

UM ESTUDO DE SEGMENTAÇÃO DE IMAGENS BASEADO EM UM MÉTODO DE
COMPUTAÇÃO EVOLUCIONÁRIA

Valeriana Gomes Roncero

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS
EM ENGENHARIA ELÉTRICA

Aprovada por:

Prof. Amit Bhaya, Ph.D.

Prof. Nelson Francisco Favilla Ebecken, D.Sc.

Profa. Marley Maria Bernardes Rebuzzi Vellasco, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

JUNHO DE 2005

RONCERO, VALERIANA GOMES

Um Estudo de Segmentação de Imagens
Baseado em um Método de Computação Evolu-
cionária [Rio de Janeiro] 2005

X,70p. 29,7 cm (COPPE/UFRJ, M.Sc., En-
genharia Elétrica, 2005)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

1. processamento de imagens, segmentação de
imagens, algoritmo genético

I. COPPE/UFRJ II. Título (série)

Agradecimentos

Quero expressar o meu agradecimento ao professor Amit Bhaya pelos ensinamentos, pela confiança e pelas críticas, que foram fundamentais a minha aprendizagem e amadurecimento.

Agradeço aos meus amigos da sala H-238 (Myrian, Leonardo, Paula, Angelo, Oumar, Magno e Miguel) e aos meus amigos particulares (Roberta, Cíntia e Flávia) pelo apoio que sempre me deram.

Um agradecimento aos meus pais (Maria e Esteban), irmã (Anita) e ao meu namorado (Oberdan Jr.) que entenderam a minha ausência e estiveram sempre ao meu lado me estimulando.

Quero também agradecer às secretarias do Programa da Engenharia Elétrica e do Laboratório do NACAD pela atenção que sempre me deram. Ao NACAD e à CAPES pela oportunidade de desenvolver o projeto.

Enfim, a todos aqueles que de alguma forma contribuíram para a realização deste trabalho.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UM ESTUDO DE SEGMENTAÇÃO DE IMAGENS BASEADO EM UM MÉTODO DE COMPUTAÇÃO EVOLUCIONÁRIA

Valeriana Gomes Roncero

Junho/2005

Orientador: Amit Bhaya

Programa: Engenharia Elétrica

Verifica-se a utilização cada vez maior do computador nas tarefas de extração de informações de imagens. Uma dessas tarefas é a segmentação ou subdivisão de uma imagem em suas partes ou objetos constituintes. Essa tese realiza um estudo sobre um algoritmo de segmentação utilizando um algoritmo genético, sendo que dois tipos de seleção são definidos e utilizados com dois diferentes tipos de cromossomos. O primeiro tipo de cromossomo armazena somente o rótulo da região, e o segundo tipo armazena a coordenada do *pixel* e o rótulo da região. Os resultados obtidos são comparados em relação à qualidade da segmentação.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

A STUDY OF IMAGE SEGMENTATION BASED ON A METHOD OF
EVOLUTIONARY COMPUTATION

Valeriana Gomes Roncero

June/2005

Advisor: Amit Bhaya

Department: Electrical Engineering

Image segmentation is an essential part of image processing and refers to the subdivision of an image into its parts or constituent objects. This thesis carries out a study of an image segmentation algorithm using a genetic algorithm. Two types of selection and two different types of chromosomes are used and compared. The first chromosome type stores only the region label, while the second type stores the pixel coordinate as well as the region label. Results obtained are compared in terms of quality of segmentation.

Sumário

Lista de Figuras	viii
Lista de Tabelas	x
1 Introdução	1
2 Processamento de Imagens	4
2.1 Introdução	4
2.2 Representação da Imagem Digital	4
2.2.1 Amostragem e Quantização	5
2.3 Elementos de um Sistema de Processamento de Imagens	6
2.4 Etapas do Processamento de Imagens	6
2.5 Introdução à Topologia Digital	8
2.5.1 Vizinhança	8
2.5.2 Relação Binária	9
2.5.3 Métrica	9
2.5.4 Relação de Adjacência e Grafos	10
2.5.5 Relação de Conectividade	10
2.5.6 Componente Conexo	10
2.6 Histograma	11
2.6.1 Critérios para o Estabelecimento de Limiares	11
2.7 Segmentação de Imagens	12
2.8 Técnicas Baseadas em Descontinuidades	14
2.8.1 Detecção de Pontos Isolados	15
2.8.2 Detecção de Linhas	16
2.8.3 Detecção de Bordas	16
2.8.3.1 Operador Gradiente	16
2.8.3.2 Operador Laplaciano	18
2.9 Técnicas Baseadas em Similaridades	18
2.9.1 Limiarização (<i>Thresholding</i>)	18
2.9.2 Crescimento de Regiões (<i>Region Growing</i>)	19
2.9.3 Divisão e Conquista (<i>Split & Merge</i>)	20

3	Algoritmos Genéticos	22
3.1	Introdução	22
3.2	Processos Básicos de AG's	24
4	Algoritmo de Segmentação de Imagem baseado em Algoritmo Genético	29
4.1	Introdução	29
4.2	Implementação do Algoritmo de Segmentação	30
4.2.1	Leitura e Armazenamento dos Dados	32
4.2.2	Geração e codificação dos rótulos	33
4.2.3	Cálculo da vizinhança	33
4.2.4	Cálculo do valor médio	35
4.2.5	Cálculo da função de desempenho	35
4.2.6	Seleção	36
4.2.6.1	Seleção por Aleatoriedade	36
4.2.6.2	Seleção por Vizinhança	38
4.2.7	Aplicação do cruzamento	39
4.2.8	Aplicação da mutação	39
4.2.9	Geração da população final	39
4.2.10	Critério de Parada	40
5	Resultados do Algoritmo Proposto	41
5.1	Resultados	42
5.1.1	Imagens Humanas	43
5.1.2	Imagem de Animal	51
5.1.3	Imagens de Mapas	55
5.1.4	Conclusão	63
6	Conclusão	65
	Referências Bibliográficas	67

Lista de Figuras

2.1	Imagem monocromática de uma mulher.	5
2.2	Etapas do processamento de imagens.	7
2.3	(a) Vizinhança do tipo 4 (b) Vizinhança do tipo 8 e (c) Vizinhança do tipo 6.	9
2.4	O conjunto X possui dois componentes conexos-4: $\{p_1, p_2, p_3\}$ e $\{p_4\}$	11
2.5	Histogramas de níveis de cinzas que podem ser particionados por (a) um único limiar e (b) por múltiplos limiares.	12
2.6	Imagem da Lena, em tonalidades de cinza (240×240 <i>pixels</i>), utilizada em centenas de artigos sobre processamento de imagens.	14
2.7	Convolução sobre uma imagem resulta em outra de mesmas dimensões. No processo ilustrado, para cada nova posição da máscara, apenas o <i>pixel</i> central da região afetada recebe o valor da operação aritmética ou resposta da máscara.	15
2.8	Máscara usada para detecção de diferentes pontos isolados.	16
2.9	Máscaras de linhas.	16
2.10	Máscara utilizada para computar o Laplaciano.	18
2.11	(a) Imagem de uma superfície. (b) Imagem da superfície limiarizada.	19
2.12	(a) Imagem particionada; (b) Quadtree correspondente.	21
3.1	População Inicial.	23
3.2	Ciclo básico do Algoritmo Genético.	25
3.3	Processo de Cruzamento.	25
3.4	Exemplo de Mutação.	26
3.5	Exemplificação da roleta com as probabilidades de seleção dos 4 indivíduos.	27
3.6	Cruzamento com ponto de corte igual a 2.	27
4.1	Algoritmo de Segmentação Genético.	31
4.2	Cromossomo com 8 <i>bits</i> utilizado para armazenar o rótulo.	33
4.3	Vizinhança com 8 conexões.	34
4.4	Exemplo da vizinhança do <i>pixel</i> que se encontra na coordenada (0, 0).	35
4.5	Seleção do tipo 1 para a escolha dos indivíduos.	37
5.1	Imagem do pingüim e o arquivo da imagem aberto em um editor de texto.	41
5.2	(a) Imagem original da Lena. (b) Imagem com contorno aplicado.	43
5.3	Imagem segmentada por limiarização.	43
5.4	Segmentação com Seleção por Aleatoriedade: (a) Tipo 1 (b) Tipo 2 e Segmentação com Seleção por Vizinhança: (c) Tipo 3 (d) Tipo 4.	44

5.5	(a) Imagem segmentada utilizando a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.	45
5.6	Imagem Lena: Gráfico da Variância com a seleção por vizinhança do tipo 3.	46
5.7	Imagem Lena: Gráfico da Aptidão com a seleção por vizinhança do tipo 3. .	46
5.8	(a) Imagem original de uma criança. (b) Imagem com contorno aplicado. .	47
5.9	Imagem segmentada por limiarização.	47
5.10	Segmentação com Seleção por Aleatoriedade: (a) Tipo 1 (b) Tipo 2 e Segmentação com Seleção por Vizinhança: (c) Tipo 3 (d) Tipo 4.	48
5.11	(a) Imagem segmentada utilizando a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.	49
5.12	Imagem criança: Gráfico da Variância com a seleção por vizinhança do tipo 3.	49
5.13	Imagem criança: Gráfico da Aptidão com a seleção por vizinhança do tipo 3.	50
5.14	(a) Imagem original do pingüim. (b) Imagem com contorno aplicado. . . .	51
5.15	Imagem segmentada por limiarização.	51
5.16	Resultado da imagem segmentada exibida no artigo [17]	52
5.17	Segmentação com Seleção por Aleatoriedade: (a) Tipo 1 (b) Tipo 2 e Segmentação com Seleção por Vizinhança: (c) Tipo 3 (d) Tipo 4.	52
5.18	(a) Imagem segmentada utilizando a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.	53
5.19	Imagem pingüim: Gráfico da Variância com a seleção por vizinhança do tipo 3.	54
5.20	Imagem pingüim: Gráfico da Aptidão com a seleção por vizinhança do tipo 3.	54
5.21	(a) Imagem original da cidade do Rio de Janeiro. (b) Imagem com contorno aplicado.	55
5.22	Imagem segmentada por limiarização.	55
5.23	Segmentação com Seleção por Aleatoriedade: (a) Tipo 1 (b) Tipo 2 e Segmentação com Seleção por Vizinhança: (c) Tipo 3 (d) Tipo 4.	56
5.24	(a) Imagem segmentada utilizando a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.	57
5.25	Imagem Rio de Janeiro: Gráfico da Variância com a seleção por vizinhança do tipo 3.	58
5.26	Imagem Rio de Janeiro: Gráfico da Aptidão com a seleção por vizinhança do tipo 3.	58
5.27	(a) Imagem original da cidade de Paris. (b) Imagem com contorno aplicado.	59
5.28	Imagem segmentada por limiarização.	59
5.29	Segmentação com Seleção por Aleatoriedade: (a) Tipo 1 (b) Tipo 2 e Segmentação com Seleção por Vizinhança: (c) Tipo 3 (d) Tipo 4.	60
5.30	(a) Imagem segmentada utilizando a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.	61
5.31	Imagem Paris: Gráfico da Variância com a seleção por vizinhança do tipo 3.	61
5.32	Imagem Paris: Gráfico da Aptidão com a seleção por vizinhança do tipo 3. .	62
5.33	(a) Imagem segmentada com a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.	64

Lista de Tabelas

2.1	Operadores 3x3 utilizados para estimar a magnitude do gradiente através de uma borda	17
3.1	Codificações do Genótipo e do Fenótipo.	23
3.2	Codificações do Genótipo e Fenótipo; e o valor da Aptidão de cada indivíduo.	24
3.3	Codificações do Genótipo e Fenótipo; o valor da Aptidão e a Probabilidade de ser sorteado cada indivíduo.	26
3.4	Codificações do Genótipo e Fenótipo; e o valor da Aptidão de cada indivíduo da nova população.	28
4.1	Exemplo da seleção do tipo 1.	37
4.2	Exemplo da seleção do tipo 2.	38
4.3	Exemplo da seleção do tipo 3.	38
4.4	Exemplo da seleção do tipo 4.	39
5.1	Representação dos resultados obtidos com: o tipo de seleção, a geração em que o critério de parada foi alcançado e o tempo médio das gerações.	63

Capítulo 1

Introdução

As técnicas de processamento de imagens começaram a ter aplicação no início do século XX com o envio de imagens por meio de cabos submarinos entre Londres e Nova Iorque para publicação em jornais. As imagens eram codificadas no transmissor e decodificadas no receptor e havia a necessidade do melhoramento da qualidade da imagem no receptor. O grande salto nas técnicas de processamento de imagens ocorreu com o surgimento dos computadores de grande porte no período da corrida espacial com a necessidade da análise e melhoramento das imagens de televisão enviadas por sondas espaciais, das missões para a Lua, para Marte (missões Mariner) e dos vôos tripulados para a Lua.

Atualmente, o processamento de imagens é feito em áreas de aplicação do conhecimento humano para resolver uma variedade enorme de problemas, que, normalmente, requerem métodos capazes de melhorar a informação visual para análise e interpretação humana ou para percepção automática por máquinas. Como exemplo da análise e interpretação humana de imagens melhoradas por tratamento digital, podemos citar as aplicações na área médica com imagens de raios-X ou ressonância, na geologia, no estudo de imagens de satélites, na restauração de figuras em arqueologia, na melhoria de imagens físicas, na medicina nuclear, na melhoria de imagens transmitidas por ondas de rádio etc. Como exemplo da percepção por máquinas, podemos citar o reconhecimento automático de caracteres, processamento automático de impressões digitais, reconhecimento automático de rostos, visão computacional, guiagem de robôs, processamento automático de imagens de satélites para reconhecimento de queimadas etc.

O processamento de imagens digitais envolve uma série de conceitos que incluem o estudo da estrutura do olho humano; o estudo de transformadas de imagens, como a Transformada de Fourier e outras transformadas separáveis de imagens; técnicas de realce, como filtragem e utilização de máscaras, cujo propósito é processar a imagem de modo que o resultado seja mais apropriado para uma aplicação do que a imagem original; restauração de imagens, como técnicas de filtragem inversa, Filtro de Wiener (filtro de mínimo médio quadrático), restauração interativa e interpolação de níveis de cinza, cujo propósito é reconstruir ou recuperar uma imagem degradada; compressão de imagens, como compressão livre de erro e com perdas, cujo propósito é reduzir a quantidade de dados necessária para representar uma imagem digital; segmentação de imagens, como detecção de bordas, de pontos, de linhas e descontinuidade, cujo propósito é a divisão da imagem em partes para sua análise; reconhecimento e interpretação de imagens, como técnicas de análise baseada em conhecimento e redes neurais, cujo propósito é a análise da imagem e o reconhecimento de padrões e de características da imagem.

Nesta tese é estudado a segmentação de imagens que desempenha um papel chave nos sistemas automáticos visuais, isto porque a percepção das partes interessantes de uma cena é uma etapa preliminar para o reconhecimento e a interpretação da imagem. Além da segmentação também é estudado um método evolucionário, o algoritmo genético. Na literatura existem vários métodos para segmentar uma imagem, alguns utilizam o agrupamento de regiões, como por exemplo, em Di Gesú [1] que utiliza as informações espacial e de intensidade que são utilizadas pela percepção humana; outros combinam técnicas de segmentação, como crescimento de regiões e detecção de bordas [2]. E existe também alguns que utilizam sistemas estocásticos de otimização que visam um máximo global, como anelamento simulado [3, 4, 5] e algoritmos evolucionários [6, 7], como o algoritmo genético [8, 9, 10, 11, 12, 13, 14, 15, 16]. O artigo do Lo Bosco [17] aborda a junção do algoritmo genético que é um algoritmo evolucionário com as informações espacial e de intensidade contidas em uma imagem.

Esta tese descreve um estudo sobre um método de segmentação de imagens utilizando algoritmo genético através da análise de valores de intensidade dos *pixels* (níveis de cinza) abordado por Lo Bosco [17]. Nesta tese foi considerado o problema de segmentação de imagens como um problema de otimização global (GOP). Esse método utiliza algoritmo genético para otimizar uma função de similaridade que é baseada na função de intensidade e

na posição espacial do *pixel*.

Esta tese está organizada como segue. No Capítulo 2 serão descritos os principais conceitos relacionados a processamento de imagens. No Capítulo 3 serão descritos os principais conceitos relacionados ao algoritmo genético. No Capítulo 4, será descrito o algoritmos utilizado e no Capítulo 5 os resultados experimentais obtidos. Finalmente, as conclusões obtidas com este trabalho serão descritas no Capítulo 6.

Capítulo 2

Processamento de Imagens

2.1 Introdução

Devido ao grande avanço tecnológico dos últimos tempos, verifica-se a utilização cada vez maior do computador nas tarefas de extração de informações de imagens para auxiliar na solução de diversos problemas práticos, tais como contagem de glóbulos brancos em amostras de sangue, sensoriamento remoto via satélite, e outras.

Para que seja possível a utilização de imagens na solução desses problemas, elas devem ser adquiridas e digitalizadas (amostradas e quantizadas). Após a digitalização, podem ser feitas diversas transformações na imagem até conseguir o resultado desejado para solução do problema. Essas transformações podem ser agrupadas nos seguintes conjuntos: filtragem, análise, compressão e síntese de imagens.

Ao longo deste capítulo serão abordados os principais conceitos relacionados a processamento de imagens.

2.2 Representação da Imagem Digital

Uma imagem monocromática é uma função de intensidade de luz bidimensional $f(x, y)$, onde x e y denotam coordenadas espaciais e o valor de f no ponto (x, y) é proporcional ao brilho (ou nível de cinza) da imagem neste ponto (Figura 2.1). Esta função também pode ser vista como uma superfície no espaço (x, y, z) , onde para cada ponto (x, y) plota-se na coordenada z o valor de $f(x, y)$.

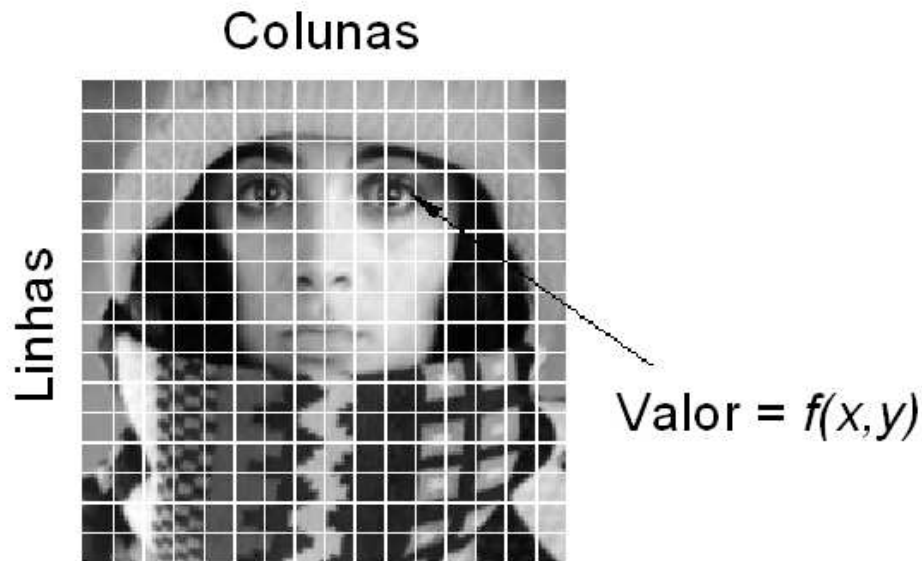


FIGURA 2.1: Imagem monocromática de uma mulher.

2.2.1 Amostragem e Quantização

Para gerar uma imagem digital, $f(x, y)$ deve ser digitalizada ao longo de x e y , e na amplitude $z = f(x, y)$. Para tanto é feita uma amostragem de $f(x, y)$ nas direções x e y , gerando uma matriz de $N \times M$ amostradas, seguida de uma quantização do valor de $f(x, y)$ em L níveis inteiros de cinza, que vão de 0 à 255. Nesta matriz, cada elemento $p(x, y)$, $x = 0, 1, \dots, M - 1$ e $y = 0, 1, \dots, N - 1$, é chamado *pixel* (*picture elements*). Dizemos então que a imagem tem dimensão M *pixels* na horizontal (eixo x) e N *pixels* na vertical (eixo y). O conceito de dimensão de um *pixel* ao longo do eixo x , ou do eixo y , está relacionado com o espaçamento físico entre as amostras. Cada *pixel* tem associado um valor $L_{min} \leq p(x, y) \leq L_{max}$, onde $L = L_{max} - L_{min} + 1$, com isso podemos assumir que $L_{min} = 0$.

2.3 Elementos de um Sistema de Processamento de Imagens

Os elementos que compõem um sistema de processamento de imagem digital são:

- Aquisição;
- Armazenamento;
- Processamento;
- Exibição;
- Comunicação.

As imagens podem ser adquiridas através de câmeras de vídeo, *scanners* de mesa, tomógrafos, satélites, etc. Estas imagens podem ser armazenadas em disco rígido ou em outros dispositivos de armazenamento, tais como, fitas magnéticas e discos ópticos. A imagem armazenada pode ser processada para cumprir os mais variados objetivos: compressão, melhoramento, reconhecimento de padrões, etc. A imagem adquirida ou processada pode ser visualizada em diversos dispositivos de exibição: filmes fotográficos, impressora, tela de computador, etc. Estas imagens também podem ser transmitidas para outras localidades através de sistemas de comunicação.

2.4 Etapas do Processamento de Imagens

Podemos entender uma imagem como uma forma compacta de representar muitas informações. Em um sistema de processamento de imagens estas informações podem passar por diversas formas de representação. Portanto, as etapas do processamento de imagens descrevem o fluxo destas informações como um dado objetivo definido pela aplicação.

A Figura 2.2 ilustra este fluxo formal geral. Após a aquisição da imagem, a etapa a ser feita é o pré-processamento que consiste na maior parte de transformações lineares e não-lineares aplicadas à imagem visando várias metas: melhoramento de contraste, remoção de ruído, regiões de interesse, decorrelação e codificação das informações para transmissão da

imagem, reamostragem dos *pixels* em uma nova escala, treinamento e extração de características de imagem para segmentação, etc. Muitas aplicações requerem apenas operações de pré-processamento.



FIGURA 2.2: Etapas do processamento de imagens.

As informações de interesse podem também ser extraídas das imagens e representadas de uma outra forma. Para tanto, a segmentação de imagens particiona a imagem em regiões disjuntas com algum significado para a aplicação. Por exemplo, podemos querer separar um objeto de interesse do resto dos *pixels* da imagem particionando-a em duas regiões. A segmentação será a etapa que iremos abordar. Em geral a segmentação é uma das mais difíceis tarefas em processamento de imagem. Esse passo no processamento determina o eventual sucesso ou falha de toda a análise [18]. A saída da segmentação pode ser a fronteira do objeto com seu exterior ou os pontos de seu interior. Isto define duas formas de representação para o objeto. A representação consiste, portanto, das várias formas de armazenar a fronteira e o interior dos objetos segmentados. Esta nova representação da imagem contém informações sobre a forma e a topologia dos objetos. A descrição quantitativa destas

informações através da extração de características estruturais complementa o sentido de representação. Em seguida, com base na descrição, o reconhecimento associa um significado ao conjunto de objetos segmentados. Um exemplo em uma aplicação de leitura automática de endereços é o reconhecimento e a interpretação de um conjunto de caracteres como o código de endereçamento postal.

2.5 Introdução à Topologia Digital

Topologia digital é o estudo de propriedades de objeto em imagem digital, as quais não são afetadas por transformações geométricas, exceto aquelas que envolvem junção ou separação de partes do objeto. Em processamento de imagem digital, um objeto em uma imagem 2D (ou 3D) é aproximado por um conjunto de *pixels*. Portanto, topologia digital estuda as propriedades deste conjunto de *pixels* que correspondem às propriedades topológicas do objeto original.

2.5.1 Vizinhança

Em uma imagem digital 2D um *pixel* $p = (x, y)$ tem quatro vizinhos que compartilham uma aresta com p : $p = (x + 1, y), (x - 1, y), (x, y + 1)$ e $(x, y - 1)$ (Figura 2.3a). Este conjunto é chamado vizinhança 4 de p ($N_4(p)$). Considerando os vizinhos que compartilham pelo menos um vértice com $p = (x, y)$ temos um conjunto vizinhança 8 de p ($N_8(p)$) (Figura 2.3b), esse tipo de vizinhança será utilizado na aplicação. Este conjunto é formado pelos *pixels* de $N_4(p)$ e os *pixels* diagonais $(x + 1, y + 1), (x - 1, y + 1), (x + 1, y - 1)$ e $(x - 1, y - 1)$. Um tratamento especial é normalmente dado aos *pixels* que pertencem às bordas da imagem, pois alguns de seus vizinhos vão estar fora da imagem.

Os conceitos de vizinhança são facilmente estendidos para imagens 3D, onde teremos vizinhanças de 6, 18 e 26, considerando os *pixels* que compartilham com o *pixel* central uma face, pelo menos uma aresta e pelo menos um vértice, respectivamente. A Figura 2.3c, por exemplo, ilustra o caso vizinhança 6. Neste caso os vizinhos do *pixel* $v = (x, y, z)$ são $(x - 1, y, z), (x + 1, y, z), (x, y - 1, z), (x, y + 1, z), (x, y, z - 1)$ e $(x, y, z + 1)$.

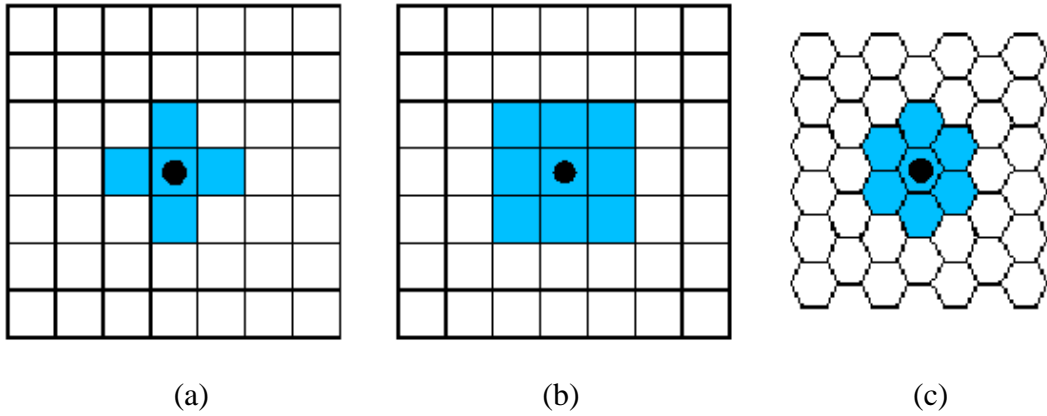


FIGURA 2.3: (a) Vizinhaça do tipo 4 (b) Vizinhaça do tipo 8 e (c) Vizinhaça do tipo 6.

De forma similar ao que foi feito para vizinhaça, os conceitos definidos nas próximas seções também se estendem para uma imagem 3D, mas será considerado apenas as imagens 2D.

2.5.2 Relação Binária

Uma relação binária R aplicada a um conjunto X é um subconjunto do produto cartesiano $X \times X$.

Uma relação binária é dita reflexiva se $(a, a) \in R$, para todo $a \in X$, simétrica se $(a, b), (b, a) \in R$, para todo $a, b \in X$, e transitiva se $(a, b), (b, c) \in R$ implica que $(a, c) \in R$, para todo $a, b, c \in X$. Neste caso R é dita de equivalência.

2.5.3 Métrica

Uma função d de distância, entre *pixels* é uma métrica se: $d(p, q) \geq 0$ ($d(p, q) = 0$, se $p = q$), $d(p, q) = d(q, p)$, $d(p, r) \leq d(p, q) + d(q, r)$, onde $p = (x_p, y_p)$, $q = (x_q, y_q)$, e $r = (x_r, y_r)$ são três *pixels* da imagem. As métricas mais usadas são:

- **Euclidiana:** $d(p, q) = ((x_p - x_q)^2 + (y_p - y_q)^2)^{\frac{1}{2}}$,
- **City-block:** $d(p, q) = |x_p - x_q| + |y_p - y_q|$,
- **Chessboard:** $d(p, q) = \max\{|x_p - x_q|, |y_p - y_q|\}$,
- **Chamfer:** $d_{a,b}(p, q) = a \cdot \max\{|x_p - x_q|, |y_p - y_q|\} + (b - a) \cdot \min\{|x_p - x_q|, |y_p - y_q|\}$, onde a, b são constantes (ex: $a = 5$ e $b = 7$).

2.5.4 Relação de Adjacência e Grafos

Uma relação de adjacência A é uma relação binária entre *pixels*, a qual depende de suas posições, e opcionalmente de outras propriedades locais da imagem. Dizemos que $A(p)$ é o conjunto dos *pixels* adjacentes ao *pixel* p de acordo com A . Isto é, $q \in A(p)$ é o mesmo que $(p, q) \in A$. Uma relação de adjacência leva, portanto, à definição de um grafo $G = (D, A)$ para imagem, onde D é conjunto de *pixels*. Neste grafo, um caminho π é uma seqüência de *pixels* adjacentes $\langle p_1, p_2, \dots, p_n \rangle$, onde $(p_i, p_{i+1}) \in A, i = 1, 2, \dots, n - 1$. Exemplos:

- $(p, q) \in A$ se $d(p, q) \leq p$, onde d é a distância Euclidiana e p é um escalar,
- $(p, q) \in A$ se $q - p \in \{(-1, 1), (1, -1)\}$,
- $(p, q) \in A$ se $|x_p - x_q| + |y_p - y_q| \leq 1$ e $|f(p) - f(q)| \leq l$, onde l é um limiar de brilho.

Observe que $p = 1$ é vizinhança-4, $p = \sqrt{2}$ é vizinhança-8, e $p = \sqrt{5}$ faz com que *pixels* distantes sejam vizinhos no grafo. Esta relação é simétrica e invariante à translação. Note também que o segundo exemplo está relacionado com a definição de elemento estruturante planar usada em morfologia matemática, e portanto uma relação de adjacência pode ser assimétrica.

2.5.5 Relação de Conectividade

Um *pixel* p é conexo a um *pixel* q se existir um caminho de p à q no grafo definido por A .

2.5.6 Componente Conexos

Um componente conexo de um conjunto X de *pixels* é um subconjunto $Y \subset X$, onde todos os pares (p, q) de *pixels* em Y são conexos (i.e. existe um caminho de p à q e um caminho de q à p , que não necessariamente são os mesmos).

Por exemplo, seja R uma relação binária "conexidade-4" aplicada sobre o conjunto $X = \{p_1, p_2, p_3, p_4\}$ da Figura 2.4; isto é $R = \{(p_1, p_2), (p_2, p_1), (p_2, p_3), (p_3, p_2), (p_1, p_3), (p_3, p_1)\}$. Note que p_4 não é conexo-4 com nenhum outro elemento X .

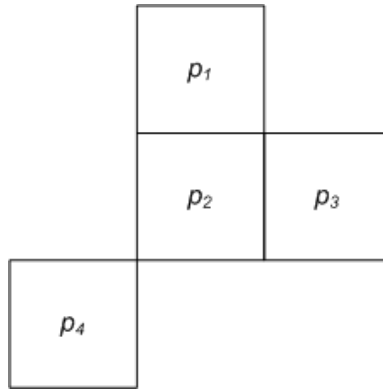


FIGURA 2.4: O conjunto X possui dois componentes conexos-4: $\{p_1, p_2, p_3\}$ e $\{p_4\}$.

2.6 Histograma

O histograma de uma imagem é um conjunto de números indicando o percentual de *pixels* naquela imagem que apresenta um determinado nível de cinza [19].

Através da visualização do histograma de uma imagem obtemos uma indicação de sua qualidade quanto ao nível de contraste e quanto ao seu brilho médio (se a imagem é predominantemente clara ou escura).

Cada elemento desse conjunto é calculado como:

$$p_r(r_k) = \frac{n_k}{n} \quad (2.1)$$

onde:

$$0 \leq r_k \leq 1;$$

$k = 0, 1, 2, \dots, N - 1$, onde N é o número de níveis de cinza da imagem digitalizada;

n = número total de *pixels* na imagem;

$p_r(r_k)$ = probabilidade do k -ésimo nível de cinza;

n_k = número de *pixels* cujo nível de cinza corresponde a k .

2.6.1 Critérios para o Estabelecimento de Limiares

Matematicamente, a operação de limiarização pode ser descrita como uma técnica de processamento de imagens na qual uma imagem de entrada $f(x, y)$ de N níveis de cinza produz uma imagem de saída $g(x, y)$, chamada de imagem limiarizada, cujo número de níveis de cinza é menor que N . Caso a imagem limiarizada $g(x, y)$ apresentar dois níveis de

cinza, tem-se

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) \geq T; \\ 0, & \text{se } f(x, y) < T. \end{cases}$$

onde os *pixels* rotulados com 1 correspondem aos objetos e os *pixels* rotulados com 0 correspondem ao fundo e T é um valor de tom de cinza predefinido, denominado limiar. Este caso é denominado limiarização simples.

A limiarização múltipla admite dois ou mais limiares; a seguir iremos admitir dois limiares: T_1 e T_2 .

$$g(x, y) = \begin{cases} 1, & \text{se } T_1 \leq f(x, y) \leq T_2; \\ 0, & \text{se } f(x, y) < T_1 \text{ ou } f(x, y) > T_2. \end{cases}$$

Quando o histograma possui um vale como na Figura 2.5a pode-se escolher o limiar como sendo o valor mínimo da curva do histograma. Caso haja mais de um vale tem-se o caso de múltiplos limiares (Figura 2.5b). A desvantagem dessa técnica, que combina o histograma com a limiarização, é que a mesma não resolve todos os problemas de segmentação, pois não leva em consideração, por exemplo, a forma dos objetos na imagem, isto é, dois objetos de formatos diferentes podem ser indistinguíveis usando-se esta técnica.

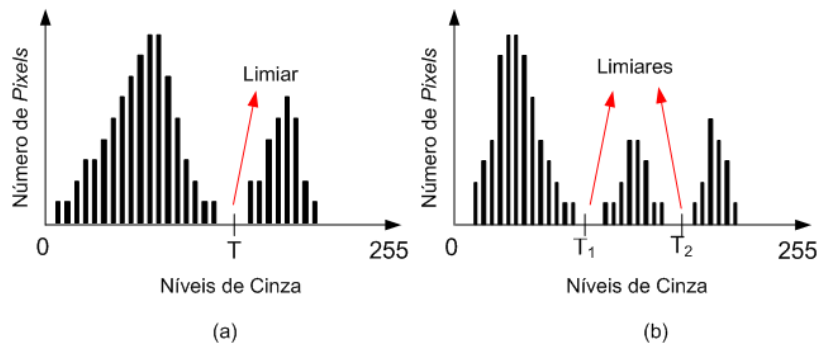


FIGURA 2.5: Histogramas de níveis de cinzas que podem ser particionados por (a) um único limiar e (b) por múltiplos limiares.

2.7 Segmentação de Imagens

A segmentação de imagem desempenha um papel chave nos sistemas automáticos visuais. Isto, porque a percepção das partes interessantes de uma cena é um passo preliminar

para o reconhecimento e a interpretação da imagem. A segmentação consiste da subdivisão de uma imagem em suas partes constituintes ou objetos, onde um objeto neste contexto refere-se a uma componente convexa. Após a segmentação, cada objeto é descrito com relação a suas propriedades geométricas e topológicas. Baseando-se nesta descrição, o reconhecimento de cada objeto resulta em uma imagem rotulada, onde *pixels* que pertencem a objetos diferentes têm associados rótulos distintos. A interpretação semântica de um conjunto de objetos rotulados no domínio de uma aplicação finaliza o processo de análise dando lugar a uma ação. Por exemplo, um sistema pode reconhecer dígitos em uma imagem de código de barras e interpretá-los como o número de acesso a um dado produto em um supermercado. A ação neste caso pode ser a contabilização do valor do produto na compra.

A segmentação é uma das principais e mais estudadas operações em processamento de imagens. A segmentação pode ser vista como um problema de classificar N elementos em K regiões, tal que $K \leq N$; $\bigcup_K^{i=1} K_i = \text{imagem}$; $K_i \cap K_j = \emptyset$, se $i \neq j$; e elementos em uma mesma região K têm propriedades similares entre si e distintas das propriedades dos elementos de outra regiões. Assim, a segmentação também pode ser modelada como um problema de otimização combinatória, no qual procura-se uma região ótima de acordo com algum critério de similaridade entre os elementos de uma mesma região.

Métodos de segmentação podem ser classificados como interativos [20, 21, 22, 23] ou automáticos [24, 25, 26, 27] levando em conta, respectivamente, a intervenção ou não intervenção do usuário na segmentação, ou podem ter sua classificação de acordo com a estratégia de representação dos objetos a serem segmentados: métodos orientados a bordas [21, 22, 23, 28] e métodos orientados a regiões [20, 29, 30]. Métodos orientados a bordas procuram detectar as bordas de cada objeto de interesse na imagem. Enquanto métodos orientados a regiões procuram identificar todos os *pixels* que pertencem ao interior de cada objeto de interesse.

Os algoritmos de segmentação de imagens em níveis de cinza geralmente são baseados em uma ou duas propriedades dos valores de níveis de cinza: descontinuidade e similaridade. Na primeira categoria, a abordagem é particionar uma imagem baseando-se nas mudanças abruptas nos níveis de cinza. As principais áreas de interesse dentro dessa categoria são a detecção de pontos isolados, detecção de linhas e detecção de bordas numa imagem através de máscaras de convolução. As principais abordagens da segunda categoria são baseadas em limiarização (*thresholding*), crescimento de regiões (*region growing*), divisão e conquista

(*split & merge*) e aglomeração (*clustering*). Estas técnicas serão discutidas nas próximas seções.



FIGURA 2.6: Imagem da Lena, em tonalidades de cinza (240×240 pixels), utilizada em centenas de artigos sobre processamento de imagens.

Além da tradicional imagem da Lena mostrada na Figura 2.6, serão utilizadas imagens em níveis de cinza de vários tipos para a validação dos algoritmos implementados.

2.8 Técnicas Baseadas em Descontinuidades

Os objetos que podem ser localizados por técnicas baseadas em descontinuidades são de três tipos: pontos, linhas e bordas. Essas técnicas são definidas por uma operação orientada a vizinhança.

Operações lógicas e aritméticas orientadas a vizinhança utilizam o conceito de convolução com máscaras (ou janelas, também denominadas, em inglês como: *templates*, *kernel* ou *masks*) [18].

A convolução envolve uma imagem original, a ser convolvida por uma máscara que deve visitar toda a extensão da imagem original (Figura 2.7), sendo que a cada posicionamento desta, uma operação deve ocorrer. Tal operação envolve os pesos da máscara e os *pixels* na imagem original, de forma que o resultado (normalmente numérico) desta operação é "gravado" na imagem resultante, usualmente, na posição que corresponde ao *pixel* central da sombra da máscara, ou seja, se o centro da máscara estiver em uma posição (x, y) na

imagem original, o nível de cinza do *pixel* posicionado em (x, y) (na imagem resultante) será substituído por R (eq. 2.2) que é conhecido como a resposta da convolução para aquela vizinhança. Após isso a máscara é movida para a próxima posição de *pixel* na imagem original e o processo se repete. Isso continua até que todas as posições de *pixels*, na imagem original, tenham sido cobertos. No final do processo a imagem original é preservada sem modificações e uma imagem resultante é gerada.

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9 = \sum_{i=1}^9 w_i z_i \quad (2.2)$$

A Figura 2.7 ilustra a convolução: os níveis de cinza de cada *pixel* são substituídos por um valor de resposta, calculado sobre cada vizinhança de cada *pixel*.

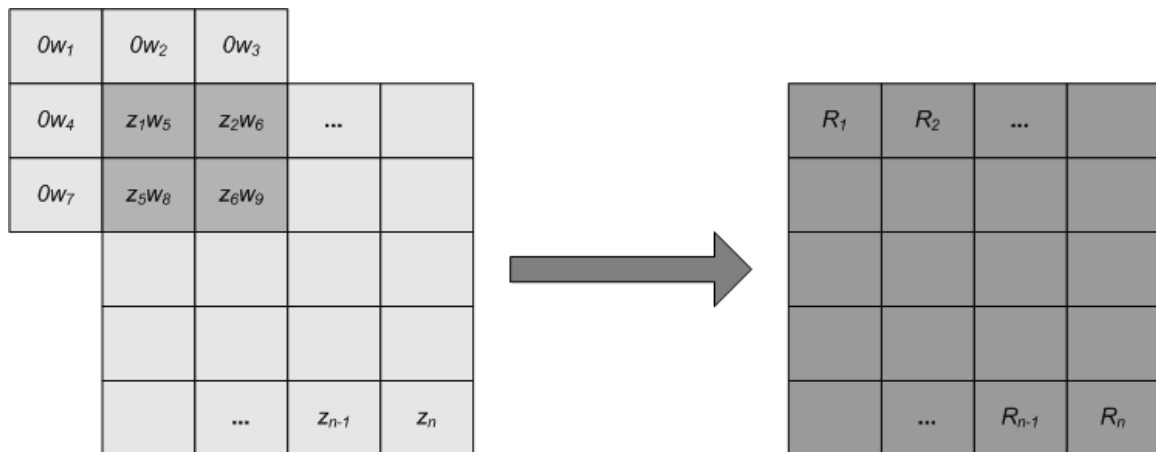


FIGURA 2.7: Convolução sobre uma imagem resulta em outra de mesmas dimensões. No processo ilustrado, para cada nova posição da máscara, apenas o *pixel* central da região afetada recebe o valor da operação aritmética ou resposta da máscara.

2.8.1 Detecção de Pontos Isolados

A detecção de pontos isolados em uma imagem é direta. Usando a máscara mostrada na Figura 2.8, diz-se que um ponto é detectado na localização em que a máscara está centrada se

$$|R| > T \quad (2.3)$$

onde T é um limiar não negativo, e R é dado pela equação (2.2). Basicamente, a aplicação da máscara mede as diferenças de pesos entre o ponto central e seus vizinhos. O nível de cinza de um ponto é bastante diferente dos níveis de cinza de seus vizinhos, de modo que a

simples aplicação da máscara detecta esta diferença.

-1	-1	-1
-1	8	-1
-1	-1	-1

FIGURA 2.8: Máscara usada para detecção de diferentes pontos isolados.

2.8.2 Detecção de Linhas

Da mesma forma que se descobre um ponto isolado, também pode ser detectada uma linha, desde que a máscara de peso usada seja adequada para realçar as partes de uma reta. Utilizam-se as máscaras da Figura 2.9.

<table border="1" style="border-collapse: collapse; width: 60px; height: 60px;"> <tr><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>2</td><td>2</td><td>2</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td></tr> </table>	-1	-1	-1	2	2	2	-1	-1	-1	<table border="1" style="border-collapse: collapse; width: 60px; height: 60px;"> <tr><td>-1</td><td>-1</td><td>2</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>2</td><td>-1</td><td>-1</td></tr> </table>	-1	-1	2	-1	2	-1	2	-1	-1	<table border="1" style="border-collapse: collapse; width: 60px; height: 60px;"> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> </table>	-1	2	-1	-1	2	-1	-1	2	-1	<table border="1" style="border-collapse: collapse; width: 60px; height: 60px;"> <tr><td>2</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>2</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>2</td></tr> </table>	2	-1	-1	-1	2	-1	-1	-1	2
-1	-1	-1																																					
2	2	2																																					
-1	-1	-1																																					
-1	-1	2																																					
-1	2	-1																																					
2	-1	-1																																					
-1	2	-1																																					
-1	2	-1																																					
-1	2	-1																																					
2	-1	-1																																					
-1	2	-1																																					
-1	-1	2																																					
Horizontal	+ 45°	Vertical	- 45°																																				

FIGURA 2.9: Máscaras de linhas.

2.8.3 Detecção de Bordas

Para a detecção de bordas, aplicam-se filtros espaciais lineares de dois tipos: baseados no gradiente da função luminosidade da imagem, e baseados no laplaciano [31].

2.8.3.1 Operador Gradiente

O gradiente de uma imagem $f(x, y)$ na localização (x, y) é o vetor:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Sabe-se da análise vetorial que o vetor gradiente aponta na direção da máxima taxa de variação de f em (x, y) . A magnitude deste vetor é:

$$\|\nabla f\| = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2}$$

Essa magnitude é igual a máxima taxa de crescimento de $f(x, y)$ por unidade de distância na direção de ∇f [18]. A magnitude do gradiente pode ser avaliada por:

$$\|\nabla f\|_\infty = \max(|G_x|, |G_y|)$$

ou

$$\|\nabla f\|_1 = |G_x| + |G_y|$$

o qual é mais simples de se implementar.

Para uma imagem digital $f(x, y)$, a magnitude do gradiente pode ser aproximada por operadores 3x3 ou máscaras de convolução. Na Tabela 2.1 são mostrados alguns importantes exemplos dessas máscaras.

Operador	Vertical ($ G_y $)	Horizontal ($ G_x $)
Roberts	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Sobel	$\frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

TABELA 2.1: Operadores 3x3 utilizados para estimar a magnitude do gradiente através de uma borda

2.8.3.2 Operador Laplaciano

O Laplaciano de uma função 2-D $f(x, y)$ é uma derivada de segunda ordem definida como:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Como no caso do gradiente pode ser aproximado por máscaras 3x3 como mostrado na Figura 2.10 .

0	-1	0
-1	-4	-1
0	-1	0

FIGURA 2.10: Máscara utilizada para computar o Laplaciano.

Embora o laplaciano seja insensível à rotação, e portanto capaz de realçar ou detectar bordas em qualquer direção, seu uso é restrito devido a sua grande suscetibilidade a ruído [31].

2.9 Técnicas Baseadas em Similaridades

2.9.1 Limiarização (*Thresholding*)

A limiarização (*thresholding*) é a técnica mais simples, mais antiga e ainda bastante utilizada. Essa técnica consiste basicamente em subdividir o histograma da imagem em K intervalos disjuntos I_k , $k = 1, 2, \dots, K$, (i.e. K regiões) usando $K - 1$ valores de limiar $T_1 < T_2 \dots < T_{K-1}$ e associar cada *pixel* da imagem com intensidade $f(x, y) \in I_k$ à região k . A Figura 2.11 representa a limiarização aplicada à uma imagem de uma superfície.

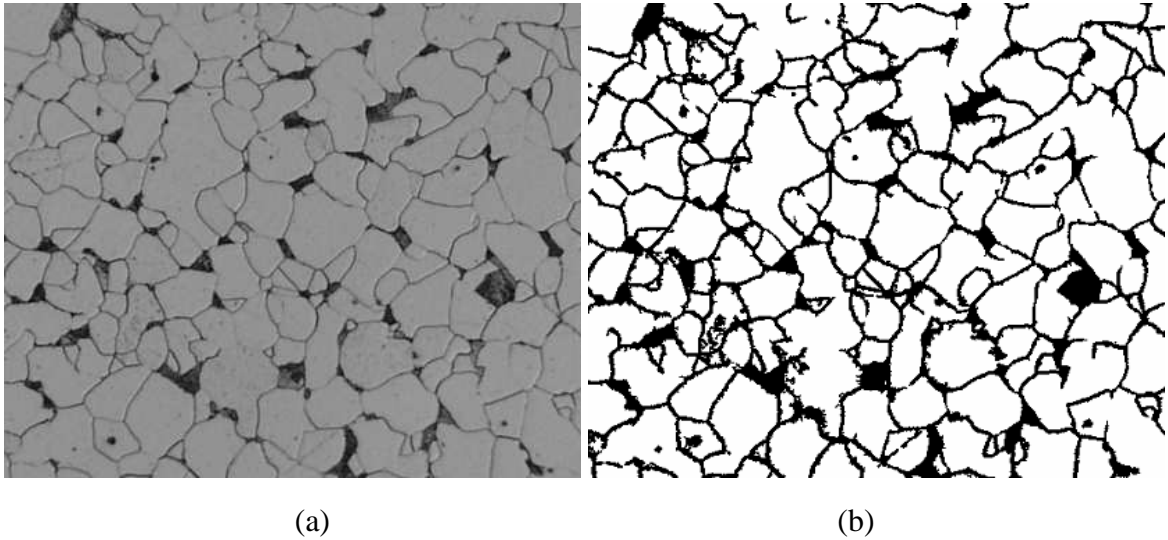


FIGURA 2.11: (a) Imagem de uma superfície. (b) Imagem da superfície limiarizada.

2.9.2 Crescimento de Regiões (*Region Growing*)

Enquanto os métodos de detecção de bordas (Seção 2.8.3) utilizam a diferença de valores de *pixel*, ou seja, descontinuidades no nível de cinza da imagem, algoritmos como o de crescimento de regiões realizam uma busca por grupos de *pixels* com alguma propriedade de similaridade, caracterizando os *pixels* de uma área com nível de cinza similar como pertencentes a uma mesma região.

Inicialmente, o método começa com um *pixel*, e a partir deste *pixel*, examina seus vizinhos, numa seqüência, para decidir se eles possuem níveis de cinza similares, segundo o critério de similaridade escolhido. Se os *pixels* vizinhos analisados forem aceitos como similares, então eles são agrupados ao *pixel* inicial (semente) para formar uma região. Desta forma, as regiões vão sendo desenvolvidas a partir de *pixels* sementes (únicos) [32].

Uma forma melhorada é não começar com *pixels*, mas com uma região da imagem em um conjunto de pequenas regiões. Um teste de uniformidade é, então, aplicado a cada região. Se o teste falhar, a região é subdividida em elementos menores até que todas regiões fiquem uniformes. Então, as regiões se desenvolveriam de regiões menores. A vantagem em usar pequenas regiões ao invés de *pixels* é a redução da sensibilidade ao ruído [18].

2.9.3 Divisão e Conquista (*Split & Merge*)

Supondo que R representa a região que seja a imagem inteira. O processo de segmentação pode ser visto como um processo que particiona R em n sub-regiões, R_1, R_2, \dots, R_n , tal que:

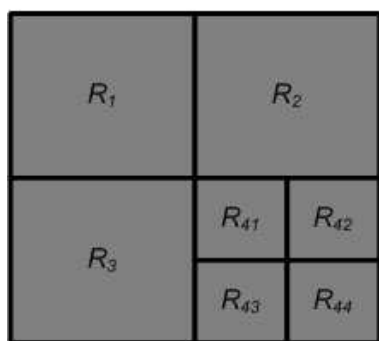
- (a) $\bigcup_{i=1}^n R_i = R$;
- (b) R_i é uma região conectada, $i = 1, 2, 3, \dots, n$;
- (c) $R_i \cap R_j = \emptyset$ para todo i e j , $i \neq j$;
- (d) $P(R_i) = \text{VERDADEIRO}$ para $i = 1, 2, 3, \dots, n$; e
- (e) $P(R_i \cup R_j) = \text{FALSO}$ para $i \neq j$,

onde $P(R_i)$ é um predicado lógico sobre os pontos no conjunto R_i e \emptyset é o conjunto vazio.

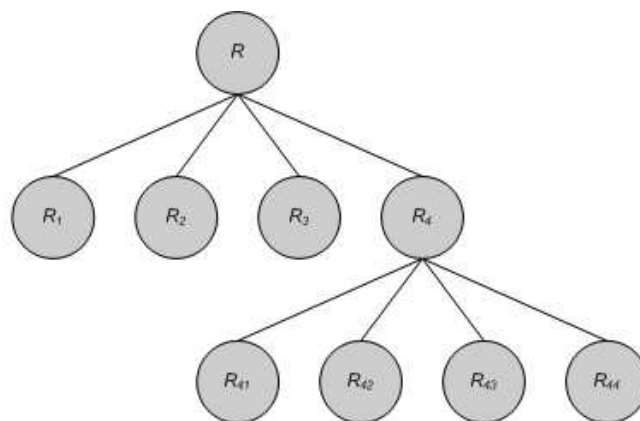
A condição (a) indica que a segmentação deve ser completa, isto é, todo *pixel* deve estar em alguma região. A segunda condição requer que cada região seja conectada. A condição (c) indica que as regiões devem ser disjuntas. A condição (d) lida com as propriedades que devem ser satisfeitas pelos *pixels* numa região segmentada, por exemplo, $P(R_i) = \text{VERDADEIRO}$ se todos os *pixels* em R_i tiverem a mesma intensidade. Finalmente, a condição (e) indica que as regiões R_i e R_j são diferentes no sentido do predicado P .

Para a segmentação de uma imagem uma alternativa é subdividi-la em um conjunto arbitrário de regiões disjuntas e então juntar ou separar essas regiões, de acordo com algum algoritmo em uma tentativa de satisfazer as condições acima discutidas.

Suponha que R representa a imagem inteira e um determinado predicado P . Para uma imagem quadrada, uma maneira de segmentar R é subdividi-la sucessivamente em porções de quadrantes mais e mais pequenos, tal que para alguma região R_i , $P(R_i) = \text{VERDADEIRO}$. Isto é, se $P(R) = \text{FALSO}$, divide-se a imagem em quadrantes. Se $P = \text{FALSO}$ para algum quadrante, subdivide-se aquele quadrante em um subquadrante, e assim por diante. Esta técnica particular de separação tem uma conveniente representação na forma *quadtrees*, como ilustrado na Figura 2.12. Nota-se que a raiz da árvore corresponde à imagem inteira e que cada nó corresponde a uma subdivisão. Neste caso, somente R_4 foi subdividida posteriormente.



(a)



(b)

FIGURA 2.12: (a) Imagem particionada; (b) Quadtree correspondente.

As técnicas de Divisão e Conquista podem ser computacionalmente intensivas se o processo de divisão for iniciado pela imagem inteira. Por outro lado, se o processo começar com a árvore excessivamente dividida (ramificada), tal que as folhas são os pontos da imagem, podem ocorrer erros devido às medidas da uniformidade de uma região estarem baseadas, inicialmente, apenas em pontos vizinhos. Assim, a Divisão e Conquista inicia, normalmente, num estágio intermediário entre estes dois extremos.

Capítulo 3

Algoritmos Genéticos

3.1 Introdução

Algoritmos genéticos (AG's) [33] constituem, em sua forma mais simples, em um método de busca de valores ótimos para funções. Desde que foram criados por John Holland eles têm sido cada vez mais usados para resolver problemas em pesquisas numérica, otimização e aprendizagem de máquina.

Para uma breve introdução ao tema pode-se consultar o artigo de Holland [34]. Para uma descrição mais abrangente pode-se consultar o livro de Goldberg [35]. A maneira mais fácil de entender o funcionamento básico de um algoritmo genético é através de um exemplo de otimização de uma função. A meta consiste em encontrar um valor inteiro, em um certo intervalo, que consiste no valor máximo para uma determinada função.

Um algoritmo genético inicia com uma população de cromossomos de valores aleatórios, que são codificados através de cadeias de *bits*, também chamadas de cromossomos [36, 37]. O algoritmo genético usa esta população inicial para produzir uma nova população que, em média, é "melhor" do que a população anterior. Seja, por exemplo, uma função objetivo $f(x) = 2x$, onde x é um inteiro entre 0 e 31. Uma possível população inicial seria a da Figura 3.1.

0010
1101
1011
1010

FIGURA 3.1: População Inicial.

Os indivíduos com os quais o AG trabalha representam uma solução candidata à resolução do problema em questão. Encontrar a melhor representação para uma solução candidata de acordo com o problema é sempre desejável.

Existem várias maneiras de representar um indivíduo em um AG. A mais simples e comumente utilizada é a representação binária de tamanho fixo, em que um indivíduo é uma cadeia de *bits* que assumem valores 0 ou 1 [38]. Porém, essa representação torna-se problemática quando as variáveis a serem representadas assumem valores contínuos [39].

Em casos em que a representação binária não é a mais natural nem a mais apropriada, outros tipos de representação podem ser utilizados, como sugerido no livro do Michalewicz [33].

É interessante ressaltar que, independente do tipo de codificação utilizada, em um algoritmo genético convencional um indivíduo nunca sabe o significado das informações que ele carrega [40].

Estas cadeias de *bits* correspondem ao genótipo dos indivíduos da população. O genótipo codifica um fenótipo (Tabela 3.1), que neste caso consiste na conversão dos valores binários entre *0000* e *1111* em valores inteiros entre 0 e 31.

Genótipo	Fenótipo
0010	2
1101	13
1011	11
1010	10

TABELA 3.1: Codificações do Genótipo e do Fenótipo.

O grau de adaptação ou aptidão, em inglês o termo é *fitness*, de cada indivíduo é dado pela aplicação da função objetivo ao fenótipo. Como o objetivo é maximizar a função, a aptidão será diretamente proporcional ao valor de $f(x)$. Aplicando a função $2x$ aos fenótipos da população inicial tem-se a Tabela 3.2.

Genótipo	Fenótipo: x	Aptidão: $f(x)$
0010	2	4
1101	13	26
1011	11	22
1010	10	20

TABELA 3.2: Codificações do Genótipo e Fenótipo; e o valor da Aptidão de cada indivíduo.

O segundo elemento, *1101*, representa o valor 26 para a função no ponto $x = 13$ e é o melhor indivíduo desta população. O terceiro e o quarto elementos também são bons indivíduos. Já o primeiro elemento é um indivíduo com fraco grau de adaptação.

A idéia é usar os melhores indivíduos para gerar a próxima população. Isto é, os mais bem adaptados podem deixar descendentes, reproduzindo e/ou recombinação seu material genético, enquanto que os menos adaptados não devem deixar descendentes. Se este processo for bem conduzido, pode-se esperar que a população seguinte seja, em média, melhor do que a população anterior.

3.2 Processos Básicos de AG's

Três processos básicos são usados para gerar uma nova população [41]:

- Seleção
- Cruzamento
- Mutação

O ciclo básico do algoritmo genético é mostrado na Figura 3.2.

O primeiro passo é o processo de seleção. Ele determina quais indivíduos da população velha serão usados para gerar os indivíduos da população nova. Esta seleção é feita por uma roleta ponderada. Ou seja, os indivíduos que vão se reproduzir são sorteados na população original. Mas aqueles que têm um valor de aptidão (*fitness*) melhor têm maiores chances de serem selecionados do que os que têm valor de aptidão menor. No exemplo da Tabela 3.2 a cadeia *1101* tem a maior chance de ser selecionada, enquanto que *0010* tem a menor chance.

O segundo passo é o processo de cruzamento. O cruzamento usa um par de indivíduos selecionados na etapa anterior. Estes indivíduos serão os pais de dois indivíduos da po-

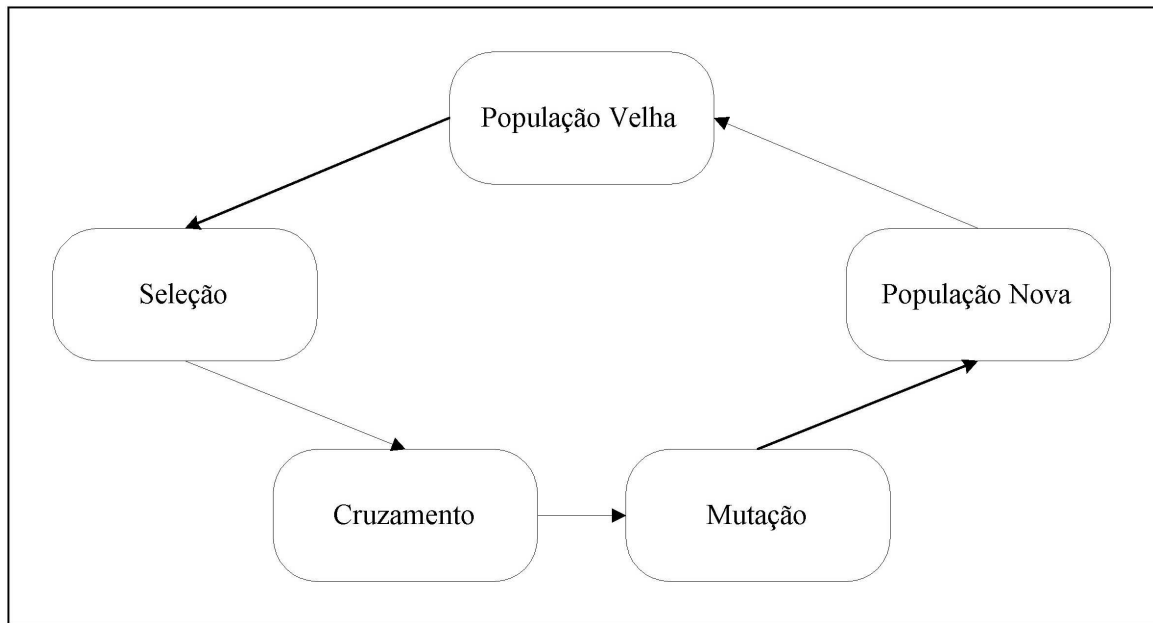


FIGURA 3.2: Ciclo b sico do Algoritmo Gen tico.

pula o nova. O cruzamento ocorre com uma probabilidade pc . Se para um determinado par selecionado n o ocorrer o cruzamento, ent o o par   simplesmente copiado na pr xima popula o. Caso ocorra o cruzamento ent o haver  um sorteio do ponto de corte dos cromossomos. Se o comprimento da cadeia de *bits* do cromossomo de cada um dos pais   r , ent o um n mero aleat rio s entre 1 e $r-1$ ser  selecionado. O processo de cruzamento consiste em formar dois novos indiv duos. O gen tipo do primeiro indiv duo ser  formado pelos s primeiros *bits* do primeiro pai e os $r - s$  ltimos *bits* do segundo pai. O gen tipo do segundo indiv duo ser  formado pelos r primeiros *bits* do segundo pai e os $r - s$  ltimos *bits* do primeiro pai. A Figura 3.3 ilustra esta situa o.

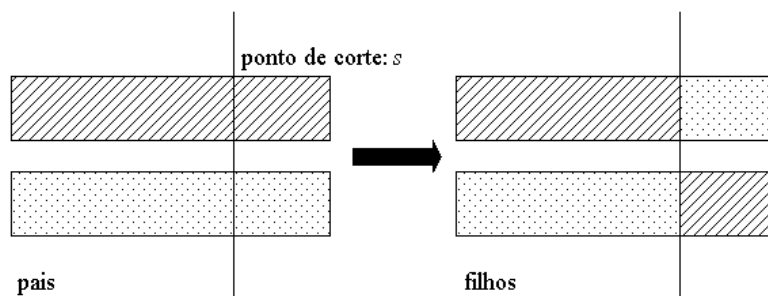


FIGURA 3.3: Processo de Cruzamento.

O  ltimo passo do processo gen tico   a muta o. Uma probabilidade de muta o pm muito pequena deve ser definida. Os *bits* de todas as cadeias da nova popula o est o sujeitos a esta probabilidade de muta o. Evidentemente, um *bit* s  pode mudar de 1 para 0 ou de 0

para 1. A Figura 3.4 ilustra a ocorrência de duas mutações numa cadeia, uma no *bit 6* e outra no *bit 9*.

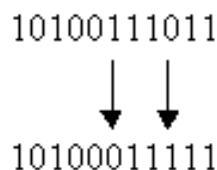


FIGURA 3.4: Exemplo de Mutação.

Os três passos, descritos acima, são repetidos para criar novas gerações até que alguma condição de parada seja obtida. Esta condição pode ser, por exemplo, atingir um número máximo de gerações. Este algoritmo tem provado ser muito efetivo para resolver vários problemas de pesquisa numérica. Embora seja uma espécie de pesquisa aleatória, o ganho qualitativo em cada geração indica que o algoritmo efetua uma pesquisa bem direcionada. Para ilustrar o processo de geração de uma nova população, vamos utilizar como população inicial o exemplo da Tabela 3.2. Define-se uma probabilidade de mutação $pm = 0.05$ e uma probabilidade de cruzamento $pc = 0.7$. O primeiro passo é a seleção de dois casais para cruzamento. O somatório das aptidões da população é igual a 72. Portanto, cada indivíduo tem a probabilidade de ser sorteado igual a sua aptidão dividido pelo somatório das aptidões.

Genótipo	Fenótipo: x	Aptidão: $f(x)$	Probabilidade de Sorteio(%)
0010	2	4	5.56
1101	13	26	36.10
1011	11	22	30.56
1010	10	20	27.78

TABELA 3.3: Codificações do Genótipo e Fenótipo; o valor da Aptidão e a Probabilidade de ser sorteado cada indivíduo.

A roleta (Figura 3.5) é montada a partir da probabilidade de sorteio de cada indivíduo da população.

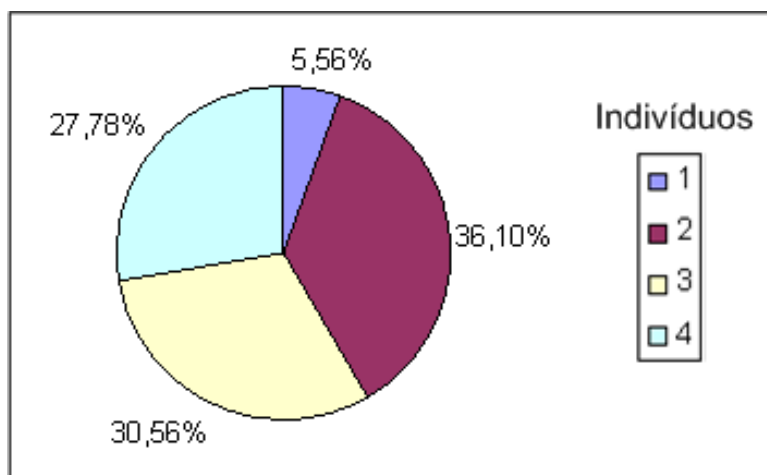


FIGURA 3.5: Exemplificação da roleta com as probabilidades de seleção dos 4 indivíduos.

Suponha-se que o sorteio ponderado através da roleta tenha selecionado os seguintes casais:

Casal 1: *1101* e *1011*

Casal 2: *1010* e *1101*

Deve-se observar que o indivíduo *1101* aparece em dois casais distintos. Mas não há nenhum problema quanto a isso. Já o indivíduo *0010* não aparece em nenhum casal. Isto ocorre porque ele é um indivíduo fraco com poucas chances de se reproduzir. Supondo que para o primeiro casal não haja cruzamento, então os indivíduos *1101* e *1011* são simplesmente copiados para a população nova. Para o segundo casal, suponha-se que houve cruzamento. Então deve ser selecionado um ponto de corte, que consiste em um valor inteiro entre 1 e 3. Se este ponto for 2, por exemplo, então o cruzamento ocorrerá conforme mostrado na Figura 3.6.

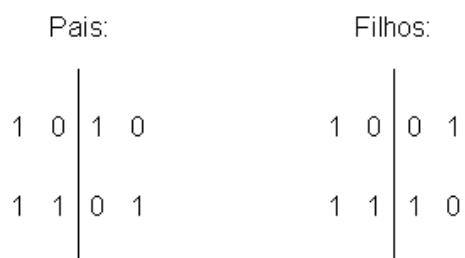


FIGURA 3.6: Cruzamento com ponto de corte igual a 2.

A nova população consiste, por enquanto, dos indivíduos gerados a partir do primeiro e do segundo casal. Ou seja:

Filhos do primeiro casal: *1101* e *1011*

Filhos do segundo casal: *1001* e *1110*

O passo seguinte é a mutação, que pode ser definida, por exemplo, com probabilidade de 5%. Isto significa que de cada 100 *bits* da nova população, em média 5 *bits* mudarão de 0 para 1 ou de 1 para 0. Suponha que apenas o terceiro *bit* do terceiro indivíduo sofra mutação. Este indivíduo, que originalmente era *1001* passa a ser *1011*. A população final fica então:

Genótipo	Fenótipo: x	Aptidão: $f(x)$
1101	13	26
1011	11	22
1011	11	22
1110	14	28

TABELA 3.4: Codificações do Genótipo e Fenótipo; e o valor da Aptidão de cada indivíduo da nova população.

Pode-se observar que a média das aptidões desta população nova é bem superior à média da população anterior. Enquanto a população anterior tinha aptidão média de 18, a nova tem 24.5.

Eventualmente, a aptidão média pode piorar de uma população a outra. Mas, com o passar das gerações ela tende a um ótimo.

Capítulo 4

Algoritmo de Segmentação de Imagem baseado em Algoritmo Genético

4.1 Introdução

A segmentação da imagem é feita pela rotulagem dos *pixels* pertencentes a uma mesma região, aonde o critério de classificação para que o *pixel* pertença a uma determinada região é: o nível de cinza ou uma cor comum; ou uma textura em comum. Tradicionalmente, a segmentação da imagem envolve a rotulagem de cada *pixel* com o rótulo da região. Este problema de segmentação, a rotulagem dos *pixels*, já foi abordado por vários autores com diferentes perspectivas. Jianbo e Malik [42] resolveram o problema da segmentação da imagem 2D como um problema de otimização global. A idéia principal é solucionar o problema como um problema de particionamento de grafo. Lo Bosco [17] propõe a utilização do Algoritmo Genético (A.G.) na estratégia de busca que nos permite encontrar soluções aproximadas satisfatórias quando tratamos o problema com técnicas de otimização global. Existem outras propostas de utilização do algoritmo genético para a segmentação de imagens [43, 44, 45, 46] em diversas áreas, como por exemplo, nas áreas: médica para tomografias [47, 48] e engenharia civil para análise de superfícies [49].

Esta tese apresenta um algoritmo de segmentação de imagens que se baseia nos valores de níveis de cinza e na posição espacial dos *pixels* abordado por Lo Bosco [17]. Além disso foi modificado a codificação do cromossomo original do artigo e desenvolvido dois métodos de seleção dos pais para o cruzamento, em cada método há duas variações de seleções, no

primeiro método a escolha dos pais é aleatória e no segundo o conceito de vizinhança é considerado.

4.2 Implementação do Algoritmo de Segmentação

A solução do problema de segmentação não é única e está relacionada à característica espacial e ao propósito da segmentação; nesta tese a intensidade e a posição espacial do *pixel* [1] são utilizados.

A evolução do A.G. será conduzida pela similaridade que será definida entre os elementos de I , sendo que I representa todos os *pixels* da imagem.

Dada uma imagem de entrada, I , de tamanho $n \times m$, $I(x, y)$ irá indicar o valor do nível de cinza do *pixel* localizado na posição (x, y) .

O algoritmo é formado pela estrutura exemplificada na Figura 4.1, esse algoritmo contém as seguintes rotinas:

- Leitura e armazenamento dos dados.
- Geração e codificação dos rótulos.
- Cálculo da vizinhança.
- Cálculo do valor médio.
- Cálculo da função de desempenho.
- Seleção.
- Aplicação do cruzamento.
- Aplicação da mutação.
- Geração da população final.
- Critério de parada da segmentação.

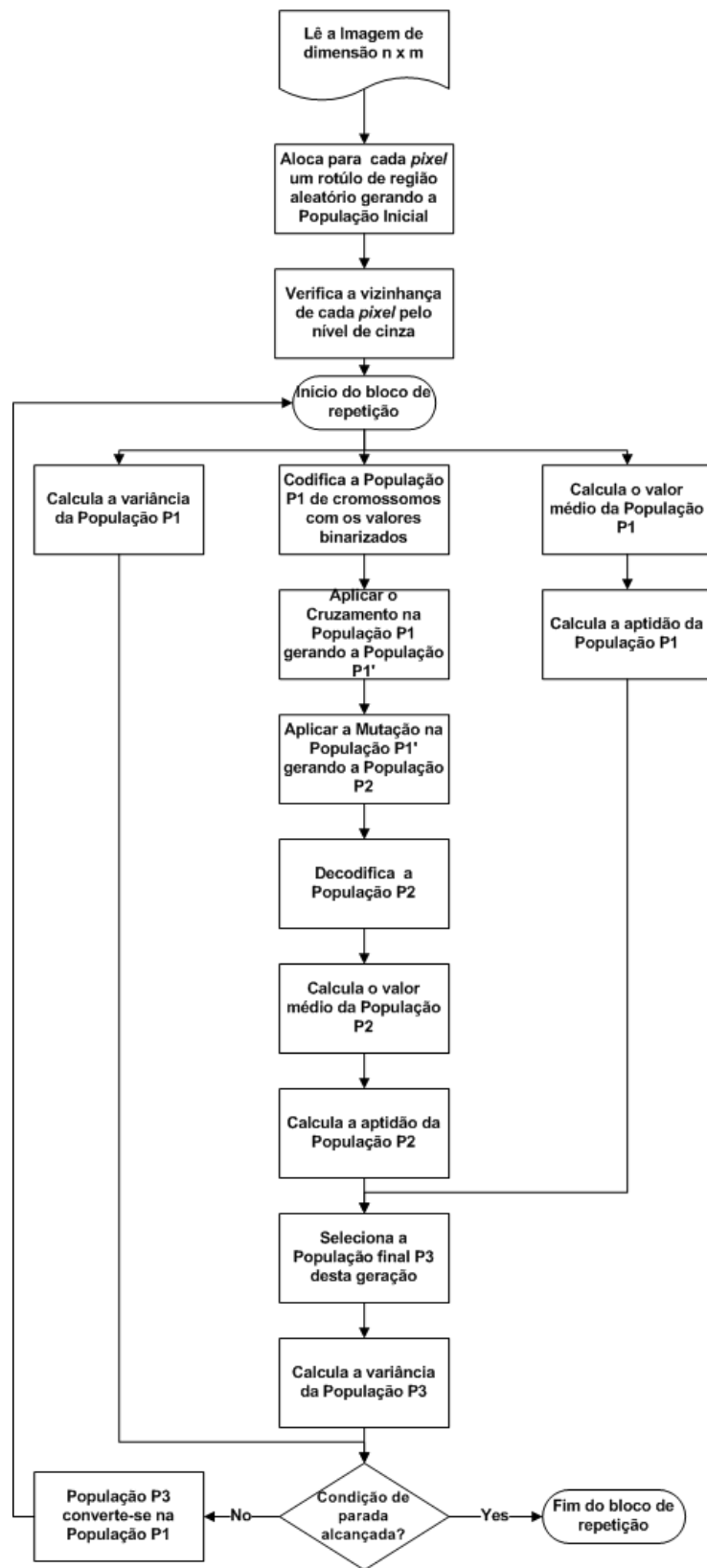


FIGURA 4.1: Algoritmo de Segmentação Genético.

4.2.1 Leitura e Armazenamento dos Dados

Inicialmente é pré-determinado: o número de partições (K), os percentuais de cruzamento (pc) e mutação (pm).

A imagem é armazenada em forma de matriz de duas dimensões denominada *matriz imagem_original*, contendo em cada coordenada uma estrutura de informações sobre o *pixel* pertencente a esta coordenada. A função *armazena_matriz* lê e armazena essas informações em uma estrutura (*matriz_imagem*) que contém:

- nível de cinza \Rightarrow intensidade luminosa que vai de 0 a 255.
- *pixels* vizinhos \Rightarrow indicação dos *pixels* que são vizinhos do *pixel* analisado, será explicado com mais detalhe na seção 4.2.3.
- rótulo1 \Rightarrow valor referente à região que o *pixel* pertence na População 1;
- rótulo2 \Rightarrow valor referente à região que o *pixel* pertence na População 2;
- aptidão1 \Rightarrow valor referente à aptidão do *pixel* na População 1;
- aptidão2 \Rightarrow valor referente à aptidão do *pixel* na População 2;
- codificação binária do rótulo1 \Rightarrow valor do rótulo1 codificado para binário;
- codificação binária do rótulo2 \Rightarrow valor do rótulo2 codificado para binário.

Podemos observar que na estrutura de armazenamento cada *pixel* armazena dois valores de regiões (*rótulo1* e *rótulo2*), o *rótulo1* recebe o valor da região deste *pixel* na População 1, e o *rótulo2* recebe o valor da região deste *pixel* na População 2, indicando que este *pixel* já sofreu cruzamento e mutação. Os dois valores de aptidão também se referem ao mesmo *pixel* em cada População.

O tamanho da população é definido pela multiplicação do número de linhas (n) pelo o número de colunas (m) da imagem.

4.2.2 Geração e codificação dos rótulos

A função *gera_rotulos* provê uma seleção aleatória de seqüências do espaço de representação. Cada *pixel* recebe um valor de rótulo (k) dentro do intervalo $[0, K]$, sendo K o número máximo de regiões na imagem. Cada rótulo é codificado, formando um indivíduo da *População Inicial (PI)*.

No cromossomo original constava os valores da coordenada e o rótulo *pixel*, mas após a aplicação do cruzamento e da mutação foi observado que alguns *pixels* são eliminados da população. Esses *pixels* não poderiam ser eliminados pois fazem parte da imagem e são necessários para a segmentação. O novo cromossomo armazena somente o rótulo, e este rótulo está associado a um determinado *pixel*. Na Figura 4.2 podemos observar a alteração do cromossomo original para o cromossomo utilizado.

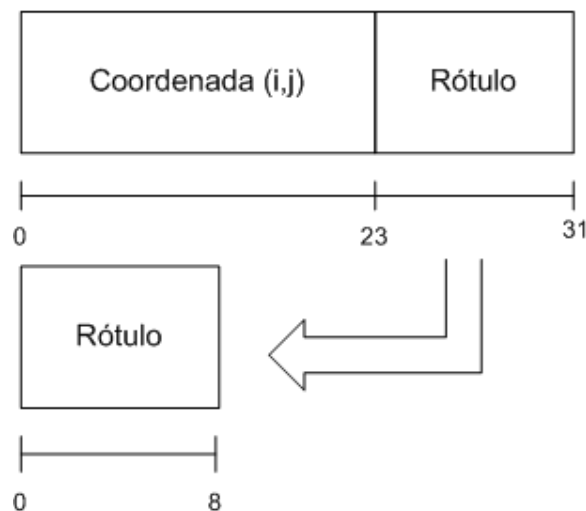


FIGURA 4.2: Cromossomo com 8 bits utilizado para armazenar o rótulo.

Nesta implementação o valor máximo de regiões que poderá ser utilizado na segmentação é igual a 256, que se refere ao valor máximo de níveis de cinza (0 – 255). Por este motivo é utilizado a codificação de 8 bits (2^8) para o rótulo.

4.2.3 Cálculo da vizinhança

A estrutura da vizinhança de um *pixel* é formada por 8 *pixels*, como pode ser observado na Figura 4.3, sendo que x indica a linha e y a coluna. A função *calc_viz* calcula a vizinhança de cada *pixel* retornando os níveis de cinza de cada vizinho do *pixel* analisado.

O armazenamento é feito do seguinte modo: é criado um vetor com tamanho 8 e cada posição representa um vizinho.

$$vizinho[1] = \text{direita } (x, y + 1);$$

$$vizinho[2] = \text{direita-superior } (x - 1, y + 1);$$

$$vizinho[3] = \text{superior } (x - 1, y);$$

$$vizinho[4] = \text{esquerda-superior } (x - 1, y - 1);$$

$$vizinho[5] = \text{esquerda } (x, y - 1);$$

$$vizinho[6] = \text{esquerda-inferior } (x + 1, y - 1);$$

$$vizinho[7] = \text{inferior } (x + 1, y);$$

$$vizinho[8] = \text{direita-inferior } (x + 1, y + 1).$$

Para representar uma determinada posição utilizamos a variável i no intervalo de 1 a 8, por exemplo, o $vizinho[2]$ representa o *pixel* vizinho que se encontra na posição direita-superior, ou $(x - 1, y + 1)$ como na Figura 4.3.

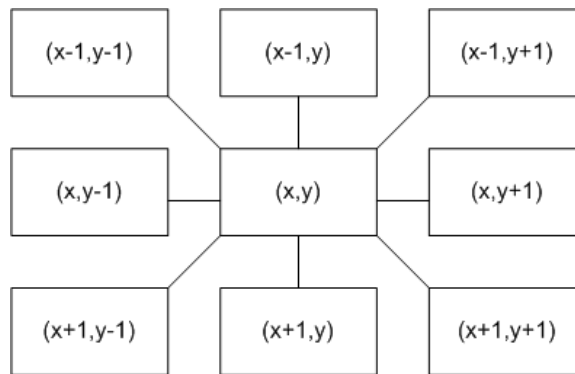


FIGURA 4.3: Vizinhança com 8 conexões.

Utilizando como exemplo, o *pixel* que está na coordenada $(0, 0)$ (Figura 4.4), iremos analisar como ficará a sua vizinhança.

$$vizinho[1] = \text{nível de cinza do } pixel I(0, 1);$$

$$vizinho[2] = 0;$$

$$vizinho[3] = 0;$$

$$vizinho[4] = 0;$$

$$vizinho[5] = 0$$

$$vizinho[6] = 0;$$

$$vizinho[7] = \text{nível de cinza do } pixel I(1, 0);$$

$$vizinho[8] = \text{nível de cinza do } pixel I(1, 1);$$

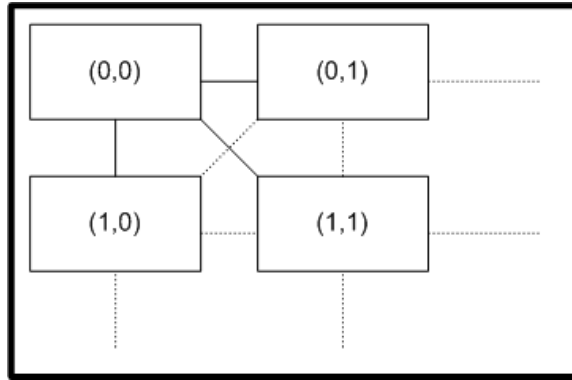


FIGURA 4.4: Exemplo da vizinhança do *pixel* que se encontra na coordenada $(0, 0)$.

Como o *pixel* $(0, 0)$ se encontra na borda da imagem, podemos observar que alguns vizinhos receberam o valor zero por não existirem.

4.2.4 Cálculo do valor médio

Cada região P_k é caracterizada pelo valor médio (mv_k) dos níveis de cinza.

$$mv_k = \frac{\sum_{p \in P_k} I(p)}{|P_k|}, \quad (4.1)$$

sendo que:

p - representa o *pixel* que está sendo analisado;

$I(p)$ - representa o nível de cinza do *pixel* que está sendo analisado;

P_k - representa o número de *pixels* que pertencem a região k .

4.2.5 Cálculo da função de desempenho

A função aptidão f (eq. 4.2) foi definida baseada na função de similaridade ρ (eq. 4.3).

$$f(\underline{p}) = \rho(\underline{p}, mv_{k_p}) \quad (4.2)$$

$$\rho(\underline{p}, mv_{k_p}) = a \times d_g(I(p), mv_{k_p}) + b \times d_E(\underline{p}, mv_{k_p}) \quad (4.3)$$

onde,

\underline{p} são os *pixels* da população.

a é o peso referente a intensidade do *pixel*, $a \geq 0$;

b é o peso referente a posição espacial do *pixel*, $b \geq 0$.

Sendo que $a + b = 1$.

Nos testes o valor utilizado para a e b foi igual a 0,5, não foram utilizados outros valores. A função d_g representa a distância entre o nível de cinza de um determinado *pixel* com a média de níveis de cinza da região que ele pertence (eq. 4.4).

$$d_g(I(\underline{p}), mv_{k_{\underline{p}}}) = \frac{|I(\underline{p}) - mv_{k_{\underline{p}}}|}{\max(I(\underline{p}), mv_{k_{\underline{p}}})} \quad (4.4)$$

A função d_E representa a distância euclidiana do *pixel*, isto é, onde, $V(\underline{p})$ é a vizinhança do *pixel* p e \underline{z} representa o nível de cinza de cada vizinho do *pixel* p .

$$d_E(\underline{p}, mv_{k_{\underline{p}}}) = \sum_{z \in V(\underline{p})} \frac{|I(\underline{z}) - mv_{k_{\underline{p}}}|}{\max(I(\underline{z}), mv_{k_{\underline{p}}})} \quad (4.5)$$

Em palavras, podemos dizer que a função de similaridade pondera (em combinação convexa) a similaridade em intensidade, bem como a proximidade espacial.

4.2.6 Seleção

Nesta tese foi implementado dois tipos de seleção:

- Seleção por Aleatoriedade:
 - Seleção tipo 1.
 - Seleção tipo 2.
- Seleção por Vizinhança:
 - Seleção tipo 3.
 - Seleção tipo 4.

4.2.6.1 Seleção por Aleatoriedade

Seleção tipo 1

Na seleção tipo 1 o pa_i^1 é selecionado seqüencialmente a partir de um módulo de repetição que abrange o primeiro indivíduo até o último indivíduo da população. O pa_i^2

é selecionado aleatoriamente dentro do mesmo intervalo que o pai^1 é selecionado (Figura 4.5).

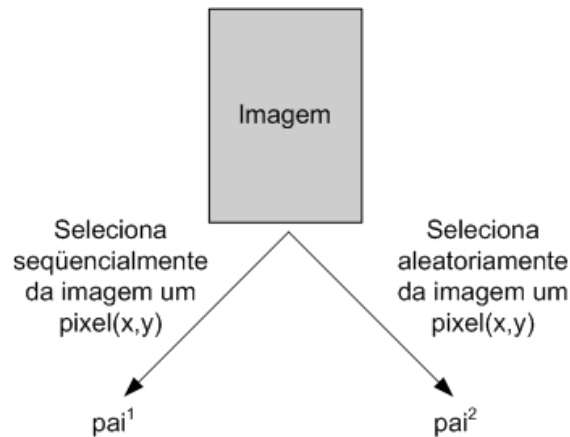


FIGURA 4.5: Seleção do tipo 1 para a escolha dos indivíduos.

Na Tabela 4.1 podemos observar como são escolhidos os indivíduos para o cruzamento. No primeiro casal o primeiro indivíduo a ser escolhido seqüencialmente é o *pixel* que está na coordenada (0, 0) e o segundo indivíduo é selecionado aleatoriamente da população.

Primeiro Casal:	Segundo Casal:
pai^1 - Seqüencial	pai^1 - Seqüencial
$pai^1 = p(0, 0)$	$pai^1 = p(0, 1)$
pai^2 - Aleatória	pai^2 - Aleatória
$pai^2 = p(8, 2)$	$pai^2 = p(3, 6)$

TABELA 4.1: Exemplo da seleção do tipo 1.

Seleção tipo 2

Na seleção tipo 2 os pais são selecionados utilizando o método de seleção da roleta, ou seja, cada pai é sorteado aleatoriamente de uma roleta ponderada que é montada a partir da aptidão dos indivíduos (Tabela 4.2).

Primeiro Casal:	Segundo Casal:
pai^1 - Roleta	pai^1 - Roleta
$pai^1 = p(3, 2)$	$pai^1 = p(2, 10)$
pai^2 - Roleta	pai^2 - Roleta
$pai^2 = p(4, 8)$	$pai^2 = p(9, 5)$

TABELA 4.2: Exemplo da seleção do tipo 2.

4.2.6.2 Seleção por Vizinança

Seleção tipo 3

Na seleção tipo 3 o pai^1 é selecionado seqüencialmente a partir de um módulo de repetição que abrange o primeiro indivíduo até o último indivíduo da população. O pai^2 é selecionado a partir de uma roleta que contém somente os vizinhos do pai^1 . Por exemplo, caso o pai^1 seja o *pixel* que está na coordenada (0, 0), então o pai^2 poderá ser um dos *pixels* que estão nas coordenadas a seguir: (0, 1), (1, 0) e (1, 1) (Tabela 4.3).

Primeiro Casal:	Segundo Casal:
pai^1 - Seqüencial	pai^1 - Seqüencial
$pai^1 = p(0, 0)$	$pai^1 = p(0, 1)$
pai^2 - Roleta com os vizinhos de $p(0, 0)$	pai^2 - Roleta com os vizinhos de $p(0, 1)$
$pai^2 = p(1, 0)$	$pai^2 = p(0, 2)$

TABELA 4.3: Exemplo da seleção do tipo 3.

Seleção tipo 4

A seleção tipo 4 é semelhante a seleção tipo 3. A diferença é que o pai^1 é selecionado igual ao pai^2 , ou seja, cada pai é representado por um dos vizinhos do *pixel* que está sendo analisado (Tabela 4.4).

Primeiro Casal:	Segundo Casal:
$pai^1 - pixel_1 = (0,0)$ Roleta com os vizinhos de $p(0, 0)$	$pai^1 - pixel_3 = (0,2)$ Roleta com os vizinhos de $p(0, 2)$
$pai^1 = p(1, 0)$	$pai^1 = p(1, 1)$
$pai^2 - pixel_2 = (0,1)$ Roleta com os vizinhos de $p(0, 1)$	$pai^2 - pixel_4 = (0,3)$ Roleta com os vizinhos de $p(0, 3)$
$pai^2 = p(0, 2)$	$pai^2 = p(1, 4)$

TABELA 4.4: Exemplo da seleção do tipo 4.

4.2.7 Aplicação do cruzamento

Na implementação do cruzamento foram utilizados dois tipos de cromossomos, gerando dois algoritmos com algumas diferenças. O cromossomo original é composto pela coordenada do *pixel* e o rótulo da região, e o cromossomo modificado somente contém o rótulo. No cruzamento é utilizado um par de pais, pai^1 e pai^2 , para gerar um par de filhos. A análise do percentual de cruzamento é igual a $pc = 0,7$, ocorrerá o cruzamento caso esse percentual seja menor do que o gerado aleatoriamente para cada cruzamento. Após a aplicação do cruzamento de um ponto de corte aleatório nos pais selecionados, são gerados dois filhos que serão os novos indivíduos da População Intermediária ($P1'$).

4.2.8 Aplicação da mutação

A aplicação da mutação será feita em cada indivíduo da População Intermediária $P1'$ se o percentual de mutação ($pm = 0,01$) for menor do que o valor gerado aleatoriamente para cada *bit* de um cromossomo. A mutação modificará somente um ou vários *bits* do cromossomo selecionado, gerando a População Intermediária $P2$.

4.2.9 Geração da população final

A População Final $P3$ é determinada por um processo de seleção entre a População Inicial $P1$ e a População Intermediária $P2$, todas com o mesmo tamanho *linha* \times *coluna* ($n \times m$). Sendo β_r o indivíduo da População Intermediária $P2$ e α_r o indivíduo da População Inicial $P1$, representando o mesmo *pixel*.

$$P3 = \{\gamma_1, \gamma_2, \dots, \gamma_{n*m}\}$$

Tal que

$$\gamma_r = \begin{cases} \beta_r & \text{se } f(\beta_r) < f(\alpha_r); \\ \text{senão } \alpha_r. \end{cases}$$

Sendo f a função de aptidão, o melhor indivíduo é selecionado a partir do menor valor de aptidão.

4.2.10 Critério de Parada

Os operadores genéticos e o processo de seleção são aplicados até que o critério de parada (eq. 4.6) seja alcançado, sendo ele baseado na convergência da variância total ($Var_t = \sum_k^K \sigma_t(k)$).

$$\text{se } |Var_{P1} - Var_{P3}| \leq \phi \quad (4.6)$$

onde $\sigma_t(k)$ representa a variância interna do grupamento k na População que está sendo calculada a variância; e $\phi \geq 0$. A condição $\phi = 0$ não é normalmente alcançada e o valor de ϕ é determinado por pela equação 4.7.

$$\phi \approx \sqrt{\min(Var_{P1}, Var_{P3})} \quad (4.7)$$

A variância (eq. 4.8) é definida como o desvio quadrático médio da média e é calculada de uma amostra de dados como:

$$var^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1} \quad (4.8)$$

Tal que, se x_1, x_2, \dots, x_n são valores dos dados de uma determinada amostra e n o número de elementos desse amostra, assim podemos escrever a média como:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{\sum_{i=1}^n x_i}{n} \quad (4.9)$$

Capítulo 5

Resultados do Algoritmo Proposto

As imagens utilizadas são do formato *PGM* (*Portable GrayMap*). Segundo este formato, o arquivo contém um cabeçalho e a matriz correspondente à imagem (Figura 5.1). A primeira linha do arquivo contém uma palavra-chave *P2* que é obrigatória. A terceira linha contém dois números que correspondem ao número de linhas e colunas da matriz. A quarta linha contém um número que é o maior número da matriz, no caso das imagens com níveis de cinza o valor correspondente é 255. As demais linhas do arquivo correspondem às linhas da matriz (imagem). Esse formato foi utilizado por não necessitar de nenhuma ferramenta adicional para lê-lo ou escrevê-lo. Algumas imagens originalmente coloridas tiveram que ser transformadas para imagens com tons de cinza, mas essa transformação não alterou a qualidade da imagem.

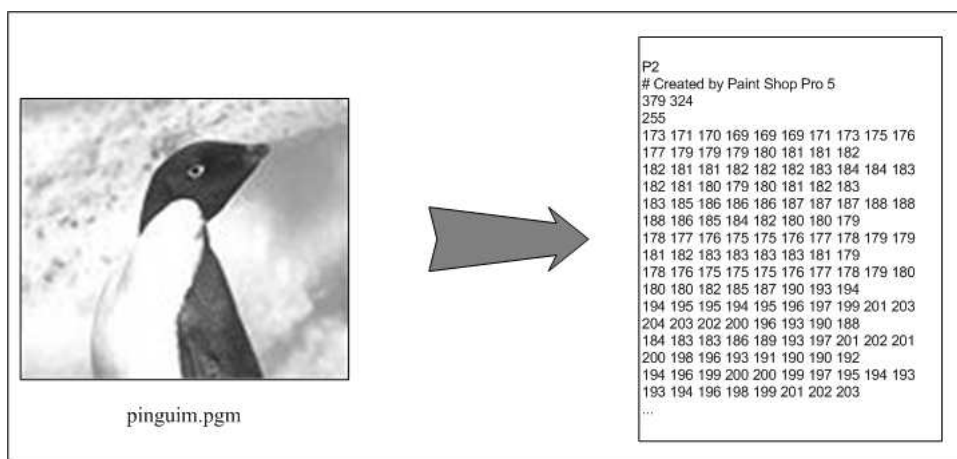


FIGURA 5.1: Imagem do pinguim e o arquivo da imagem aberto em um editor de texto.

O objetivo da segmentação é particionar a imagem em regiões homogêneas. Na utilização apenas da técnica de limiarização os resultados deixam restos de informações desnecessárias ou perdem informações. Para superar estas limitações no domínio do valor são utilizados alguns métodos que operam no domínio do espaço e utilizam informações sobre: vizinhança do *pixel*, variação do valor do *pixel* em uma vizinhança, estatísticas sobre agrupamento de *pixels*, ou neste trabalho, baseado na intensidade e na posição espacial do *pixel* para gerar uma imagem resultante dividida em regiões.

Para avaliar a metodologia estudada foram utilizadas imagens de diferentes tipos:

- Humana: Lena, e de uma criança;
- Animal: pingüim;
- Mapa: cidade do Rio de Janeiro e Paris.

5.1 Resultados

A segmentação foi realizada por detecção de regiões pelo algoritmo descrito no Capítulo 4, que utiliza a similaridade entre os *pixels* para encontrar regiões diferentes. Não foi aplicado nenhum tipo de filtragem ou pré-processamento das imagens estudadas. Nas imagens foram aplicadas a segmentação com os dois tipos de seleção (seção 4.2.6), com o cromossomo contendo a codificação do rótulo da região que o *pixel* pertence e nos testes foram utilizados o número de rótulos igual a dois, que indica que a imagem será segmentada em duas regiões. Para o cruzamento o percentual utilizado foi de $pc = 0,7$ e para a mutação o percentual foi de $pm = 0,01$. Os testes de segmentação foram executados em uma máquina com processador intel pentium III de 1GHz com 256Mb de memória RAM.

Ao longo deste capítulo serão exibidas as imagens resultantes da segmentação e os gráficos da variância e da aptidão da população final de cada geração. O gráfico da variância representa a variância de cada população final em cada geração e o gráfico da aptidão representa a média das aptidões dos indivíduos da população final em cada geração. O valor da aptidão utilizado para calcular a média é o valor resultante da aplicação da função em cada indivíduo da população.

5.1.1 Imagens Humanas

A primeira imagem a ser analisada é a imagem da Lena por ser uma imagem quase obrigatória na área de processamento de imagens (Figura 5.2(a)). A Figura 5.2(b) exibe a aplicação da detecção de borda na imagem original da Lena.

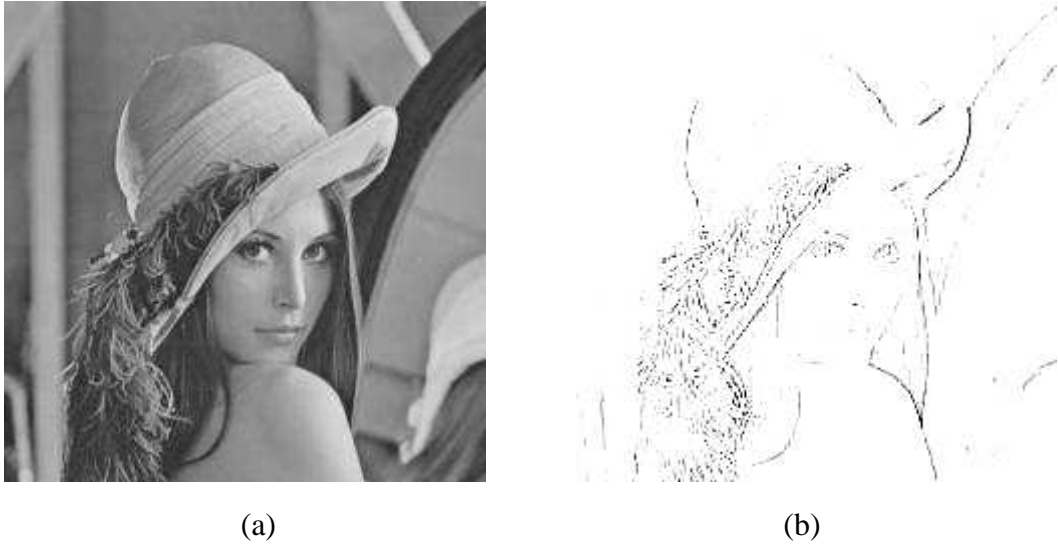


FIGURA 5.2: (a) Imagem original da Lena. (b) Imagem com contorno aplicado.

A partir da imagem original (Figura 5.2(a)) foi aplicada a limiarização. Como podemos observar a imagem após a limiarização (Figura 5.3), resulta na ausência de algumas características importantes na imagem.



FIGURA 5.3: Imagem segmentada por limiarização.

Na Figura 5.4 podemos observar a segmentação da imagem da Lena com os quatro tipos de seleção. As Figuras 5.4 (a) e (b) representam a segmentação utilizando a seleção por aleatoriedade tipo 1 e tipo 2 respectivamente. Nas Figuras 5.4 (a) e (b) os pares de pais são escolhidos sem considerar a aptidão dos indivíduos. Nesse tipo de seleção todos os *pixels* são escolhidos para o cruzamento. Na Figura 5.4(a) alguns detalhes foram realçados, como os olhos, a borda do espelho e o detalhe do chapéu. Na Figura 5.4(b) os detalhes dos objetos na imagem não foram muito bem caracterizados.

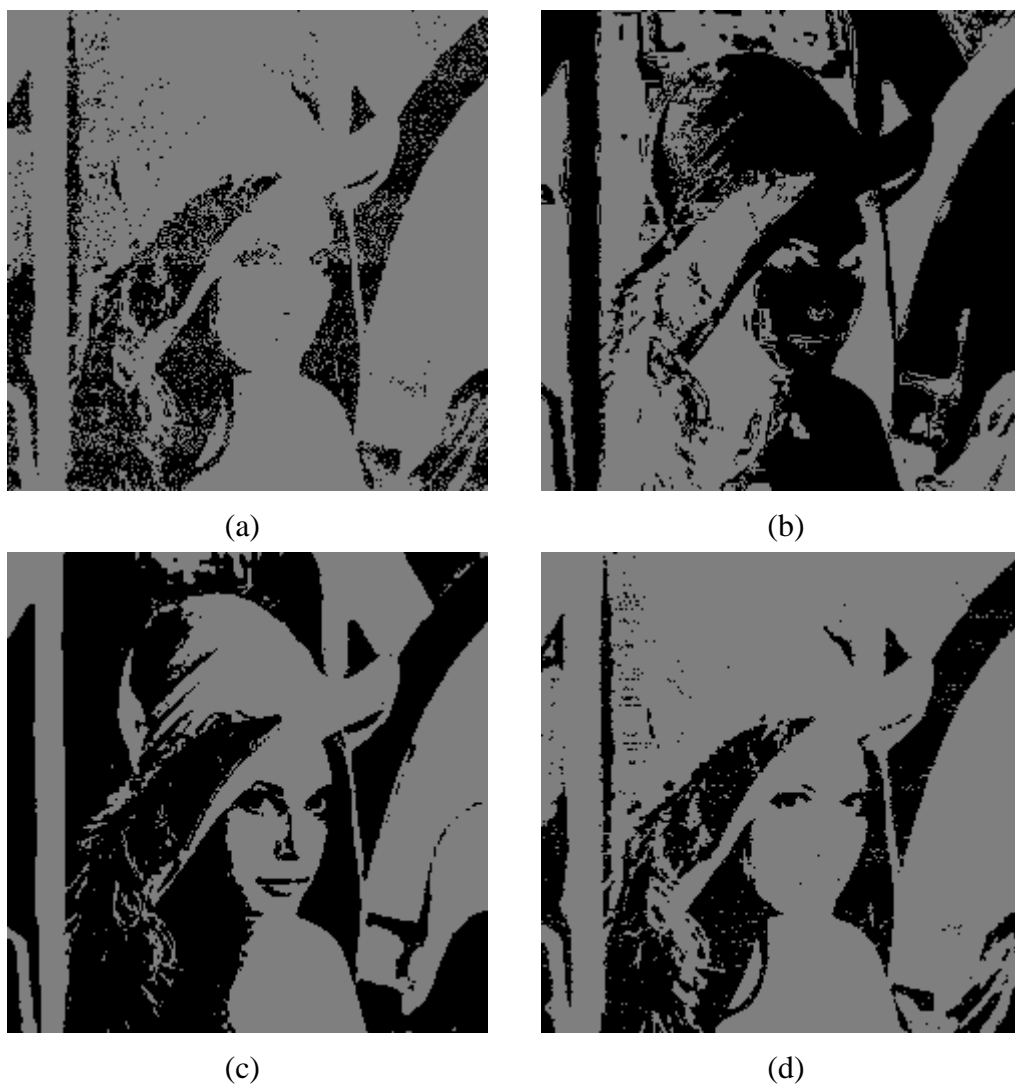


FIGURA 5.4: Segmentação com Seleção por Aleatoriedade: (a) Tipo 1 (b) Tipo 2 e Segmentação com Seleção por Vizinhança: (c) Tipo 3 (d) Tipo 4.

As Figuras 5.4 (c) e (d) representam a segmentação utilizando a seleção por vizinhança tipo 3 e tipo 4 respectivamente. Na Figura 5.4(c) os pares de pais são escolhidos de acordo

com a vizinhança dos *pixels* analisados. Um pai é representado pelo *pixel* analisado e o outro por um dos vizinhos desse *pixel*. Na Figura 5.4(c) podemos observar que os detalhes dos objetos na imagem foram realçados. Na Figura 5.4(d) os pares de pais são escolhidos de acordo com a vizinhança dos *pixels* analisados, mas neste tipo diferente da seleção por vizinhança do tipo 3, os vizinhos dos *pixels* analisados são utilizados no cruzamento. Na Figura 5.4(d) podemos observar que alguns detalhes dos objetos na imagem foram realçados e outros detalhes não foram detectados.

A melhor segmentação obtida da imagem da Lena é a segmentação por vizinhança do tipo 3 (Figura 5.5(a)). Neste resultado foi aplicado a detecção de borda (Figura 5.5(b)) para um melhor detalhamento da imagem resultante da segmentação.

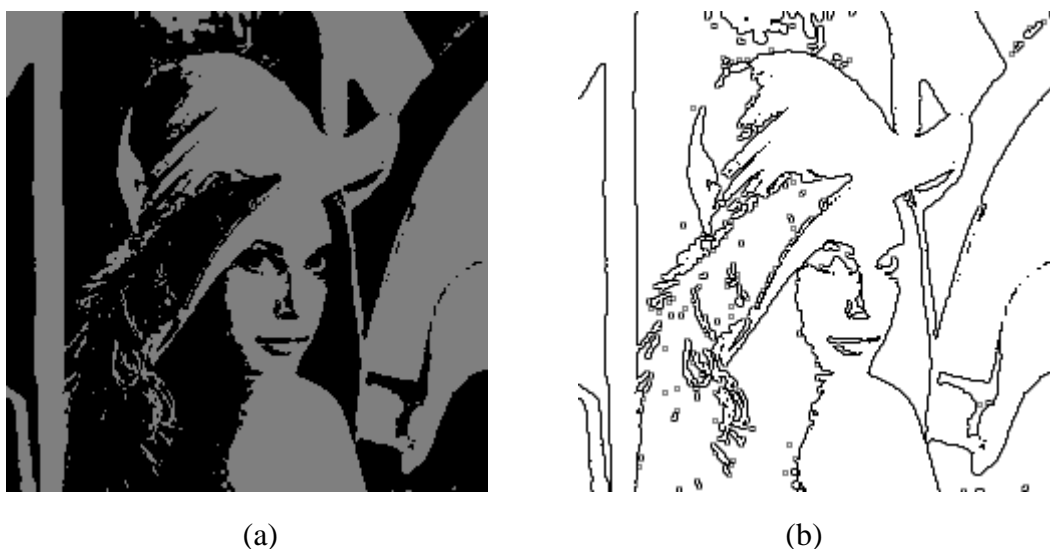


FIGURA 5.5: (a) Imagem segmentada utilizando a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.

Os gráficos da variância e da aptidão representam a evolução da segmentação que utiliza a seleção por vizinhança do tipo 3 ao longo das gerações. O gráfico da variância representa a variância da população final em cada geração. Nos gráficos da variância e da aptidão (Figuras 5.6 e 5.7) podemos observar que na geração 8 o critério de parada foi alcançado. O critério de parada não está relacionado ao número de gerações como foi detalhado no capítulo anterior (seção 4.2.10).

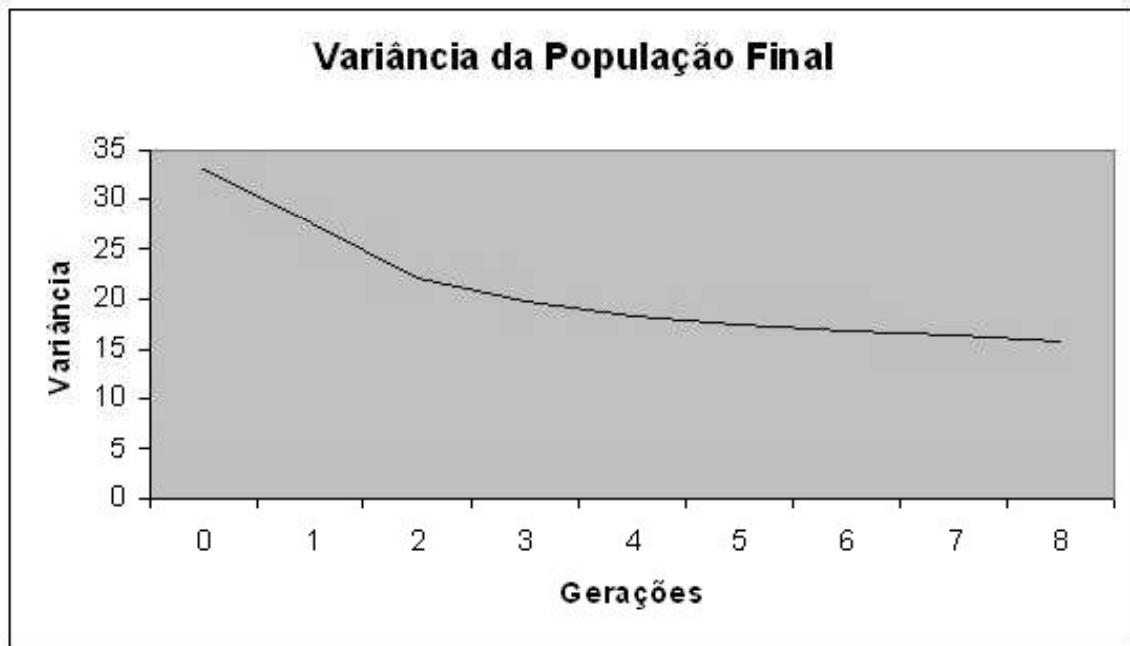


FIGURA 5.6: Imagem Lena: Gráfico da Variância com a seleção por vizinhança do tipo 3.

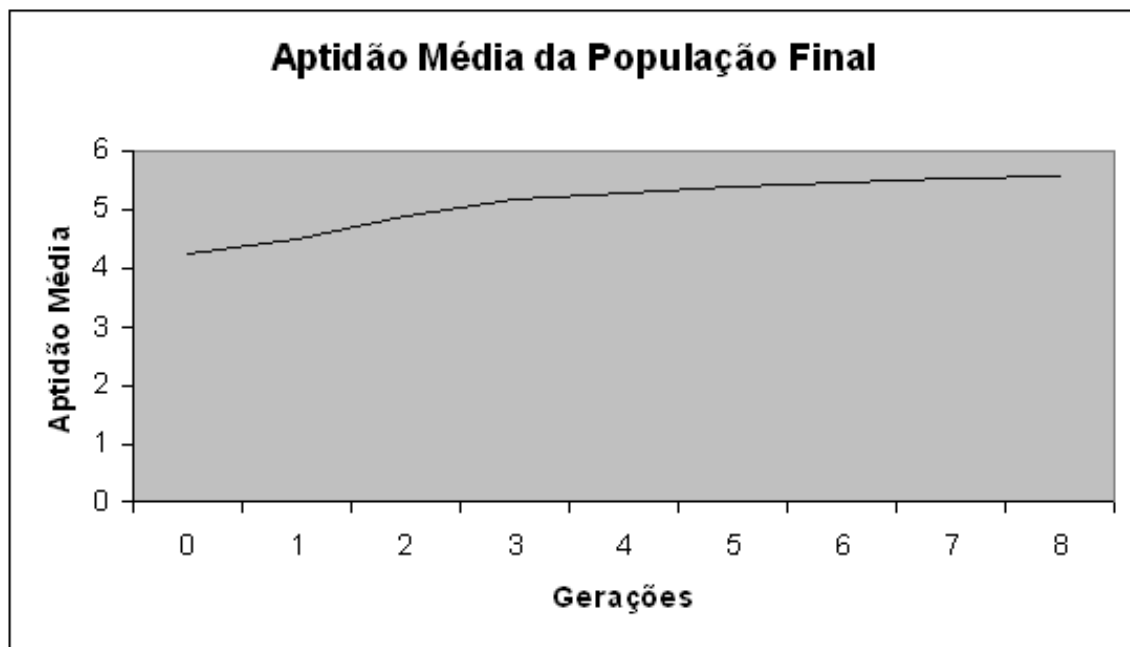


FIGURA 5.7: Imagem Lena: Gráfico da Aptidão com a seleção por vizinhança do tipo 3.

Também foi analisada uma imagem de uma criança (Figura 5.8(a)) que contém objetos ao fundo.



FIGURA 5.8: (a) Imagem original de uma criança. (b) Imagem com contorno aplicado.

Na limiarização (Figura 5.9) e nas segmentações que utilizam a seleção por aleatoriedade dos tipos 1 e 2 (Figuras 5.10 (a) e (b)), e por vizinhança dos tipos 3 e 4 (Figuras 5.10 (c) e (d)) podemos verificar que a diferença é muito pequena, a segmentação fica difícil pois a imagem tem muitas alterações na intensidade dos *pixels*, por influência da claridade do sol e a sombra que contém na imagem. Neste caso, essa imagem precisaria de um pré-processamento para distinguir os objetos de fundo da imagem da criança.



FIGURA 5.9: Imagem segmentada por limiarização.

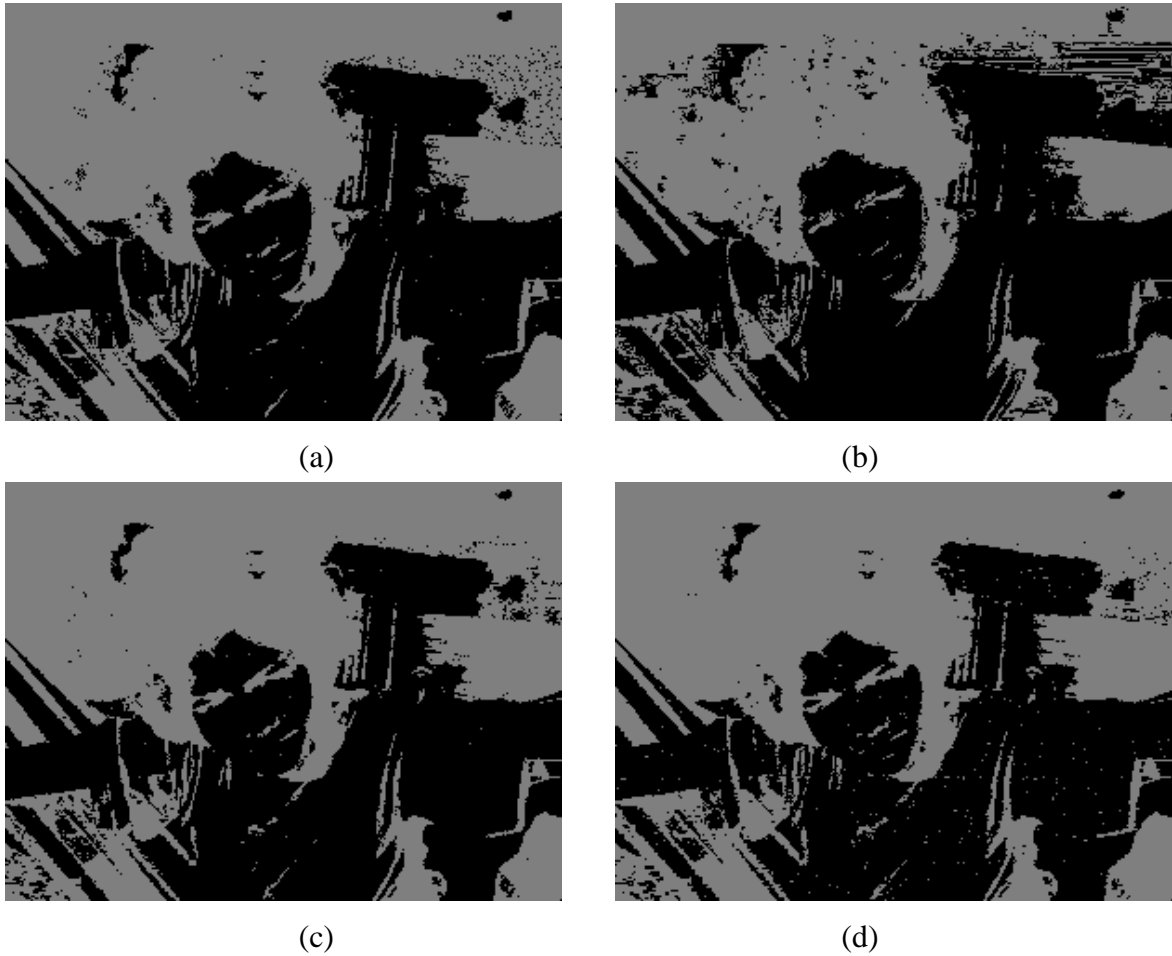


FIGURA 5.10: Segmentação com Seleção por Aleatoriedade: (a) Tipo 1 (b) Tipo 2 e Segmentação com Seleção por Vizinhança: (c) Tipo 3 (d) Tipo 4.

A melhor segmentação obtida da imagem da criança é a segmentação por vizinhança do tipo 3 (Figura 5.11(a)). Neste resultado foi aplicado a detecção de borda (Figura 5.11(b)) para um melhor detalhamento da imagem resultante da segmentação. Como na imagem da Lena o melhor resultado foi obtido com a utilização da seleção por vizinhança do tipo 3.

Os gráficos da variância e da aptidão representam a evolução da segmentação que utiliza a seleção por vizinhança do tipo 3 ao longo das gerações. Neste resultado em particular, o critério de parada foi alcançado após a 45^o geração, como podemos observar nas Figuras 5.12 e 5.13. Isto ocorre quando as regiões estão muito diversificadas em relação aos níveis de cinza dos indivíduos que pertencem a estas regiões, necessitando de um número maior de gerações para se obter um resultado de segmentação satisfatório.



(a)



(b)

FIGURA 5.11: (a) Imagem segmentada utilizando a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.

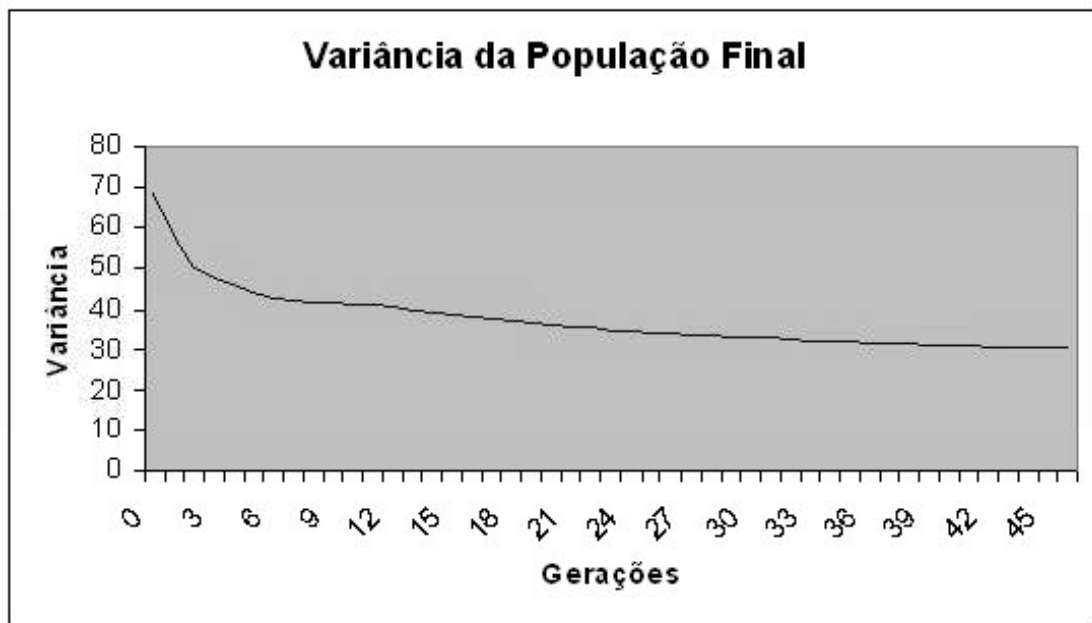


FIGURA 5.12: Imagem criança: Gráfico da Variância com a seleção por vizinhança do tipo 3.

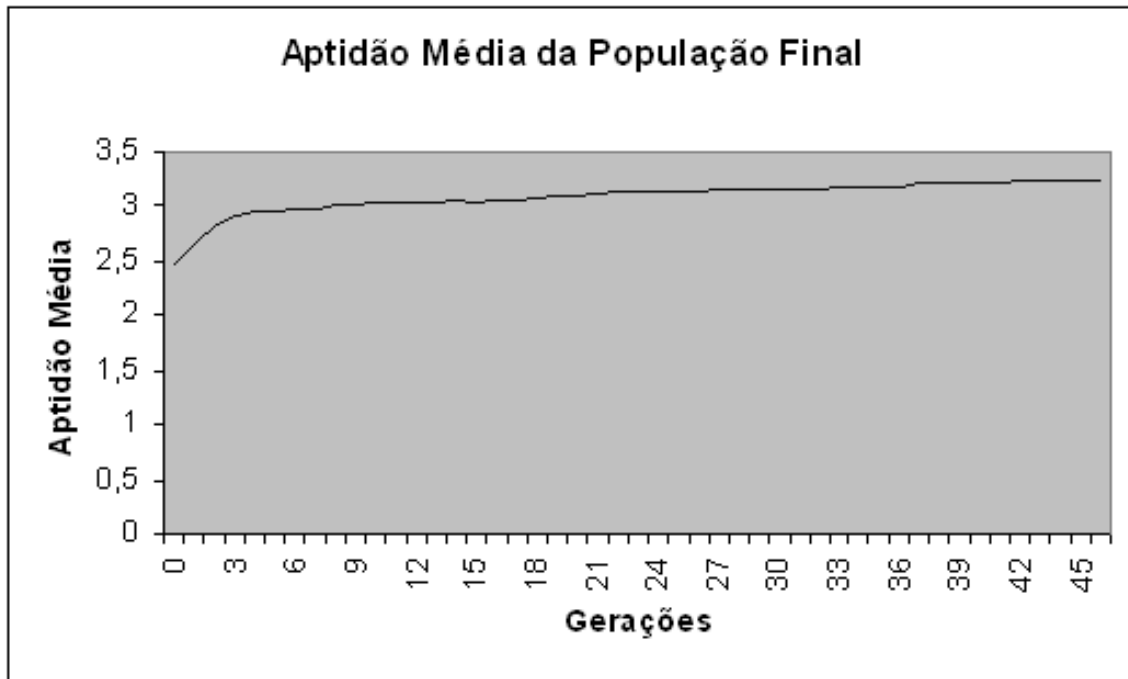


FIGURA 5.13: Imagem criança: Gráfico da Aptidão com a seleção por vizinhança do tipo 3.

5.1.2 Imagem de Animal

A imagem do pingüim (Figura 5.14(a)) foi utilizada para teste no artigo do Lo Bosco [17], porém não consta nesse artigo quantos rótulos foram utilizados, e se houve algum pré-processamento das imagens. A imagem da limiarização (Figura 5.15) é a que mais se aproxima da segmentação que seria feito por uma pessoa sendo um resultado mais subjetivo.

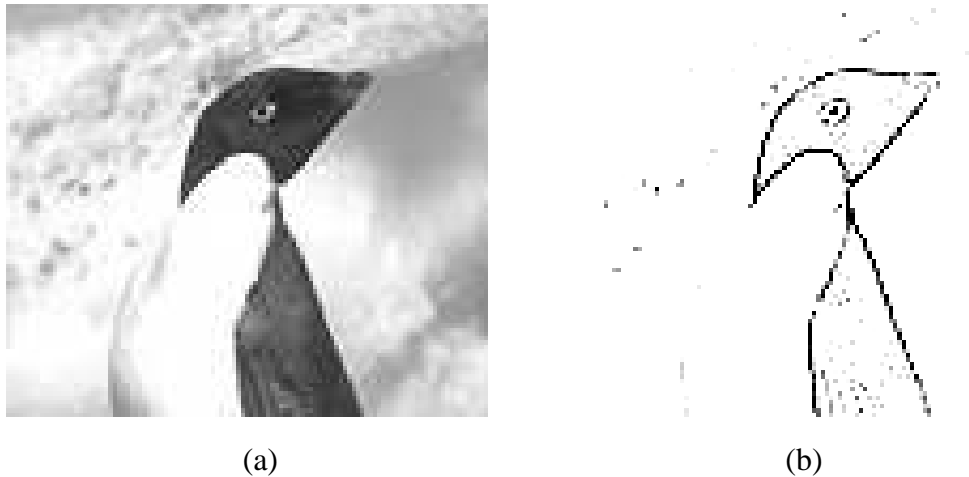


FIGURA 5.14: (a) Imagem original do pingüim. (b) Imagem com contorno aplicado.

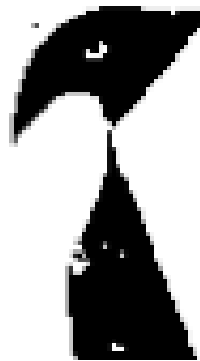


FIGURA 5.15: Imagem segmentada por limiarização.

A Figura 5.16 representa o resultado da segmentação obtido pelo artigo, nesta imagem podemos observar que a segmentação detecta o pingüim e as nuvens ao fundo da imagem. A segmentação desta imagem é mais subjetiva, pois podemos detectar somente a imagem do pingüim ou a cena toda com o pingüim e as nuvens.



FIGURA 5.16: Resultado da imagem segmentada exibida no artigo [17]

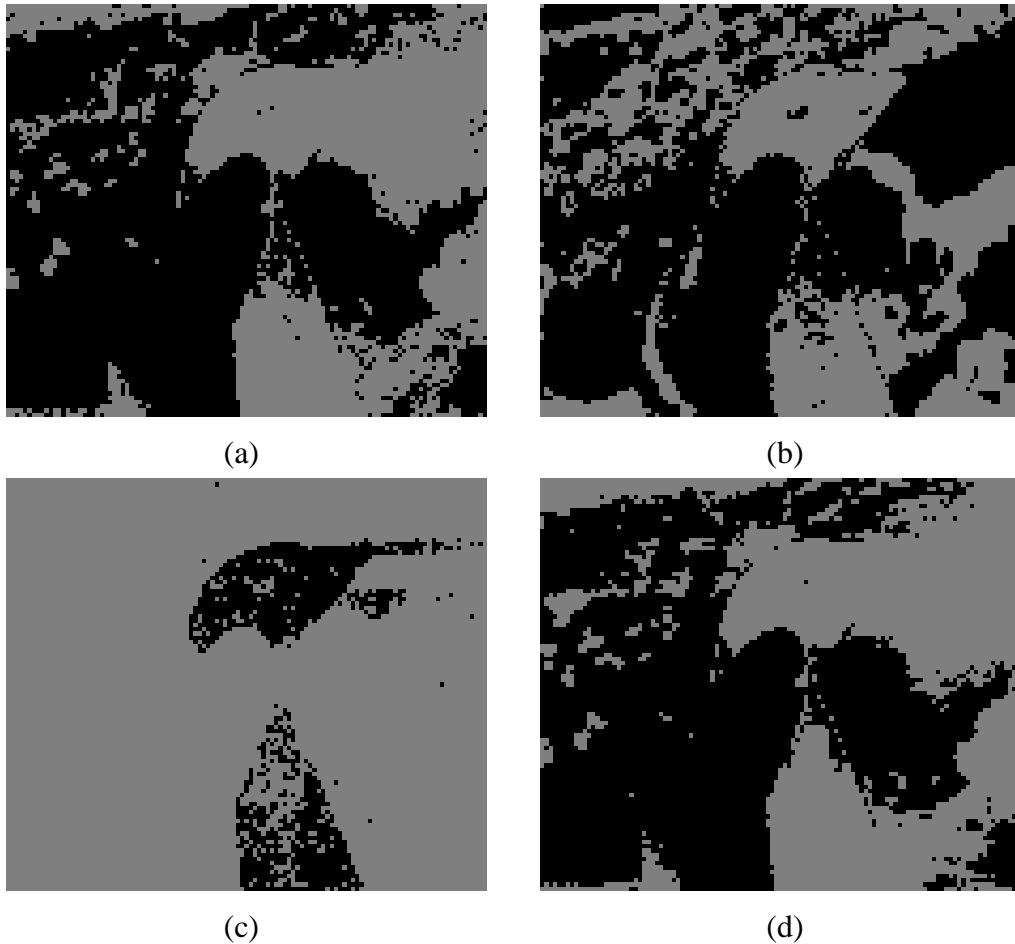


FIGURA 5.17: Segmentação com Seleção por Aleatoriedade: (a) Tipo 1 (b) Tipo 2 e Segmentação com Seleção por Vizinhaça: (c) Tipo 3 (d) Tipo 4.

Na segmentação utilizando as seleções dos tipos: 1 (Figura 5.17(a)), 2 (Figura 5.17(b)), e 4 (Figura 5.17(d)) podemos verificar que as nuvens ao fundo da imagem interferem na segmentação, como na imagem resultante Figura 5.16 retirada do artigo. Caso o resultado esperado da segmentação seja somente a imagem do pingüim.

A melhor segmentação obtida da imagem do pingüim é a segmentação por vizinhança do tipo 3 (Figura 5.18(a)). Neste resultado foi aplicado a detecção de borda (Figura 5.18(b)) para um melhor detalhamento da imagem resultante da segmentação.



FIGURA 5.18: (a) Imagem segmentada utilizando a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.

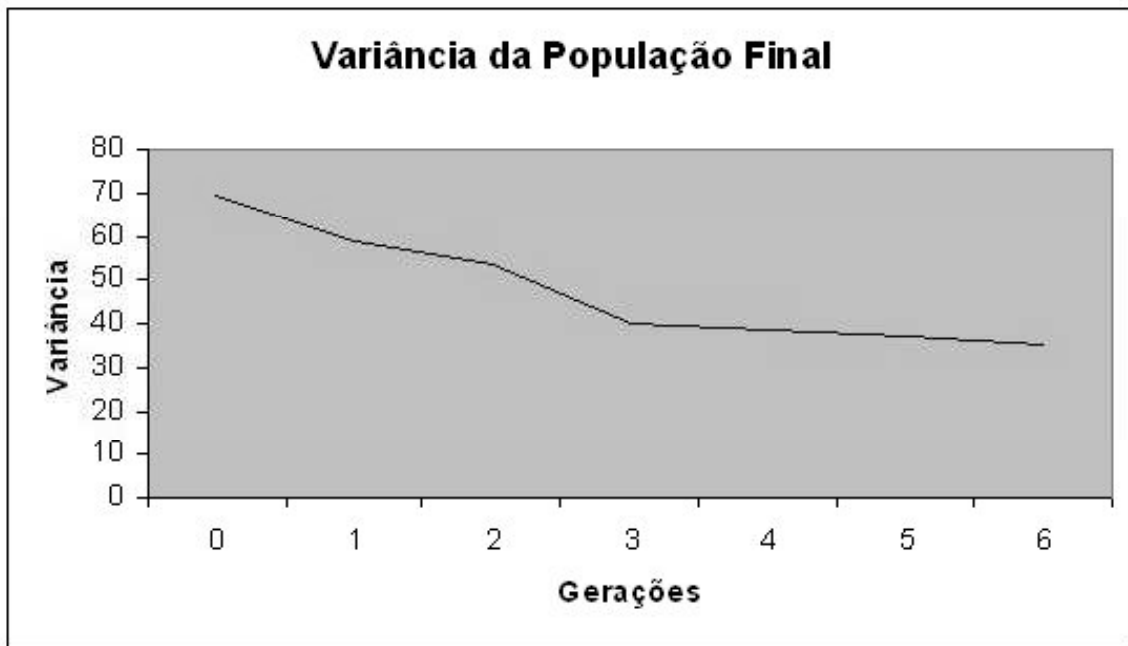


FIGURA 5.19: Imagem pingüim: Gráfico da Variância com a seleção por vizinhança do tipo 3.

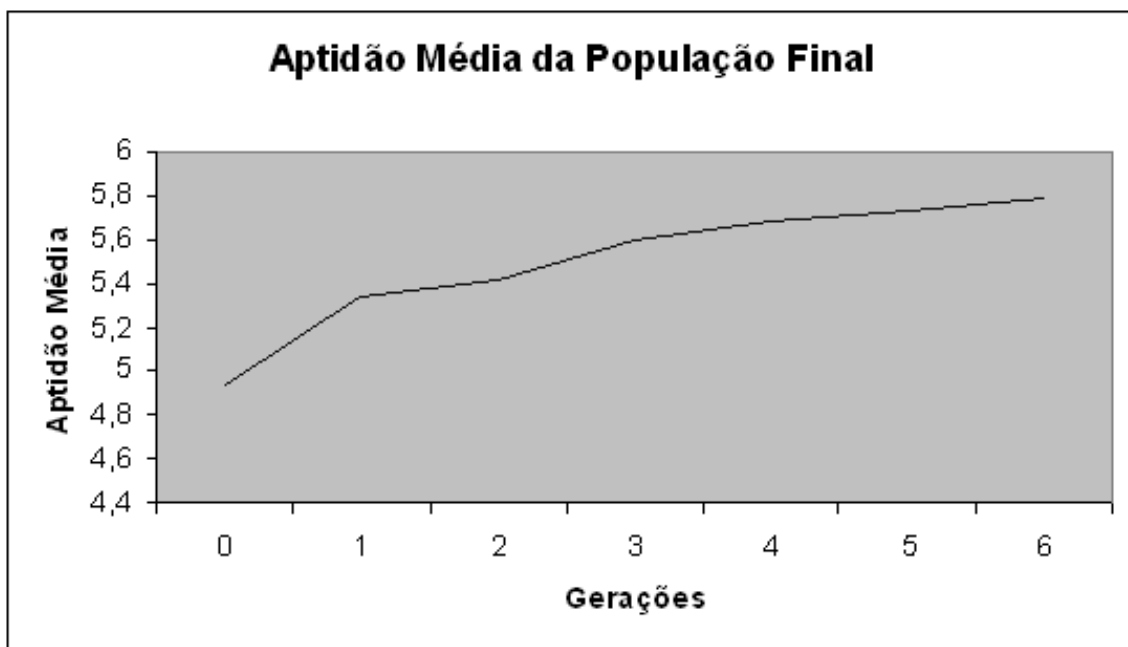


FIGURA 5.20: Imagem pingüim: Gráfico da Aptidão com a seleção por vizinhança do tipo 3.

5.1.3 Imagens de Mapas

Nesta seção serão exibidos os resultados das segmentações das imagens da cidade do Rio de Janeiro (Aterro do Flamengo) e de Paris.

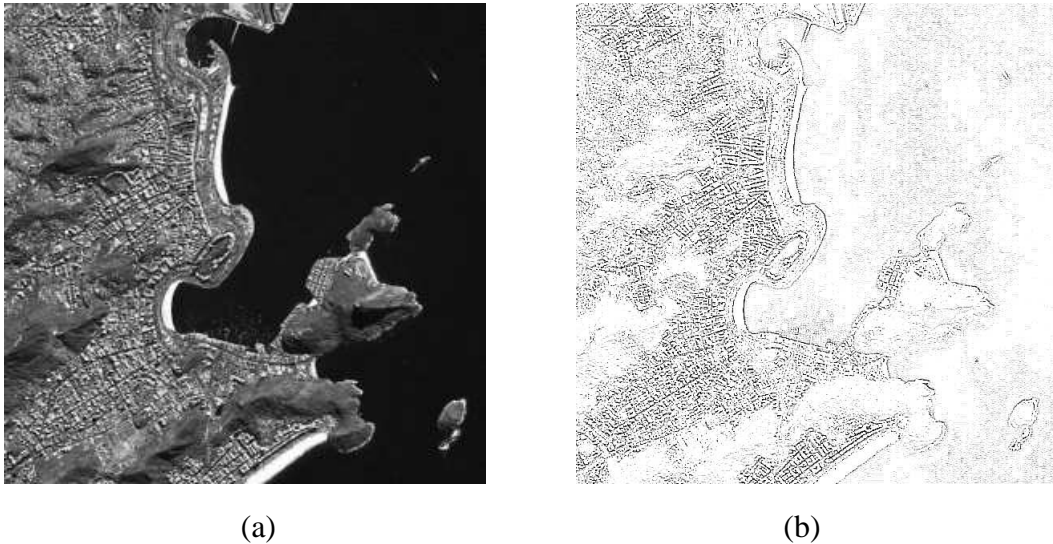


FIGURA 5.21: (a) Imagem original da cidade do Rio de Janeiro. (b) Imagem com contorno aplicado.

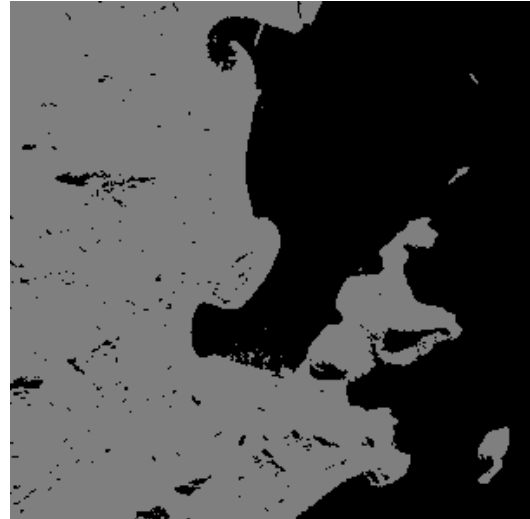
A aplicação da limiarização na imagem (Figura 5.22) obteve um resultado mais detalhado da superfície do que nas segmentações utilizando as seleções por aleatoriedade dos tipos: 1 (Figura 5.23(a)) e 2 (Figura 5.23(b)).



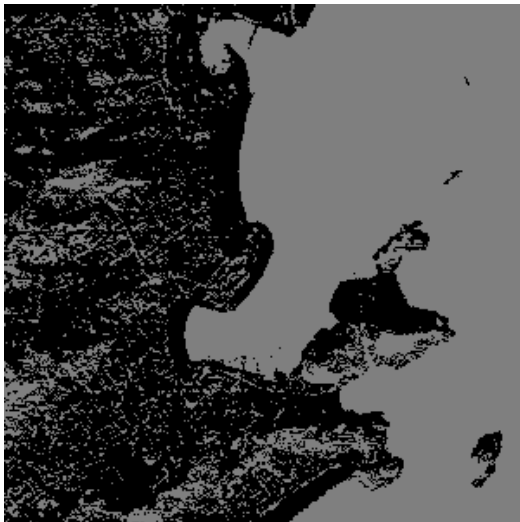
FIGURA 5.22: Imagem segmentada por limiarização.



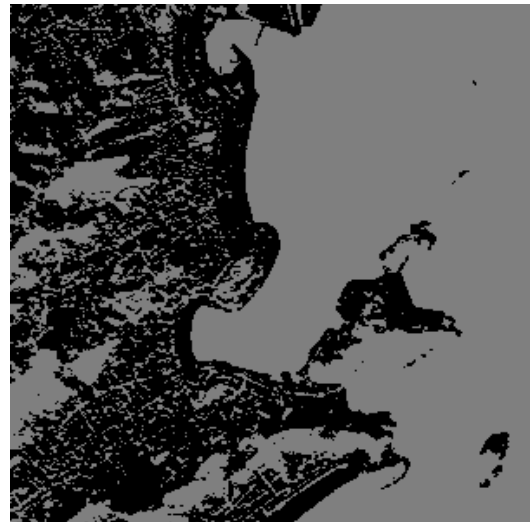
(a)



(b)



(c)



(d)

FIGURA 5.23: Segmentação com Seleção por Aleatoriedade: (a) Tipo 1 (b) Tipo 2 e Segmentação com Seleção por Vizinhança: (c) Tipo 3 (d) Tipo 4.

Na segmentação que utiliza a seleção do tipo 3 (Figura 5.24) os detalhes da superfície do mapa do Rio de Janeiro estão mais realçados, atingindo um resultado mais próximo à segmentação esperada. Nas segmentações que utilizam as seleções dos tipos 1 e 2 os resultados não obtiveram uma boa qualidade, mas na segmentação que utiliza a seleção do tipo 4 que utiliza a característica da vizinhança do *pixel* obteve um resultado um pouco melhor.

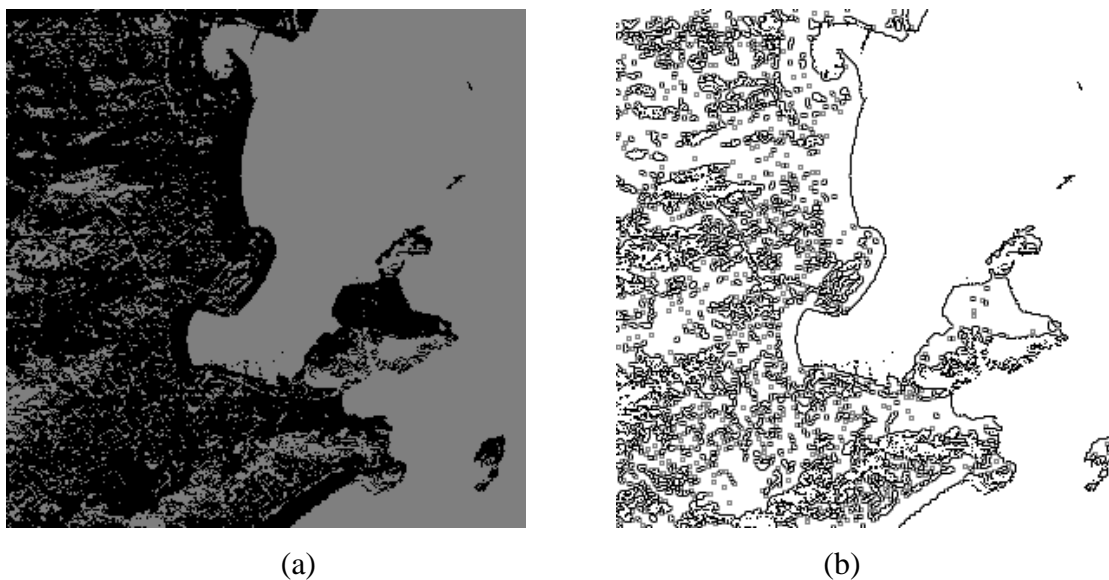


FIGURA 5.24: (a) Imagem segmentada utilizando a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.

Os gráficos da variância e da aptidão (Figuras 5.25 5.26) representam a evolução da segmentação que utiliza a seleção por vizinhança do tipo 3 ao longo das gerações.

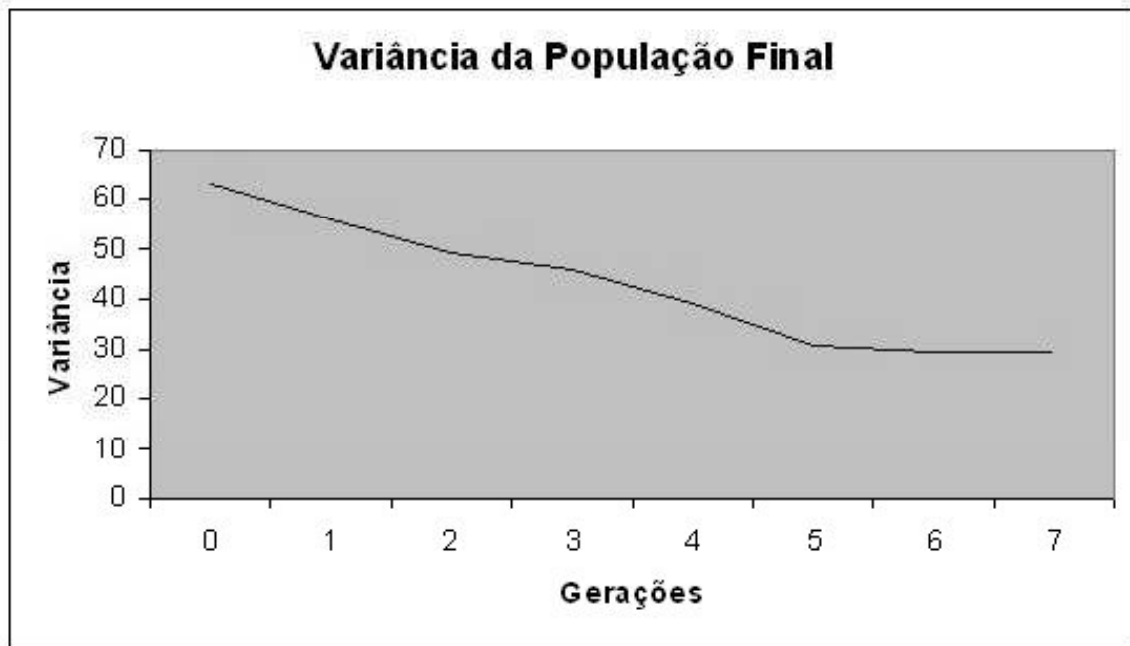


FIGURA 5.25: Imagem Rio de Janeiro: Gráfico da Variância com a seleção por vizinhança do tipo 3.

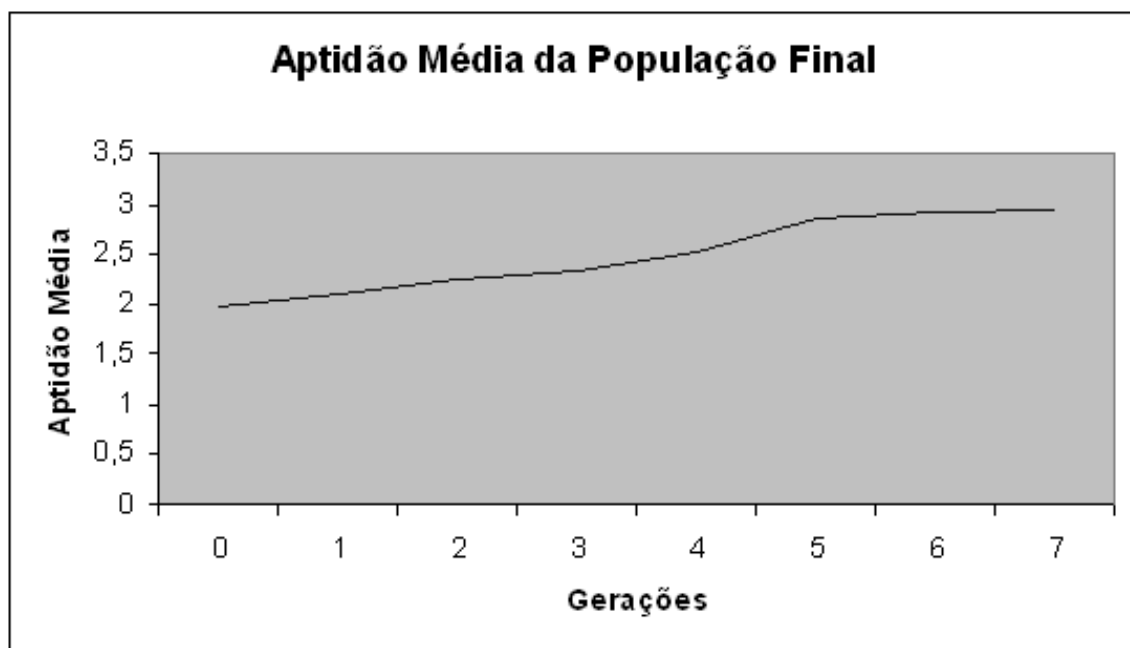


FIGURA 5.26: Imagem Rio de Janeiro: Gráfico da Aptidão com a seleção por vizinhança do tipo 3.

A imagem da cidade de Paris (Figura 5.27(a)) com aplicação da detecção de borda (Figura 5.27(b)) contém alguns detalhes não muito nítidos. Isso também interfere no resultado das segmentações, pois a imagem original necessitaria de um pré-processamento.

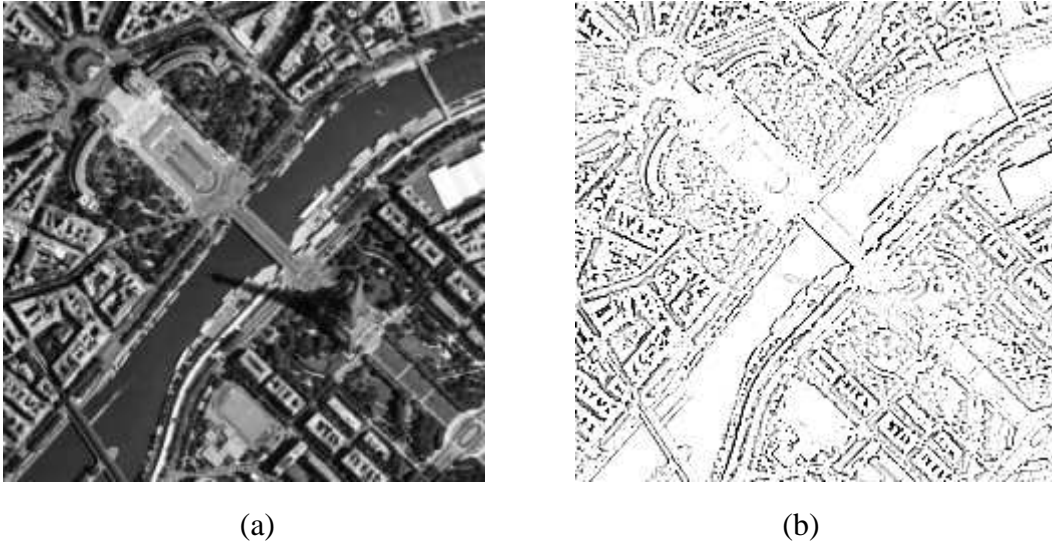


FIGURA 5.27: (a) Imagem original da cidade de Paris. (b) Imagem com contorno aplicado.

A aplicação da limiarização na imagem de Paris obteve um resultado (Figura 5.28) mais detalhado da superfície do que nas segmentações utilizando as seleções por aleatoriedade (tipos: 1 (Figura 5.29(a)) e 2 (Figura 5.29(b))) e por vizinhança (tipos: 3 (Figura 5.29(c)) e 4 (Figura 5.29(d))).



FIGURA 5.28: Imagem segmentada por limiarização.

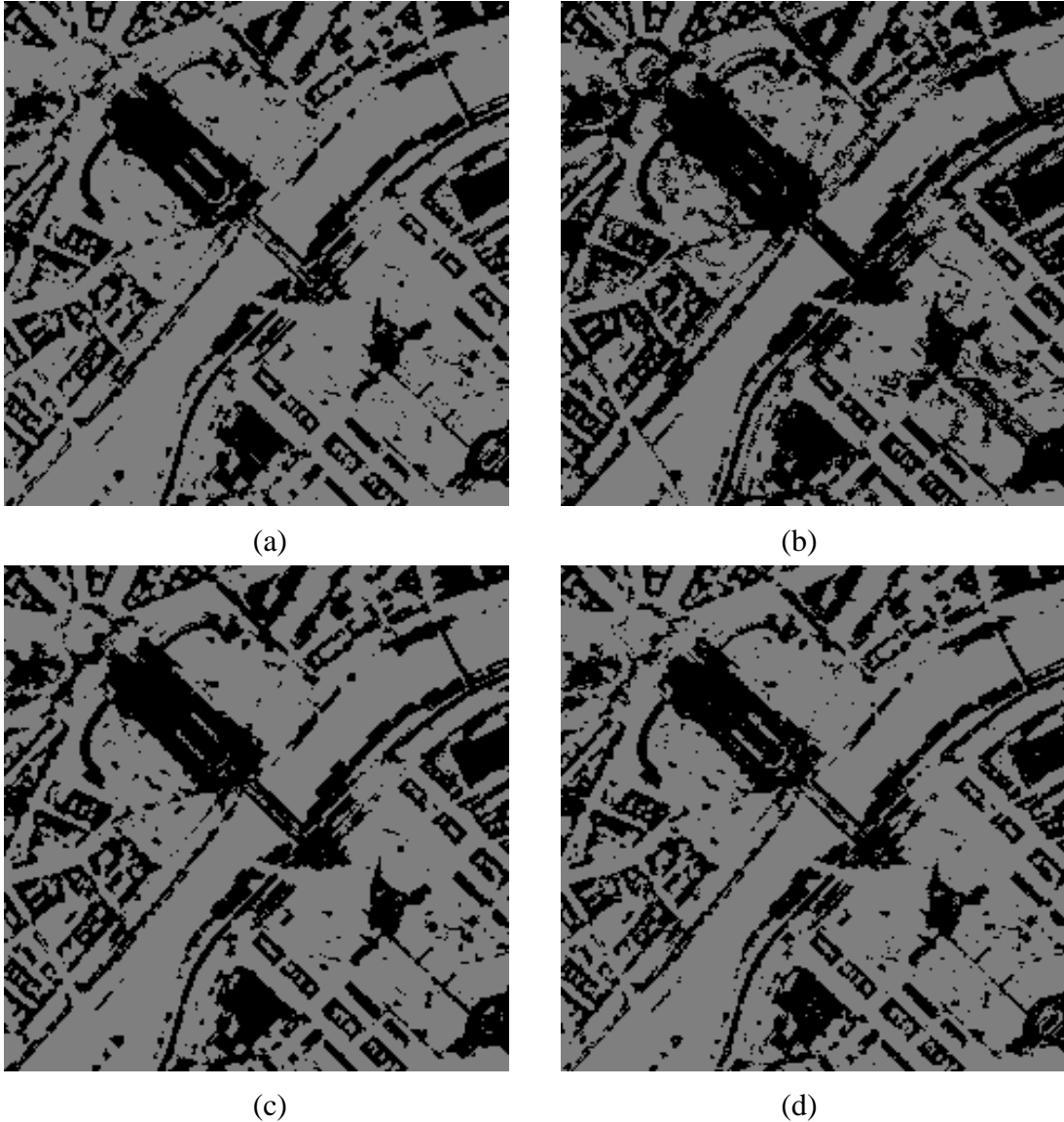
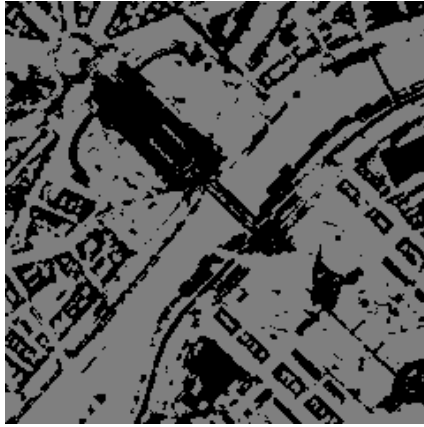


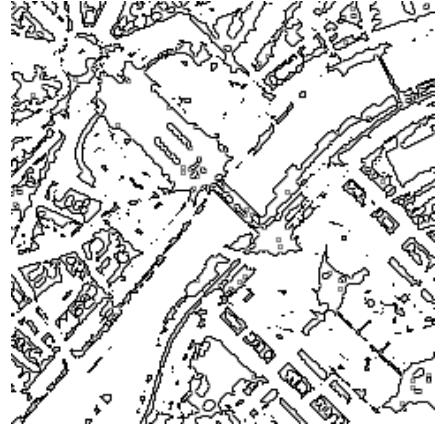
FIGURA 5.29: Segmentação com Seleção por Aleatoriedade: (a) Tipo 1 (b) Tipo 2 e Segmentação com Seleção por Vizinhança: (c) Tipo 3 (d) Tipo 4.

A segmentação da imagem da cidade de Paris (Figura 5.27(a)) tem alguns objetos que dependendo da segmentação utilizada não são destacados. O melhor resultado foi obtido pelo método da limiarização, já que na aplicação do algoritmo de segmentação os resultados obtidos são semelhantes como podemos observar nas Figuras 5.29(a), (b), (c) e (d).

Como na segmentação os resultados foram semelhantes, os gráficos da variância (Figura 5.31) e da aptidão média (Figura 5.32) que serão apresentados serão da segmentação que utiliza a seleção por vizinhança do tipo 3, já que nas imagens anteriormente apresentadas os melhores resultados foram com este tipo de seleção.



(a)



(b)

FIGURA 5.30: (a) Imagem segmentada utilizando a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.

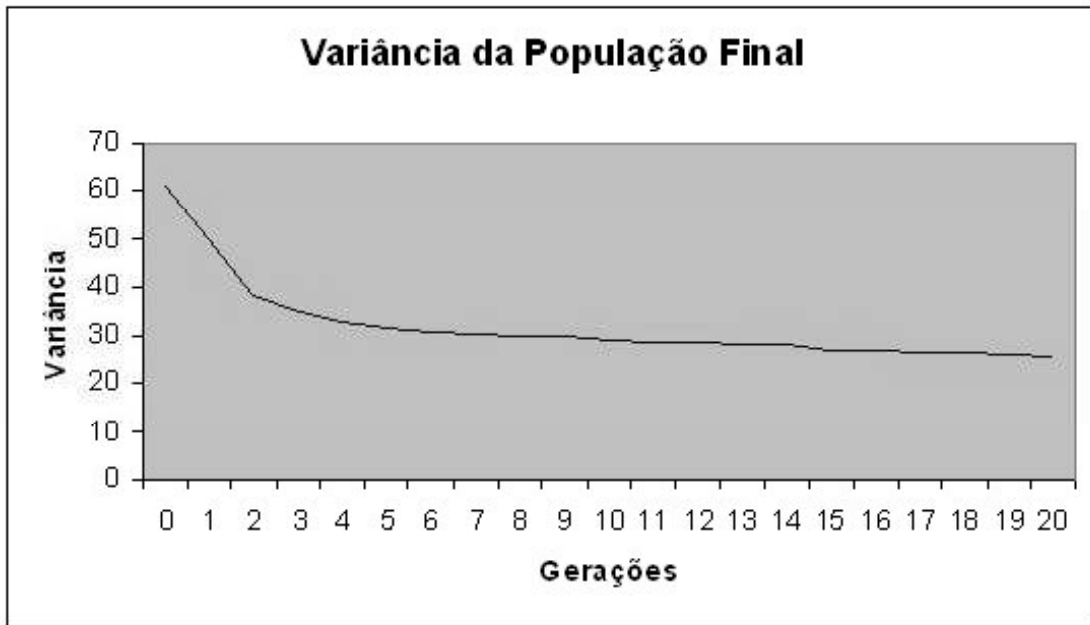


FIGURA 5.31: Imagem Paris: Gráfico da Variância com a seleção por vizinhança do tipo 3.

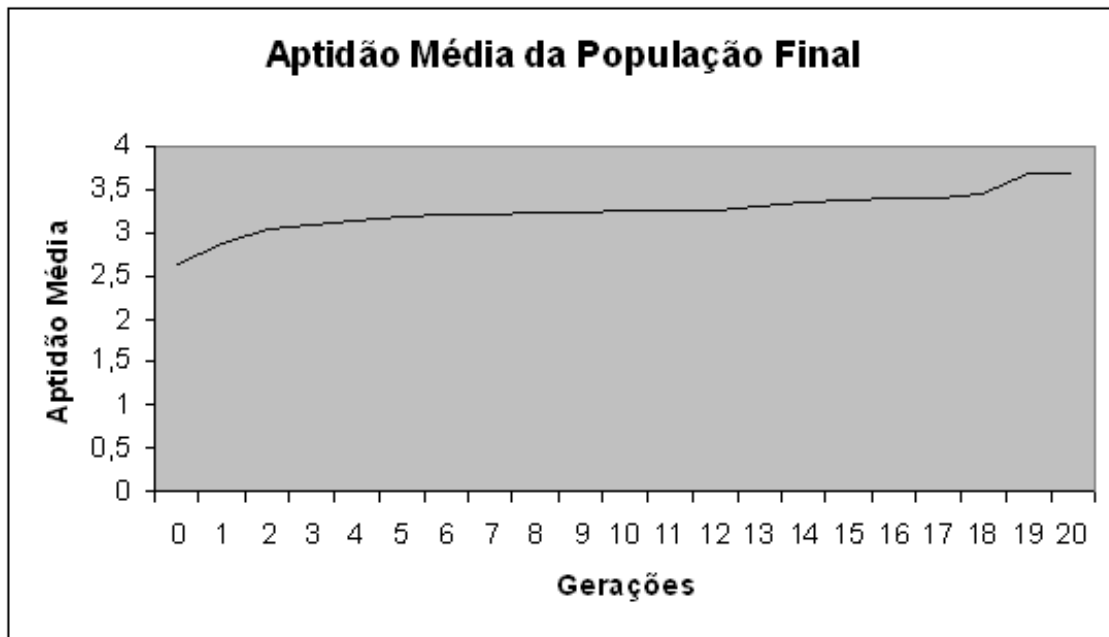


FIGURA 5.32: Imagem Paris: Gráfico da Aptidão com a seleção por vizinhança do tipo 3.

5.1.4 Conclusão

Nos gráficos foram observados que mesmo alcançando o critério de parada o valor da variância continua decrementando. Essa característica pode ser alterada acrescentando ao critério de parada outras restrições que aumente o número de rodadas e considere o que se deseja segmentar na imagem, além de somente utilizar a convergência da variância.

A Tabela 5.1 representa os resultados da segmentação de cada imagem com a utilização de 2 rótulos: com cada tipo de seleção, com o valor da geração em que o critério de parada foi alcançado e o tempo médio em segundos da evolução.

Imagem(resolução)	Tipo de Seleção	Geração do Resultado	Tempo Médio por Geração (s)
Lena(240 × 240)	1	9	1.85
Lena(240 × 240)	2	21	1.21
Lena(240 × 240)	3	8	1.52
Lena(240 × 240)	4	9	2.08
Criança(300 × 225)	1	13	2.16
Criança(300 × 225)	2	7	2.25
Criança(300 × 225)	3	46	2.24
Criança(300 × 225)	4	10	2.16
Pingüim(113 × 97)	1	7	0.71
Pingüim(113 × 97)	2	15	0.31
Pingüim(113 × 97)	3	6	0.78
Pingüim(113 × 97)	4	10	0.81
Rio de Janeiro(300 × 300)	1	10	2.17
Rio de Janeiro(300 × 300)	2	7	2.53
Rio de Janeiro(300 × 300)	3	6	3.03
Rio de Janeiro(300 × 300)	4	12	2.83
Paris(240 × 240)	1	6	1.85
Paris(240 × 240)	2	13	1.91
Paris(240 × 240)	3	20	2.1
Paris(240 × 240)	4	18	2.06

TABELA 5.1: Representação dos resultados obtidos com: o tipo de seleção, a geração em que o critério de parada foi alcançado e o tempo médio das gerações.

Os algoritmos de segmentação foram implementados em C. O algoritmo que utiliza o cromossomo original, que codifica a coordenada e o rótulo da região baseado no artigo [17], não obteve resultados satisfatórios em todos os tipos de imagens testadas como, por exemplo, a imagem da criança representada na Figura 5.33 utilizando a seleção por vizinhança do tipo 3. Alguns motivos que possivelmente influenciaram nesse resultado não satisfatório foram: a falta de informação sobre a implementação, como o número de rótulos utilizado inicialmente e se houve algum tipo de pré-processamento. Neste algoritmo a coordenada e o rótulo eram codificados em um único cromossomo, se na aplicação do cruzamento a coordenada do *pixel* for alterada, esse indivíduo que contém essa coordenada poderá ser totalmente colocado em uma outra posição espacial influenciando no resultado.

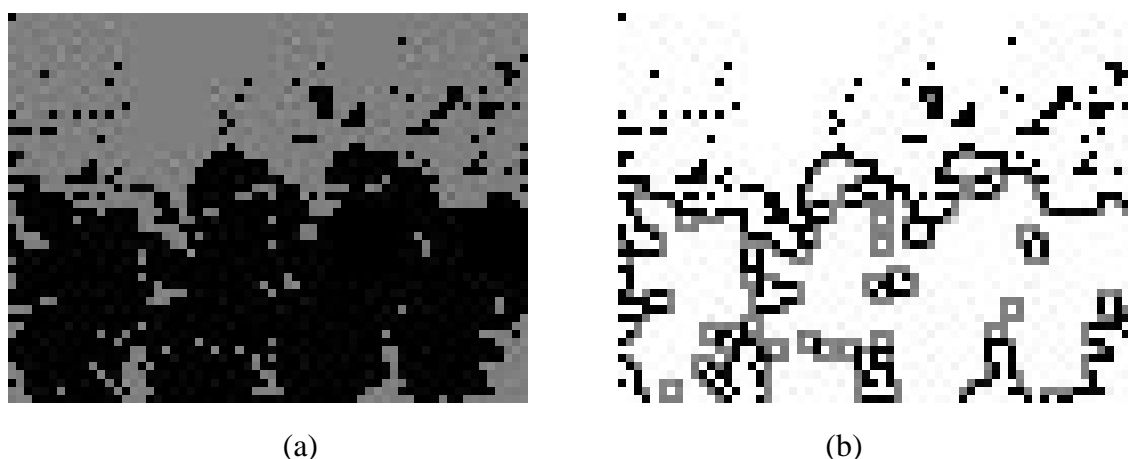


FIGURA 5.33: (a) Imagem segmentada com a seleção por vizinhança do tipo 3. (b) Imagem com contorno aplicado.

A imagem original do pingüim (Figura 5.14(a)) segmentada pelos tipos de seleção: 1 (Figura 5.17(a)), 2 (Figura 5.17(b)) e 4 (Figura 5.17(d)); chegaram a uma semelhança do resultado do artigo (Figura 5.16). O algoritmo implementado que obteve os resultados exibidos ao longo desse capítulo utilizou o cromossomo somente com o rótulo codificado, e não a coordenada e o rótulo codificados em um único cromossomo. As segmentações que utilizaram a seleção por vizinhança do tipo 3 (seção 4.2.6.2) obtiveram os melhores resultados, pois utilizam a informação sobre a vizinhança para a seleção dos indivíduos que irão sofrer o cruzamento, relacionando assim cada indivíduo com seus vizinhos.

Capítulo 6

Conclusão

Este trabalho realiza um estudo de um método de segmentação utilizando um algoritmo genético. No presente estudo, embora as imagens (de face, animal, e mapa) utilizadas tenham sido de diversos tipos, os resultados foram satisfatórios, mesmo sem tratamento especial para um determinado tipo de imagem. No algoritmo proposto foram implementados:

1. dois tipos diferentes de seleção: por aleatoriedade (seção 4.2.6.1) e por vizinhança do *pixel* (seção 4.2.6.2);
2. a modificação do cromossomo proposto em Lo Bosco [17] com duas informações codificadas: coordenada e rótulo da região que cada indivíduo da população pertence, para um cromossomo (seção: 4.2) que armazena uma única informação o rótulo da região que cada indivíduo da população pertence.

O algoritmo genético utilizando o cromossomo modificado obteve bons resultados, mas evidenciou-se a necessidade de um pré-processamento das imagens para obter um bom resultado no caso de imagens com “clutter” [50]. “Clutter” é um termo que se refere à presença de muitos objetos no fundo da imagem, gerando uma espécie de “ruído” e criando a necessidade de algum tipo de pré-processamento “grosseiro” para facilitar a tarefa mais “fina” de segmentação. Para se obter um resultado satisfatório na segmentação, devemos primeiro decidir quais tipos de imagens iremos trabalhar, pois cada uma tem características em especial, que indicará quais serão os parâmetros que terão mais enfoque, como também quantos rótulos são necessários para um determinado tipo de imagem. Os resultados da segmentação são muito importantes na determinação do eventual sucesso ou falha no processo de análise de imagem.

Como sugestões para a continuidade e aperfeiçoamento do presente trabalho pode-se propor:

1. Efetuar um pré-processamento nas imagens;
2. Utilizar algoritmo genético para escolher um valor inicial de rótulos para as imagens;
3. Paralelizar o algoritmo de segmentação para utilizar imagens com dimensões maiores;
4. Além da segmentação, gerar um algoritmo para detectar os objetos da imagem resultante.

Existem diversas técnicas de segmentação de imagens, mas não existe nenhum método único que seja capaz de segmentar todos os tipos de imagem. Para uma implementação mais completa de segmentação é necessário um aperfeiçoamento dos métodos, para serem rápidos e garantirem um resultado satisfatório para uma posterior análise dos objetos que pertencerem às imagens segmentadas.

Referências Bibliográficas

- [1] V. D. Gesú, “A clustering approach to texture classification,” in *Real Time Object and Environment Measurement and Classification* (A. K. Jain, ed.), vol. 42 of *NATO ASI Series F*, Springer Verlag, 1988.
- [2] T. Pavlidis and Y.-T. Liow, “Integrating region growing and edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 225–233, March 1990.
- [3] D. E. Brown and C. L. Huntley, “A practical application of simulated annealing to clustering,” *Pattern Recognition*, vol. 25, no. 4, pp. 401–412, 1992.
- [4] R. W. Klein and R. C. Dubes, “Experiments in projection and clustering by simulated annealing,” *Pattern Recognition*, vol. 22, no. 2, pp. 213–220, 1989.
- [5] S. Z. Selim and K. Alsultan, “A simulated annealing algorithm for the clustering problem,” *Pattern Recognition*, vol. 24, no. 10, pp. 1003–1008, 1991.
- [6] C. J. Veenman, M. J. T. Reinders, and E. Backer, “A cellular coevolutionary algorithm for image segmentation,” *IEEE Transactions on Image Processing*, vol. 12, pp. 304–316, March 2003.
- [7] S. M. Bhandarkar and H. Zhang, “Image segmentation using evolutionary computation,” *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 1–21, April 1999.
- [8] A. do Nola, V. Loia, and A. Staiano, “Genetic-based spatial clustering,” in *Proceedings of the 9th IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 953–956, 2000.
- [9] L. Zhao, Y. Tsujimura, and M. Gen, “Genetic algorithm for fuzzy clustering,” in *Proceedings of IEEE International Conference on Evolutionary Computation, Nayoya, Japan*, pp. 716–719, May 1996.
- [10] M. Yu, N. Eua-anant, A. Saudagar, and L. Udpa, “Genetic algorithm approach to image segmentation using morphological operations,” in *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 775–779, October 1998.
- [11] L. O. Hall, I. B. Özyurt, and J. C. Bezdek, “Clustering with a genetically optimized approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 103–112, July 1999.
- [12] K. Krishna and M. N. Murty, “Genetic k-means algorithm,” *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, vol. 29, pp. 433–439, June 1999.

- [13] T. V. Le, “Evolutionary fuzzy clustering,” in *IEEE International Conference on Evolutionary Computation* (I. C. S. Press, ed.), vol. 2, pp. 753–758, November 1995.
- [14] T. M. Yuukou Horita and M. Miyahara, “Region segmentation using k-mean clustering and genetic algorithms,” in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, pp. 1016–1020, 1994.
- [15] Y. Motohide and S. Oe, “Edge detection of texture image using genetic algorithms,” in *Proceedings of the 36th SICE Annual Conference International Session, Tokushima, Japan*, pp. 1261–1266, July 1997.
- [16] B. Bhanu, S. Lee, and S. Das, “Adaptive image segmentation using genetic and hybrid search methods,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, pp. 1268–1291, October 1995.
- [17] G. L. Bosco, “A genetic algorithm for image segmentation,” in *Proceedings of the IEEE 11th International Conference on Image Analysis and Processing, Palermo, Italy*, pp. 262–266, September 2001.
- [18] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice-Hall, Upper Saddle River, NJ, second ed., 2002.
- [19] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Thomson, second ed., 1998.
- [20] W. E. Higgins and E. J. Ojard, “Interactive morphological watershed analysis for 3d medical images,” *Computerized Medical Imaging and Graphics*, vol. 17, pp. 387–395, May 1993.
- [21] I. Carlbom, D. Terzopoulos, and K. Harris, “Computer-assisted registration, segmentation, and 3d reconstruction from images of neuronal tissue sections,” *IEEE Transactions on Medical Imaging*, vol. 13, pp. 351–362, 1994.
- [22] L. Cohen and R. Kimmel, “Fast marching the global minimum of active contours,” in *Proceedings of IEEE International Conference on Image Processing, (Lausanne)*, pp. 473–476, September 1996.
- [23] A. X. Falcão, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. A. Lotufo, “User-steered image segmentation paradigms: live-wire and live-lane,” *Graphical Models and Image Processing*, vol. 60, pp. 233–260, July 1998.
- [24] A. E. O. Boudraa, J. Mallet, J. E. Besson, S. E. Bouyoucef, and J. Champier, “Left ventricle automated detection method in gated isotopic ventriculography using fuzzy clustering,” *IEEE Transactions on Medical Imaging*, vol. 12, pp. 451–465, September 1993.
- [25] A. R. Cohen, B. Roysam, and J. N. Turner, “Automated tracing and volume measurements of neurons from 3d confocal fluorescence microscopy data,” *Journal of Microscopy*, vol. 173, pp. 103–114, February 1994.

- [26] E. L. D. K. P. Philip, D. D. McPheron, N. L. Gotteiner, M. J. Vonesh, W. Stanford, J. E. Reed, J. A. Rumberg, and K. B. Chandran, "Automatic detection of myocardial contours in cine-computed tomographic images," *IEEE Transactions on Medical Imaging*, vol. 13, pp. 241–253, June 1994.
- [27] A. Goshtasby and D. A. Turner, "Segmentation of cardiac cine mr images for extraction of right and left ventricular chambers," *IEEE Transactions on Medical Imaging*, vol. 14, pp. 56–64, March 1995.
- [28] A. Martelli, "Edge detection using heuristic search methods," *Computer Graphics and Image Processing*, vol. 1, pp. 169–182, August 1972.
- [29] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 641–647, June 1994.
- [30] L. Najman and M. Schmitt, "Geodesic saliency of watershed contours and hierarchical segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 1163–1173, December 1996.
- [31] K. R. Castleman, *Digital Image Processing*. Prentice-Hall, Upper Saddle River, NJ, 1996.
- [32] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Rockville, Maryland, USA. Computer Science Press, 1982.
- [33] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York, 3 ed., 1996.
- [34] J. H. Holland, "Genetic algorithms," *Scientific American*, pp. 44–50, July 1992.
- [35] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [36] J. H. Holland, *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- [37] D. E. Goldberg and K. Deb, "A comparison of selection schemes used in genetic algorithms," in *Foundations of Genetic Algorithms* (G. J. E. Rawlins, ed.), Morgan Kaufmann, 1991.
- [38] R. Hinterding, "Representation, mutation and crossover issues in evolutionary computation," in *Proceedings of the Conference on Evolutionary Computation*, pp. 916–923, 2000.
- [39] A. A. Freitas, *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer-Verlag, Berlin, 2002.
- [40] V. Dhar and R. Stein, *Seven Methods for Transforming Corporative Data into Business Intelligence*. Prentice Hall, Londres, 1997.
- [41] M. Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, 1996.

- [42] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico*, pp. 731–737, June 1997.
- [43] Y. Motohide and S. Oe, "A method of distance texture image segmentation by genetic algorithms," in *Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan*, pp. 125–130, May 1996.
- [44] P. Andrey, "Selectionist relaxation: Genetic algorithms applied to image segmentation," *Image Vision Computing*, vol. 17, pp. 175–187, 1999.
- [45] G. G. Yen and N. Nithianandan, "Facial feature extraction using genetic algorithm," in *Proceedings of the Congress on Evolutionary Computation*, pp. 1895–1900, IEEE Press, 2002.
- [46] S. W. Hwang, E. Y. Kim, S. H. Park, and H. J. Kim, "Object extraction and tracking using genetic algorithms," in *Proceedings of the International Conference on Image Processing*, pp. 383–386, October 2001.
- [47] J. H. Xue, S. Ruan, B. Moretti, M. Revenu, and D. Bloyet, "Knowledge-based segmentation and labeling of brain structures from mri images," *PRL*, vol. 22, pp. 395–405, March 2001.
- [48] Y. Yokoo and M. Hagiwara, "Human faces detection method using genetic algorithm," in *Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan*, pp. 113–118, May 1996.
- [49] H. Zheng, L. X. Kong, and S. Nahavandi, "Automatic inspection of metallic surface defects using genetic algorithms," *Journal of Materials Processing Technology, Elsevier, UK*, vol. 125-126, pp. 427–433, February 2002.
- [50] Y. Aq and C. D., "Morphological and wavelet transforms for object detection and image-processing," *Applied Optics*, vol. 33, pp. 8226–8239, December 1994.