

O USO DO ALGORITMO GENÉTICO EM SEGMENTAÇÃO DE IMAGENS DIGITAIS

Ítalo de Oliveira Matias

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 DOUTOR EM CIÊNCIAS DE ENGENHARIA CIVIL.

Aprovada por:

Prof. Nelson Francisco Favilla Ebecken, D.Sc.

Prof. Alexandre Gonçalves Evsukoff, D.Sc.

Prof. Beatriz de Souza Leite Pires de Lima, D.Sc.

Prof. Ricardo Cordeiro de Farias, Ph.D.

Prof. Luiz Marcos Garcia Gonçalves, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

ABRIL DE 2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

MATIAS, ITALO DE OLIVEIRA

O Uso do Algoritmo Genético em
Segmentação de Imagens Digitais [Rio de
Janeiro] 2007

VII, 89p. 29,7 cm (COPPE/UFRJ, D.Sc.,
Engenharia Civil, 2007)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

Dedicatória:

À Deus, meus Pais, minha filha Júlia e a minha esposa Márcia

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

O USO DO ALGORITMO GENÉTICO EM SEGMENTAÇÃO DE IMAGENS DIGITAIS

Ítalo de Oliveira Matias

Abril/2007

Orientadores: Alexandre Gonçalves Evsukoff
Nelson Francisco Favilla Ebecken

Programa: Engenharia Civil

O objetivo deste trabalho é aplicar a técnica de Algoritmo Genético (AG) com diferentes funções de avaliação (funções objetivo) na geração de algoritmos para a segmentação de imagens digitais. Para tanto, três métodos de segmentação foram testados: limiarização, geração por subdivisão de quadrantes (quadtree) e por crescimento de regiões. Todos estes procedimentos visam minimizar uma função de avaliação (entropia, segmentação excedente, entre outras) ou maximizar tal função (pixels corretamente agrupados), buscando uma maior similaridade entre os objetos a serem segmentados. Para efeito comparativo, foram gerados como resultados várias imagens que utilizam o AG no algoritmo de segmentação e outras imagens geradas com o mesmo algoritmo, mas sem utilizar o AG. Dessa maneira, pode-se verificar que o uso do AG com uma função ou métrica de avaliação adequada, pode trabalhar muito bem em conjunto com algoritmos de segmentação, na busca de bons parâmetros de entrada iniciais (limiar, área dos objetos, forma dos objetos, entre outros) para estes algoritmos, facilitando o uso para o usuário final.

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

THE GENETIC ALGORITHM METHOD FOR DIGITAL IMAGE SEGMENTATION

Italo de Oliveira Matias
April/2007

Advisors: Alexandre Gonçalves Evsukoff
Nelson Francisco Favilla Ebecken

Department: Civil Engineering

The purpose of this work is to use the Genetic Algorithm (GA) with different evaluation functions (objective functions), and to apply it to the generation of algorithms for the segmentation of digital images. For this, the three tested segmentation methods have been: generation by subdivision of quadrants (quadtree), thresholding and region growing. In all these procedures, the main objective for the Genetic Algorithm is to minimize an evaluation function (entropy, grouping of pixels, exceeding segmentation, among others), that searches for a greater similarity between the objects to be segmented. For comparative purposes some images have been generated, upon which the segmentation has been performed by means of the use of GA, and others without GA. This way, it can be verified that the use of the GA with an evaluation or metric function, can work very well with segmentation algorithms, in the search for good initial parameters (threshold, area of objects, forms of objects, among others) for these algorithms, thus facilitating their use for the final user.

Sumário

1. INTRODUÇÃO	1
1.1 Organização do trabalho.....	3
2. IMAGEM DIGITAL.....	6
2.1 Amostragem e Quantização.....	8
2.2 Imagem Monocromática.....	8
2.3 Equalização e Expansão de Histograma.....	9
2.4 Distância entre pixels	10
2.5 Controle de Contraste.....	11
2.6 Convolução Bidimensional Discreta	12
2.7 Textura	13
3. SEGMENTAÇÃO DE IMAGENS DIGITAIS	14
3.1 Segmentação por Descontinuidades	17
3.1.1 Detecção de Pontos.....	19
3.2 Segmentação por Região	20
3.2.1 Segmentação por Limiarização (<i>Thresholding</i>).....	21
3.2.2 Divisão e fusão	24
3.2.3 Segmentação por Crescimento de Regiões	26
3.3 Resumo.....	36
4. MÉTRICAS UTILIZADAS PARA AVALIAÇÃO DE SEGMENTAÇÃO.....	37
4.1 Avaliação Segundo Zhang.....	38
4.1.1 Caracterização da performance.....	40
4.1.2 Avaliação da segmentação.....	40
4.2 Avaliação Segundo Huang e Dom	41
4.3 Avaliação Segundo Oliver e Ortiz.....	42
4.3.1 Avaliação da performance	42
4.3.2 Avaliação da Segmentação por Matriz de Confusão	43
4.3.3 Novas estratégias de Avaliação baseadas em discrepância	44
4.4 Avaliação Segundo Udupa et al	47
4.5 Conclusão sobre as métricas utilizadas	48
5. OTIMIZAÇÃO DE SEGMENTAÇÃO EM IMAGENS DIGITAIS	49
5.1 Características dos AGs.....	50
5.1.1 Representação cromossomial.....	51
5.1.2 Função de Avaliação	52
5.1.3 Seleção dos pais.....	52
5.1.4 Funcionamento do AG.....	53
5.2 Algoritmos Implementados	53
5.2.1 AGs para a Segmentação por Quadtree	54
5.2.2 Segmentação por Limiar utilizando os AGs.....	57
5.2.3 Segmentação por crescimento de regiões utilizando os AGs	58
6. TESTES E RESULTADOS.....	65

6.1	Introdução.....	65
6.2	Resultados Experimentais	66
6.2.1	Segmentação de Quadtree com AG.....	66
6.2.2	Segmentação por Limiar com AG	70
6.2.3	Segmentação por Crescimento de Regiões com AG.....	73
7.	CONCLUSÃO E TRABALHOS FUTUROS	81
7.1	Trabalhos Futuros.....	82

Capítulo 1

Introdução

Nos últimos anos tem havido um crescente interesse em dados (imagens) enviados por satélites e posteriormente para aplicações de sensoriamento remoto. A enorme procura e disponibilidade destes dados gera uma forte necessidade de automatização das técnicas de processamento e análise destas imagens para estas aplicações. Para isso, é utilizado um processo de identificação de objetos, sendo que o cérebro humano é capaz de em fração de segundos diferenciar (classificar) cada objeto pertencente a uma cena de um modo eficiente e exato. Chama-se a este processo de segmentação. Dados de imagens de sensoriamento remoto (Tso *et al.* [1]) são amplamente utilizados em aplicações de modelagem de terreno, monitoramento ambiental, terrestre e aquático, bases de dados georeferenciados, aplicações que manipulam dados atmosféricos (Liou [2]) etc.

Segundo Gonzales e Wintz [3], um dos primeiros passos do processo de análise de uma imagem digital consiste em particioná-la em seus elementos constituintes. A Segmentação é um dos fatores mais importantes relacionados a uma imagem ou a uma cena. O objetivo da segmentação é a divisão da imagem em regiões. Essas regiões são encontradas a partir de um processo de agrupamento de pixels com base em alguma propriedade intrínseca da imagem tal como intensidade média, contraste ou textura. Muito embora a segmentação permita representar objetos da imagem como entidades únicas, ela não é *a priori* uma classificação e sim um reconhecimento de estruturas desta imagem.

Os algoritmos de segmentação de imagem são baseados nas propriedades dos seus níveis de cinza: descontinuidade e similaridade (Gonzalez e Woods [4], Jain [5] e Baxes [6]). Descontinuidade significa uma mudança abrupta de níveis de cinza entre duas regiões homogêneas e a similaridade é uma medida de homogeneidade entre regiões com mesmo nível de cinza baseada em limiares. Uma região em uma imagem é um conjunto de pontos conectados um ao outro. A detecção dessas regiões pode ser feita com um objetivo: dividir a imagem em regiões separadas que quando juntas representam a imagem original. A segmentação pode ser manual ou automática. Segmentação manual tem como principal vantagem a possibilidade de contar com outros dados (percepção humana, mapas, etc.) além dos contidos na imagem. A segmentação automática (feita a partir de algoritmos) é adaptada de acordo com a finalidade da imagem em observação.

Este trabalho visa a segmentação automática com a utilização dos Algoritmos Genéticos (algoritmos que usam modelos computacionais dos processos naturais de evolução como uma ferramenta para resolver problemas) para determinação de parâmetros iniciais. Parâmetros estes (atributos da imagem) que são entradas para o algoritmo de segmentação.

Trabalhos como os de Pappa *et al.* [7], Santos *et al.* [8] e Melkemi *et al.* [9] utilizam o Algoritmo Genético para selecionar atributos para a geração da imagem segmentada. Yoshimura e OE [10], [11] trabalham com um método de segmentação de imagens compostas por vários tipos de textura, utilizando os AGs para considerar um número “ótimo” de áreas segmentadas no qual é definida uma função de avaliação que mede a qualidade da clusterização (agrupamento). Há também trabalhos como o de Jin e Davis [12] que usam os AGs em segmentação de imagens usando uma função de avaliação baseada em conjuntos fuzzy.

O diferencial do presente trabalho em relação aos algoritmos que usam os AGs em segmentação de imagens digitais é que geralmente estes utilizam os AGs como parte integrante da segmentação, isto é, os AGs são usados internamente ao algoritmo de segmentação. Nesta Tese o que ocorre é exatamente o contrário, isto é, o algoritmo de segmentação está inserido dentro dos AGs. Como consequência, pode-se utilizar qualquer tipo de algoritmo de segmentação, pois os AGs se adaptam, a partir de uma função de avaliação apropriada, a cada tipo de algoritmo. Uma outra particularidade deste trabalho é o uso de novas métricas (Ortiz e Oliver [13]) para a função de avaliação dos AGs.

Para testar o uso da técnica dos AGs, na segmentação de imagens digitais, utilizou-se métodos de segmentação de imagens baseados em limiarização (Gonzalez e Woods [4]), métodos baseados em divisões, isto é, segmentação por quadrees (Matias e Evsukoff [14]) e métodos que utilizam o crescimento de regiões (Bins *et al.* [15]). Para comparar os resultados destes algoritmos com e sem o uso dos AGs foram utilizadas algumas medidas de avaliação de segmentação propostas por Matias e Evsukoff [14] e por Ortiz e Oliver [13], as quais serão discutidas no capítulo sobre Métricas de Avaliação para Algoritmos de Segmentação.

1.1 Organização do trabalho

Esta Tese está organizada da seguinte maneira:

- **Capítulo 2 - Imagem Digital**

Neste capítulo, inicialmente, dar-se-á uma definição de imagem digital e, em seguida, serão conceituadas várias descrições intimamente ligadas à imagem como um todo, juntamente com algumas notações que serão utilizadas neste trabalho.

• **Capítulo 3 - Segmentação de Imagens Digitais**

Este capítulo descreve o processo de segmentação de uma imagem digital, definindo o que é segmentação e quais passos devem ser tomados para obtenção da mesma com a definição de alguns métodos existentes na literatura. São descritos os algoritmos que foram implementados para este trabalho: limiarização, quadtree e crescimento de regiões (ressaltando-se a definição de um dos métodos utilizados na Tese e que compõe a biblioteca *TerraLib*¹); e mais um método que foi bastante estudado: a segmentação em multi-resolução que é aplicada num Sistema de Análise Orientada a Objetos denominado *eCognition*².

• **Capítulo 4 - Métricas Utilizadas para Avaliação de Segmentação**

Neste capítulo são desenvolvidos alguns estudos, propostos por diferentes autores, cujo intuito é mostrar métricas que podem ser utilizadas para avaliar se o quanto o resultado fornecido por um algoritmo de segmentação é satisfatório. Para isso são descritas algumas definições teóricas sobre as métricas e várias equações que buscam atingir um resultado comum: a avaliação da segmentação. Algumas dessas métricas são utilizadas no desenvolvimento deste trabalho.

• **Capítulo 5 – Otimização de Segmentação em Imagens Digitais**

Este capítulo se divide em três momentos: Primeiramente fornece uma introdução inicial sobre os Algoritmos Genéticos (AGs), levando em consideração alguns aspectos descritivos dos mesmos.

¹ Biblioteca *OpenSource* encontrado em <http://www.terralib.org/>

² GIS Comercial encontrado em <http://www.definiens.com/>

Em seguida descreve alguns trabalhos que utilizaram os AGs para auxiliarem algoritmos de segmentação de imagens. Por fim, são descritos sucintamente os algoritmos de segmentação usados nesta Tese com o emprego dos AGs.

• **Capítulo 6 – Testes e Resultados**

Neste capítulo, são mostrados alguns testes realizados com algoritmos de segmentação por quadtree, limiarização e crescimentos de regiões. São comparados os resultados de cada algoritmo sem o uso prévio dos AGs e com o uso dos AGs. Para isso, foram utilizadas algumas funções de avaliação com base nas métricas definidas no Capítulo 4.

• **Capítulo 7 – Conclusão e Trabalhos Futuros**

Este capítulo descreve como foi realizado o trabalho, aqui descrito, incluindo alguns detalhes de implementação. São também introduzidas algumas novas idéias sobre o tema desta Tese.

Capítulo 2

Imagem Digital

Uma imagem é uma função bidimensional $f(x, y)$ de intensidade da luz (Gonzalez e Woods [4]), onde x e y denotam as coordenadas espaciais de um ponto e o valor da função f (Figura 2.1) é proporcional ao brilho (tons de cinza, caso a imagem seja monocromática) da imagem naquele ponto. Uma imagem digital é uma função $f(x, y)$ discretizada tanto em coordenadas espaciais quanto em brilho. Essa função $f(x, y)$ é um produto da luminância pela reflectância em cada ponto (x, y) .

$$f(x, y) = i(x, y) * r(x, y) \quad (2.1)$$

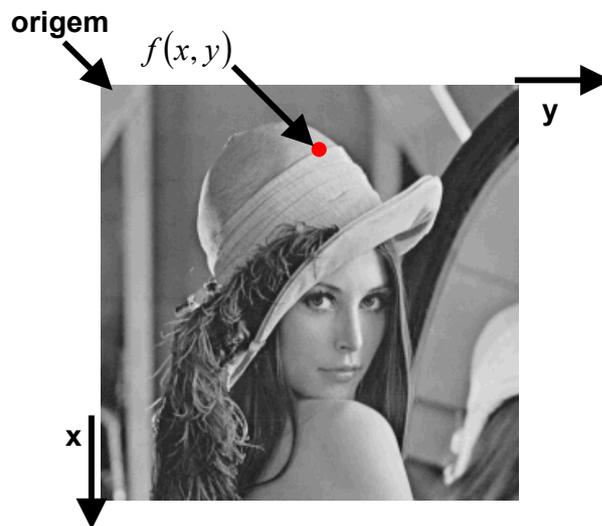


Figura 2.1 - Convenção dos eixos para representação de uma imagem digital na imagem da Lena

O modelo dessa função $f(x, y)$ pode ser exibido conforme a Figura 2.2, no qual o valor da função exprime o brilho em cada ponto da imagem.



Figura 2.2 - Representação de imagem em forma 3D com tons de cinza

Segundo a equação 2.1 tem-se:

- $i(x, y) \rightarrow$ luminância no ponto (x, y)
- $r(x, y) \rightarrow$ reflectância no ponto (x, y)

Com:

$$\begin{aligned} 0 < i(x, y) < \infty \\ e \\ 0 < r(x, y) < 1 \end{aligned} \tag{2.2}$$

A luminância $i(x, y)$ está diretamente associada a quantidade de luz que incide no objeto, e a reflectância $r(x, y)$ está associada a quantidade de luz refletida pelo objeto.

A Tabela 2.1 mostra os valores da iluminação sobre diferentes condições ambientais e a Tabela 2.2 apresenta a reflectância associada a diferentes objetos.

Condição	$i(x, y)$
Dia de Sol	9000
Dia Nublado	1000
Lua cheia	0,01

Tabela 2.1 – Iluminação sob diferentes condições ambientais

Material	$r(x, y)$
Neve	0,93
Parede Branca	0,80
Aço Inoxidável	0,65
Parede Opaca	0,01

Tabela 2.2 – Reflectância de diferentes objetos

2.1 Amostragem e Quantização

Para que se possa efetuar o processamento computacional, uma função $f(x, y)$ precisa ser digitalizada tanto espacialmente quanto em amplitude. A digitalização das coordenadas espaciais (x, y) é denominada amostragem da imagem e a digitalização da amplitude é denominada quantização. A Equação 2.3, abaixo, mostra uma imagem representada por uma função $f(x, y)$ contínua que é aproximada por amostras igualmente espaçadas, representada em uma matriz de ordem $N \times M$.

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, M-1) \\ f(1,0) & f(1,1) & \cdots & f(1, M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1, M-1) \end{bmatrix} \quad (2.3)$$

2.2 Imagem Monocromática

Uma imagem monocromática apresenta unicamente graduações de cinzas, com pixels mais claros representando valores mais elevados; pixels pretos possuem valores que se aproximam ou são iguais a zero, enquanto pixels brancos se aproximam ou são iguais a $L - 1$, onde L geralmente é 256. (Figura 2.3)



Figura 2.3 – Tons de cinza

2.3 Equalização e Expansão de Histograma

Se o nível de cinza l ocorre n_l vezes em uma imagem com n pixels, pode-se estimar a probabilidade de ocorrência do nível l na imagem por intermédio de sua frequência relativa:

$$P(l) = \frac{n_l}{n} \quad (2.4)$$

O histograma da imagem é uma representação gráfica de n_l ou de $P(l)$ em função de l . A análise do histograma permite tecer considerações a respeito da distribuição de níveis de cinza de uma imagem. A Figura 2.4 apresenta histogramas de imagens escuras e claras.

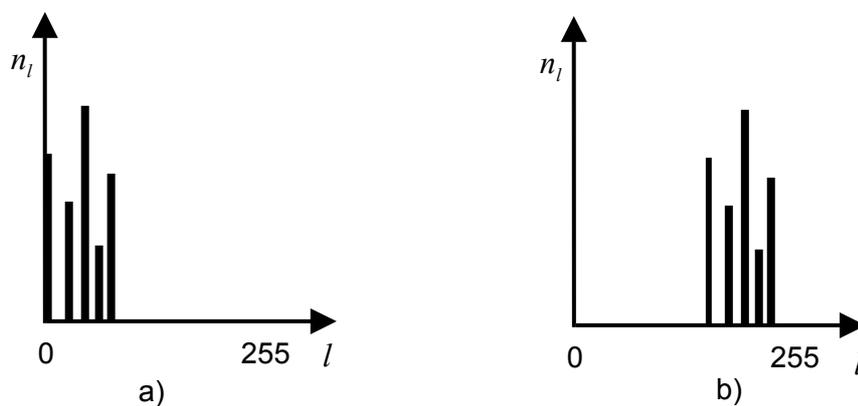


Figura 2.4 - Histogramas: (a) Imagens escuras; (b) Imagens claras

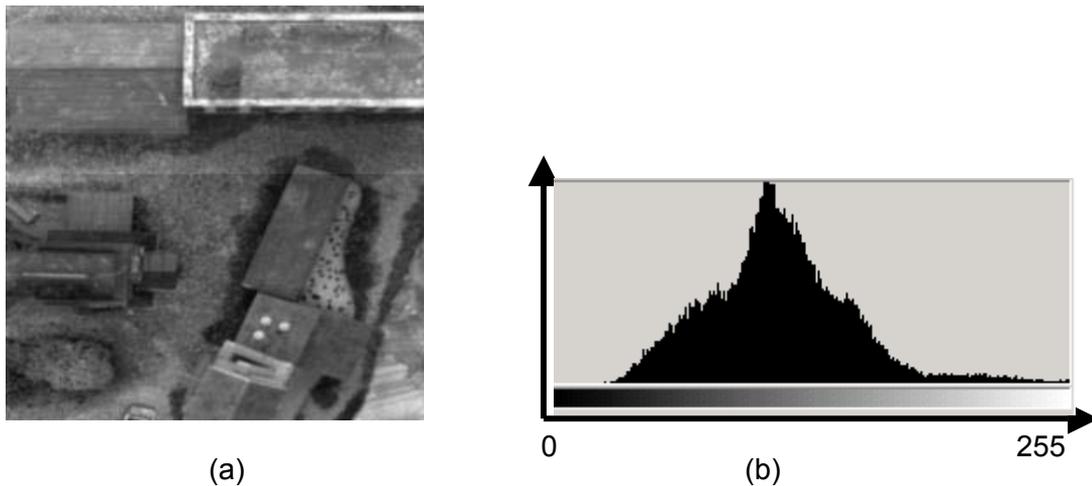


Figura 2.5 - (a) Imagem Original (Gonçalves *et al.* [16])
 (b) Histograma correspondente

Quando se tem uma faixa reduzida de níveis de cinza, pode ser utilizada a técnica de expansão do histograma para produzir uma imagem mais rica em detalhes. A expansão do histograma consiste na seguinte transformação pontual:

$$s = T(r) = \text{round} \left(\frac{r - r_{\min}}{r_{\max} - r_{\min}} * (L - 1) \right) \quad (2.5)$$

onde r_{\min} e r_{\max} representam, respectivamente, o menor e o maior nível de cinza presentes na imagem de entrada, com $r_{\min} < r_{\max}$, L é o limite superior da escala de cinzas e round significa arredondamento para o inteiro mais próximo. A operação atribui zero aos pixels com nível de cinza r_{\min} e valor $L - 1$ aos pixels com nível de cinza r_{\max} , e distribui proporcionalmente os níveis intermediários entre 0 e $L - 1$. Assim, essa operação leva à utilização de toda a escala de cinzas disponível.

2.4 Distância entre pixels

Dados dois pixels p_1 e p_2 de coordenadas (x_1, y_1) e (x_2, y_2) respectivamente, a distância entre eles é calculada por:

$$\|p_2 - p_1\| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.6)$$

Também é válido observar as propriedades abaixo:

$$\begin{aligned} \|p_2 - p_1\| = 0 &\Leftrightarrow p_2 = p_1; \\ &e \\ \|p_2 - p_1\| &= \|p_1 - p_2\|. \end{aligned} \quad (2.7)$$

2.5 Controle de Contraste

A importância e o significado do contraste começam no nível básico da visão através da ausência ou presença de luz. O contraste é de todas as técnicas, a mais importante para o controle visual de uma imagem bi ou tridimensional. Na verdade, assim que adicionamos qualquer elemento a uma página em branco já estamos utilizando o contraste. Ele é automático.

Imagens de baixo contraste podem resultar de iluminação deficiente, tempo de exposição demasiadamente curto, problemas eletrônicos no equipamento de digitalização, etc. Um contraste reduzido dificulta o discernimento dos objetos em uma cena. O objetivo do aumento do contraste é aumentar a faixa de níveis de cinza utilizada. Em algumas ocasiões, no entanto, pode ser esteticamente interessante reduzir o contraste em uma cena. A operação de controle de contraste pode ser feita de diversas maneiras. Uma forma simples é mostrada na Figura 2.6.

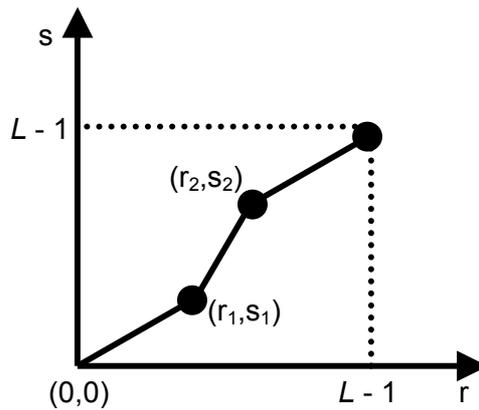


Figura 2.6 - Operação de Controle de Contraste

Se $r_1 = s_1$ e $r_2 = s_2$ não há nenhuma alteração nos níveis de cinza. Se a declividade do primeiro e do último segmento de reta é menor que 1 e a do segmento central maior que 1 tem-se um aumento de contraste. Por outro lado, se a declividade do primeiro e do último segmento de reta é maior que 1 e a do segmento central menor que 1, tem-se redução do contraste.

2.6 Convolução Bidimensional Discreta

Sejam duas imagens digitais de resolução $M \times N$ e representadas pelas funções $f(x, y)$ e $g(x, y)$, sendo $x=0,1,\dots,M-1$ e $y=0,1,\dots,N-1$. A convolução é uma outra função que é calculada por:

$$Conv(x, y) = \left(\frac{1}{M * N} \right) * \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) * g(x - m, y - n) \quad (2.8)$$

O resultado da convolução de $f(x, y)$ por $g(x, y)$ em um ponto p é uma média ponderada (Equação 2.9) dos pontos de $f(x, y)$, onde os pesos são dados pela imagem $g(x, y)$. Ou seja, dado um ponto p de $f(x, y)$ os pontos

correspondentes que servem de peso na imagem $g(x, y)$ é uma vizinhança de p na imagem $f(x, y)$, a qual chamamos de filtro ou máscara (Figura 2.7).

$$z = w_1 * z_1 + w_2 * z_2 + \dots + w_9 * z_9 = \sum_{i=1}^9 w_i * z_i \quad (2.9)$$

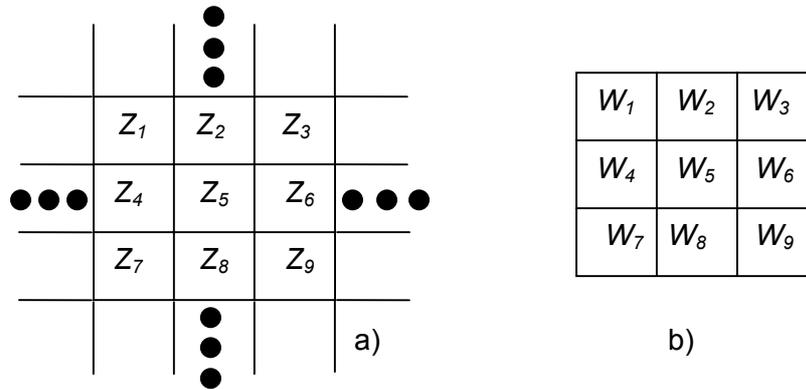


Figura 2.7- (a) Subárea de uma imagem mostrando valores de pixels;
 (b) Uma máscara 3x3 com coeficientes genéricos.

2.7 Textura

A textura pode ser observada em padrões estruturais de superfícies de objetos tais como madeira, grãos, areia e tecidos. O termo textura geralmente refere-se a repetição de elementos de textura básicos denominado *texels* (Jain [5]). Uma textura contém vários pixels cuja localização pode ser periódica, quase-periódica ou aleatória. Texturas naturais são geralmente aleatórias, e texturas artificiais são freqüentemente determinísticas ou periódicas.

Capítulo 3

Segmentação de Imagens Digitais

Normalmente, informações sobre objetos, dados ou cenas são obtidas com o uso de algoritmos computacionais (baseados em formas matemáticas ou estatísticas) que operam sobre dados digitais, captados a partir de sensores. Um exemplo típico é o uso de imagens que podem ser obtidas a partir de câmeras filmadoras, câmeras digitais, equipamentos de ressonância magnética e de ultra-sonografia (Gonçalves *et al.* [16]). Um sistema de aquisição de imagens possui dispositivos sensíveis à luz refletida ou emitida pelos objetos. Esses (foto)-sensores determinam valores para a luminância de pequenas regiões da superfície. Os valores da luminância são geralmente quantizados através da divisão do espaço de percepção em vários níveis, cuja quantidade depende da capacidade de sensibilidade do dispositivo. Para sistemas monocromáticos, os níveis de intensidade da luminância são normalmente denominados de níveis de cinza ou tons de cinza, sendo 256 níveis suficientes para a maioria das aplicações. Geralmente estas pequenas regiões são definidas por formas retangulares ou quadradas agrupadas lado a lado num plano de projeção, sendo a imagem resultante composta por uma matriz 2D.

O pixel (de “*ixmap element*”) é a denominação usual para cada pequena região deste plano. Assim, cada pixel nada mais é que uma codificação ou uma quantificação dos atributos de luminância, cor, ou brilho de uma pequena porção da superfície de um objeto discretamente amostrado. É a partir deste atributo contido no pixel que se torna possível a identificação de regiões e estabelecimento de subdivisões na imagem na sua unidade básica para que a mesma possa ser

interpretada de acordo com uma finalidade específica. A essa identificação de regiões ou segmentos (não sobrepostos) na imagem dá-se o nome de segmentação.

O processo de segmentação tem a finalidade de detectar objetos em uma imagem. Esses objetos podem ser entendidos como regiões que possuem alguma propriedade em comum. Essa propriedade, em imagens monocromáticas, está relacionada à intensidade dos tons de cinza em cada pixel pertencente a cada região correspondente. Gonzalez e Woods [4], Jain [5], Baxes [6] e Fu e Mui [17], citam que os algoritmos de segmentação para imagens monocromáticas (tons de cinza) são geralmente baseados em uma das seguintes propriedades básicas: métodos baseados em descontinuidades (bordas ou contornos), métodos orientados ao pixel e métodos baseados em similaridades (regiões homogêneas). Os métodos baseados em descontinuidade têm como abordagem principal a partição da imagem em função de mudanças bruscas nos tons de cinza. Os métodos orientados aos pixels são baseados em informações locais e comparados com um valor de threshold (limiar) e os métodos baseados na similaridade visam a detecção de regiões homogêneas. Muitos autores tratam os dois últimos métodos em um único tipo de segmentação que é a segmentação por regiões.

Os métodos que utilizam o limiar são sem dúvida os mais populares na área de segmentação de imagens por serem de processamento rápido e de mais fácil implementação, porém seus resultados podem possuir qualidade limitada. Pal e Pal [19] e Sahoo *et al.* [20] propõem um método baseado em limiarização. Punam e Jayaram [21], propõem um método baseado em limiar denominado minimização da energia baseada em incertezas e homogeneidades cuja idéia principal é observar quais pixels com altas taxas de incerteza ficam acumulados, isto é, agrupados, mais ao redor das bordas. Pal e Pal [22], [23] e Yin [24] utilizaram métodos baseados na variação de entropia, considerando a distribuição de probabilidade entre os pixels

vizinhos. Xie e Tisui [25] propõem uma segmentação obtida a partir de uma maximização do modelo usando uma distribuição baseada na entropia ótima da imagem. Pham e Prince [26] obtêm uma segmentação via classificação por lógica fuzzy e Tao *et al.* [27] desenvolveu uma segmentação de imagem utilizando a lógica fuzzy juntamente com a técnica de algoritmo genético.

Há trabalhos, como o de Hu e Chang [28], que realizam a segmentação usando uma estrutura de dados do tipo quadtree, na qual diferentes estratégias de codificação são adotadas para a geração de blocos de diferentes significâncias perceptuais. Um outro trabalho que utiliza a segmentação por quadtree é o proposto por Yueni e To [29]. O método de Zouagui *et al.* [30] constrói um modelo a partir de segmentação funcional (Matias e Evsukoff [14]). Apesar das considerações anteriores, grande parte dos algoritmos de segmentação utilizam o procedimento de crescimento de regiões ([15], [31], [32], [33] e [34]) em que as regiões são formadas a partir do agrupamento dos pixels em função de alguma similaridade (tons de cinza, forma, textura entre outras). Baatz e Schäpe [35] e Darwish *et al.* [36] utilizam a textura como medida de similaridade. O método descrito por Bins *et al.* [15] é um outro algoritmo que foi estudado para o desenvolvimento desta Tese no qual foram aplicados como medida de similaridade: o tom de cinza e a distância euclidiana entre os objetos a serem detectados.

O processo de segmentação consiste em uma divisão de uma imagem em regiões de atributos similares. Para isso, conforme dito anteriormente, a detecção destas regiões podem ser feitas de dois modos: através da determinação da fronteira (borda) da região ou através do agrupamento de pontos com o uso da similaridade entre pixels.

3.1 Segmentação por Descontinuidades

Segmentações produzidas por técnicas de detecção de descontinuidades (bordas da imagem) procuram encontrar limites entre regiões homogêneas de diferentes propriedades, e necessitam de operadores específicos que realcem estas descontinuidades antes da segmentação.

Uma borda em uma imagem monocromática é uma variação brusca no nível de cinza entre duas regiões relativamente homogêneas. Cada região (Haralick e Shapiro [37]) é uniforme ou homogênea de acordo com alguma propriedade da imagem. Existem três tipos básicos de descontinuidades em imagens digitais: pontos, linhas e bordas. A maneira mais prática de encontrar uma descontinuidade em uma imagem digital, é a partir do uso de operadores diferenciais. Existem duas grandes classes de operadores diferenciais:

- Diferenciador de Primeira Ordem
- Diferenciador de Segunda Ordem

O Diferenciador de Primeira Ordem utilizado é o gradiente. A partir dele, pode ser feita detecção de bordas (contornos), da seguinte maneira: o valor do gradiente em cada pixel é comparado com um limiar (*threshold*) em tons de cinza. A borda surgirá se o gradiente da função no pixel de coordenadas (x, y) exceder este valor de limiar. Os operadores gradientes mais usados para a detecção de contornos, linhas verticais e linhas horizontais, são os operadores de Roberts e Sobel. As máscaras abaixo mostram estes operadores no mundo discreto e a Equação 3.1 representa o gradiente de uma função de duas variáveis no espaço contínuo.

0	-1
1	0

-1	0
0	1

Figura 3.1 - (a) Máscaras para o operador de Roberts

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Figura 3.1 - (b) Máscaras para o operador de Sobel

As máscaras da Figura 3.1 (a) e (b) são utilizadas sobre uma imagem digital com uma convolução bidimensional discreta definida no item 2.6.

Seja $f(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$ uma função contínua, então o Gradiente é dado por:

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (3.1)$$

Os Diferenciadores de Segunda Ordem utilizam a diferenciação espacial de segunda ordem para realçar as bordas. Uma borda existirá se ocorrer uma mudança brusca na segunda derivada (Marr [38]). Um dos diferenciadores de segunda ordem mais utilizados na área de visão computacional é o Laplaciano.

Seja $f(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}$ uma função contínua, então o Laplaciano da função $f(x, y)$ é dado por:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3.2)$$

No domínio discreto o laplaciano da função $f(x, y)$ pode ser descrito pela matriz de ordem 3x3 representado na Figura 3.3 e que é aplicado conforme a convolução bidimensional discreta definida no item 2.6.

0	1	0
1	-4	1
0	1	0

Figura 3.3 - Máscara com os valores do Laplaciano

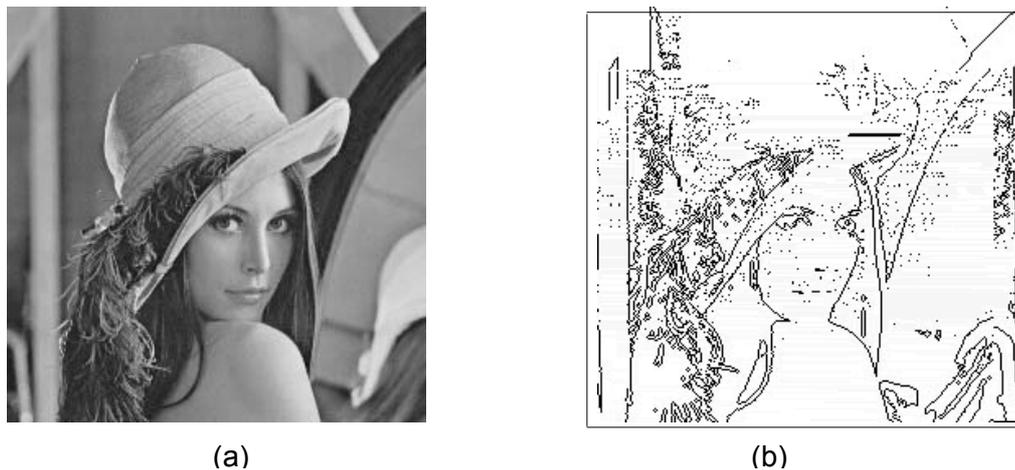


Figura 3.4 – (a) Imagem original (b) Laplaciano discreto

Técnicas de detecção de descontinuidades têm sido largamente empregadas para segmentação de imagens digitais.

3.1.1 Detecção de Pontos

A forma mais simples de segmentar uma imagem sob a forma de contornos é de segmentá-la na forma de pontos particulares relevantes. Isto supõe, muitas vezes, a eliminação de forma eficiente do ruído para evitar falsas detecções. Esses pontos particulares relevantes pesquisados são pontos isolados que devem conter muita

informação e não devem ser deteriorados por um ruído indesejado. Usando a máscara mostrada na Figura 3.5, diz-se que um ponto foi detectado na posição da máscara se:

$$|f(x, y)| > L \quad (3.3)$$

em que L é um limiar não negativo.

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

Figura 3.5 - Uma máscara usada para a detecção de pontos isolados a partir de um fundo constante.

3.2 Segmentação por Região

A detecção de regiões numa imagem pode ser feita com dois objetivos: 1) extrair uma determinada região ou 2) dividir (particionar) a imagem num conjunto de regiões disjuntas cuja união representa a imagem por inteiro. Uma região de uma imagem é um conjunto de pontos onde, de qualquer ponto da região, pode-se chegar a qualquer outro ponto por um caminho inteiramente contido nesta região. As regiões a serem detectadas, em geral, são bastante homogêneas, ou seja, apresentam alguma propriedade local aproximadamente constante. Essa propriedade de continuidade, em imagens monocromáticas, geralmente é o nível de cinza. A textura também pode ser utilizada como uma medida de similaridade (Baatz e Schäpe [35] e Darwish *et al.* [36]), assim como a cor e a intensidade (Zouagui *et al.* [30]). Os diferentes métodos para o particionamento em regiões e que foram desenvolvidos neste trabalho são:

- Limiarização (*threshold*)
- Divisão e fusão (*split and merge* ou *cluster*)
- Crescimento de regiões (*region-growing*)

3.2.1 Segmentação por Limiarização (*Thresholding*)

A limiarização consiste em converter imagens em tons de cinza para imagens binárias, por isso essa técnica também é conhecida como binarização. A forma mais simples de binarização consiste na partição do histograma, convertendo os pixels cujo tom de cinza é maior ou igual a um certo valor de limiar L em brancos e os demais em pretos.

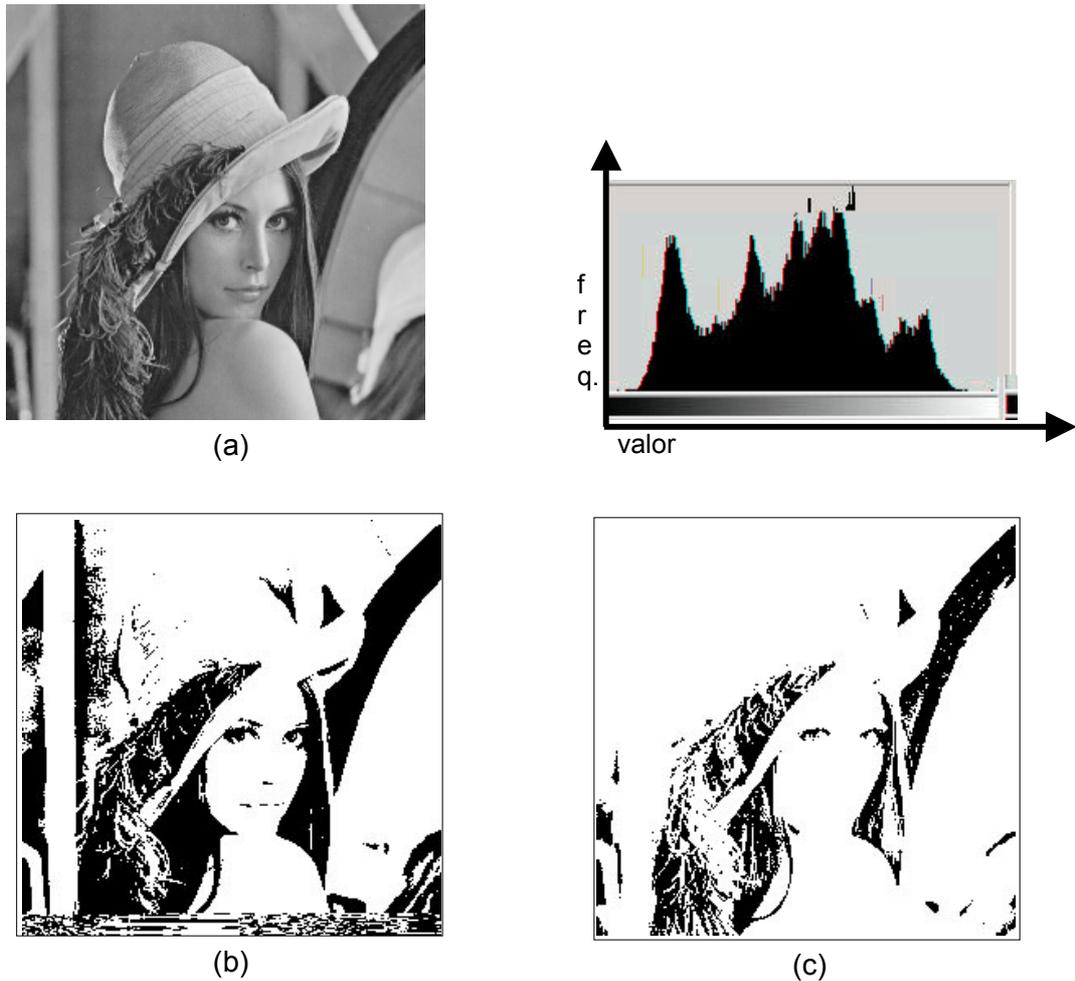


Figura 3.6 (a) Imagem Original e seu respectivo histograma.
(b) Imagem com Limiar L_1 igual a 100
(c) Imagem com Limiar L_2 igual a 60.

Na operação de limiarização, uma imagem de entrada $f(x, y)$ com N tons de cinza produz na saída uma imagem $g(x, y)$ chamada imagem limiarizada ou binarizada, sendo:

$$g(x, y) = 1 \text{ se } f(x, y) \geq L \text{ ou } g(x, y) = 0 \text{ se } f(x, y) < L \quad (3.4)$$

onde L é um valor de nível de cinza denominado limiar. A qualidade de uma imagem limiarizada depende do valor de L . Assim, é importante definir um valor ótimo para L de forma que a imagem não sofra limiarização inadequada. Quando a limiarização é definida de forma manual, o valor de L pode ser escolhido visualmente. No entanto, a maioria das aplicações requerem que valor de L seja obtido de forma automática. Existem diversas técnicas para se definir o valor de L , porém não se pode afirmar qual será a melhor, pois depende das condições da imagem (contraste, iluminação, complexidade da cena) e também do resultado esperado pela limiarização. A técnica de limiarização “ótima” parte de uma imagem da qual se conhecem as principais propriedades estatísticas (supondo que seu histograma possua uma distribuição normal (gaussiana)). Por exemplo:

- μ_1 : média dos tons de cinza da região de interesse
- μ_2 : média dos tons de cinza da região do fundo
- σ_1 : desvio padrão da região de interesse
- σ_2 : desvio padrão da região do fundo
- P_1, P_2 : probabilidade de ocorrência dos pixels pertencentes a cada região

Existe um valor ótimo de limiar L dado pelas raízes da equação 3.5.

$$AL^2 + BL + C = 0 \quad (3.5)$$

Onde:

$$A = \sigma_1^2 - \sigma_2^2 \quad (3.6)$$

- $B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2)$ (3.7)

- $C = \mu_2^2\sigma_1^2 - \mu_1^2\sigma_2^2 + 2\sigma_1^2\sigma_2^2 \ln\left(\frac{\sigma_2 P_1}{\sigma_1 P_2}\right)$ (3.8)

Duas raízes reais e positivas indicam que a imagem pode requerer dois valores de limiar para a obtenção da solução ótima. Conforme já foi citado, Pal e Pal [19] e Sahoo *et al.* [20] propuseram algoritmos de segmentação baseados na limiarização.

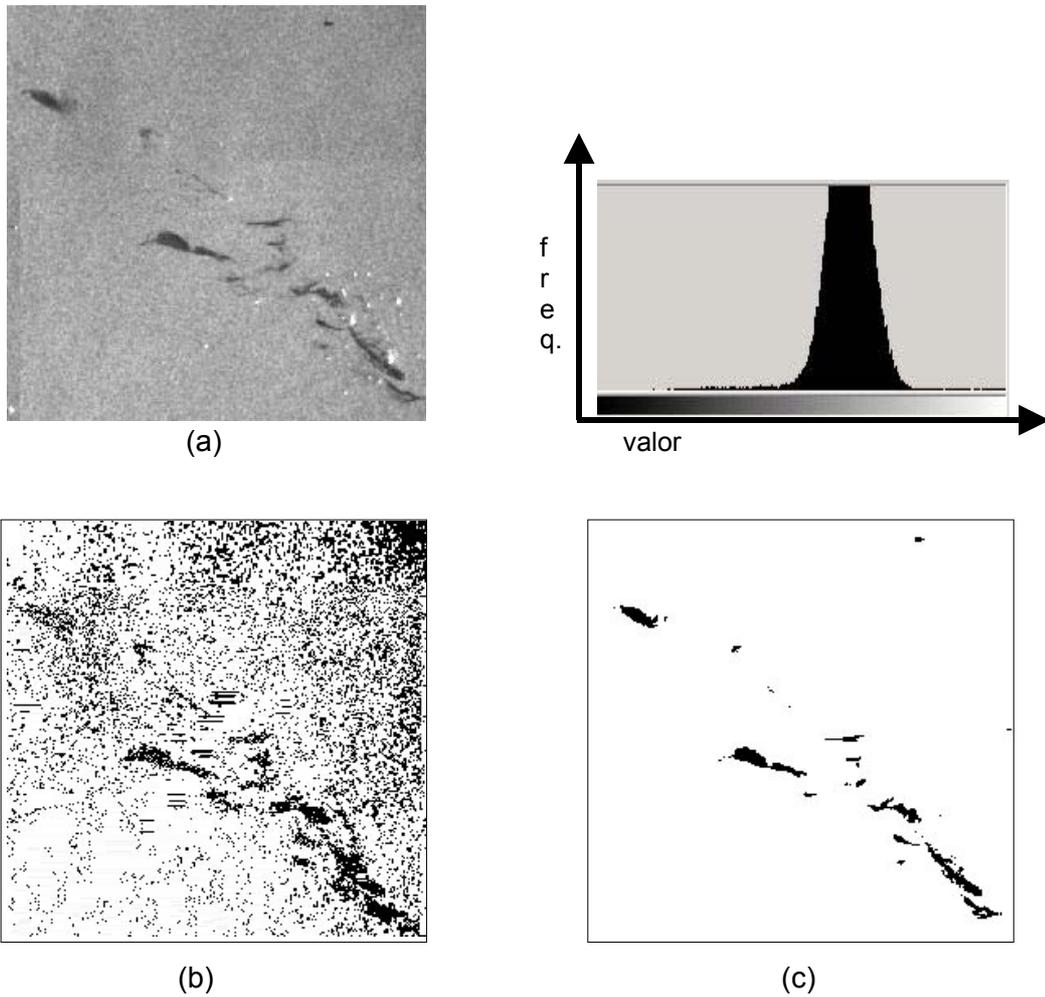


Figura 3.7 (a) Imagem Original e seu respectivo histograma.

(b) Imagem com Limiar L_1 igual a 128 .

(c) Imagem com limiar L_2 igual a 111.

3.2.2 Divisão e fusão

É um método que consiste em agrupar pixels com características em comum, como os seus níveis de cinza, e, dessa forma, criam regiões sobre as quais um critério de homogeneidade seja verificado. A imagem será manipulada através de divisões e fusões até que obtenham regiões que satisfaçam o critério de homogeneidade. Segundo Gonzalez e Wintz [3], uma abordagem para segmentar uma imagem a partir desta técnica consiste em subdividir a imagem de entrada sucessivamente em quadrantes que satisfaçam certa propriedade. Esta abordagem será descrita nesta Tese pela segmentação em quadtree.

3.2.2.1 Segmentação por Quadtree

Samet [39], Sayood [40] e Manouvier *et al.* [41] usam a quadtree para segmentação de imagens. Uma quadtree é uma estrutura de dados em árvore utilizada para codificar imagens, sendo que a quadtree é um tipo especial de árvore onde todos os nós ou são nós folha ou têm quatro nós filhos. Sua utilização principal consiste no armazenamento de uma decomposição recursiva do espaço.

A Figura 3.8 mostra a subdivisão de duas imagens e suas quadtrees correspondentes. A primeira imagem contém uma única cor, logo sua quadtree corresponde será um único nó. Denomina-se o nó que não sofre mais subdivisão de nó cinza. No primeiro exemplo desta figura tem-se uma imagem de cor única, logo será representada por uma quadtree com um único nó.

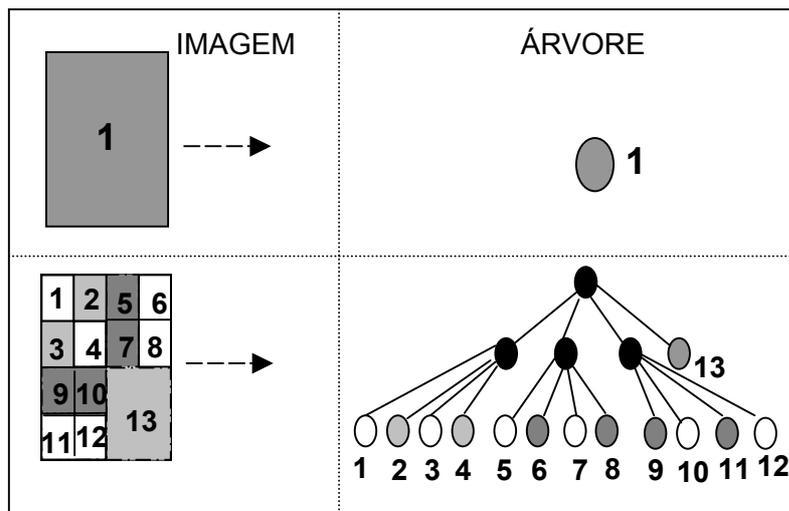


Figura 3.8 - Modelo estrutural da Quadtree

A figura 3.9, abaixo, mostra a geração da quadtree em uma imagem em tons de cinza. Devido a problemas de iluminação e representação do objeto, que possui algumas particularidades difíceis de segmentar, a quadtree foi gerada por várias subdivisões sucessivas.

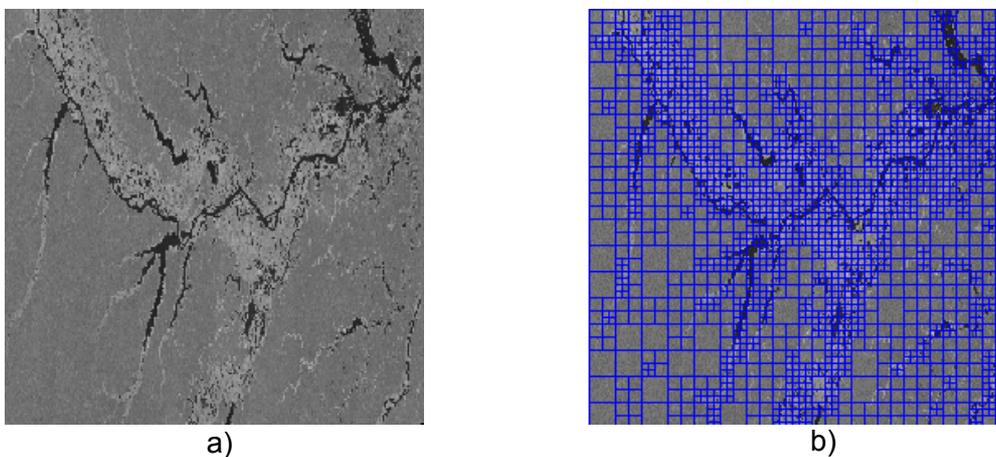


Figura 3.9 - (a) Imagem Original

(b) Quadtree gerada

Esta abordagem por quadrantes pode vir a gerar no resultado da segmentação final regiões com formas quadradas.

Algoritmo 1 Criação de nó em quadtree

```
tipo registro nó {
```

```

BYTE cor; /* Valores de 0 a 255 */
Registro Nó ↑nw, ↑ne, ↑sw, ↑se;
} Nó;

1: Nó ↑NovoNó (inteiro cor)
2:  Nó ↑ret; ; /* Ponteiro para um novo nó criado */ ;
3:  Se ( !(ret = ( Nó ↑ ) alloca ( sizeof(Nó) ) ) )
4:      Aborta();
5:  Fim do Se
6:  ret↑cor = cor;
7:  ret↑nw = ret↑ne = ret↑sw = ret↑se = 0;
8:  retorna ret;

```

3.2.3 Segmentação por Crescimento de Regiões

O crescimento de regiões é um procedimento que reúne (agrupa) pixels ou sub-regiões em regiões maiores (Gonzalez e Woods [4] e Bins *et al.* [15]). Uma das abordagens mais simples por crescimento de regiões é a agregação de pixels, que começa com um conjunto de pontos sementes (Figura 3.10) e, a partir deles, as regiões vão crescendo ao anexar a cada ponto semente os pontos que possuem alguma similaridade entre si. Para que a segmentação seja efetuada com sucesso, é necessário que cada região R siga as seguintes regras:

1. $\bigcup_{i=1}^n R_i = R$,
2. R_i é uma região conexa, $i = 1, 2, \dots, n$,
3. $R_i \cap R_j = \emptyset \quad \forall i \neq j$,
4. $P(R_i) = \text{Verdadeiro}$ para $i = 1, 2, \dots, n$,
5. $P(R_i \cup R_j) = \text{Falso}$ para $i \neq j$.

em que $P(R_i)$ é um predicado lógico sobre os pontos do conjunto R_i e ϕ é o conjunto vazio.

A primeira condição indica que cada pixel deve pertencer a uma única região. A segunda condição requer que os pixels pertencentes a uma região sejam conexos. A terceira cláusula indica que as regiões são distintas entre si. A penúltima cláusula postula que os pixels pertencentes a uma mesma região devem possuir uma propriedade em comum, por exemplo, a homogeneidade. A última condição indica que duas regiões distintas não possuem uma propriedade em comum no sentido do predicado lógico P . A maioria dos algoritmos por crescimento de regiões utiliza as condições acima, mas Bins *et al.* [15] cita em seu artigo as modificações propostas por Schoenmakers *et al.* [42] que são restrições para adaptar os algoritmos de segmentação por crescimento de regiões a uma heurística apropriada a um determinado problema em consideração.

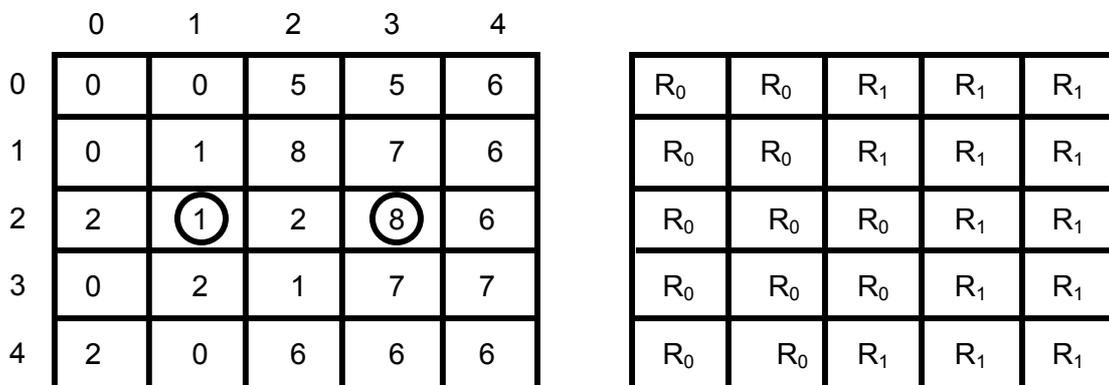


Figura 3.10 – Algoritmo de Crescimento de Regiões: Imagem Original e o resultado por uma similaridade de valor 3.

Conforme mostrado na Figura acima, há dois pixels que estão marcados com um círculo e que correspondem ao que denominamos de sementes do algoritmo. É a partir dessas sementes iniciais que os pixels vão se agrupando ao seu redor e desta forma vão gerando as regiões. O problema está exatamente na quantidade de

sementes e como encontrá-las. Isso depende da resolução da imagem e também de finalidade da mesma. Outro fator a ser levado em consideração é o fator de crescimento das regiões que está relacionado a algum valor de similaridade como mostrado na Figura 3.11 abaixo.

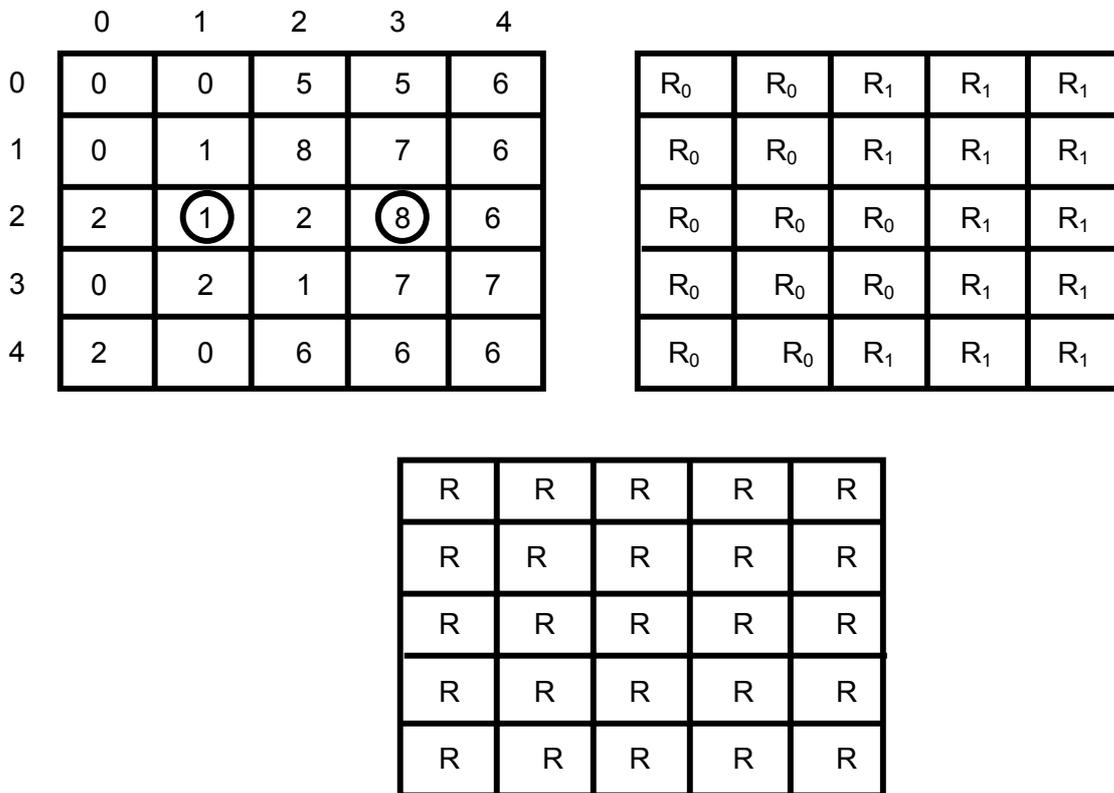


Figura 3.11 – Imagem Original, resultado parcial e resultado final utilizando um valor de similaridade igual 8.

A escolha dos pontos sementes pode ser feita tanto manualmente quanto automaticamente. Quando ela é feita manualmente, o usuário que está manuseando a imagem, geralmente coloca os pontos sementes nas regiões em que se deseja obter uma segmentação de um determinado objeto. Quando a escolha dos pontos sementes é feita de modo automático pode-se usar técnicas baseadas no gradiente e na análise de histograma (Gonzalez e Woods [4], Adams e Bishof [31]). A Figura 3.12 exibe um exemplo prático da técnica de crescimento de regiões.

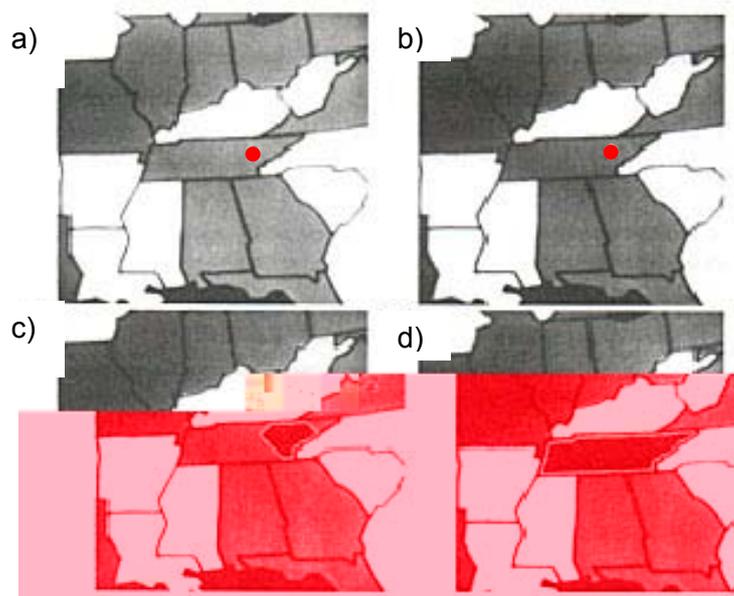


Figura 3.12 – (a) Imagem Original com a semente (b) Início do Algoritmo
(c) Passo Intermediário (d) Região encontrada

Para representar as imagens correspondentes aos algoritmos de crescimento de regiões, dois tipos de algoritmos de carácter generalizado serão descritos a seguir: Crescimento de Regiões Segundo a Varredura e Crescimento de Regiões por Busca em Todas as Direções. No primeiro caso, a imagem é percorrida de cima para baixo e da esquerda para a direita. Cada pixel p de coordenadas (x, y) é comparado com seus vizinhos anteriores (já rotulados) e adicionado ou não a região de um dos vizinhos.

Algoritmo 2 Crescimento_de_regiões_Segundo_Varredura

- 1: $k = 0$;
- 2: Percorra a imagem segundo a varredura;
- 3: Compare cada pixel $p(x, y)$ com seus vizinhos anteriores; duas situações podem acontecer:
- 4: $p(x, y)$ não pode ser adicionado a região de nenhum vizinho anterior por violar a condição de similaridade (homogeneidade, por exemplo); neste caso inicie uma nova região: $k = k + 1$;
 $R(p) = k$;

5: $p(x,y)$ pode ser adicionado a região de $q(x,y)$ (seu vizinho anterior); neste caso adicione p a região de q : $R(p)=R(q)$.

6: **Fim**

Os vizinhos anteriores ao pixel p são os pixels com coordenadas $(x-1,y)$, $(x,y-1)$, $(x-1,y-1)$, $(x-1,y+1)$; e $R(p)$ indica a região do pixel p . Ao contrário do crescimento de regiões segundo a varredura, onde várias regiões são “crescidas” simultaneamente, no crescimento de regiões por busca em todas as direções, uma única região é crescida por vez. O algoritmo abaixo mostra o que foi citado.

Algoritmo 3 Crescimento_de_regiões_por_buca_em_todas_as_direções

1. $k = 0$;
2. Escolha um pixel $p(x,y)$, tal que $R(p)=0$, se não existe ponto nesta condição, então pare; fim.
3. **Inicie** nova região: $k = k + 1$; $R(p) = k$;
4. Encontre todos os pontos da vizinhança de $p(x,y)$, tal que $R(q)=0$ e q possa ser adicionado a região de p sem violar o critério de similaridade: faça $R(p)=R(q)$ e armazene estes pixels em uma região Z ;
5. **Se** Z estiver vazio, então vá para passo 2; **senão** escolha e extraia um pixel p de Z ;
6. **Repita** passos 4 e 5.
7. **Fim**

É válido observar que para ambos algoritmos os pixels são conectados por conexões simples, isto é os pixels vizinhos que possuem alguma similaridade são simplesmente agrupados. Existe um outro tipo de conexão entre pixels que se baseia no centróide. Nesta conexão os pixels não são agrupados por uma medida de similaridade, e sim por um valor médio da região.

3.2.3.1 O Algoritmo de Bins *et al.*

Bins *et al.* [15] utilizam uma técnica de crescimento de regiões tradicional, com algumas modificações que ajudam a resolver o problema da dependência dos *merges* (junção de duas regiões). Antes de descrever o esquema de segmentação algumas notações e definições são fornecidas. \mathfrak{R} é usado para denotar o conjunto de regiões da imagem e $R \in \mathfrak{R}$ é um elemento deste conjunto. Seja $L(t)$ o valor de limiar inferior ao qual duas regiões são consideradas similares no instante t , e seja M_i o vetor de valor médio da região R_i . Seja $D(R_i, R_k) = \|M_i - M_k\|$ a distância euclidiana entre os valores médios espectrais das regiões R_i e R_k , e seja $N(R)$ o conjunto de regiões vizinhas a R (não incluindo ela mesma). A região R_k é a região adjacente mais similar a região R_i se $D(R_i, R_k) \leq D(R_i, R_l) \forall R_l \in N(R_i)$.

Este algoritmo que é o mesmo utilizado na biblioteca *TerraLib* (biblioteca open-source para GIS) e que é implementado nesta Tese, é descrito da seguinte maneira:

Algoritmo 4 Crescimento_de_regiões_TerraLib

1. No início do processo de segmentação uma lista de regiões $\{R_i, i=1, \dots, n\}$ é criada (n é o número de pixels na imagem). Inicialmente cada região é composta por um único pixel, chamado "semente". Para cada região R_i , seu vetor de valor médio e as regiões vizinhas são armazenadas;
2. Para cada região R_i suas regiões vizinhas $N(R_i)$ são examinadas e:
 - . A região vizinha mais similar $R_k \in N(R_i)$ é escolhida. Se $D(R_i, R_k) < L(t)$ então R_k é denominado "melhor vizinho" de R_i .
 - . Se o melhor vizinho de R_k existe e é R_i , então ambas as região são agrupadas (*merge*).

3. A cada passo uma região é agregada e uma outra é retirada da lista;
4. O valor médio é recalculado toda vez que ocorrer um agrupamento entre regiões;
5. O procedimento se repete até que não haja mais regiões a serem agregadas;
6. No final, pequenas regiões são agregadas com regiões adjacentes maiores, de acordo com um valor de área que é limitado por um atributo fornecido pelo usuário.
7. **Fim**

No algoritmo de Bins *et al* $T(t) = \alpha^t * T(0)$, com $T(0) > 0$, $t = 0,1,2,\dots$ e $\alpha < 1$.

Isto impõe que os merges iniciais sejam mais difíceis de serem realizados do que os *merges* no fim do processo de segmentação, isto é, somente regiões muito similares são agrupadas primeiramente.

3.2.3.2 Segmentação por Multiresolução

O procedimento de segmentação em multiresolução pode ser descrito como uma técnica de *region merging* na direção *bottom-up* (modelo hierárquico de baixo para cima, isto é, do pixel até o SuperObjeto [35], [36]). Vale lembrar que os dois trabalhos citados acima se baseiam na técnica de segmentação que foi implementada para o software de Análise Orientada a Objetos (*eCognition*). Este algoritmo é baseado em dados texturizados, pois o mesmo analisa melhor imagens aéreas, de Radar ou de Satélites, isto é, imagens onde a textura aparece como fonte principal de áreas “homogêneas”.

Este algoritmo de segmentação que está incluso no *eCognition* e que é proposto por Battz e Schäpe [35] e Darwish *et al.* [36] inicia-se com cada pixel

formando um objeto ou uma região. A cada passo do algoritmo um par de objetos é comparado para estabelecer a condição de *merging* (junção) e, caso a condição seja verdadeira, cria-se um novo objeto. Para que um objeto se junte a outro é necessário que estes satisfaçam a uma determinada similaridade, ou melhor, a um critério de homogeneidade. Para cada possível junção entre os objetos é calculado um grau de ajuste, denominado h , que é avaliado para o caso de ocorrer a junção. Isto é, dado um certo espaço denominado de característico, dois objetos da imagem são similares se eles são vizinhos no espaço característico ct .

Dessa forma o grau de ajuste h é calculado pela Equação 3.9 abaixo:

$$h = \sqrt{\sum_d (ct_{1d} - ct_{2d})} \quad (3.9)$$

Muitas vezes a distância entre os objetos é normalizada pelo desvio padrão.

Desta maneira a Equação 3.9 pode ser substituída pela Equação 3.10:

$$h = \sqrt{\sum_d \frac{(ct_{1d} - ct_{2d})}{\sigma_{cd}}} \quad (3.10)$$

O algoritmo pára quando é satisfeita a condição de parada, isto é, quando não é mais possível se obter nenhuma junção entre os objetos. Quanto menor for o grau de ajuste menor será a quantidade de objetos a serem agrupados. Segundo Battz e Schãpe [35] e Darwish *et al.* [36] a segmentação em multiresolução engloba dois componentes principais:

- Decisões heurísticas para determinar quais objetos são agrupados a cada passo do algoritmo;
- Definição de uma homogeneidade entre os objetos para computar o grau de ajuste para um determinado par de objetos.

Para se determinar se um dado objeto A será agrupado a um objeto B há quatro diferentes possibilidades para uma decisão heurística. Os quatro graus de liberdade estão divididos em:

- Ajuste (*Fitting*): O objeto A agrupa-se a um objeto B caso o critério de homogeneidade seja satisfeito;
- Melhor Ajuste (*Best Fitting*): O objeto A agrupa-se a um objeto B caso seja encontrado o melhor critério h;
- Melhor Ajuste Mutuamente Local: Se o objeto A encontra o objeto B pelo qual o critério h é estabelecido e se o objeto B encontra o objeto C também pelo critério h, então reciprocamente A se agrupa a C;
- Melhor Ajuste Mutualmente Global: Agrupar aos pares os objetos da cena que satisfazem o critério h.

Battz e Schãpe [35] e Darwish *et al.* [36] utilizaram como critério de segmentação, isto é, para que se tenha um *merge* (agrupamento) virtual entre dois objetos uma métrica em que a modificação da heterogeneidade (h_{diff}) entre os objetos deve ser minimizada que é dada por:

$$h_{diff} = h_m - \frac{(h_1 + h_2)}{2} \quad (3.11)$$

Onde:

h_m → Heterogeneidade Média dos objetos

h_1, h_2 → Heterogeneidade entre objetos adjacentes

Para este modelo proposto pelos autores a heterogeneidade pode estar relacionada com os valores médios da variância ou do desvio padrão.

Conforme já foi citada, a segmentação em multiresolução do software *Ecognition* é utilizada para construir um modelo hierárquico dos objetos que representam a informação da imagem simultaneamente em diferentes resoluções espaciais. Os objetos na imagem são interconectados, de tal forma que cada objeto reconheça sua vizinhança, seus superobjetos e seus subobjetos, conforme Figura 3.13. Assim, é possível definir relações entre objetos e utilizar este tipo de informação para o contexto local. Dessa maneira, um superobjeto (pai) pode ser dividido em vários subobjetos (filhos) e vice-versa, assim como os subobjetos (pais deste nível) de nível superior também possam ser divididos em vários subobjetos (filhos deste nível), seguindo a mesma definição anteriormente descrita.

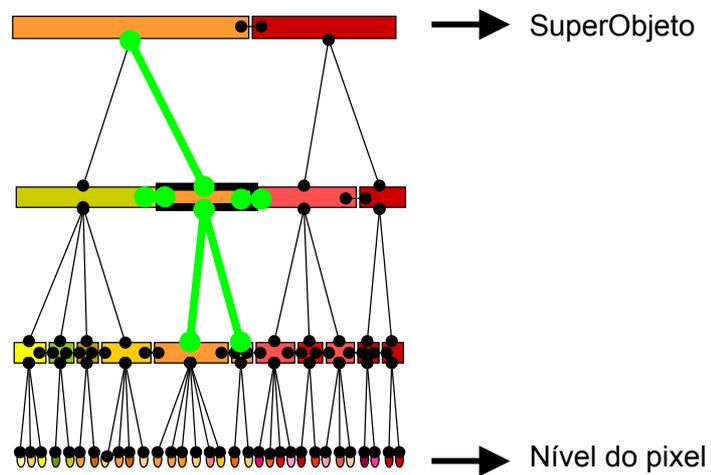


Figura 3.13 - (a) Rede hierárquica dos objetos na imagem.

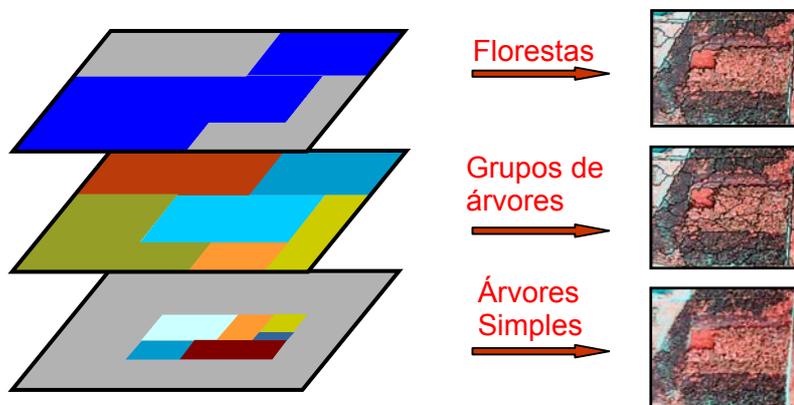


Figura 3.13 - (b) Segmentação para cada nível de resolução

3.3 Resumo

Neste capítulo, foram descritos os tipos de algoritmos de segmentação estudados para o desenvolvimento deste trabalho, ressaltando os algoritmos baseados em limiar, divisão e fusão e crescimento de regiões que foram implementados para a Tese. A utilização destes três tipos de algoritmos se deu ao fato de que cada um ser bem distinto do outro, podendo dessa maneira aplicar a metodologia proposta nesta Tese para cada um deles.

Capítulo 4

Métricas Utilizadas para Avaliação de Segmentação

O critério de performance para validação e comparação de algoritmos de segmentação tem sido um interesse crescente para pesquisadores nesta área. No artigo de Zhang [43], a descrição desta comparação é dividida em duas técnicas: avaliação analítica e empírica. Segundo Zhang [43] os métodos analíticos se referem à análise dos algoritmos de segmentação com base em suas propriedades, tais como:

- O conhecimento prévio que foi incorporado ao algoritmo;
- A estratégia de processamento se é paralela, interativa ou ambas;
- A eficiência e/ou complexidade do processamento.

Os métodos empíricos, de outro modo, avaliam o resultado, verificando somente a sua saída. Entre eles, alguns são orientados a avaliar a boa qualidade da saída (métodos de boa qualidade empírica) enquanto outros produzem alguma discrepância entre o resultado gerado pelo algoritmo (métodos de discrepância empírica) e o resultado esperado. Devido a isso, esse capítulo tem como objetivo descrever algumas diferentes medidas de avaliação, mas com maior enfoque na métrica de avaliação usada para esta Tese.

4.1 Avaliação Segundo Zhang

Na classificação de Zhang [43], [44], os métodos analíticos evitam a implementação do algoritmo, e assim, reduzem a influência das combinações de experimentos. Contudo, nem todas propriedades de um algoritmo de segmentação podem ser capturadas por estudos analíticos. Dessa forma, os métodos empíricos possuem maior escopo e são mais abrangentes. Devido a essa classe de métodos, os métodos empíricos de boa qualidade validam algumas propriedades das regiões e as comparam com alguns valores considerados bons para as regiões de saída. Em estudos realizados (Levine e Nazif [46]; Pal e Bhandari [47]; Zhang [45]; Rosenberger [48]; Cardoso e Real [49]) a uniformidade intra-região e o contraste inter-região são boas medidas de qualidade, orientadas a regiões, para uma boa segmentação.

Um método de discrepância empírica, quando comparado a um método empírico de boa qualidade (*goodness*), pode até ser mais generalizado, desde que a avaliação seja baseada em diferenciar a saída gerada pelo algoritmo com uma imagem de referência já segmentada. O número de pixels incorretamente classificados como pixels de arestas ou o número de pixels incorretamente segmentados, sua posição e o número de regiões estão entre as diferentes medidas de discrepâncias existentes na literatura. (Huang e Dom [50], Lewis e Brown [51]). Os métodos baseados em discrepância .4(meâ)5.2ializem a difâ uma(imaú)-5.7zem tqamde uma(l)-632(m)3.4aebass de eados dententes])

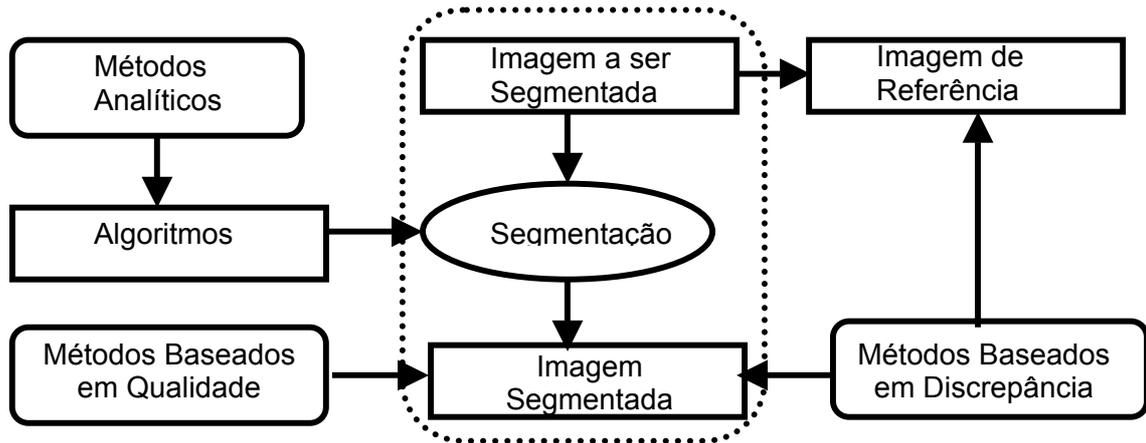


Figura 4.1 – Esquema simplificado para avaliar a segmentação

Segundo Zhang [44] os métodos baseados em qualidade e os métodos baseados em discrepância podem ser classificados conforme a Tabela abaixo:

Método	Métricas de Avaliação
Boa Qualidade	Uniformidade intra-região
Boa Qualidade	Contraste inter-região
Boa Qualidade	Forma da região
Discrepância	Número de pixels segmentados com erro
Discrepância	Posição dos pixels segmentados com erro
Discrepância	Número de objetos na imagem
Discrepância	Características dos valores dos objetos segmentados
Discrepância	Características mistas

Tabela 4.1 – Métricas utilizadas em avaliações empíricas

Em outro estudo de Zhang [52] é descrito um método quantitativo e objetivo de algoritmos de segmentação. Este estudo é diferente de outros, pois considera o aspecto de avaliação da segmentação. A avaliação e a comparação de algoritmos de segmentação tende a ser indispensável para que possa usá-los de forma adequada. Zhang [52] avalia o resultado da segmentação a partir de imagens geradas por diferentes algoritmos de segmentação e descreve métricas de discrepância baseadas nas características da região como: área, excentricidade ou perímetro.

4.1.1 Caracterização da performance

O estudo do critério quantitativo e objetivo discutido em Zhang e Gerbrands [53] é utilizado para caracterizar a performance do algoritmo de segmentação. Seja S_r a notação de um valor característico obtido em uma imagem de referência e seja S_f a notação de um valor cuja característica é uma medida de uma imagem segmentada, o critério de performance é denominado por: *Relative Ultimate Measurement Accuracy*. Esta medida pode ser calculada por:

$$RUMA_f = \left(\frac{|S_r - S_f|}{S_r} \right) \times 100\% \quad (4.1)$$

Os valores de $RUMA_f$ são inversamente proporcionais aos resultados da segmentação: o menor dos valores, o melhor é a qualidade. As Características utilizadas na equação 4.1 podem ser selecionadas de acordo com as metas do algoritmo de segmentação assim como os requerimentos de avaliação e comparação. Na prática dos experimentos de avaliação e comparação, os algoritmos selecionados são primeiros aplicados a imagem teste, e então, S_f pode calculado a partir das imagens segmentadas. S_r também pode ser obtido por um processo similar da imagem de referência produzido durante o processo de geração da imagem. Logo, com S_f e S_r computados, $RUMA_f$ pode ser calculado.

4.1.2 Avaliação da segmentação

Segundo Zhang [52] a avaliação da segmentação pode ser dividida em duas etapas. Uma delas é a utilização do mesmo parâmetro de ajuste do algoritmo para segmentar as várias imagens e a outra é fornecer valores diferentes aos parâmetros do algoritmo

para segmentar algumas imagens a serem comparadas e então avaliar a influência destes valores distintos utilizados neste algoritmo sobre seu desempenho.

4.2 Avaliação Segundo Huang e Dom

O método de avaliação proposto por Huang e Dom [50] baseia-se em dois conjuntos de medidas para avaliar uma imagem segmentada, são elas: o primeiro é quando se tem uma base de dados de resultados de imagens segmentadas (*ground truth*) para comparar com a imagem gerada por algum algoritmo de segmentação, e o segundo caso é quando não se tem a base de dados para a comparação. Segundo [50] quando se tem uma base de dados de imagens disponível, a performance de avaliação é dividida em três partes: baseado por parâmetros, baseados por fronteiras e baseado em regiões. De acordo com Huang e Dom, quando se trabalha com *ground truth*, a segmentação pode ser avaliada em termos de exatidão e robustez. Neste caso, a exatidão reflete a precisão da segmentação com respeito a imagem modelo (imagem de comparação) e a robustez está relacionada com a degradação da exatidão com respeito a qualidade do dado testado.

Quando não se tem uma imagem modelo, segundo [50] avaliar a segmentação é similar a avaliar a validade dos clusters (regiões). Um algoritmo de segmentação particiona uma imagem I em várias regiões não sobrepostas, $\{R_1, \dots, R_M\}$ tal que $\bigcup_{i=1}^M R_i = I$, $R_i \cap R_j = \phi$, $\forall i \neq j$. Um critério de criar tal partição seria a homogeneidade entre as regiões segmentadas e a heterogeneidade entre regiões distintas serem simultaneamente maximizadas. Seja H_w a heterogeneidade total interna a região e H_b a heterogeneidade entre regiões. A primeira medida genérica proposta por [50] para avaliar dados sem *ground truth* é:

$$\rho = \left(\frac{H_b + 1}{H_b + H_w + 1} \right) \quad (4.2).$$

4.3 Avaliação Segundo Oliver e Ortiz

O método de avaliação utilizado nesta Tese baseia-se no artigo de Ortiz e Oliver [13] que descreve o comportamento de algumas medidas para checar a performance de um algoritmo de segmentação, quando imagens de qualquer tipo de cena são envolvidas na comparação, e também, propõe uma nova medida de performance que é independente da imagem a ser testada ou classificada na análise da performance.

4.3.1 Avaliação da performance

Ortiz e Oliver [13], em seu artigo, comentam que às vezes uma simples métrica de avaliação pode não ser o bastante para julgar o resultado obtido por um algoritmo de segmentação. Devido a isso, segundo [13], Liu e Yang [54] propuseram uma função (Equação 4.3) baseada em critérios heurísticos criados por Haralick e Shapiro [37] e que são relacionados a:

- As formas dos objetos (regiões) devem ser simples e com poucos buracos;
- As regiões devem ser uniformes e homogêneas;
- As regiões adjacentes devem possuir características distintas.

$$F(I) = \frac{1}{1000 * (N * M)} * \sqrt{R} * \sum_{i=1}^R \frac{e_i^2}{\sqrt{A_i}}, \quad (4.3)$$

onde $(N * M)$ é a resolução da imagem, R é o número de regiões, A_i é a área da região i e e_i é o erro relacionado a cor. Porém, em trabalhos mais recentes (Borsoti *et*

a). [55]) melhoraram a Equação 4.3 após provarem empiricamente que essa expressão, em geral, não satisfazia os critérios de Haralick e Shapiro [37].

4.3.2 Avaliação da Segmentação por Matriz de Confusão

Uma gama de métodos de discrepância empírica são baseados em uma matriz C que contém informações sobre a distribuição dos pixels entre as diferentes regiões da imagem. Cada entrada da matriz C , isto é, um elemento $C(i, j)$ contém o número de pixels da região i na imagem de referência (imagem segmentada para comparação) que são rotuladas como pertencentes a região j na imagem de saída. Um caso particular dessa matriz no qual, entre outras características, C é quadrada, é denominada como matriz de confusão, matriz de erros ou matriz de contingência.

No contexto de classificação, a percentagem de pixels classificados corretamente ou a percentagem de pixels classificados incorretamente são medidas de discrepâncias mais fáceis de serem obtidas a partir da matriz de confusão C . Em um caso de N classes (multi-classe) C é uma matriz de ordem $N \times N$ e a percentagem de pixels classificados corretamente é dado por:

$$M = \frac{\sum_{i=1}^N C_{ii}}{\sum_{i,j=1}^N C_{ij}} \quad (4.4)$$

Assim, para o caso de pixels serem classificados incorretamente, as percentagens de falsos positivos (erro tipo I para multi-classe) para a classe k é expressa por:

$$M_I^{(k)} = \frac{\left(\sum_{i=1}^N C_{ki} \right) - C_{kk}}{\sum_{i=1}^N C_{ki}} * 100 \quad (4.5)$$

enquanto a percentagem de falsos negativos (erro tipo *II* para multi-classe) para a classe *k* é obtida por:

$$M_{II}^{(k)} = \frac{\left(\sum_{i=1}^N C_{ik} \right) - C_{kk}}{\sum_{i=1}^N \sum_{j=1}^N C_{ij} - \sum_{i=1}^N C_{ki}} * 100 \quad (4.6)$$

4.3.3 Novas estratégias de Avaliação baseadas em discrepância

Quando se fala em avaliar um resultado obtido, por um algoritmo de segmentação, em muitos casos, é utilizada uma imagem de referência (imagem para comparação). Muitas vezes o resultado gerado pelo algoritmo gera uma quantidade de regiões que diferente do número de regiões da imagem de referência. Isso ocorre devido a dois comportamentos errôneos que podem acontecer sozinhos ou em combinações: (1) uma região na imagem de referência pode aparecer dividida em várias regiões, o qual é conhecida pelo termo *over-segmentation* (segmentação excedente); e (2) várias regiões na imagem de referência podem ser agrupadas em uma única região na imagem resultado, e isto é conhecido como *under-segmentation* (baixa segmentação). Devido a isso, foi criada uma nova matriz, decorrente da matriz de confusão, porém não necessariamente ela é quadrada. A essa nova matriz foi denominada de Matriz de Sobreposição de áreas (*MSA*).

Um conjunto de novas métricas é introduzido com características especiais da MSA. Para efeito de notação a região i é denotada por R_i , enquanto a região j será denotada por R_j . Cada métrica de discrepância está descrita pelas equações abaixo:

Tamanho da região

$$\text{Tamanho da região } \hat{R}_j: n(\hat{R}_j) = \sum_{k=1}^{N_r} C_{kj} \quad (4.8)$$

$$\text{Tamanho da imagem: } = \sum_{=}^r () = \sum_{k=}^{N_s} \quad (4.9)$$

$$US(p) = \frac{\sum_{j=1}^{N_s} (1 - SR_b(\hat{R}_j, p)) * n(\hat{R}_j)}{n(I)} * 100 \quad (4.13)$$

Percentagem de segmentação excedente (*over-segmentation*) no nível p:

$$OS(p) = \frac{\sum_{i=1}^{N_r} (1 - SO(R_i, p)) * n(R_i)}{n(I)} * 100 \quad (4.14)$$

N_r é o número de regiões de referência, enquanto N_s é o número de regiões de saída.

De acordo com as equações, acima descritas, vale a pena descrever as três últimas métricas: *CG*, *US* e *OS*.

- *CG*: A percentagem de pixels corretamente agrupados tem como objetivo contabilizar a quantidade de pixels da região de referência R_i que são computados na região de saída \hat{R}_j ;
- *US*: A percentagem de baixa segmentação representa a quantidade de pixels da imagem que foram atribuídos a região \hat{R}_j que envolve várias regiões de referência R_i . O parâmetro p é um atributo de relaxamento;
- *OS*: A percentagem de segmentação excedente contabiliza os pixels da região de saída \hat{R}_j que dividem a região de referência R_i .

Huang e Dom [50] apresentaram duas quantidades muito similares ao *US* e *OS*, respectivamente, mas não são idênticas. Elas são chamadas de e_R^m (*missing rate* – taxa faltante) e e_R^f (*false alarm rate* – taxa de alarme falsa) e são dadas por:

$$e_R^m = \sum_{i=1}^{N_r} \sum_{\substack{j \neq \arg \max \\ k=1, \dots, N_s} \{C_{kj}\}} C_{ij} \quad (4.15)$$

$$e_R^f = \sum_{j=1}^{N_s} \sum_{\substack{i \neq \arg \max \\ k=1, \dots, N_r} \{C_{kj}\}} C_{kj} \quad (4.16)$$

Segundo a equação 4.15 o valor de e_R^m recai em determinar, para toda a região de referência, a região de saída para qual a comparação é máxima, enquanto e_R^f necessita da correspondência entre toda região de saída e sua região de referência máxima. Uma vez determinada esta correspondência, ambas as métricas acumulam os pixels com valores de interseções não máximos entre a região de referência e a região de saída. Conforme Levine e Nazif [46] essas métricas não pretendem computar o nível de fragmentação de regiões de referência em regiões de saída e vice-versa. Além disso, se uma correspondência é estabelecida entre uma região de referência R_i e uma região de saída \hat{R}_j em e_R^m , \hat{R}_j não necessariamente possui interseção máxima com R_i ao calcular o valor de e_R^f ; como conseqüência disto, o que pode ser considerado como uma classificação incorreta por uma das métricas pode ser considerado como uma classificação correta por outra métrica.

4.4 Avaliação Segundo Udupa *et al*

De acordo com Udupa *et al.* [56] a segmentação da imagem consiste de dois processos que se relacionam: reconhecimento e representação. O reconhecimento é o processo de alto nível para determinar a localização de um objeto de interesse na imagem. A representação consiste em um processo de baixo nível para determinar a precisão espacial ponto a ponto do objeto de interesse na imagem.

Para satisfazer os processos de reconhecimento e representação, Udupa *et al.* [56] utilizaram métodos que avaliam o resultado de uma segmentação baseados em três fatores: precisão, exatidão e eficiência. Cada um destes fatores exerce uma influência em relação ao outro, isto é, é difícil melhorar um fator sem afetar o outro.

4.5 Conclusão sobre as métricas utilizadas

Tomando como base as métricas de avaliação de segmentação descritas ao decorrer deste capítulo, as métricas *OS*, *US* e *CG* (Ortiz e Oliver [13]) são utilizadas nesta Tese para a avaliação do algoritmos de segmentação por crescimento de regiões. O uso dessas métricas se deu devido a tais fatores:

- Conforme testes comparativos realizados por Ortiz e Oliver [13] as métricas *OS*, *US* e *CG* tendem a avaliar com mais exatidão o nível de fragmentação da imagem de saída;
- Todas estas três métricas se baseiam em uma Matriz de Sobreposição de áreas (*MSA*);
- As métricas *OS*, *US* e *CG* são calculadas em função de um parâmetro de tolerância p e que serve como um parâmetro de relaxação para a imagem segmentada;
- Segundo (Ortiz e Oliver [13]) as outras métricas de avaliação podem ser utilizadas como métricas secundárias para fornecer informações adicionais para *OS*, *US* e *CG*.

Capítulo 5

Otimização de Segmentação em Imagens Digitais

Antes que sejam discutidos alguns trabalhos que tratam da segmentação de imagens, com o uso do AG, e, posteriormente, as definições dos algoritmos implementados nesta Tese, vale a pena dar uma introdução inicial sobre os Algoritmos Genéticos. Os AGs, desenvolvidos por Holland [57], constituem uma técnica bastante utilizada em problemas de otimização, e baseiam a lógica de seu funcionamento nas leis de evolução natural propostas por Charles Darwin.

O termo genético diz respeito à maneira como as possíveis soluções para o problema são codificadas em genomas, uma estrutura composta por uma cadeia finita de elementos, os genes. Os AGs são capazes de resolver problemas cuja solução possa ser codificada numa string de valores pertencente a um alfabeto pré-definido. Segundo Goldberg [58] e Vieira *et al.* [59], a base de funcionamento dos algoritmos genéticos pode facilmente ser adaptada à solução de uma grande variedade de problemas, e em muitos casos a parte de codificação efetivamente dependente do domínio destes pode ser limitada à escolha de representação (um ponto crítico para seu bom funcionamento) e da função de avaliação dos indivíduos (também chamada de função objetivo).

Os algoritmos genéticos são amplamente utilizados como otimizadores de funções, nos casos em que os métodos de busca exaustiva falham e é necessário o uso de busca controlada em espaço. Sua vantagem sobre os outros métodos neste

aspecto é seu alto grau de adaptabilidade, robustez e paralelismo. O objetivo desta Tese está em resolver o problema de otimização que em modo geral pode ser representado pelo modelo da Equação 5.1 abaixo:

$$X^* = \arg \min_x f(x), \text{ onde:}$$
$$X^* \in \mathfrak{R} \text{ e } f(x): \mathfrak{R}^n \longrightarrow \mathfrak{R} \quad (5.1)$$

5.1 Características dos AGs

Devido a maneira particular como os AGs operam, são destacadas algumas características:

- Generalidade – Os algoritmos genéticos simulam a natureza em um de seus mais fortes atributos: a adaptabilidade;
- Busca codificada – De acordo com Geyer [60], “os AGs não trabalham sobre o domínio do problema, mas sim sobre representações de seus elementos.” Tal fator impõe ao seu uso uma restrição: para resolver um problema é necessário que o conjunto de soluções viáveis para este possa ser de alguma forma codificado em uma população de indivíduos;
- Paralelismo explícito – o alto grau de paralelismo intrínseco aos AGs pode ser facilmente verificado se considerarmos o fato de que cada indivíduo da população existe como um ente isolado e é avaliado de forma independente;
- Busca estocástica – Segundo Geyer [60], ao contrário de outros métodos de busca de valores ótimos, os AGs não apresentam um comportamento determinístico.

No contexto deste trabalho os AGs são usados em segmentação de imagens digitais no qual o algoritmo de segmentação está inserido dentro dos AGs pelo uso da função de avaliação (função objetivo). A cada nova geração são criados novos filhos, através de mutações e cruzamentos, no qual são avaliados um a um pela função objetivo adequada a cada algoritmo de segmentação.

5.1.1 Representação cromossomial

A representação cromossomial é fundamental para os AGs (Linden [61]). Basicamente ela consiste em uma maneira de traduzir a informação do problema a ser tratado em uma maneira viável de ser produzida pelo computador. A cada pedaço indivisível dessa representação dar-se o nome de gene. Por analogia as partes fundamentais de um cromossomo biológico.

Segundo Linden [61] algumas regras tem que serem seguidas para uma boa representação cromossomial, são elas:

- 1) A representação deve ser a mais simples possível;
- 2) Se houver soluções proibidas ao problema, então elas não devem ter uma representação;
- 3) Se o problema impuser condições de algum tipo, estas devem ser implícitas dentro da representação.

A Figura 5.1 mostra um cromossomo de oito bits onde são representados duas variáveis inteiras s_1 e s_2 . Cada uma dessas variáveis é calculada dividindo o cromossomo em duas partes.

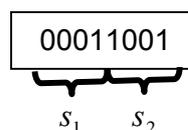


Figura 5.1 – Cromossomo binário

5.1.2 Função de Avaliação

A função de avaliação é a maneira utilizada pelos AGs para determinar a qualidade de um indivíduo como solução do problema em questão. Dada a generalidade dos AGs, a função de avaliação, em muitos casos, é a única ligação verdadeira da aplicação com o problema geral (Linden [61]). A função de avaliação deve embutir todo o conhecimento sobre o problema a ser resolvido, tanto suas restrições quanto seus objetivos de qualidade.

Levando em consideração os algoritmos de segmentação de imagens digitais, a função de avaliação serve para otimizar o resultado a partir de alguma métrica apropriada. Essas métricas podem variar dependendo do tipo de imagem utilizada e de algoritmo de segmentação escolhido. As funções utilizadas neste trabalho foram baseadas na entropia e baseadas nas métricas *CG*, *US* e *OS*.

5.1.3 Seleção dos pais

O método de seleção dos pais deve simular o mecanismo de seleção natural que atua sobre as espécies biológicas, em que os pais mais capazes podem gerar mais filhos, ao mesmo tempo em que os pais menos aptos também podem gerar descendentes (Goldberg [58] e Linden [61]). Devido a isso, tem-se que dar privilégio a aqueles indivíduos com o valor da função de avaliação alto, sem desprezar completamente aqueles indivíduos com o valor da função de avaliação baixo. Esta decisão é bastante razoável, pois até indivíduos que possuem uma má avaliação podem ter características genéticas que sejam favoráveis à criação de um indivíduo que seja a melhor solução para o problema.

5.1.4 Funcionamento do AG

Um algoritmo genético opera sobre uma população, fazendo com que esta evolua de acordo com um grau de adaptação. Seu funcionamento começa com a inicialização de uma população, e depois o processo de seleção, reprodução e mutação ocorre a cada geração até que seja atingido algum critério para o fim da evolução. Sua estrutura de funcionamento segue o esquema mostrado na Figura 5.2.

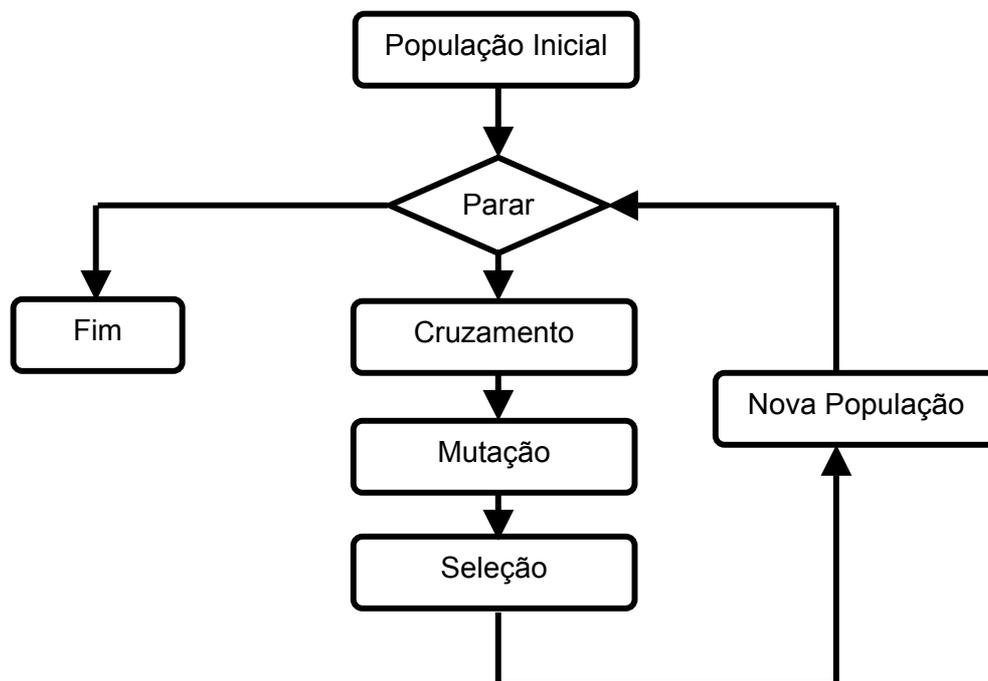


Figura 5.2 - Fluxograma que mostra o funcionamento de um AG

5.2 Algoritmos Implementados

Os algoritmos implementados nesta Tese foram os três abaixo:

- Segmentação por Quadtree com AG
- Segmentação por Limiar com AG
- Segmentação por Crescimento de regiões com AG

5.2.1 AGs para a Segmentação por Quadtree

O primeiro método de segmentação implementado para esta Tese foi o algoritmo por quadtree, que utilizou o Algoritmo Genético para a geração dos valores de cada nível da quadtree (Matias e Evsukoff [14]). O objetivo desta técnica é gerar de forma automática uma imagem segmentada por quadtree de forma que esse resultado obtido satisfaça uma determinada aplicação. A Quadtree é gerada por cada subdivisão de níveis. Samet em [39] mostra que cada subdivisão ou nível tem uma codificação. O método é descrito da seguinte maneira: Inicialmente é gerada uma família de cromossomos, de forma randômica, que contém todos os níveis para a geração da população inicial de quadtrees (Figura 5.3 exibe a representação cromossômica binária).

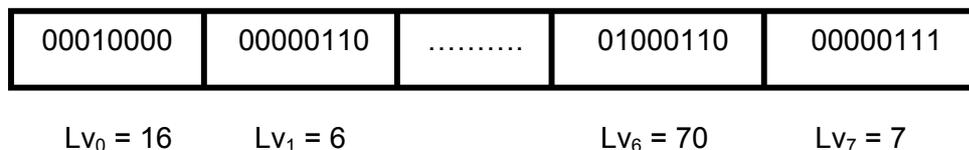


Figura 5.3 – Cromossomo binário onde L corresponde a cada nível da quadtree.

Esses níveis da quadtree são representados pelo valor da entropia que é computada e avaliada pelos AGs. Após cada nível receber o seu valor de entropia, então é feito um segundo passo dos AGs, que é a atribuição dos códigos para os quatro primeiros filhos da quadtree e assim recursivamente. Para calcularmos a quantidade de bytes necessários para a geração de um cromossomo, baseamos em uma imagem digital com resolução quadrática, isto é, número linhas igual a o número de colunas. Se representarmos esta imagem por uma matriz de ordem $N \times N$ temos que o número de níveis (L_v) da quadtree é dado pela Figura 5.4 .

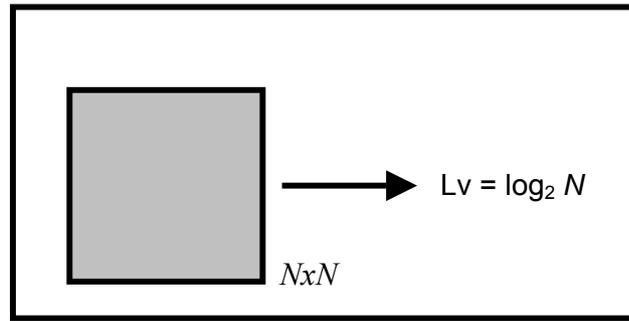


Figura 5.4 – Imagem quadrada exibindo o número de níveis da quadtree.

Conforme citado, a função de avaliação (função de *fitness*), para este algoritmo, é baseada na minimização da entropia, pois é comum definir a entropia como uma medida de desordem e algumas vezes a atribuímos como medida de complexidade. Matematicamente a entropia é definida como uma medida para a quantidade de informações contidas em uma série de número de símbolos. Em uma imagem digital, a entropia está associada à falta de correlação entre os pixels. Shannon em [18] define uma quantidade denominada $i(k)$. Se $p(k)$ é a probabilidade que o evento k possa ocorrer, então a informação associada com k é dada por:

$$i(k) = -\log p(k) \quad (5.2)$$

Logo, define-se a entropia como a medida associada à falta de correlação.

Como sendo:

$$S = \sum p(k)i(k) \quad (5.3)$$

Substituindo a equação 5.2 na 5.3 obtém-se a equação 5.4:

$$S = -\sum p(k) \log(p(k)) \quad (5.4)$$

Onde S é a entropia e $p(k)$ é a frequência associada a cada pixel da imagem. Logo, a entropia calculada minimiza uma função de *fitness* segundo essa equação:

$$Fitness = \arg \min S, \text{ onde:} \quad (5.5)$$

$$Fitness \in \mathfrak{R} \text{ e } S : \mathfrak{R} \longrightarrow \mathfrak{R}$$

Embora a entropia S possa atingir picos em alguns níveis da quadtree, o somatório global tem que ser minimizado. A Figura 5.5 mostra a população de quadtrees que é originada pelos valores de entropia gerados a cada iteração do AG. Quanto maior for a resolução da imagem, maior será o número de níveis da quadtree e conseqüentemente maior será o tempo de processamento do AG.

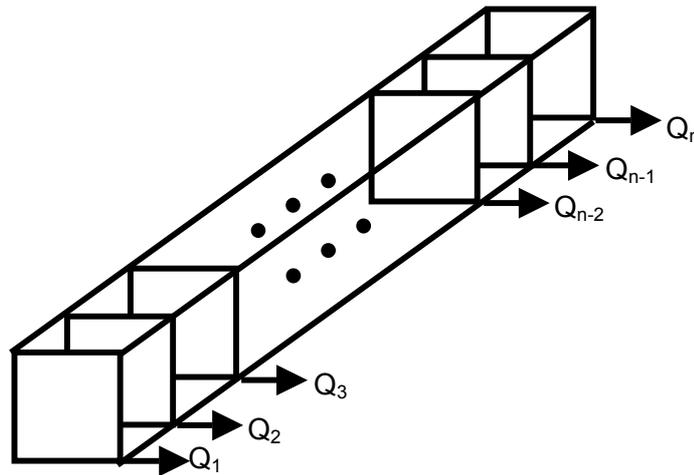


Figura 5.5 – População do AG para a Quadtree.

O pseudo-código para o algoritmo de segmentação por quadtree usando os AGs e dado por:

Algoritmo 5 GAQuadtree

1. Inicialização das quadtrees;
2. Validação();
3. **Enquanto** (Falso)
4. **Inicio_Enquanto**
5. Seleção();
6. Crossover();
7. Nova População de Quadtree();//Novos níveis
8. **Se** (Entropia for minimizada)
9. **Return** Image_Result;
10. **Caso Contrário**
11. **Fim_Enquanto**
12. **Fim.**

5.2.2 Segmentação por Limiar utilizando os AGs

Uma outra técnica utilizada foi o uso dos Algoritmos Genéticos em um método de segmentação utilizando o limiar (*Threshold*). O objetivo dos AGs, neste tipo de algoritmo, é gerar de forma automática uma imagem segmentada por atribuição de um *threshold*. A função de avaliação utilizada, para este algoritmo, também foi a entropia.

Neste algoritmo a representação cromossômica contém informação sobre um limiar da imagem (Figura.5.6). Vale ressaltar que este algoritmo foi o mais simples dos três implementados na Tese e que obtém os resultados menos interessantes, pois os métodos de segmentação baseados no limiar geralmente são focados para um tipo de aplicação que o suporta adequadamente.



00110000

$L_{v_0} = 48$

Figura 5.6 – Representação binária do cromossomo

Para calcularmos a quantidade de bytes necessários para a geração de um cromossomo, baseamos em uma imagem digital com resolução quadrática, isto é, número linhas igual a o número de colunas. Se representarmos esta imagem por uma matriz de ordem $N \times N$ temos que o valor de limiar máximo da imagem é o número máximo de tons de cinza da mesma.

A Figura 5.7 exibe uma população de imagens diferentemente segmentadas que são originadas pelos valores de limiares gerados a cada iteração do AG. Esses novos limiares são originados a partir da avaliação da entropia global da imagem.

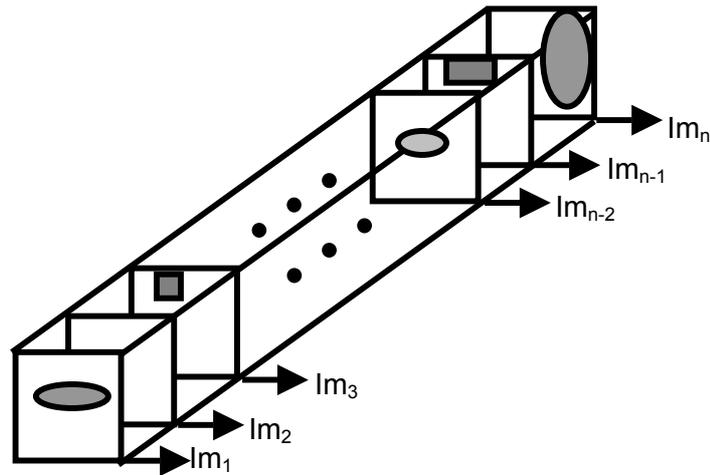


Figura 5.7 - População de imagens com diferentes limiares e objetos

O pseudo-código para o algoritmo de segmentação por limiar usando os AGs e dado por:

Algoritmo 6 *GATreshold*

1. Inicialização da população;
2. Validação();
3. **Enquanto** (Falso)
4. **Inicio_Enquanto**
5. Seleção();
6. Crossover();
7. Nova População () //Novos valores de limiar
8. **Se** (Entropia for minimizada)
9. **Return** Image_Result;
10. **Caso Contrário**
11. **Fim_Enquanto**
12. **Fim**

5.2.3 Segmentação por crescimento de regiões utilizando os AGs

Um terceiro tipo de algoritmo de segmentação com AGs, usado nesta Tese, foi o algoritmo de segmentação por crescimento de regiões (Bins *et al.* [15]) que consta na

TerraLib (biblioteca OpenSource). Este algoritmo proposto por Bins *et al.* [15] está direcionado a imagens de sensoriamento remoto, porém pode ser utilizado em outras imagens, sendo que para este outro tipo de imagens torna-se necessário algumas operações para retirar problemas indesejáveis (iluminação, ruído, ..) na imagem de entrada.

Conforme já foi descrito anteriormente, o objetivo destes algoritmos de segmentação por crescimento de regiões é inicialmente após serem criadas algumas sementes (pontos iniciais) estas crescem, a partir de alguma medida de similaridade (homogeneidade, textura, etc..), agrupando-se a pixels vizinhos de forma que vão estabelecendo uma região em comum. No algoritmo proposto por Bins *et al.* [15], e utilizado aqui, é necessário a inclusão de dois parâmetros de entrada para o algoritmo: um limiar de similaridade e a área máxima dos objetos a segmentar (Figura 5.8). O limiar (*threshold*) é utilizado para agrupar os pixels que estejam dentro deste valor e a área máxima é definida para não ter o problema de *under-segmentation* (várias regiões agrupadas em uma única).

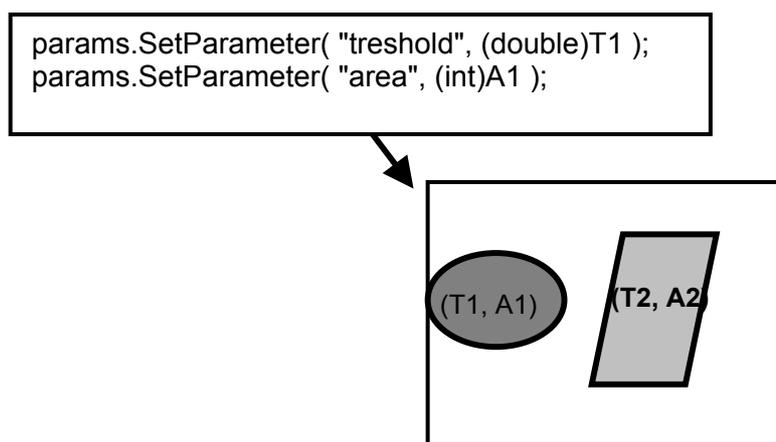


Figura 5.8 – Imagem de inicialização do algoritmo

Embora a segmentação deste algoritmo seja feita de modo automático é necessário incluir manualmente os dois atributos referenciados na Figura 5.8. O intuito

de utilizar os AGs, neste tipo de algoritmo, é a partir da representação cromossômica destes e da sua função de avaliação (função objetivo), gerar populações dentro deles (dos AGs), de tal maneira que o resultado gerado imposto pela sua função de avaliação, seja um resultado pelo menos adequado a uma determinada aplicação. Com isso, possa ser que esses atributos iniciais, calculados pelos AGs, sejam bons atributos para uma determinada imagem, de modo que possa ser utilizado em aplicações específicas de segmentação de imagens digitais.

5.2.3.1 Representação Cromossômica

Neste algoritmo a representação cromossômica contém informação sobre um *threshold* (limiar) de similaridade entre as regiões e o tamanho da área dessas regiões a serem segmentadas (Figura.5.9). Para calcularmos a quantidade de bytes necessários para a geração de um cromossomo, baseamos em uma imagem digital com resolução quadrática, isto é, número linhas igual a o número de colunas. Se representarmos esta imagem por uma matriz de ordem $N \times N$ temos que a o valor de limiar máximo da imagem é o número máximo de tons de cinza da mesma e a área do objeto é de no máximo 2 bytes.

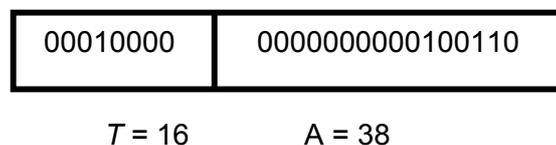


Figura 5.9 – Representação binária do cromossomo

5.2.3.2 Função de avaliação

O capítulo 4 descreveu várias métricas de avaliação para algoritmos de segmentação de imagens digitais. No algoritmo, aqui implementado, foram utilizados três métricas

referentes ao trabalho de Bins *et al.* [15] que são: *CG*, *OS* e *US*. Estas métricas já foram definidas, anteriormente, mas serão definidas mais uma vez.

$$CG(p) = \frac{\sum_{i=1}^{N_r} \sum_{j=1}^{N_s} SR_a(R_i, \hat{R}_j, p) * C_{ij}}{n(I)} * 100 \quad (5.6)$$

$$US(p) = \frac{\sum_{j=1}^{N_s} (1 - SR_b(\hat{R}_j, p)) * n(\hat{R}_j)}{n(I)} * 100 \quad (5.7)$$

$$OS(p) = \frac{\sum_{i=1}^{N_r} (1 - SO(R_i, p)) * n(R_i)}{n(I)} * 100 \quad (5.8)$$

Logo, a função de *fitness* é otimizada segundo essa equação:

$$Fitness = \arg \min Mt, \text{ onde:} \quad (5.9)$$

$$Fitness \in \mathfrak{R} \text{ e } Mt : \mathfrak{R}^2 \longrightarrow \mathfrak{R} \text{ e } Mt \text{ é uma métrica}$$

De acordo com o algoritmo proposto acima as métricas *US* e *OS* são minimizadas e a métrica *CG* tem que ser maximizada.

De maneira ilustrativa de *CG*, *OS* e *US* as Figuras 5.10 (a) e (b) mostram como é armazenado em memória o modelo representativo das medidas de avaliação *CG*, *OS* e *US* e a Figura 5.12 exhibe alguns exemplos de segmentação para uma simples imagem. Os cinco casos explicitados na Figura 5.12 estão representados a seguir:

1. Este caso corresponde a uma segmentação perfeita com a percentagem de pixels agrupados corretamente no valor de 100%. Enquanto a percentagem de *over-segmentation* e *under-segmentation* são nulas.
2. Neste exemplo, a região de referência R_1 aparece dividida dentro das regiões de saída de \hat{R}_1 a \hat{R}_4 . Nenhuma das regiões \hat{R}_j misturam pixels de diversas regiões

verdadeiras R_i , isto conduz a um valor de CG de 100%, mas 25% da imagem está sobre-segmentada (*over-segmented*) e 0% da imagem está *under-segmentation*.

3. Este é um caso extremo onde todas as regiões de referência aparecem divididas no resultado, o que faz obter uma taxa de *over-segmentation* de 100%. Contudo os pixels estão corretamente agrupados, logo o valor de CG é de 100%. E, não há taxa de *under-segmentation*, isto é, US é nula.
4. Neste exemplo, duas regiões de referência tem sido agrupadas e, conseqüentemente, um certo grau de *under-segmentation*. Esta duas regiões cobrem 50% dos pixels da imagem, o grau de *under-segmentation* é também de 50%. Os restantes dos 50% dos pixels estão corretamente agrupados.
5. Este é o exemplo mais complexo, pois somente os pixels da região \hat{R}_1 e \hat{R}_4 estão agrupados corretamente (18,75%) e o resto pertencem a regiões da imagem *under-segmenting*. Como a área de todas as regiões estão compartilhadas por mais que uma região \hat{R}_j , a imagem contém um valor de $OS = 100\%$.

Baseado neste exemplo, da para verificar que CG, OS, e US produzem medidas de avaliação com uma acurácia muito boa para agrupamento entre pixels, segmentação excendente e baixa segmentação, respectivamente. E, além disso, o parâmetro p oferece uma boa tolerância para a dispersão de pixels entre as regiões.

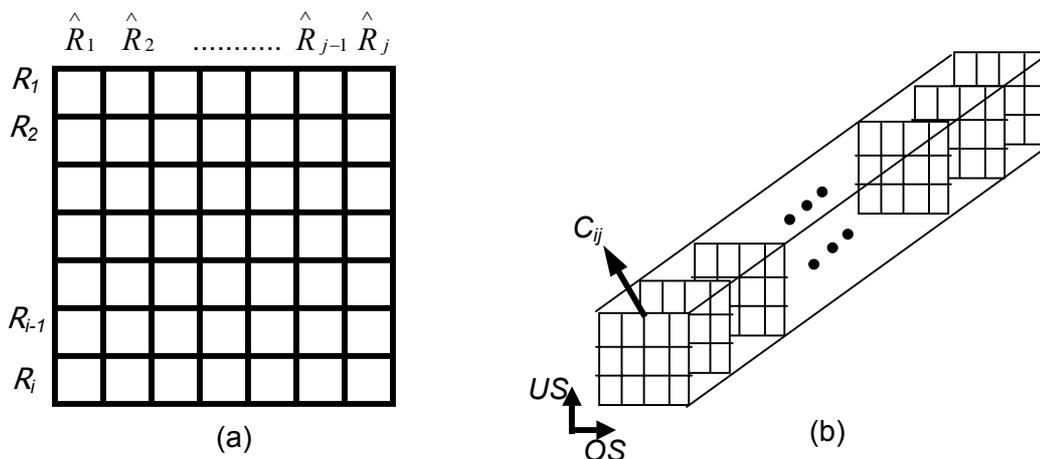
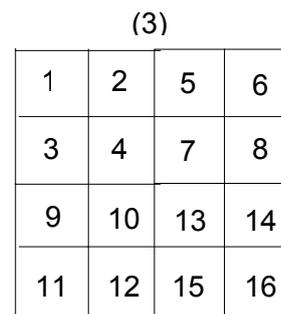
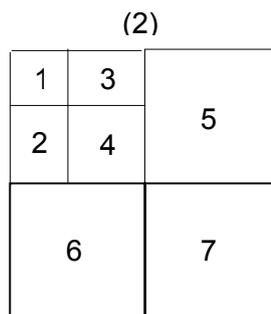
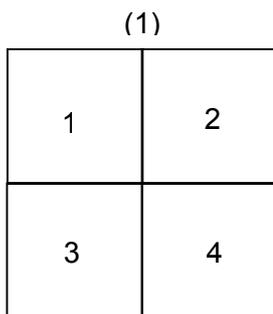
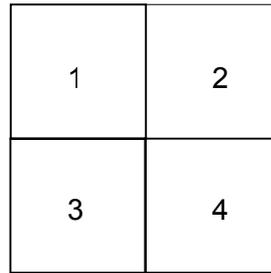
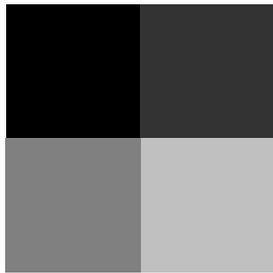


Figura 5.11 – (a) CG é computado pelo somatório dos pixels desta matriz
 (b) Representação de OS e US para cada matriz de confusão gerada



(4)

(5)

O algoritmo de crescimento de regiões usando os AGs é dado por:

Algoritmo 7 GA_RegionGrowing

1. Inicialização da população;
2. Region_Growing(Cromossomo[i]);
3. Validação();
4. **Enquanto** (Falso)
5. **Inicio_Enquanto**
6. Seleção();
7. Crossover();
8. Nova População () //Novos valores de limiar e área;

```
9.   Se (Fitness for Minimizada)
10.   Return Image_Result;
11.   Caso Contrário
12.   Fim_Enquanto
13. Fim
```

Capítulo 6

Testes e Resultados

6.1 Introdução

Conforme foi visto nos capítulos anteriores, a implementação deste trabalho está baseada nos algoritmos de segmentação por *quadtree*, limiar (*threshold*) e crescimento de regiões. Todos os algoritmos foram implementados sem e com o uso do AG. A metodologia desta Tese se refere ao uso do AG em Segmentação de Imagens Digitais, para que o mesmo possa gerar parâmetros de entrada para os Algoritmos de Segmentação, a partir de uma função de avaliação, de forma que tais parâmetros configurem um ponto de partida satisfatório para a imagem gerada. Convém ressaltar que, quando se trata de segmentação, nada é melhor do que a acuidade visual humana em classificar os objetos de forma automática e precisa.

Em nossos experimentos, foram utilizadas as linguagens de programação C e C++, além do VTK que é uma biblioteca de visualização 3D usada na geração de algumas imagens. Foi implementado o Algoritmo Genético baseado em Goldberg [58] com o Algoritmo de Crescimento de Regiões da biblioteca de Processamento de Imagens *TerraLib* que pertence ao INPE (Instituto Nacional de Pesquisas Espaciais). A plataforma utilizada foi o Sistema Operacional Linux.

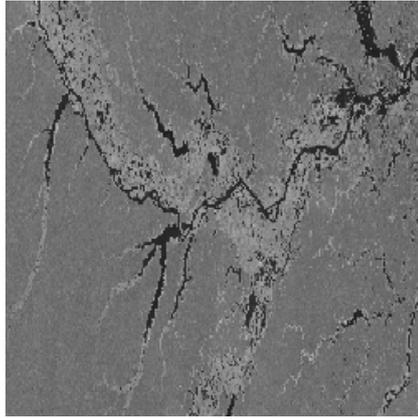
6.2 Resultados Experimentais

Para demonstrar a validade da utilização do AG nos algoritmos propostos, foram selecionadas cinco imagens distintas. Para os resultados gerados com segmentação por quadtree e limiar, utilizando o AG, foram usadas funções de avaliação baseadas apenas na entropia da imagem. Por outro lado, para os resultados gerados com o algoritmo de Crescimento de Regiões da biblioteca *TerraLib*, foram utilizadas como função de avaliação do AG as novas métricas de avaliação para algoritmos de segmentação: *CG*, *US* e *OS*. Para validar estas métricas *CG*, *US* e *OS*, foram realizadas comparações entre o resultado da segmentação sem o uso do AG (sem conhecimento prévio do usuário) e o resultado da segmentação com o uso do AG. Por fim foi realizado um estudo comparativo com o trabalho de Ortiz e Oliver [13] utilizando uma de suas imagens para validar o resultado final obtido no presente trabalho.

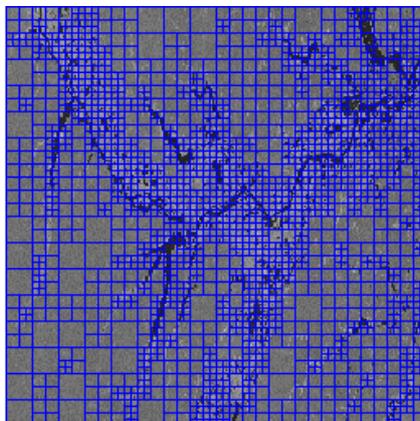
6.2.1 Segmentação de Quadtree com AG

Uma das imagens utilizadas neste tipo de algoritmo foi uma foto de satélite do Rio Solimões. A resolução dela é de 256x256. Tratando-se de uma imagem de resolução quadrática, isto facilitou a criação da quadtree, pois neste caso a mesma possui no máximo 8 níveis, sendo que cada nível possui um valor de entropia gerado pelo AG.

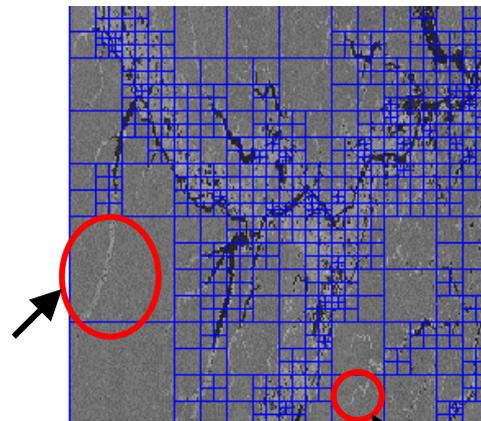
Para esta imagem foi executado um Algoritmo Genético com uma população inicial de 300 quadtrees. A quadtree que representou uma melhor *fitness* (função de avaliação) foi a representada pela Figura 6.1 (b). A probabilidade de mutação utilizada foi de 0.1, a probabilidade de cruzamento (*crossover*) foi de 0.6 e o número de gerações do AG foi 12.



(a)



(b)



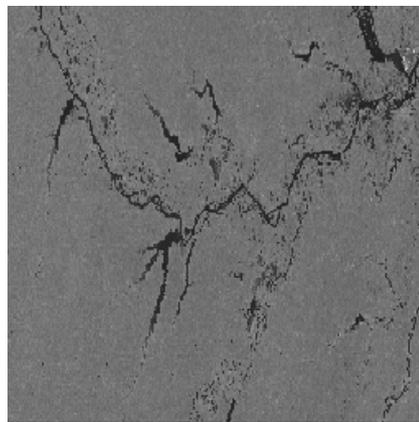
(c)

Figura 6.1 – (a) Imagem original (baixa-resolução)
(b) Quadtree sem o uso do AG
(c) Quadtree com o uso do AG

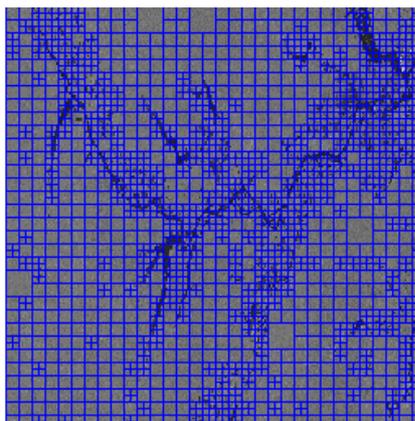
Comparando-se, por exemplo, a imagem que representa o Rio Solimões (Figura 6.1 a) em época de cheia e alguns de seus afluentes, observa-se que neste caso o método proposto (uso do AG) pode não ter sido tão eficiente quanto o método usual. Contudo, para alguns especialistas da área representada por esta foto de satélite, a imagem obtida com o uso do AG possui uma importância enorme devido a segmentação de algumas áreas muito importantes (como as apontadas na Figura 6.1 (c)). A imagem gerada com o uso do AG pode ser utilizada em aplicações de

compressão de dados, por exemplo, pois permitem expandir áreas de maior interesse e comprimir áreas de menor interesse.

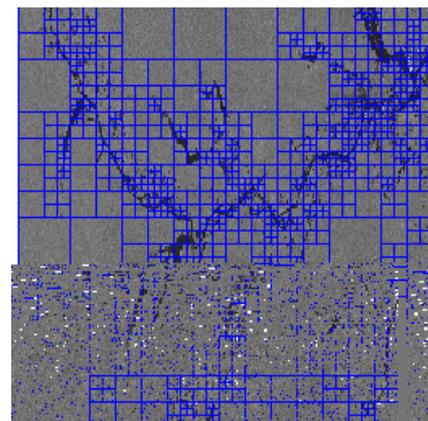
Uma outra imagem usada para testar os resultados foi a do Rio Solimões em época de seca com alguns de seus afluentes. Para esta imagem foi executado um Algoritmo Genético com uma população de 300 quadrees. A quadtree que representou uma melhor *fitness* (função de avaliação) foi a representada pela Figura 6.2 c. A probabilidade de mutação utilizada foi de 0.1, a probabilidade de cruzamento (*crossover*) foi de 0.6 e o número de gerações do AG foi 9.



(a)



(b)



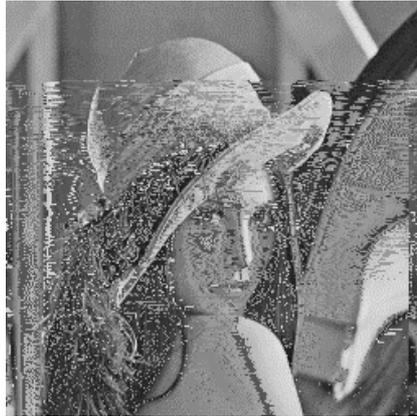
(c)

Figura 6.2 – (a) Imagem original (baixa-resolução)
(b) Quadtree sem o uso do AG
(c) Quadtree com o uso do AG

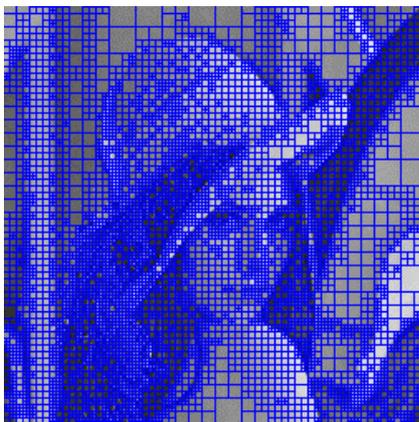
Na imagem acima (Figura 6.2) pode-se verificar que o método proposto (quadtree com o uso do AG) foi mais eficiente que o método usual (quadtree sem o uso do AG). Com efeito, a quantidade de áreas homogêneas geradas com o uso do AG é muito menor do que sem o uso dele. Da mesma forma que a imagem segmentada anteriormente (Figura 6.1 (c)) este resultado também pode ser utilizado em compressão de imagens digitalizadas.

Uma das primeiras imagens utilizadas para testar a segmentação por quadtree usando o AG foi a imagem da Lena. Como essa imagem é uma fotografia, possui uma iluminação e vários objetos com diversos contrastes, inclusive um espelho. Com isso, não podemos inferir se um dado resultado é bom ou não, mas de qualquer forma essa imagem foi usada como uma primeira experiência de um dos algoritmos propostos nesta Tese. Para esta imagem foi executado um Algoritmo Genético com uma população de 100 quadtrees. A quadtree que representou uma melhor *fitness* (função de avaliação) está representada na Figura 6.3 (c). A probabilidade de mutação utilizada foi de 0.1, a probabilidade de cruzamento (*crossover*) foi de 0.4 e o número de gerações foi 12. É válido observar que esses parâmetros de entrada (tamanho da população, probabilidade de mutação e probabilidade de cruzamento) influenciam significativamente a geração do resultado obtido.

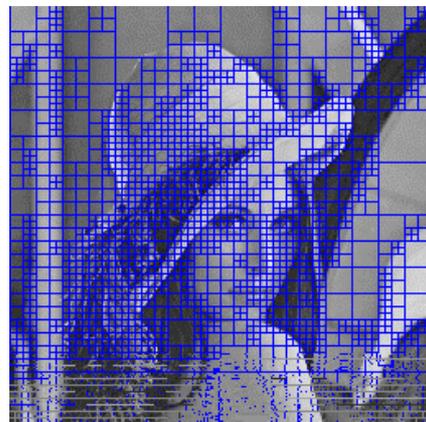
Vale ressaltar que foram utilizados, para esses testes por segmentação em quadtree, baixos valores do número de gerações do AG e altos valores para a população dos indivíduos, devido ao fato deste algoritmo de segmentação por quadtree com AG ter sido a tentativa inicial para a geração desta Tese.



(a)



(b)



(c)

Figura 6.3 – (a) Imagem original
(b) Quadtree sem o uso do AG (Matias e Evsukoff [14])
(c) Quadtree com o uso do AG (Matias e Evsukoff [14])

6.2.2 Segmentação por Limiar com AG

Este tipo de algoritmo consiste essencialmente em um ponto de corte, isto é, os valores da imagem que estiverem acima ou abaixo de um determinado valor (limiar) são representados na imagem de saída. O método de segmentação por limiar implementado nesta Tese foi relativamente simples. A segmentação gerada é representada por uma imagem binária formada pelo resultado do ponto de corte, conforme foi citado na seção 3.2.1 do Capítulo 3. A resolução das imagens utilizadas neste algoritmo é de 256x256 pixels.

Para a imagem da Figura 6.4 (a), foi executado um Algoritmo Genético com uma população inicial de 25 indivíduos. A imagem que representou uma melhor *fitness* (função de avaliação) foi a representada pela Figura 6.4 (b). A probabilidade de mutação utilizada foi de 0.05 e a probabilidade de cruzamento (*crossover*) foi de 0.6. O número máximo de gerações do AG a ser utilizado foi 1000.

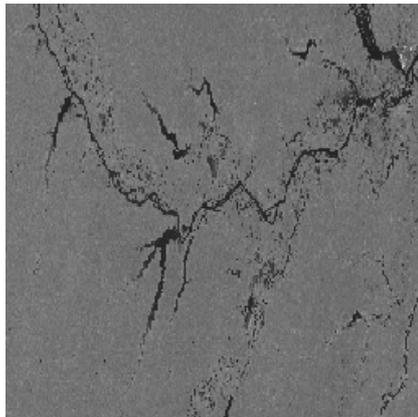


Figura 6.4 – (a) Imagem original da Lena
(b) Imagem da Lena com o uso do AG

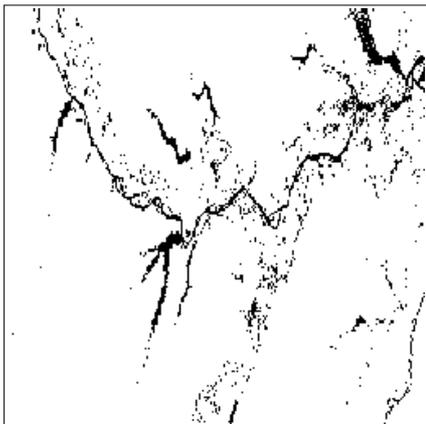
Como mostra a Figura 6.4 (b), os métodos de segmentação baseados em limiar geram apenas imagens binárias. Quando se trabalha com imagens deste tipo (fotografias em geral) ou imagens com vários detalhes, tais métodos não geram resultados satisfatórios, pois, como pode-se ver, a imagem segmentada perde todas as informações da imagem original. Isso ocorre porque quando se usa uma imagem binária resultante de uma imagem em tons de cinza ou colorida, perde-se todo o detalhe da imagem. Mais adiante será mostrado um contexto de visualização mais adequada para este método.

Uma outra imagem utilizada para testar os resultados foi a do Rio Solimões em época de seca com alguns de seus afluentes. Para esta imagem (Figura 6.5 (a)), foi executado um Algoritmo Genético com uma população inicial de 25 indivíduos. A imagem que representou uma melhor *fitness* (função de avaliação) está representada

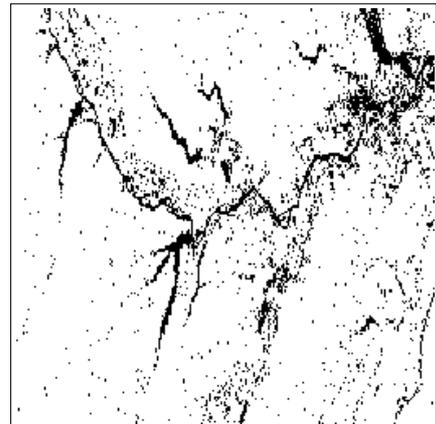
pela Figura 6.5 (c). A probabilidade de mutação utilizada foi de 0.05 e a probabilidade de cruzamento (*crossover*) foi de 0.6. O número máximo de gerações do AG a ser utilizado foi 1000. Pode-se verificar que como essa é uma imagem de satélite em tons de cinza, a segmentação por limiar, com ou sem AG fornece um bom resultado final.



(a)



(b)



(c)

Figura 6.5 – (a) Imagem original
(b) Imagem segmentada sem o uso do AG
(c) Imagem segmentada com o uso do AG

6.2.3 Segmentação por Crescimento de Regiões com AG

Neste método de segmentação foram utilizadas métricas de avaliação que não usam a minimização da entropia (função de avaliação da segmentação por quadrees e limiar). Conforme foi explicado na seção 5.2.3, as métricas aqui estudadas e implementadas foram a *CG*, *US* e *OS*. Para que essas métricas sejam calculadas, torna-se necessário que se tenha uma imagem de referência já segmentada (*ground truth*).

Este algoritmo por Crescimento de Regiões, implementado na biblioteca de processamento de imagens *TerraLib*, necessita de dois parâmetros de entrada: um limiar e uma área mínima dos objetos a serem classificados. O diferencial deste trabalho está na tentativa de gerar, com o uso do AG em função das métricas de avaliação acima citadas, os parâmetros de entrada anteriormente mencionados. Isso vem informar que os valores de limiar e de área dos objetos gerados pelo AG podem ser quantidades favoráveis a uma boa segmentação inicial. O fato de se usar uma imagem de referência serve como medida comparativa para se distinguir entre os vários tipos de algoritmos de segmentação existentes na literatura.

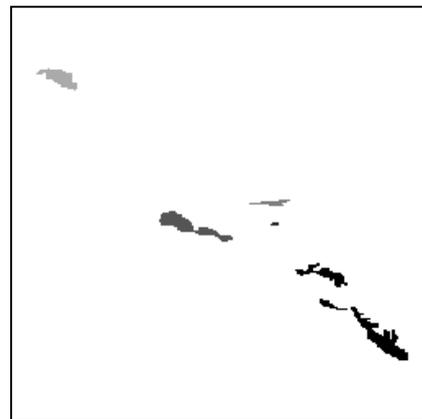
As imagens segmentadas utilizadas neste método e que serviram como imagens de referência, foram obtidas gerando um histograma avaliado com base no conhecimento dos dados e na inspeção visual. A primeira imagem a ser analisada foi uma foto de satélite com resolução de 256x256 pixels e que exibe uma exudação de petróleo no mar (Figura 6.6). Como o que interessa nesta fotografia são as manchas de óleo, torna-se necessário a visualização destas áreas em específico.



(a)



(b)



(c)

Figura 6.6 – (a) Imagem original
(b) Imagem segmentada sem AG
(c) Imagem segmentada com o uso do AG

Para obter o resultado da Figura 6.6 (c) foi executado um Algoritmo Genético com uma população inicial de 25 indivíduos. A probabilidade de mutação utilizada foi de 0.05, a probabilidade de cruzamento (*crossover*) foi de 0.6, o número máximo de gerações do AG a ser utilizado foi 1000. O valor de *CG* foi de 90.44%, *US* igual a 3.74% e *OS* foi de 4.71%. O tempo de processamento foi de 6 horas e 33 minutos.

A imagem da Figura 6.6 (b) foi gerada sem AG e sem nenhum conhecimento prévio do usuário, o que ocasionou um resultado de segmentação inicial insatisfatório.

Para efeito comparativo foi criada a Tabela 6.1 com os dois tipos de segmentação (sem o uso e com o uso do AG).

Caso	CG	US	OS
Segmentação sem AG	10,17%	18.33%	82.21%
Segmentação com AG	90.44%	3.74%	4.71%

Tabela 6.1 – Estudo comparativo entre a segmentação sem AG e a segmentação com AG

A Tabela 6.1 mostra que o resultado foi muito bom em comparação ao resultado usual, pois os valores de *US* e *OS* estão abaixo dos 15% de tolerância e o valor de *CG* está acima dos 80%.

Com a mesma imagem da Figura 6.6 (a) foi executado o Algoritmo Genético com valores de entrada distintos, isto é, a população inicial foi de 25 indivíduos. A probabilidade de mutação utilizada foi de 0.05, a probabilidade de cruzamento (*crossover*) foi de 0.3, o número máximo de gerações do AG foi 1000. O valor de *CG* foi de 84.97%, *US* igual a 9.3% e *OS* foi de 8.36%. O tempo de processamento foi de 5 horas e 36 minutos.



(a)

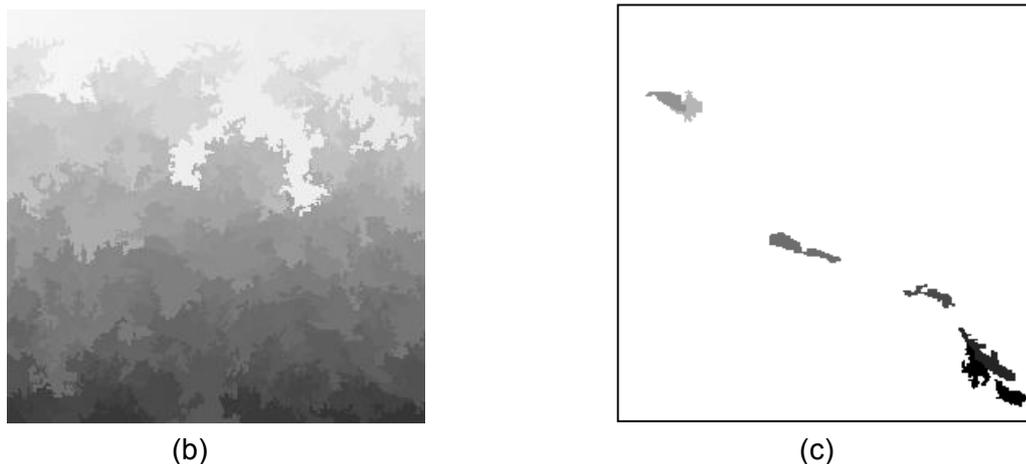


Figura 6.7 – (a) Imagem original
 (b) Imagem segmentada sem AG
 (c) Imagem segmentada com o uso do AG

Para efeito comparativo foi criada a Tabela 6.2 com os dois tipos de segmentação (sem e com o uso do AG). A imagem da Figura 6.7 (b) foi gerada sem AG e sem nenhum conhecimento prévio do usuário, e que ocasionou um resultado de segmentação inicial insatisfatório.

Caso	CG	US	OS
Segmentação sem AG	10,17%	18.33%	82.21%
Segmentação com AG	84.97%	9.3%	8.36%.

Tabela 6.2 – Estudo comparativo entre a segmentação sem AG e a segmentação com AG

Os valores finais da Tabela 6.2 não foram tão bons quando comparados aos da Tabela 6.1, porém o resultado final ainda é satisfatório, pois os valores de *US* e *OS* continuam abaixo dos 15% de tolerância e o valor de *CG* está acima dos 80%.

É bom lembrar que nem sempre uma maior probabilidade de cruzamento irá fornecer melhores valores de *CG*. Há outros fatores peculiares a própria imagem e

também aos parâmetros de entrada dos AGs que influenciam a computação de qualquer uma das métricas de avaliação.

Em um outro teste foi utilizada a imagem da Lena com resolução de 256x256 pixels. Para construir a imagem segmentada de referência foi gerado um histograma avaliado com base no conhecimento dos dados e também foi realizada uma inspeção visual. Para obter o resultado da Figura 6.8 (c) foi executado o Algoritmo Genético com uma população inicial de 25 indivíduos. A probabilidade de mutação utilizada foi de 0.05, a probabilidade de cruzamento (*crossover*) foi de 0.6, o número máximo de gerações do AG a ser utilizado foi 1000. O valor de *CG* foi de 90.57%, *US* igual a 17.31% e *OS* igual a 16.1%. Embora o valor da métrica *CG* tenha sido excelente para este tipo de imagem, os valores de *OS* e *US* estão um pouco acima do esperado. O tempo de processamento foi de 5 horas e 12 minutos.

Os valores de *OS* e *US* obtidos nesta imagem, são devidos a imagem original ser uma fotografia real que contém diversos detalhes como cabelo, iluminação, espelho, entre outros o que dificulta bastante o computo destas duas métricas.

Para efeito comparativo foi criada a Tabela 6.3 com os dois tipos de segmentação (sem e com o uso do AG). A imagem da Figura 6.8 (b) foi gerada sem AG e sem nenhum conhecimento prévio do usuário. Apesar disto, esta imagem seria uma segmentação aceitável.



(a)



(b)



(c)

Figura 6.8 – (a) Imagem original
 (b) Imagem segmentada sem AG
 (c) Imagem segmentada com o uso do AG

Caso	CG	US	OS
Segmentação sem AG	61.06%	39.41%	24.32%
Segmentação com AG	90.57%	17.31%	16.1%.

Tabela 6.3 – Estudo comparativo entre a segmentação sem AG e a segmentação com AG

A Tabela 6.3 mostra que este tipo de imagem (fotografia real) sem nenhum pré-processamento anterior leva a uma grande fragmentação da imagem segmentada, independente do uso ou não do AG, devido ao alto valor de OS.

Conforme foi descrito na introdução deste capítulo, para validação do algoritmo proposto nesta Tese, foi realizado um estudo dos resultados obtidos no trabalho de Ortiz e Oliver [13]. Para isso, comparamos o nosso resultado com o obtido por eles. Verifica-se que o uso do AG funcionou muito bem com as métricas de avaliação *CG*, *US* e *OS*. Porém, é válido lembrar que a nossa imagem de referência segmentada é diferente da deles (Ortiz e Oliver [13]), devido a não possuímos tal imagem.

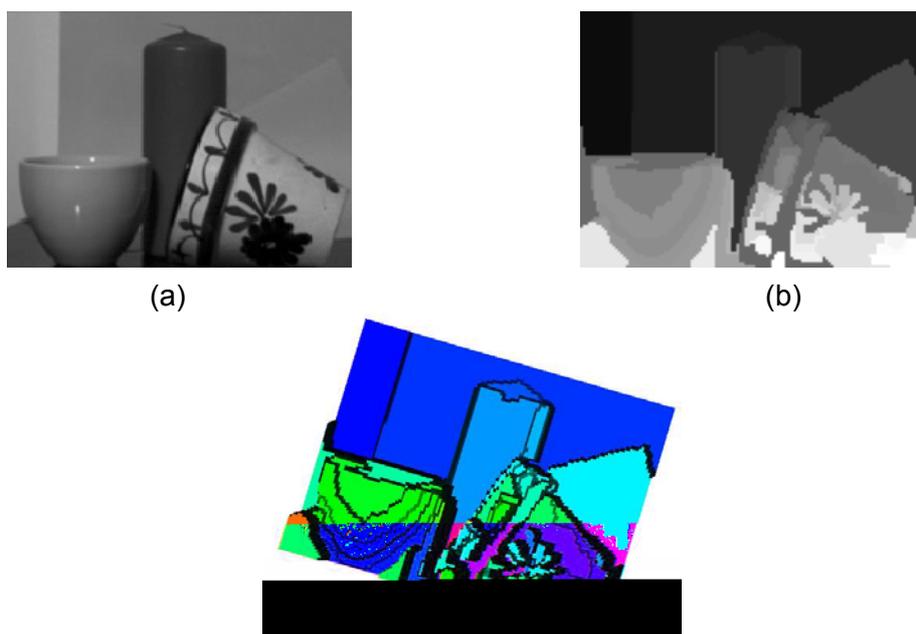


Figura 6.9 – (a) Imagem original (b) Imagem segmentada usando os AGs
(c) Perspectiva de (b) em 15 graus

Para obter o resultado da Figura 6.9 (b) foi executado o Algoritmo Genético com uma população de 25 indivíduos. A probabilidade de mutação utilizada foi de 0.05, a probabilidade de cruzamento (*crossover*) foi de 0.6, o número máximo de gerações do AG a ser utilizado foi 1000. Os valores obtidos para as métricas *CG*, *US* e *OS*, são fornecidos na Tabela 6.4, comparando-se com os dados de Ortiz e Oliver [13]. Para os dois casos explicitados (Tabela 6.4) o valor para o parâmetro de tolerância p foi de 90%, o tempo de processamento foi de 3 horas e 22 minutos. Vale lembrar que embora se tenha como entrada, para os dois algoritmos, a mesma imagem (Figura 6.9

(a)), a imagem de referência segmentada utilizada não é a mesma, porém para efeitos comparativos o que vale é o resultado das métricas para as respectivas imagens.

Caso	CG	US	OS
Ortiz e Oliver	95.84%	3.83%	26.21%
Algoritmo proposto	91.27%	5.53%	29.46%

Tabela 6.4 – Estudo comparativo entre o algoritmo proposto e o algoritmo de Ortiz e Oliver [13], baseado na Figura 6.9 (b).

A tabela acima mostra que o resultado gerado pelo método de segmentação utilizando os algoritmos genéticos forneceu valores mais que satisfatórios quando comparado ao resultado obtido por Ortiz e Oliver [13]. Vendo desta maneira, pode-se deduzir mais uma vez a conveniência do uso do AG em segmentação de imagens digitais.

Capítulo 7

Conclusão e Trabalhos Futuros

O objetivo desta Tese foi verificar o uso do Algoritmo Genético (AG) em Algoritmos de Segmentação de Imagens Digitais. Concluímos que, o AG se adequa a cada tipo de algoritmo de segmentação estudado, como: segmentação por quadtree, segmentação por limiar e segmentação por crescimento de regiões. Foi proposta a geração de uma segmentação de forma automática, independente de se inserir os parâmetros iniciais para cada tipo de algoritmo de segmentação. O AG realiza a inicialização destes parâmetros (limiar, níveis da quadtree, área do objeto, forma, entre outros) a partir de uma função de avaliação ou métrica ([14], [13], [50], [52], [53]) particular a cada algoritmo de segmentação. Uma função de avaliação pode ser boa para um determinado tipo de algoritmo e não ser eficiente para um outro algoritmo de segmentação.

Nos algoritmos de segmentação por quadtree e limiar, foi utilizada uma função de avaliação baseada na minimização da entropia. No algoritmo de segmentação por crescimento de regiões, foi usada uma função de avaliação baseada nas métricas *CG*, *US* e *OS*. Os resultados obtidos por estas métricas foram melhores que os obtidos pelos algoritmos com funções de avaliação baseadas na entropia da imagem. Isto ocorreu devido ao fato que as três métricas (*CG*, *US* e *OS*) se baseiam em uma Matriz de Sobreposição de áreas (*MSA* [13], [50]), o que torna a avaliação confiável. Além disto, as métricas *OS*, *US* e *CG* tendem a avaliar com mais clareza o nível de fragmentação da imagem de saída (Ortiz e Oliver [13]). As métricas de avaliação são

úteis pois com elas pode-se avaliar a robustez e a performance do algoritmo de segmentação utilizado.

Um fator muito importante para a obtenção dos resultados foi o uso de algoritmos distintos para a segmentação das imagens. Conforme pode ser visto nos resultados, um algoritmo de segmentação por limiar está muito restrito ao tipo da imagem de entrada (é amplamente utilizado em imagens médicas). Algoritmos de segmentação por quadtree têm o problema de muitas vezes gerar resultados com mosaicos (blocos quadrangulares na imagem). Os algoritmos que usam a técnica por crescimento de regiões são largamente utilizados em diferentes aplicações por permitirem vários tipos de imagem como entrada (imagens de satélite, entre outras).

No tocante à performance, isto é, tempo de processamento, pode-se dizer que o uso do AG faz este tempo aumentar consideravelmente. Este aumento ocorre porque em cada população gerada, seus filhos são testados um a um pelo algoritmo de segmentação até que sejam verificadas as condições necessárias da função de avaliação (até que haja uma convergência). Além disso, alguns algoritmos de segmentação são bem mais lentos que outros. Alguns algoritmos necessitam que as imagens sejam pré-processadas, outros não.

7.1 Trabalhos Futuros

Como sugestão para aumentar a eficiência dos algoritmos apresentados, em termos de tempo de processamento, poderia-se usar a computação distribuída com o objetivo de disparar em vários processadores os filhos gerados pelo AG e desta forma diminuir o tempo de processamento com o número de máquinas utilizadas. Isto facilitaria aplicações que trabalham com imagens de satélite em tempo real. Para utilizar a

computação distribuída, poderia ser usada a biblioteca *MPI* (Message Passing

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Tso, B. and Mather, P. M., *Classification Methods for Remotely sensed data.*, CRC, 2001.
- [2] Liou, K. N., *Radiation and cloud processes in Atmosphere.*, Oxford University Press, 1992.
- [3] Gonzalez, R.C., Wintz, P., *Digital Image Processing*, Addison-Wesley, 1987.
- [4] Gonzalez, C. R., Woods, R. E., *Processamento de Imagens Digitais.*, Edgar Bluxer Ltda, 2000.
- [5] Jain, A. K., *Fundamentals of Digital Image Processing.*, Prentice Hall, 1989.
- [6] Baxes, G. A., *Digital image processing: principles and applications.*, John Wiley & Sons, Inc., 1994.
- [7] Pappa, G. L., Freitas, A. A., and Kaestner, C. A. A., "A Multiobjective Genetic Algorithm for Attribute Selection", In Proc. 4th Int. Conf. on Recent Advances in Soft Computing., pp.116-121, 2002.
- [8] Santos, C. J., Oliveira, J. R. F., Dutra, L. V., "Uso de Algoritmos Genéticos para Seleção de Atributos para Classificação de Regiões", Instituto Nacional de Pesquisas Espaciais, INPE, 2003.
- [9] Melkemi, K. E., Batouche, M., Fougou, S., "A multiagent system approach for image segmentation using genetic algorithms and extremal optimization heuristics", Pattern Recognition Letters v. 27, pp. 1230-1238, 2006.
- [10] Yoshimura, M., Oe, S., "Texture Image Segmentation by Using Genetic Algorithms Considering Optimum Number of Segmentation Areas.", Proceedings of IEEE International Conference on Evolutionary Computation, pp.471-476, 1997.

- [11] Yoshimura, M., OE, S., "Texture Image Segmentation by Genetic Algorithms", Proceedings of IEEE International Conference on Evolutionary Computation, pp.125-130, 1996.
- [12] Jin, X., Davis, C. H., "A Genetic Image Segmentation Algorithm with a Fuzzy-Based Evaluation Function.", The IEEE International Conference on Fuzzy Systems, 2003.
- [13] Ortiz, A., Oliver, G., "On the use of the overlapping area matrix for image segmentation evaluation: A survey and new performance measures", Pattern Recognition Letters v. 27, pp. 1916-1926, 2006.
- [14] Matias, I.O., Evsukoff, A.G., "Optimization of Quad-tree Image Segmentation by Genetic Algorithm", ISIS2005, Korea.
- [15] Bins, L. S., Fonseca, L. M. G, Erthal, J. G., It, F.M., "Satellite Imagery Segmentation: a region growing approach", VIII Simpósio Brasileiro de Sensoriamento Remoto., pp. 677-680, 1997.
- [16] Gonçalves, L. M. G., Matias, I. O., Oliveira, A. A. F., "Enhancing the Volumetric Approach to Stereo Matching". XIV International Conference on Computer Graphics and Image Processing. (SIBGRAP'01), pp.218-225, 2001.
- [17] Fu, K. H., Mui, J.K., "A survey on image segmentation.", Pattern Recognition I. v.13, pp. 3-16, 1981.
- [18] Shannon, C. E. , "A Matematical Theory pf Communication. ", Bell System Technical Journal v.27, pp. 379-423, 1948.
- [19] Pal, N. R., Pal, S.K., "A review on image segmentation techniques.", Pattern Recognition v.26, pp. 1277-1294, 1993.
- [20] Sahoo, P.K., Solutani, S., Wong, A.K.C., "A survey of thresholding techniques.", Comput. Vision Graphics Image Process, v. 41, pp. 233-260, 1998.

- [21] Punam, K. S., Jayaram, K. U., "Optimum image thresholding via class uncertainty and region homogeneity.", *IEEE Trans. Pattern Recog. Mach. Intell.* v. 23, pp. 689-706, 2001.
- [22] Pal, N.R., Pal, S.K., "Entropy thresholding.", *Signal Process*, v.16, pp. 97-108, 1989.
- [23] Pal, N. R., Pal, S.K., "Entropy: A new definition and its applications.", *IEEE. Trans. Systems Man Cybernet*, v. 21, pp. 1260-1270, 1991.
- [24] Yin, P. Y., "Maximum entropy-based optimal threshold selection using deterministic reinforcement learning with controlled randomization", *Signal Processing* v. 82, pp. 993-1006, 2002.
- [25] Xie, J., Tisui, H. T., "Image Segmentation based on maximum likelihood estimation and optimum entropy-distribution.", *Pattern Recognition* v. 25, pp. 1133-1141, 2004.
- [26] Pham, L. D., Prince, L. J., "An adaptive fuzzy C-means algorithm for image segmentation in the presence of intensity inhomogeneities". *Pattern Recognition Letters* v.20, pp. 57-68, 1999.
- [27] Tao, W. B., Tian, J. W., Liu, J., "Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm.", *Pattern Recognition Letters* v. 24, n. 16, pp. 3069-3078, 2003.
- [28] Hu, Y. C., Chang, C. C., "Variable Rate Vector Quantization Scheme Based On Quadtree Segmentation.", *IEEE Trans. On Consumer Electronics*. v. 45, pp. 310-317, 1999.
- [29] Yueni, H., To, T. P. J., "Efficient Variable Rate Vector Quantization Using Quadtree Segmentation.", *IEEE Trans. On Consumer Electronics*. v. 42, pp. 212-215, 1996.
- [30] Zouagui, T., Benoit-Catin, H., Odet, C., "Image Segmentation functional model.", *Pattern Recognition* v. 37, n. 9, pp. 1785-1795, 2004.

- [31] Adams. R, Bischof, L., "Seeded region growing", IEEE Trans. Pattern Anal. Machine Intell. v.16, n. 6, pp. 641-647, 1994.
- [32] Zhu, C. S., Yuille, A., "Region Competition: Unifying snakes, region growing, and Bayer/MDL for multiband image segmentation", IEEE Trans. Pattern Anal. Machine Intell. v.18, n. 9, pp. 884-900, 1996.
- [33] Wan, S. Y., Higgins, W. E., "Symetric Region Growing", Proc. IEEE Int. Conf. Image Processing (ICIP 2000), v. 2, pp. 96-99, 2000.
- [34] Wan, S. Y., Higgins, W. E., "Symetric Region Growing", IEEE Trans. on Image Processing, v.12, n. 9, pp. 1007-1015, 2003.
- [35] Baatz, M., Schäpe, A., "Multiresolution Segmentation: an Optimization Approach for High Quality Multi-Scale Image Segmentation", Available:
http://www.definiens.com/documents/publications_earth2000.php
- [36] Darwish, A., Leukert, K., Reinhardt, W., "Image Segmentation for the Purpose of Object-Based Classification", Available:
http://www.definiens.com/documents/publications_earth2003.php
- [37] Haralick, R. M., Shapiro, L.G., *Computer and Robot Vision.*, Addison Wesley, 1993
- [38] Marr, D., *Vision, A Computacional Investigation into the Human Representation and Processing of Visual Information.* MIT Press, 1982.
- [39] Samet, Hanan., *Applications of Spatial Data Structures.*, Addison-Wesley, 1990.
- [40] Sayood, K., *Introduction to Data Compression.*, Morgan Kaufmann, 1996.
- [41] Manouvier, M., Rukoz, M., Jomier, G., "Quadtree representations for storage and manipulation of cluster of images"., Image and Vision Computing, v.20, n.7, pp. 513-527, 2002.
- [42] Schoenmakers, R. P. H. M., Wilkinson, G. G., Schouten, T. E., "Segmentation of remotely-sensed images: a re-definition for operational applications", IGARSS'91, V.II, pp. 1087-1090, 1991.

[43] Zhang, Y., "A survey on evaluating methods for image segmentation.", *Pattern Recognition* v. 29, n. 8, pp. 1335-1346, 1996.

[44] Zhang, J. Y., "A Review of Recent Evaluation methods for Image Segmentation.", *International Symposium on Signal Processing and its Application*, pp. 148-151, 2001.

[45] Zhang, Y., *Image Segmentation.*, Science Publisher, 2001.

[46] Levine, M., Nazif, A., "Dynamic measurement of computer generated image segmentations.", *IEEE Trans. Pattern Anal. Machine Intell* v. 7, pp. 155-164, 1985.

[47] Pal, N. R., Bhandari, D. "Image Thresholding: Some new techniques.", *Signal Process.* v. 33, n. 2, pp. 139-158, 1993.

[48] Rosenberg, C., Chehdi, K., "Genetic Fusion: application to multi-components image segmentation.", *Proc. ICASSP*, v.4, pp. 2223-2226, 2000.

[49] Cardoso, J. S., Corte-Real, L., "Toward a Generic Evaluation of Image Segmentation.", *IEEE Transactions on Image Processing*, v. 14, n. 11, pp.1173-1782, 2005.

[50] Huang, Q., Dom, B., "Quantitative methods of evaluating image segmentation.", *Proc. IEEE Internat. Conf. on Image Processing*, v. 3, pp. 53-56, 1985.

[51] Lewis, H., Brown, M., " A generalized confusion matrix for assessing area estimates from remotely sensed data.", *J. Remote Sens.* v. 22, n. 16, pp. 3223-3235, 2001.

[52] Zhang, J. Z., "Evaluation and comparison of different segmentation algorithms.", *Pattern Recognition Letters.* v. 18, n.10, pp. 963-974, 1997.

[53] Zhang, Y. J., Gerbrands, J. J., "Objective and quantitative segmentation evaluation and comparison.", *Signal Processing* v. 39, n. 1, pp. 43-54, 1994.

- [54] Liu, J., Yang, Y-H., "Multiresolution color image segmentation.", IEEE Trans. Pattern Anal. Machine Intell. v.6, n. 7, pp. 689-700, 1994.
- [55] Borsoi, M., Campadelli, P., Schettini, R., "Quantitative evaluation of color image segmentation results.", Pattern Recognition Letters. v. 19, n. 8, pp. 741-747, 1998.
- [56] Udupa, K. J., LeBlanc, R. V., Zhuge, Y., Imielinska, C., Schmidt, H., Currie, M. L., Hirsch, E. B., Woodburn, J., "A framework for evaluating image segmentation algorithms.", Computerized Medical Imaging and Graphics, v. 3, pp. 75-87, 2006.
- [57] Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
- [58] Goldberg, E. D., *Genetic Algorithms in Search, Optimization, and Machine Learning.*, Addison-Wesley, 1989.
- [59] Vieira, P. S. Azeredo, T. R., Neto, A. A. S., Matias, I.O, Viana, D. S., "Um controle autônomo de um carro de corrida", SBGAMES 2006.
- [60] Geyer, S. A., *Fuzzy Rule Based Expert System and Genetic Machine Learning.*, Heidelberg : Phisica-Verlag, 1997.
- [61] Linden, R., *Algoritmos Genéticos: Uma importante ferramenta da Inteligência Computacional.*, Editora Brasport, 2006.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)