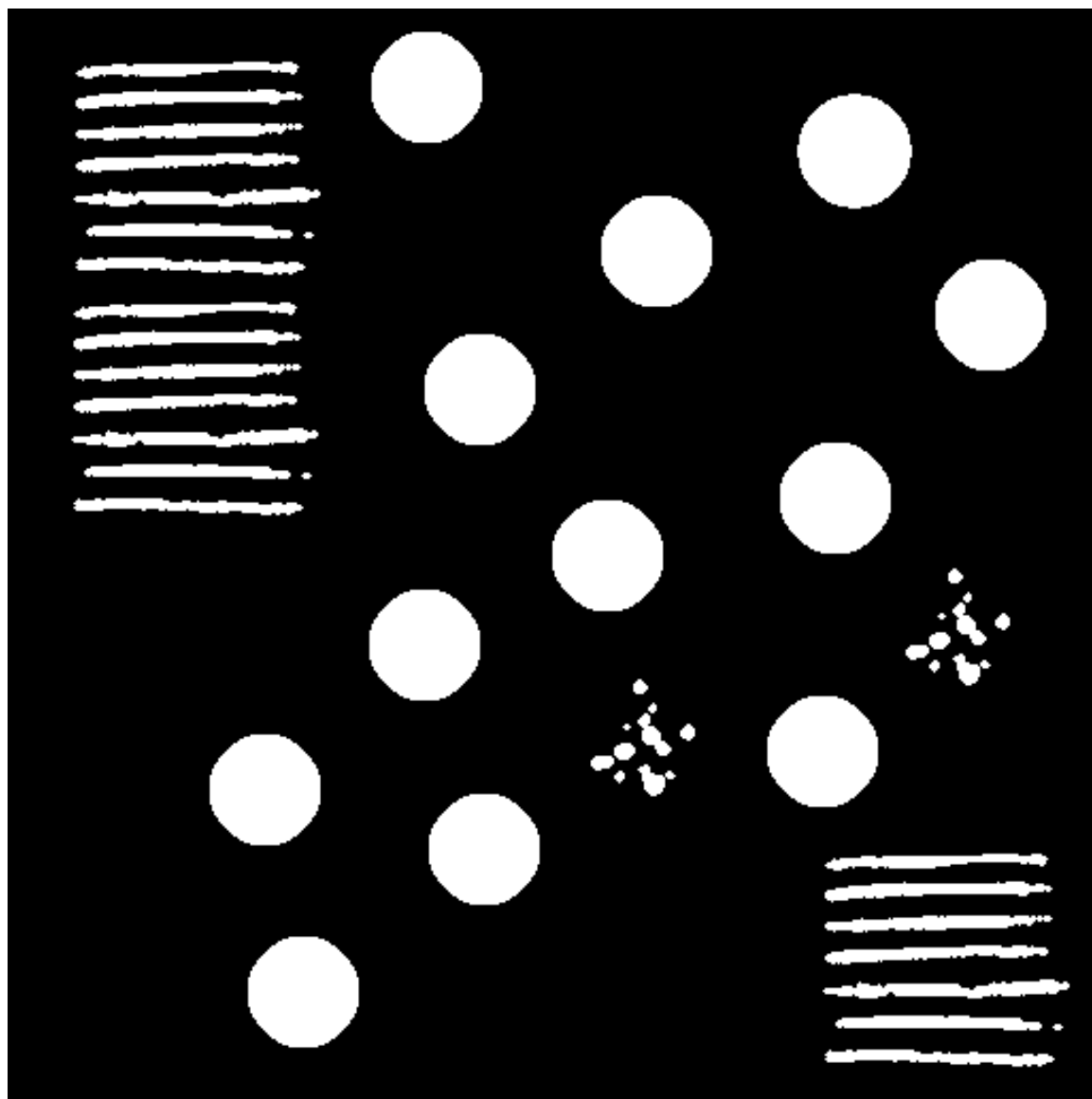
Figura 2: Imagem sintética binarizada



Fonte: Elaborado pelo Autor.

A Figura 3 apresenta a imagem sintética binária após um processo de **dilatação** por um elemento estruturante em formato de círculo (Quadro 1, linhas de comando 16-22). A **dilatação**, além de aumentar o tamanho dos objetos, nas barras horizontais, preencheu as descontinuidades internas e reduziu a irregularidade nas bordas causadas pelas descontinuidades. A redução das descontinuidades costuma ser uma vantagem, pois simplifica a construção das **máscaras de detecção** para uma forma geométrica simples, o que possibilita a varredura de áreas maiores por meio de intervalos, simplificando o algoritmo.

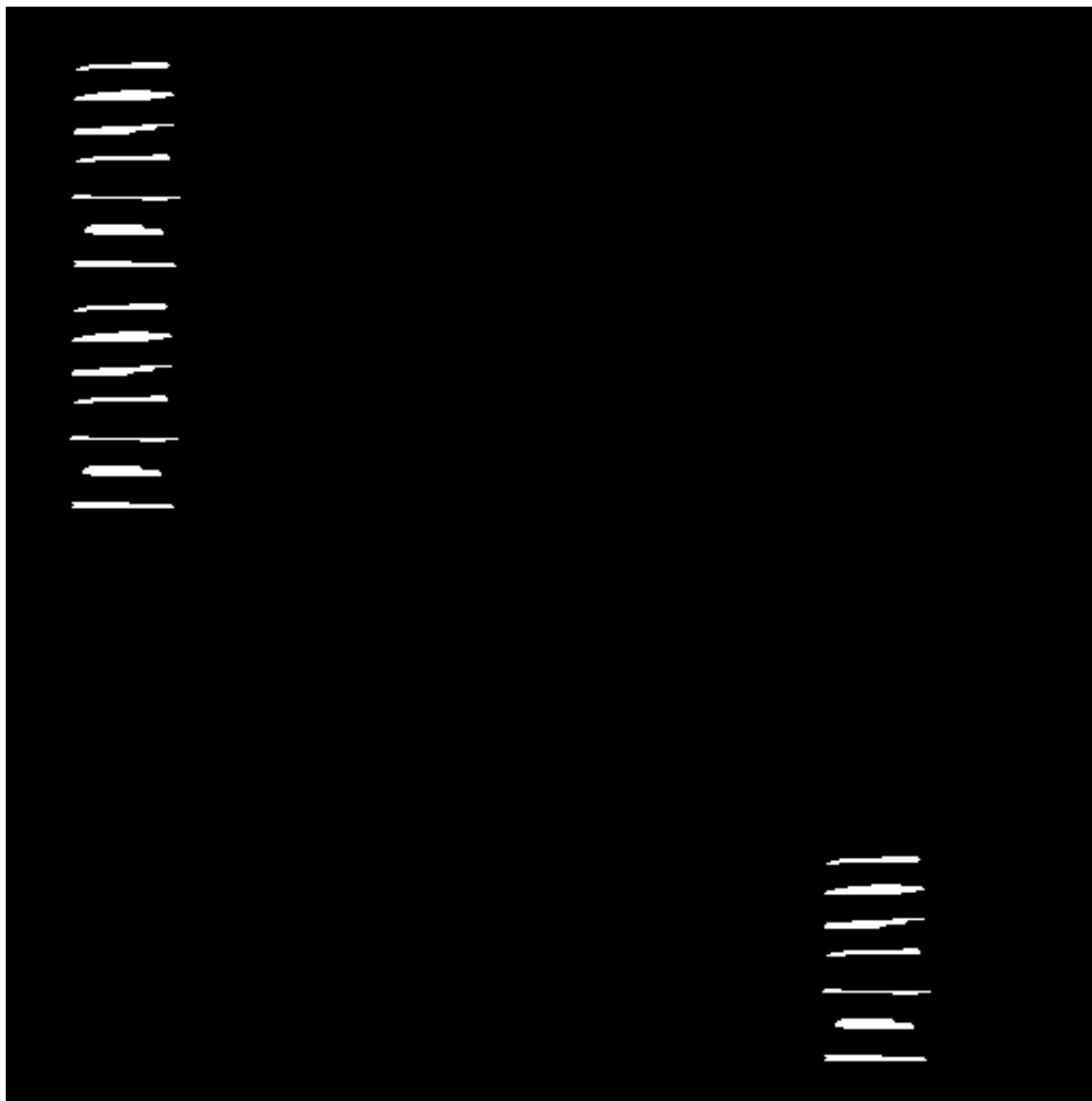
Figura 3: Dilatação da imagem sintética binarizada



Fonte: Elaborado pelo Autor.

A Figura 4 foi gerada de forma automatizada valendo-se apenas e nada mais, da simplicidade do formato geométrico da **máscara de detecção** (Quadro 1, linhas de comando 29,33-35), uma linha horizontal com comprimento de 60 (sessenta) *pixels*, cuja implementação foi possível devido à operação de **dilatação** que preencheu as descontinuidades internas e corrigiu parte das irregularidades descontinuidades das bordas.

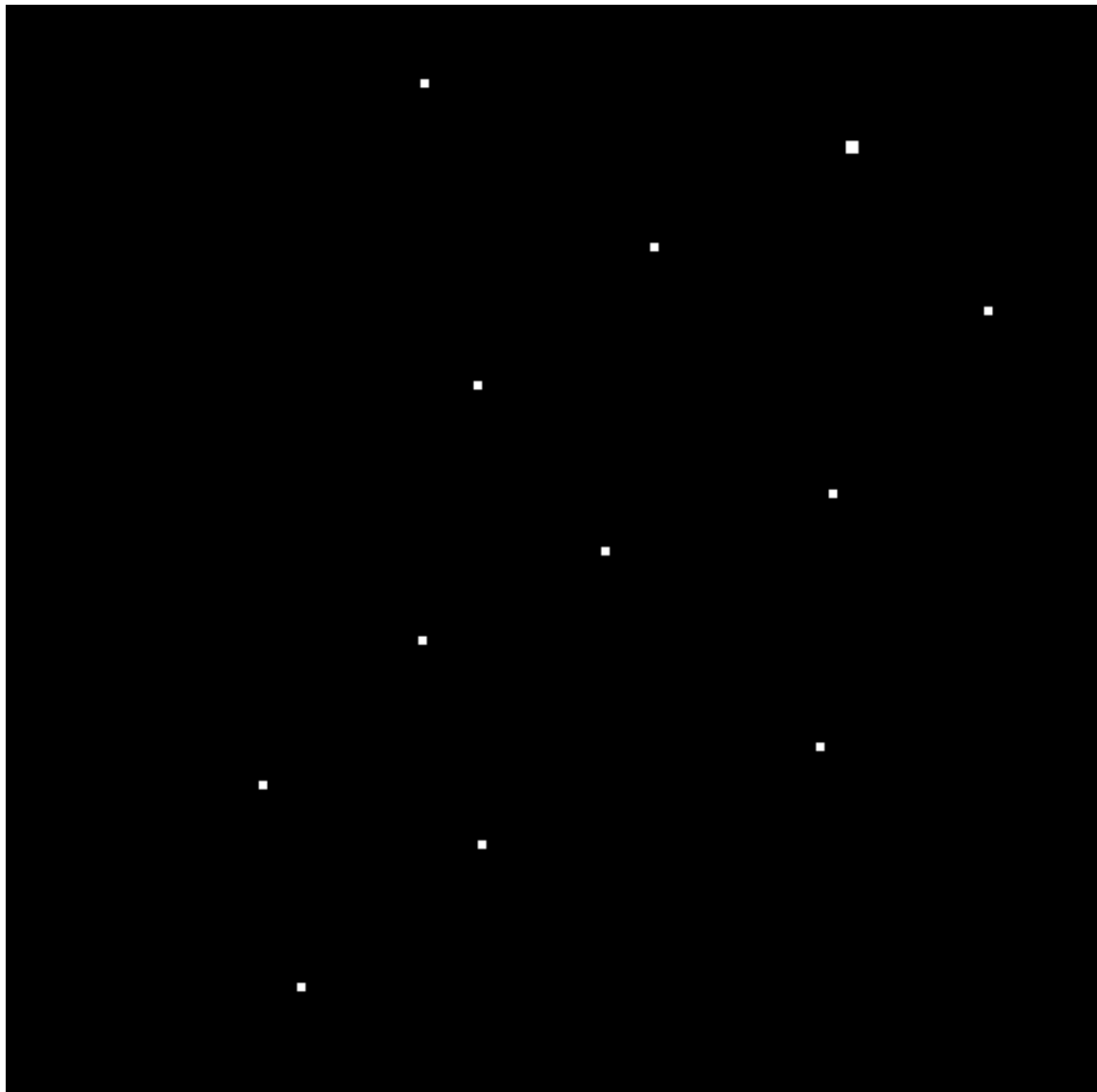
Figura 4: Segmentação da imagem sintética - detecção dos objetos horizontais



Fonte: Elaborado pelo Autor.

A Figura 5, gerada de forma automatizada, apresenta pequenas áreas indicando onde as circunferências estão radialmente localizadas na imagem sintética original. A continuidade do preenchimento das circunferências e sua propriedade geométrica de simetria são fatores que possibilitam a criação de uma **máscara de detecção** simples. Se na imagem sintética estivessem presentes apenas circunferências isométricas e isoladas, uma linha horizontal ou vertical, com comprimento do tamanho do diâmetro de uma circunferência, seria suficiente para ser uma **máscara de detecção**. Para o caso específico da imagem em questão, a máscara utilizada na detecção das circunferências foi construída em formato de cruz com dimensões menores ou iguais ao diâmetro das circunferências (Quadro 1, linhas de comando 42, 46-48). Dado o domínio do problema, sabe-se que as dimensões das circunferências são suficientes para permitir a aplicação de uma máscara em formato de cruz sem que haja interseção completa da máscara com outros objetos presentes. Os objetos em formato de circunferência são detectados como pequenas áreas e não um ponto pelo fato de seus quadrantes formarem linhas retas, o que implica em repetidas ocorrências de identidade de centro para a circunferência em análise.

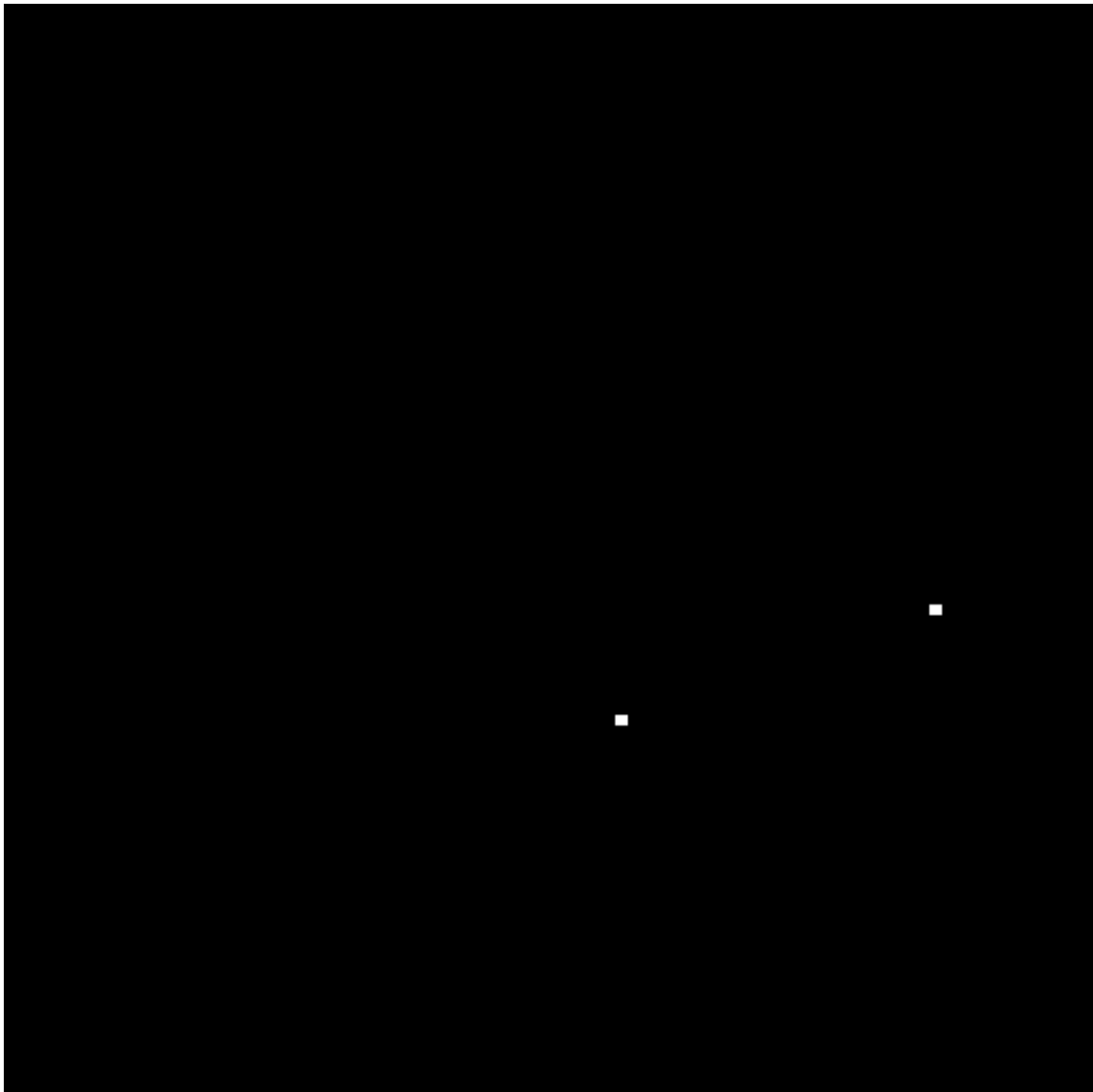
Figura 5: Segmentação da imagem sintética - detecção dos objetos circulares



Fonte: Elaborado pelo Autor.

A Figura 6, também gerada de forma automatizada, segue o mesmo raciocínio de buscar uma identidade com o objeto a ser detectado. A diferença é que a **máscara de detecção** utilizada não considera apenas *pixels* com valor 1 (um), ela é uma matriz populada com zeros e uns representando o formato de um elemento presente de forma exclusiva apenas no objeto de interesse de forma que não haja interseção de identidade com qualquer outro objeto da imagem, garantindo a exatidão na segmentação.

Figura 6: Segmentação a imagem sintética - detecção dos objetos pulverizados



Fonte: Elaborado pelo Autor.

Os resultados das operações de segmentação (Figuras 4, 5 e 6) podem ser otimizados por meio de operações de **erosão** até que as regiões que indicam a existência de um determinado objeto se tornem um único ponto.

Conhecidas as dimensões da imagem e os formatos dos objetos, e a cada formato sendo atribuído um ponto de referência (um *pixel*), a saída da operação de segmentação pode ser um arquivo com tais informações e as coordenadas de referência de cada objeto detectado na imagem, tais informações são suficientes para recuperar a imagem original.

A imagem pode ser dividida em regiões, também de acordo com a coordenada de re-

ferência cada objeto identificado. Para simplificar imagine que ao redor de cada objeto identificado seja desenhado um retângulo delimitando uma região, ou que a cada objeto identificado seja gerada uma nova imagem de saída com uma cópia da região onde o objeto foi identificado.

O Quadro 1 apresenta o código-fonte do *script* que foi utilizado na construção desta nota.

Quadro 1 - Código-fonte do *script* de segmentação utilizado no exemplo

```

1  clc;
2  clear all;
3  clf;
4  pkg load image;
5  original=imread("Bactral.png");
6  [M N] = size(original);
7  originalcomBordas = zeros(M+2,N+2);
8  originalcomBordas(2:M+1,2:N+1) = original(1:M,1:N);
9  binarizada=im2bw(originalcomBordas);
10 binarizadaDilatada=zeros(M+2,N+2);
11 ##Operação de dilatação = = = = =
12 ##Varredura com elemento estruturante lógico
13 for (l=2:M-1)
14     for (c=2:N-1)
15         ##Elemento estruturante lógico
16         if(binarizada(l,c)==1)
17             binarizadaDilatada(l-1,c)=1;
18             binarizadaDilatada(l,c-1)=1;
19             binarizadaDilatada(l,c)=1;
20             binarizadaDilatada(l,c+1)=1;
21             binarizadaDilatada(l+1,c)=1;
22         endif
23     endfor
24 endfor
25
26 ##Segmentação por similaridade = = = = =
27 saidaObjHorizontais=zeros(M+2,N+2);
28 ##Detecção do padrão de linhas grandes contínuas (objetos
   ↪ horizontais)
29 k=60;
30 for (l=2:M-1)
31     for (c=2:N-(k+1))
32         ##Máscara de detecção
33         if(binarizadaDilatada(l,c:c+k)==1)
34             saidaObjHorizontais(l,c)=1;
35         endif
36     endfor
37 endfor
38
39 ##Segmentação por similaridade = = = = =
40 saidaObjCirculares=zeros(M+2,N+2);

```



```

41  ##Detecção do padrão de linhas grandes contínuas (objetos
    → circulares)
42  k=24;
43  for (l=k+2:M-(k+1))
44      for (c=k+2:N-(k+1))
45          ##Máscara de detecção
46          if((binarizadaDilatada(l-k:l+k,c)==1) &&
              → (binarizadaDilatada(l,c-k:c+k)==1))
47              saidaObjCirculares(l,c)=1;
48          endif
49      endfor
50  endfor
51
52  ##Segmentação por similaridade = = = = =
53  saidaObjPulverizados=zeros(M+2,N+2);
54  modeloObjPulverizados=[0 0 0 0 0 0; 0 0 1 1 0 0 ; 0 1 1 1 1 0;
    → 0 0 1 1 0 0; 0 0 0 0 0 0 ];
55  ##Detecção do padrão de linhas grandes contínuas (objetos
    → pulverizados)
56  [k1 k2] = size(modeloObjPulverizados);
57  for (l=2:M-k1-1)
58      for (c=2:N-k2-1)
59          ##Máscara de detecção
60          if(binarizadaDilatada(l:l+k1-1,c:c+k2-1) ==
              → modeloObjPulverizados(1:k1,1:k2))
61              saidaObjPulverizados(l:l+k1-1,c:c+k2-1)=1;
62          endif
63      endfor
64  endfor
65
66  a=2;
67  b=3;
68  subplot(a,b,1)
69  imshow(original,[]);
70  title("Imagem original");
71  subplot(a,b,2)
72  imshow(uint8(binarizada),[0 1]);
73  imwrite(binarizada,"BactralBinarizada.png");
74  title("Imagem binarizada");
75  subplot(a,b,3)
76  imshow(uint8(binarizadaDilatada),[0 1]);
77  imwrite(binarizadaDilatada,"BactralBinarizadaDilatada.png");
78  title("Imagem binarizada dilatada");
79  subplot(a,b,4)
80  imshow(uint8(saidaObjHorizontais),[0 1]);
81  imwrite(saidaObjHorizontais,"saidaObjHorizontais.png");
82  title("Objetos horizontais");
83  subplot(a,b,5)
84  imshow(uint8(saidaObjCirculares),[0 1]);

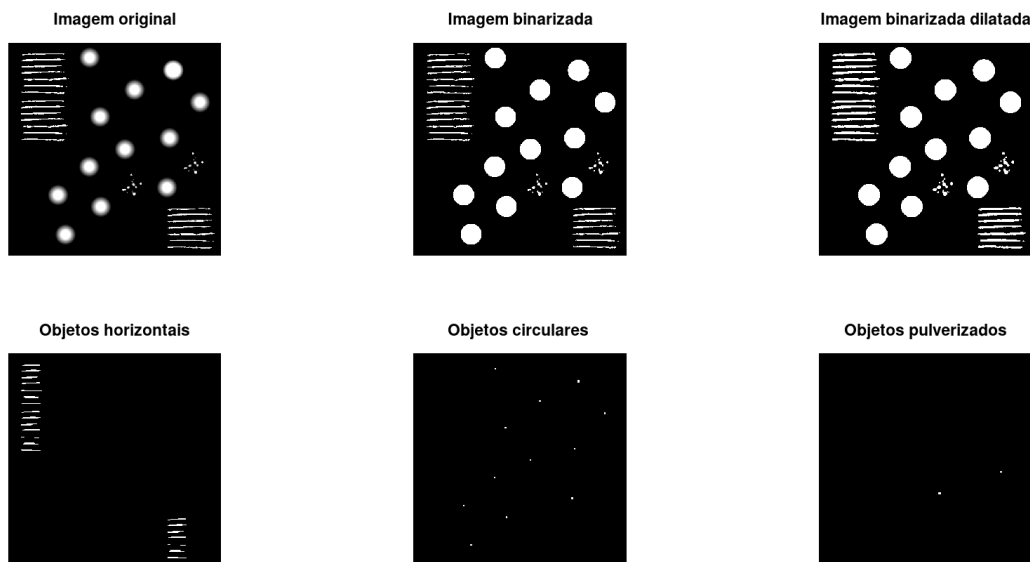
```

```
85  imwrite(saidaObjCirculares,"saidaObjCirculares.png");
86  title("Objetos circulares ");
87  subplot(a,b,6)
88  imshow(uint8(saidaObjPulverizados),[0 1]);
89  imwrite(saidaObjPulverizados,"saidaObjPulverizados.png");
90  title("Objetos pulverizados");
```

Fonte: Elaborado pelo Autor.

A Figura 7 apresenta a janela com uma saída produzida pelo *script* disponível no Quadro 1. As outras saídas, que são arquivos de imagem em formato *png* (*portable network graphics*), no momento da redação desta *nota explicativa*, foram disponibilizados na sala virtual da disciplina no tópico correspondente ao assunto abordado.

Figura 7: Saída da segmentação realizada através do *script* apresentado no Quadro 1



Fonte: Elaborado pelo Autor.

Espera-se que a abordagem desta *nota explicativa* tenha sido suficiente para iniciar o processo de estudo de segmentação de imagens. Para amplo e completo entendimento do assunto recomenda-se realizar o acesso ao tópico *Segmentação de imagens* disponível na sala virtual da disciplina e também a leitura do *capítulo 10* do livro texto da disciplina.

Referências

- [1] R. E. GONZALEZ, Rafael C.; WOODS, *Processamento Digital de Imagens*. São Paulo: Pearson Prentice Hall, 3 ed., 2010. Revisão técnica: Marcelo Vieira e Maurício Escarpinati; [tradução Cristina Yamagami e Leonardo Piamonte].