



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m19/2012/11.23.12.22-TDI

**EXTRAÇÃO DE ESTRADAS EM IMAGENS SAR
AEROTRANSPORTADAS: UMA ABORDAGEM
BASEADA EM MODELO DE CONTORNO ATIVO COM
O USO DE SEMEADOR SEMIAUTOMÁTICO**

Rodolfo Georjute Lotte

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Sidnei João Siqueira Sant'Anna, Cláudia Maria de Almeida, Corina da Costa Freitas, e José Demisio Simões da Silva, aprovada em 07 de dezembro de 2012.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3D47T6S>>

INPE
São José dos Campos
2012

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):

Presidente:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Membros:

Dr. Antonio Fernando Bertachini de Almeida Prado - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Germano de Souza Kienbaum - Centro de Tecnologias Especiais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Vivéca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m19/2012/11.23.12.22-TDI

**EXTRAÇÃO DE ESTRADAS EM IMAGENS SAR
AEROTRANSPORTADAS: UMA ABORDAGEM
BASEADA EM MODELO DE CONTORNO ATIVO COM
O USO DE SEMEADOR SEMIAUTOMÁTICO**

Rodolfo Georjute Lotte

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Sidnei João Siqueira Sant'Anna, Cláudia Maria de Almeida, Corina da Costa Freitas, e José Demisio Simões da Silva, aprovada em 07 de dezembro de 2012.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3D47T6S>>

INPE
São José dos Campos
2012

L917 Lotte, Rodolfo Georjute.
Extração de estradas em imagens SAR aerotransportadas: uma abordagem baseada em Modelo de Contorno Ativo com o uso de semeador semiautomático / Rodolfo Georjute Lotte. – São José dos Campos : INPE, 2012.
xxiv + 136 p. ; (sid.inpe.br/mtc-m19/2012/11.23.12.22-TDI)

Dissertação (Mestrado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2012.

Orientadores : Drs. Sidnei João Siqueira Sant’Anna, Cláudia Maria de Almeida, Corina da Costa Freitas, e José Demisio Simões da Silva.

1. extração de estradas. 2. imagem de radar. 3. modelo de contorno ativo. 4. sementeação semiautomática. I. Título.

CDU 004.932.72’1

Copyright © 2012 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, reprográfico, de microfilmagem ou outros, sem a permissão escrita do INPE, com exceção de qualquer material fornecido especificamente com o propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2012 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, microfilming, or otherwise, without written permission from INPE, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

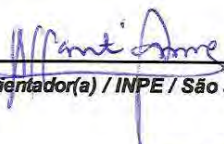
Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de **Mestre** em
Computação Aplicada

Dr. Luciano Vieira Dutra



Presidente / INPE / SJC Campos - SP

Dr. Sidnei João Siqueira Sant'Anna



Orientador(a) / INPE / São José dos Campos - SP

Dra. Cláudia Maria de Almeida



Orientador(a) / INPE / SJC Campos - SP

Dr. Gilson Alexandre Ostwald Pedro da
Costa



Convidado(a) / PUC- RIO / Rio de Janeiro - RJ

Este trabalho foi aprovado por:

() maioria simples

unanimidade

Aluno (a): **Rodolfo Georjute Lotte**

São José dos Campos, 07 de Dezembro de 2012

“Imprima o que for verdade. Escreva de forma clara. Defenda-o até o último suspiro”.

JOHANN WOLFGANG VON GOETHE
em *“Fausto”*, 1808

A meus pais Raul e Lélia, e a minha irmã Cláudia.

AGRADECIMENTOS

Agradeço a Deus pela realização deste trabalho, por tornar possível cada conquista em que obtive neste instituto. A minha mãe divina, Nossa Senhora da Aparecida, pela proteção e luz, necessárias tantas vezes nessa caminhada.

Agradeço a todos professores e companheiros do curso de Computação Aplicada do INPE. Aos meus orientadores: Dra. Cláudia M. Almeida, pela competência, gentileza, paciência e dedicação. Ao Dr. Sidnei J. S. Sant’Anna e Dra. Corina C. Freitas por aceitarem esta tarefa e pelo esforço imensurável em ajudar-me. Ao Dr. Rafael D. C. dos Santos pela paciência em solucionar minhas dúvidas. Um agradecimento especial ao professor e grande amigo Dr. José Demisio Simões da Silva (*in memoriam*), pessoa pela qual tive o privilégio de conhecer e aprender coisas que guardarei comigo sempre. Obrigado pela amizade, pelos ensinamentos, das excelentes aulas sempre acompanhadas pelo bom humor e sabedoria, por acreditar em meu trabalho e, acima de tudo, por acolher-me como filho neste instituto.

Agradeço a toda minha família, especialmente aos meus queridos pais, por me mostrarem que é necessário lutar na vida, que é possível conquistar muito com simplicidade, honestidade e persistência. Agradeço a minha querida irmã, por suas palavras amigas e por estar presente nos momentos em que não pude contar com todos. Minha avó e meu avô (*in memoriam*) pelo amor e carinho infinito.

Agradeço a todos os amigos que contribuíram pela realização deste projeto, em especial aos da minha querida cidade, Orlandia, que torceram e torcem por mim. Obrigado por terem acreditado quando tudo não passava de um sonho. Aos amigos que aqui fiz, agradeço de coração por partilharem cada minuto de dificuldade e de alegria nestes anos. Um agradecimento especial aos amigos que em muitos momentos fizeram com que tudo se tornasse mais fácil: Anna Karina, Fernando Pereira, Luiz Gustavo e Mirian Caetano. Obrigado pela amizade, por terem me escutado e me ajudado nos piores momentos. Ao “Grande” amigo Amarísio Araújo pela paciência, amizade e pela enorme contribuição neste trabalho. Ao Vitor Conrado pela sua admirável competência, por compartilhar tantas vezes seu conhecimento e nunca faltar-me. Vocês, com certeza, foram peças essenciais nesta conquista.

Por fim, agradeço ao INPE pela infraestrutura, e à CAPES e FINEP pelo apoio financeiro.

RESUMO

A pesquisa envolvendo métodos computacionais para a extração de estradas se intensificou nas últimas duas décadas. O processo geralmente é realizado sobre a análise de imagens adquiridas por sensores ópticos ou de radar (*Radio Detection and Ranging*). Muitas vezes, o mapeamento cartográfico a partir dessas imagens é realizado manualmente, exigindo tempo e esforço consideráveis por parte do intérprete. Existem atualmente muitos trabalhos envolvendo a extração de estradas de forma automática ou semiautomática deste mapeamento, porém, cada um deles apresenta soluções diferenciadas para problemas igualmente diferenciados, tornando essa tarefa uma questão científica ainda em aberto. Dentre as vantagens na utilização de imagens de radar, há a possibilidade de levantamento em áreas frequentemente recobertas por nuvens, ou regiões cujas topografias são ocultadas por copas de árvores, na aquisição de imagens independente do horário, entre outras. Este trabalho compreende a geração de pontos pré-estipulados próximos à feição de interesse, denominados pontos-sementes. A marcação desses pontos (*seeding*) é realizada de maneira semiautomática por meio do conceito de Mapas Auto-Organizáveis (*Self-Organizing Maps - SOM*). Na sequência, é utilizado um Modelo de Contorno Ativo (*Snakes*) para a extração do eixo de simetria de estradas em uma imagem de radar de abertura sintética (*Synthetic Aperture Radar - SAR*) aerotransportada. Os resultados obtidos foram avaliados quanto a sua qualidade em termos de perfeição, correção e redundância.

ROADS CENTER-AXIS EXTRACTION IN AIRBORNE SAR IMAGES: AN APPROACH BASED IN ACTIVE CONTOUR MODEL WITH THE USE OF SEMI-AUTOMATIC SEEDING

ABSTRACT

The research involving computational methods for the extraction of roads has intensified in the last two decades. The procedure is usually performed through the analysis of images acquired by optical sensors or radar (Radio Detection and Ranging). The great advantage in the usage of radar images is the possibility of surveying areas often covered by clouds, since the imaging by active sensors is independent from atmospheric conditions in the region of interest. The cartographic mapping based on these images is often done manually, requiring considerable time and effort from the interpreter. There are currently many studies involving the extraction of roads by means of automatic or semi-automatic approaches, however, each of them has different solutions for different problems, making this task a scientific issue still open. Among the advantages of using radar images, one can mention the acquisition of images regardless of atmospheric and illumination conditions, besides the possibility of surveying regions where the terrain is hidden by the vegetation canopy, among others. This work comprises the semi-automatic generation of seed points located close to the features of interest by means of Self-Organizing Maps (SOM). We then employ an active contour method (Snakes) for the extraction of the roads centre-axis in a synthetic aperture radar (SAR) airborne image. The obtained results were evaluated as to their quality with respect to perfection, correction, and redundancy.

LISTA DE FIGURAS

	<u>Pág.</u>
2.1 Exemplo de uma sementeção ótima sobre uma imagem óptica.	7
2.2 Rotação de estrada retangular sobre o eixo dos pontos num processo de Sementeção semiautomática.	10
2.3 Aplicação do operador morfológico de abertura para realce de estradas. (a) Imagem original. (b) Imagem após a aplicação do operador de abertura. (c) Imagem após a aplicação do operador de abertura direcional. (d) Imagem após a aplicação do operador de fechamento. (e) Imagem final, após a aplicação do operador <i>top-hat</i>	15
2.4 Ilustração da aplicação de <i>Snakes</i> com curva fechada (método tradicional) para a identificação de objetos: (a) Interpolação inicial. (b) Evolução final da curva.	18
2.5 Ilustração da aplicação de <i>Snakes</i> com curva aberta para a identificação de estradas: (a) Interpolação inicial. (b) Evolução final da curva em um processo de obtenção do eixo de simetria.	19
2.6 Ilustração do conceito de expansão do método <i>Ribbon-Snakes</i>	20
3.1 Geometria de imageamento de radares aerotransportados de visada lateral.	25
3.2 Ilustração da curva ζ sobre estrada em uma imagem digital, segundo propriedades matemáticas definidas por Gruen e Li (1997).	30
3.3 Fluxograma do método de extração de estradas.	31
3.4 Exemplo de arquitetura de uma rede SOM, com mapa de saída bidimensional de dimensão 3×3 neurônios e camada de entrada com dois neurônios (x_1 e x_2).	32
3.5 Arquitetura da rede SOM empregada no trabalho, em que e é igual a 21, n igual a 6 e p igual a 8.	34
3.6 Rede SOM: (a) Raio de vizinhança de um neurônio vencedor. (b) Valores de σ dos neurônios vizinhos.	36
3.7 Processo de classificação das estradas.	37
3.8 Pesos utilizados no cálculo da norma para a montagem da Matriz U.	37
3.9 Varredura da imagem para a leitura do conjunto f de amostras, em que cada amostra, são obtidos k perfis de estradas (Na ilustração, 8 perfis).	38
3.10 Exemplo da determinação dos <i>pixels</i> entre dois pontos em um dispositivo matricial (Algoritmo de Bresenham).	39

3.11	Exemplo do conjunto de amostras selecionadas pelo método de sementeação.	41
3.12	Descontinuidades da classificação dos pontos-sementes.	43
3.13	Exemplo da obtenção de contornos criados pela <i>Snakes</i> : (a) Imagem original. (b) Curva inicial. (c) Evolução intermediária da curva. (d) Curva final.	48
3.14	Avanço da <i>snake</i> com os pontos extremos fixos:(a) Curva inicial. (b) e (c) Avanços intermediários da curva. (d) Curva final.	53
3.15	Exclusão da aresta de ligação entre o extremo fixado e o vértice interno.	53
3.16	Resultados da extração para diferentes perfis de tonalidades: (a) Resultados obtidos com o termo de energia externa positivo e (b) Resultados obtidos com o termo de energia externa negativo.	54
3.17	Comparação dos resultados extraídos considerando um desvio-padrão de erro: (a) Verdadeiro positivo. (b) Falso negativo.	56
4.1	Imagem SAR da região de Paulínea-SP e respectivas amostras para as classes estrada e fundo. Imagem em composição colorida HH(R), HV(G) e VV(B).	60
4.2	Imagens sintéticas de radar: (a) <i>Phantom</i> (b) Polarização VV. (c) Polarização HH. (d) Polarização HV.	61
4.3	Imagem SAR da região de Paragominas-PA e as localizações das áreas de estudo.	62
4.4	Recorte 1: (a) Imagem SAR (513×513 <i>pixels</i>). (b) Referência cartográfica.	63
4.5	Recorte 2: (a) Imagem SAR (434×573 <i>pixels</i>). (b) Referência cartográfica.	64
4.6	Recorte 3: (a) Imagem SAR (247×707 <i>pixels</i>). (b) Referência cartográfica.	64
4.7	Recorte 4: (a) Imagem SAR (460×649 <i>pixels</i>). (b) Referência cartográfica.	65
4.8	Recorte 5: (a) Imagem SAR (357×357 <i>pixels</i>). (b) Referência cartográfica.	66
4.9	Resultados obtidos para a imagem sintética 1 (Figura 4.2(b)): (a) Pontos-sementes selecionados manualmente e (b) Resultado final após a aplicação do método <i>Snakes</i>	67
4.10	Resultados obtidos para a imagem sintética 2 (Figura 4.2(c)): (a) Pontos-sementes selecionados manualmente e (b) Resultado final após a aplicação do método <i>Snakes</i>	69
4.11	Resultados obtidos para a imagem sintética 3 (Figura 4.2(d)): (a) Pontos-sementes selecionados manualmente e (b) Resultado final após a aplicação do método <i>Snakes</i>	70
4.12	Resultados obtidos para o recorte 1 (Figura 4.4): (a) Sementeação sem a realização da poda. (b) Sementeação após a poda.	72

4.13	Resultados obtidos para o recorte 1 (Figura 4.4): (a) Resultado sobre a imagem, após a aplicação do método <i>Snakes</i> . (b) Resultado final sem a imagem. (c) Redes de estradas plotadas manualmente (referência).	73
4.14	Resultados obtidos para o recorte 2 (Figura 4.5): (a) Semeação sem a realização da poda. (b) Semeação após a poda.	75
4.15	Resultados obtidos para o recorte 2 (Figura 4.5): (a) Resultado sobre a imagem, após a aplicação do método <i>Snakes</i> . (b) Resultado final sem a imagem. (c) Redes de estradas plotadas manualmente (referência).	76
4.16	Resultados obtidos para o recorte 3 (Figura 4.6): (a) Semeação sem a realização da poda. (b) Semeação após a poda.	77
4.17	Resultados obtidos para o recorte 3 (Figura 4.6): (a) Resultado sobre a imagem, após a aplicação do método <i>Snakes</i> . (b) Resultado final sem a imagem. (c) Redes de estradas plotadas manualmente (referência).	78
4.18	Resultados obtidos para o recorte 4 (Figura 4.7): (a) Semeação sem a realização da poda. (b) Semeação após a poda.	79
4.19	Resultados obtidos para o recorte 4 (Figura 4.7): (a) Resultado sobre a imagem, após a aplicação do método <i>Snakes</i> . (b) Resultado final sem a imagem. (c) Redes de estradas plotadas manualmente (referência).	80
4.20	Resultados obtidos para o recorte 5 (Figura 4.8): (a) Semeação sem a realização da poda. (b) Semeação após a poda.	81
4.21	Resultados obtidos para o recorte 5 (Figura 4.8): (a) Resultado sobre a imagem, após a aplicação do método <i>Snakes</i> . (b) Resultado final sem a imagem. (c) Redes de estradas plotadas manualmente (referência).	82

LISTA DE TABELAS

	<u>Pág.</u>
3.1 Principais bandas do espectro de frequência em micro-ondas.	27
4.1 Medidas de desempenho do método sobre a Figura 4.2(b).	67
4.2 Medidas de desempenho do método sobre a Figura 4.2(c).	68
4.3 Medidas de desempenho do método sobre a Figura 4.2(d).	70
4.4 Parâmetros adotados no método de extração para os recortes apresentados.	71
4.5 Níveis de cinza utilizados nos padrões de treinamento.	71
4.6 Medidas de desempenho do método sobre o recorte 1.	74
4.7 Medidas de desempenho do método sobre o recorte 2.	75
4.8 Medidas de desempenho do método sobre o recorte 3.	77
4.9 Medidas de desempenho do método sobre o recorte 4.	79
4.10 Medidas de desempenho do método sobre o recorte 5.	80
4.11 Medidas de desempenho do método de extração de estradas para os re- cortes apresentados.	83
4.12 Tempo de processamento e complexidade dos algoritmos.	83

LISTA DE ABREVIATURAS E SIGLAS

<i>ACE</i>	–	<i>Anti-parallel Centerline Extractor</i>
<i>FCM</i>	–	<i>Fuzzy C Means</i>
<i>FN</i>	–	<i>False Negative</i>
<i>FP</i>	–	<i>False Positive</i>
<i>GA</i>	–	<i>Genetic Algorithm</i>
<i>GVF</i>	–	<i>Gradient Vector Flow</i>
<i>IA</i>	–	<i>Inteligência Artificial</i>
<i>LMC</i>	–	<i>Limite Mínimo de Continuidade</i>
<i>LMD</i>	–	<i>Limite Máximo de Descontinuidade</i>
<i>LSB-Snakes</i>	–	<i>Least Square B-spline Snakes</i>
<i>MLP</i>	–	<i>Multi-Layer Perceptron</i>
<i>MRF</i>	–	<i>Markov Random Field</i>
<i>pixel</i>	–	<i>Picture Element</i>
<i>QPC</i>	–	<i>Quantidade de Pontos Contínuos</i>
<i>RAM</i>	–	<i>Random Access Memory</i>
<i>RGB</i>	–	<i>Red, Blue and Green</i>
<i>RNA</i>	–	<i>Rede Neural Artificial</i>
<i>ROC</i>	–	<i>Receiver Operator Curve</i>
<i>SAR</i>	–	<i>Synthetic Aperture Radar</i>
<i>SOM</i>	–	<i>Self-Organizing Maps</i>
<i>SORM</i>	–	<i>Self-Organizing Road Mapping</i>
<i>TH</i>	–	<i>Transformada de Hough</i>
<i>TN</i>	–	<i>True Negative</i>
<i>TP</i>	–	<i>True Positive</i>

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	1
2 REVISÃO DA LITERATURA	7
2.1 Métodos de Semeação (<i>Seeding</i>)	7
2.1.1 Semeação Automática	8
2.1.2 Semeação Semiautomática	9
2.1.3 Semeação Manual	9
2.2 Principais Métodos para a Extração de Estradas	11
2.2.1 Classificação	11
2.2.2 Transformada de Hough	13
2.2.3 Morfologia Matemática	14
2.2.4 Segmentação	15
2.2.5 Modelos Deformáveis	17
2.2.6 Outras Abordagens	21
3 CONCEITOS E TECNOLOGIA	25
3.1 Fundamentos do imageamento por radar	25
3.2 Definições de Estrada em uma Imagem Digital	27
3.3 Metodologia	29
3.3.1 Semeação	30
3.3.1.1 <i>Self-Organizing Maps - SOM</i>	31
3.3.1.2 Leitura das Amostras	38
3.3.1.3 Classificação e Identificação	40
3.3.2 Processo de poda dos pontos-sementes	42
3.3.3 Modelo de Contorno Ativo (<i>Snakes</i>)	43
3.3.3.1 Energia da <i>Snake</i>	45
3.3.3.2 Energia Interna	45
3.3.3.3 Energia externa	46
3.3.3.4 Minimização da Energia Total	49
3.3.3.5 Representação Aberta da Curva <i>Snake</i>	52
3.3.4 Métricas de Qualidade	55

4	RESULTADOS E DISCUSSÃO	59
4.1	Dados Sintéticos	59
4.1.1	Obtenção do <i>Phantom</i>	59
4.2	Dados de SAR	61
4.3	Resultados e Validação	66
5	CONCLUSÕES	85
5.1	Trabalhos futuros	87
	REFERÊNCIAS BIBLIOGRÁFICAS	89
	APÊNDICE A	99
A.1	Fluxo principal de execução do código	99
A.1.1	Principal	99
A.1.2	Carregamento dos padrões de treinamento	103
A.1.3	Carregamento dos pesos da RNA	104
A.1.4	Treinamento da RNA	104
A.1.4.1	Ajuste dos pesos da RNA	106
A.1.5	Obtenção da matriz unitária	107
A.1.6	Leitura da Imagem	107
A.1.7	Leitura das Amostras	108
A.1.8	Localização dos centros no domínio de Kohonen	111
A.1.9	Semeadura	112
A.1.10	Pinta pontos-sementes	113
A.1.11	Fragmentação dos segmentos de reta	115
A.1.11.1	Ordena os índices para a fragmentação	121
A.1.12	Pinta pontos-sementes (Após a poda)	125
A.1.13	Execução do método <i>Open-Snakes</i>	127
A.2	Códigos-Fonte auxiliares no processo de extração	131
A.2.1	Converte a imagem de composição RGB para níveis de cinza	131
A.2.2	Algoritmo de Bresenham	132
A.2.3	Validação dos resultados	133
A.2.3.1	Contagem de acerto e erro	134
A.2.3.2	Dilatação dos segmetos de reta extraídos	136

1 INTRODUÇÃO

A pesquisa quanto ao uso da aerofotogrametria para a geração de produtos cartográficos vem ganhando enfoque nas últimas décadas. Aplicações como o levantamento da altimetria em regiões específicas ou a atualização de mapas na localização de estradas, edificações ou demais alvos são exemplos de aquisições por recursos de sensoriamento remoto. A formação das imagens geralmente é realizada por sensores ópticos ou de radar (*Radio Detection and Ranging*), a bordo de aeronaves ou satélites, denominadas imagens aerotransportadas e orbitais, respectivamente.

O uso de imagens de radar para o reconhecimento de padrões terrestres e o levantamento de informações acerca de mudanças nos alvos da superfície da Terra vem sendo muito utilizado em diferentes áreas de aplicação, tais como: geologia, hidrologia, oceanografia, cartografia e outras. O funcionamento dos radares imageadores é baseado na transmissão e recepção de pulsos de micro-ondas. As imagens são formadas por meio da radiação emitida pelos sensores, em que parte dessa energia é absorvida pelo objeto, parte é perdida no meio, e o restante, retroespalhada. A energia retroespalhada retorna ao sensor que, por sua vez, realiza a leitura.

As inúmeras aplicações em áreas diversas de conhecimento justificam a importância do imageamento por radar, entre essas a geologia, na análise de fendas, foliações geológicas, no estudo do relevo e na prospeção do solo para a identificação de áreas com recursos minerais, entre outras. Na hidrologia, são tratados problemas como o gerenciamento dos recursos hídricos, a detecção de umidade do solo ou pontos de alagamento. Na oceanografia, no monitoramento do mar, poluição marinha causada por derramamentos de óleo, detecção de navegações em áreas de pesca ilegal, apoio para o estabelecimento de rotas marítimas e identificação de ondas internas. Além dessas, o imageamento por radar permite ainda otimizar tarefas na cartografia, tarefas que outrora são inviáveis pela utilização de sensores ópticos, tais como o controle sistemático dos níveis de vegetação, áreas alagadas, desmatamentos, no mapeamento de áreas com vazios cartográficos, cuja topografia é ocultada por copas de árvores em regiões de alta floresta, entre outras.

Em relação ao imageamento por sensores ópticos, o imageamento por radar pode apresentar algumas desvantagens. A presença de ruído inerente ao seu imageamento coerente, distorções por interferências de sinal ou o sofisticado processamento para a geração de imagens são exemplos de alguns dos obstáculos a serem enfrentados.

O ruído proveniente deste tipo de imageamento, denominado ruído *speckle*, pode ser considerado o maior empecilho na identificação dos alvos presentes na cena, necessitando, quase sempre, que a imagem gerada seja submetida a etapas de pré-processamento. Contudo, por serem possuidores de sua própria fonte de energia, denominado sensores ativos, os radares imageadores permitem a aquisição de imagens em quaisquer horários, em regiões com adversidades climáticas, em que os alvos são recobertos por nuvens ou em regiões de mata fechada, em que a topografia é frequentemente ocultada pelo dossel florestal. No imageamento por sensores ópticos, os alvos são mais bem representados quando comparados aos alvos presentes nas imagens adquiridas por sensores de radar. Em função disso, os métodos envolvendo o reconhecimento de alvos específicos são, também, mais bem fundamentados.

Geralmente, no processo de mapeamento cartográfico são utilizados dois métodos para a obtenção da rede de estradas. O primeiro é realizado por uma equipe de especialistas que vai a campo para o levantamento de informações quanto à localização e estrutura das estradas, processo que por vezes consome muito tempo e esforço. No segundo método, a extração de estradas é realizada por meio de dados de sensoriamento remoto aerotransportado ou orbital. Este segundo processo pode ser categorizado como manual, semiautomático ou automático. A extração manual implica no uso de um operador humano para o delineamento de estradas, enquanto que o processo semiautomático requer a interferência humana, além do uso de recursos computacionais, para conduzir o processo. E, finalmente, o processo automático não requer interferências humanas (HAUPTFLEISCH, 2010).

A pesquisa sobre a extração de estradas em imagens de radar de abertura sintética (*Synthetic Aperture Radar - SAR*), surgiu na década de 1970, com o pioneirismo de Bajcsy e Tavakoli (1976) e Quam (1978), e se intensificou desde então. A extração de feições lineares, ou estradas, utilizando imagens digitais é ainda uma questão científica em aberto, e as aplicações existentes exprimem soluções diferenciadas (POZ; AGOURIS, 2001; AGOURIS et al., 2001). Tal processo é considerado uma componente de grande importância na cartografia (DOUCETTE et al., 2004).

As formas de se identificar ou extrair uma estrada presente na imagem podem variar muito. Existem métodos que dispensam qualquer pré-processamento, outros, no entanto, necessitam que a imagem seja submetida a esta etapa. A etapa de pré-processamento inclui processos como a aplicação de filtros para minimizar o efeito do ruído, realces, operadores de erosão, dilatação, afinamento, entre outros. As ca-

racterísticas da imagem *SAR*, tal como o ruído *speckle*, exige, na maioria das vezes, que esta etapa seja realizada, a fim de aumentar a eficiência do método de extração. Após o processo de identificação de estradas, é possível ainda realizar melhoramentos nos resultados e, esta etapa é caracterizada como a fase de pós-processamento, na qual é possível, por exemplo, eliminar os fragmentos identificados erroneamente, combinar segmentos de reta que apresentaram descontinuidades, entre outros.

Alguns métodos de extração de estradas necessitam não só de filtros suavizadores, como também de pontos iniciais para dar início ao processo de extração. Esses pontos, denominados pontos-sementes, representam a localização aproximada de uma estrada na imagem. Os métodos *Snakes* (KASS et al., 1987; GRUEN; LI, 1997; MAYER et al., 1997; LAPTEV et al., 2000) e delineamento automático (ZHAO et al., 2002), são exemplos de métodos que exigem que pontos-sementes sejam definidos próximos a feição de interesse que, a partir daí, evoluam ou para o eixo central ou para as bordas da via. Há dois fatores que fazem com que os trabalhos optem por utilizar semeadores automáticos ou não. Os trabalhos que utilizam a semeadura automática visam alcançar a autonomia do processo, fazendo com que o fator eficiência seja um segundo objetivo. No entanto, outros primam pela eficiência, e utilizam, assim, operadores manuais, a fim de priorizar o processo principal, a extração, e não seus complementos. Em suma, alguns trabalhos utilizam a semeadura automática (HU et al., 2007; DOUCETTE et al., 2004), ao passo que outros priorizam o método de extração e realizam o processo de semeadura manualmente (MAYER et al., 1997; GRUEN; LI, 1997; BAUMGARTNER et al., 2002).

Neste trabalho, é apresentada uma metodologia semiautomática para a extração de estradas em imagens de radar *SAR* aerotransportada da região de Paragominas-PA, abordando o uso de um semeador semiautomático combinado ao princípio de contorno ativo. Esta pesquisa se insere na Rede Brasileira de Visualização Avançada - RVA, uma encomenda da Financiadora de Estudos e Projetos - FINEP, vinculada ao Sistema Brasileiro de Tecnologia de Centros de Inovação - SIBRATEC - do Ministério da Ciência, Tecnologia e Inovação - MCTI, desenvolvida em parceria entre a Pontifícia Universidade Católica do Rio de Janeiro - PUC-Rio, o Instituto Nacional de Pesquisas Espaciais - INPE, e a empresa Orbisat Indústria e Aerolevantamento S.A.

O modelo híbrido proposto constitui-se de dois conceitos: a semeadura semiautomática e o Modelo de Contorno Ativo (*Snakes*). Essas duas abordagens são divididas

em duas etapas. Na primeira etapa, será tratado o processo de sementeação semiautomática, e em seguida, a aplicação do método de extração. Para que parte do conceito de sementeação seja automatizado, é necessário que se criem métodos específicos para a identificação de estradas. Neste caso, adotou-se a utilização das Redes Neurais Artificiais (RNA), com o método de Mapas Auto-organizáveis (*Self Organizing Maps* - SOM) (KOHONEN, 1982). A qualidade da identificação dos pontos nesta etapa influencia diretamente o processo de extração (HARVEY, 1999), e portanto, uma etapa essencial para eficiência do processo. Em um processo intermediário, é apresentado um método para a poda dos pontos-sementes identificados erroneamente pelo método de sementeação, em que são excluídos os pontos indesejáveis. A segunda etapa é caracterizada pela execução do método *Snakes* (KASS et al., 1987), em que a sua entrada refere-se a saída do método de sementeação. Os pontos-sementes obtidos pela etapa anterior são interpolados nesta, caracterizando a curva inicial próxima à estrada. Após essa etapa, encerra-se o processo de extração.

O método *Snakes* consiste de um conjunto de pontos, interpolados, de modo a formar uma curva paramétrica, que deforma-se de uma localização arbitrária inicial para uma localização final desejada, minimizando uma determinada função de energia. A minimização dessa função, dá origem às forças espaciais que agem sobre a curva para deformá-la. A medida que a energia é minimizada, sua localização tende a evoluir para as bordas de um objeto próximo. Tal processo de minimização é constituído de um problema de otimização, que pode ser tratado com diferentes métodos. No modelo tradicional apresentado por Kass et al. (1987), os autores utilizaram o método variacional de Euler-Lagrange, que permite minimizar o funcional de energia, que por sua vez é definida por uma equação diferencial parcial. Li (1997) abordou o conceito de Programação Dinâmica (BELLMAN; DREYFUS, 1962) para o problema de extração de estradas. Amini et al. (1990) apresentaram uma extensão melhorada do método, utilizando, igualmente, a Programação Dinâmica. Neste trabalho, adotou-se o método variacional de Euler-Lagrange para o ajuste da curva, porém, é realizado determinadas adequações ao método de modo que a curva seja aberta e se ajuste aos eixos de simetria do objeto, diferentemente do procedimento tradicional, em que o processo de minimização é aplicado sobre uma curva fechada, com o objetivo de adquirir as fronteiras de objetos.

O conteúdo deste trabalho é apresentado nos quatro capítulos que se seguem. No Capítulo 2, é realizada a revisão da literatura envolvendo a extração de estradas

em imagens digitais, em que são apresentados os principais trabalhos. O Capítulo 3 acompanha a sequência lógica da metodologia apresentada, em que são detalhadas as etapas dos métodos de Semeação, Poda e Modelo de Contorno Ativo. Em seguida, no Capítulo 4, são apresentados e discutidos os resultados adquiridos com o método apresentado. Por fim, no Capítulo 5, são discutidos os aspectos gerais da implementação, apresentando-se as principais contribuições, vantagens e desvantagens da metodologia utilizada.

2 REVISÃO DA LITERATURA

Atualmente, existem inúmeras técnicas para extração de estradas em imagens digitais. As pesquisas envolvendo este problema se resumem em imagens provenientes de sensores ópticos e de micro-ondas, ambos apresentando soluções diferenciadas. Com a grande disponibilidade de artigos abordando o problema, nota-se um intenso avanço no desenvolvimento de novos métodos nas últimas décadas (TUPIN *et al.*, 1998; DOUCETTE *et al.*, 2004; GRUEN; LI, 1997). Tendo em vista a enorme quantidade de trabalhos, este capítulo destina-se a apresentar as principais contribuições para o tema, descrevendo brevemente cada uma delas. A seguir, os trabalhos estão categorizados de acordo com os respectivos métodos de extração.

2.1 Métodos de Semeação (*Seeding*)

Apesar de não se tratar de um método de extração, a semeação é, muitas vezes, uma importante ferramenta nesse processo e, portanto, relacionada neste capítulo como um dos métodos de extração. Sua principal função é selecionar pontos na imagem, na qual cada um destes pontos caracterize a localização aproximada da feição de interesse. A Figura 2.1 ilustra um exemplo de como seria um resultado ótimo de semeação sobre uma imagem óptica, isto é, o número total de pontos identificados corresponde à porcentagem máxima de acertos. Nota-se também, que nenhum ponto foi identificado incorretamente, fator que também contribui para um resultado ótimo.



Figura 2.1 - Exemplo de uma semeação ótima sobre uma imagem óptica.

Geralmente, esse tipo de resultado é obtido a partir de métodos robustos de semeação. Na imagem acima, por exemplo, a identificação destas estradas possivelmente

necessitaria de recursos computacionais na qual consideraria a largura do objeto a ser identificado, seus valores espectrais, sua curvatura, variação da intensidade de cores, os possíveis obstáculos (como árvores, carros e edificações), inclinação, bifurcações, entre outras. Considerar todas essas características nem sempre é uma tarefa fácil, ainda sim, alguns pontos não são devidamente marcados ou são marcados incorretamente, fazendo com que o método sempre chegue a resultados bons, mas não ótimos. Em imagens ópticas, a acurácia dos resultados é uma questão relativamente fácil de se alcançar, quando comparada à de imagens de radar. Nessas imagens a redução do ruído *speckle* (ruído inerente de imagens de radar) é quase sempre uma etapa necessária de pré-processamento.

Uma vez que a sementeação seja aplicada, a informação das prováveis localizações das estradas torna-se conhecida, assim, o processo de extração torna-se ainda mais rápido e eficiente, justificando assim, a importância da sementeação no processo. [Harvey \(1999\)](#) destaca a importância da qualidade dos pontos iniciais para o desempenho do algoritmo de extração. Geralmente, o método de sementeação é utilizado em alguns trabalhos como predecessor ao algoritmo de extração. Algoritmos como o delineador de estradas (*road trackers*) e Modelo de Contorno Ativo (*Snakes*) são exemplos de métodos de extração que utilizam a marcação de pontos-sementes.

Na literatura, a sementeação na abordagem de extração de estradas pode ser categorizada em automática, semiautomática e manual. Alguns dos trabalhos são descritos a seguir.

2.1.1 Sementeação Automática

A sementeação automática é realizada sem nenhuma interferência humana, ou seja, tanto a definição dos parâmetros do método quanto a sua aplicação é realizada de forma autônoma.

[Hu et al. \(2007\)](#) utilizaram o conceito para o delineamento de estradas sobre imagens ópticas, onde a sementeação é realizada de forma automática. Os autores pré-estabelecem as geometrias ideais de uma estrada, definindo formas diferentes para as vias, tais como vias com interseções e contínuas. A partir destas, é realizada uma varredura na imagem, buscando os pontos em que a correlação com essas formas são menores que um determinado limiar.

[Doucette et al. \(2004\)](#) propuseram um eficiente método para a identificação do eixo

de simetria de estradas em imagens ópticas. A abordagem apresentada identifica as bordas paralelas das estradas de acordo com um operador gradiente, que a partir do *pixel* central da via, fornece tanto a orientação da borda quanto a sua localização. Tal processo é realizado pelo algoritmo denominado extrator de eixo anti-paralelo (*Anti-parallel Centerline Extractor - ACE*), proposto em Agouris et al. (2004). Após a identificação das bordas, o processo é seguido pela construção da rede de estradas com o auxílio do algoritmo de auto-organizável de mapeamento de estradas (*Self-Organizing Road Mapping - SORM*), proposto em Doucette et al. (2004).

2.1.2 Semeação Semiautomática

Este processo, realizado de forma semiautomática, diferencia-se apenas pela interferência humana em parte do processo. Essa interferência pode ocorrer no início do método (por exemplo marcação de ponto inicial ou de partida), ou no final (por exemplo ajuste na direção durante o delineamento).

Zhao et al. (2002) apresentaram um algoritmo semiautomático para a extração de estradas em imagens ópticas orbitais de alta resolução. A metodologia apresentada consiste em três etapas, na qual a imagem é submetida a um processo de classificação em uma primeira etapa, e, por meio das classes identificadas como rodovias, é definida uma máscara para utilização nas etapas posteriores. A segunda etapa é caracterizada pelo processo de semeação. Nesta, é obtido primeiramente uma imagem binária correspondente às bordas da imagem. Os pontos são marcados, conforme a análise dos vizinhos dos *pixels* de borda. Na terceira e última etapa da extração, é realizado um processo iterativo, em que, em cada iteração, um modelo de estrada retangular é rotacionado sobre o eixo dos pontos, de forma que a direção do novo ponto apresente maior correlação entre todas as direções analisadas. A rotação da estrada sobre o eixo dos pontos é ilustrada na Figura 2.2. Este método é considerado uma abordagem semiautomática, pois a cada nova direção processada pelo método, esta pode ser contestada e remarcada pelo operador, caso desejável.

2.1.3 Semeação Manual

Neste tipo de semeação, todos os pontos são marcados por um operador humano. Poz e Agouris (2001) utilizaram o conceito de programação dinâmica para a extração de rodovias em imagens ópticas aerotransportadas e orbitais. O processo é inicializado por um operador humano, que fornece, grosseiramente, pontos sementes descrevendo

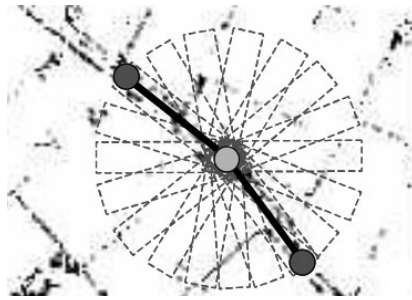


Figura 2.2 - Rotação de estrada retangular sobre o eixo dos pontos num processo de Semeação semiautomática.

Fonte: Adaptado de Zhao et al. (2002).

a região da estrada.

Fazan et al. (2003) apresentam um método de extração de estradas com o auxílio da semeação manual de pontos. Nesta semeação um operador define um conjunto de pontos-sementes, no qual o primeiro identifica o início da rodovia, e um segundo a direção. O processo de extração é realizado pelo delineamento da estrada por estratégia de teste ativo, técnica relacionada com o conceito de visão ativa (POZ; SILVA, 2002), que analisa a trajetória local do último ponto extraído, resultando, por fim, no eixo central da via.

Gruen e Li (1996) utilizaram também um semeador manual para a extração de estradas em imagens ópticas. Nesse trabalho, os autores apresentam uma nova abordagem para a extração de estradas, denominado como Linha Curvilínea Básica de Mínimos Quadrados ou *LSB-Snakes* (*Least Square B-spline Snakes*), que utiliza o ajuste dos mínimos quadrados combinados ao método *Snakes* para o delineamento das fronteiras das vias.

Com essas diversas abordagens, conclui-se que a forma de se identificar a passagem de uma estrada em um determinado ponto na imagem pode ser obtida de diversas formas. A semeação manual mostrou-se como a opção mais vantajosa para o método de extração. Com este tipo de semeação, a localização dos pontos se torna propícia à conversão, uma vez que é possível contornar obstáculos imprevistos na imagem, tais como árvores ou nuvens, no caso de imagem óptica, e estradas com ruído, no caso de imagens de radar. Esse tipo de operação faz com que o processo tenha um provável êxito, no entanto, exige ainda um operador humano.

2.2 Principais Métodos para a Extração de Estradas

2.2.1 Classificação

O processo de classificação consiste em categorizar os dados presentes nas imagens em diferentes classes de padrões, de forma que seja atribuída uma identidade ou valores de probabilidade para os dados analisados. Tais classes representam as feições e alvos terrestres, tais como: água, cerrado, área urbana, área de desmatamento, entre outras. A classificação nada mais é que o reconhecimento desses grupos, cujos membros exibem características comuns.

A classificação de imagens pode ainda ser subdividida em supervisionada e não supervisionada. Na primeira, é definido um conjunto de amostras das categorias a serem utilizadas na classificação e, posteriormente, com o recurso computacional, processa a imagem de modo que cada região analisada seja associada a uma das classes. A segunda é útil quando não se possui informações relativas às classes de interesse na área imageada. Segundo [Hauptfleisch \(2010\)](#), os classificadores podem ser utilizados para categorizar qualquer tipo de dados, o que implica que podem ser utilizados em níveis diferentes no processo de extração de estradas, seja no pré-processamento, processamento ou pós-processamento. Na identificação de estradas, um classificador é tipicamente utilizado de acordo com as propriedades da estrada presente na imagem. Para o problema de extração de estradas em imagens digitais, [Hauptfleisch \(2010\)](#) categoriza cinco classificadores ópticos e supervisionados em:

- **Classificadores Espectrais:** Os classificadores espectrais são utilizados para comparar a assinatura espectral da classe padrão com as amostras não conhecidas. Existem diversas maneiras de se aplicar esse tipo de classificador, entre essas, as mais utilizadas são os classificadores estatísticos, classificadores por Redes Neurais Artificiais (RNA) e classificadores *fuzzy*¹.
- **Classificadores Texturais:** Os classificadores texturais visam dividir as classes de padrões pelas propriedades texturais contidas na imagem. O processo de classificação se inicia definindo as classes de padrões que serão utilizadas para a classificação dos mesmos.
- **Classificadores Geométricos:** Nesse tipo de classificação, as feições são identificadas por suas propriedades estruturais ou geométricas, ou seja, no caso

¹Método matemático (computacional) utilizado no tratamento de incertezas.

da identificação de estradas, por exemplo, as feições tendem a aparecer na imagem como segmentos retos, contínuos, com largura relativamente pequena. Desta forma, estas informações estruturais são levadas em conta na definição das classes de padrões que serão utilizadas na classificação.

- **Classificadores Contextuais:** Na classificação contextual, é necessário que se conheçam as informações dos *pixels* vizinhos para o desenvolvimento do processo de extração. A informação contextual, como é chamada, permite definir regiões onde as estradas possuem características similares, tais como curvatura, largura e níveis de intensidade. Sendo assim, é possível otimizar o processo de extração em diferentes regiões da imagem, a exemplo de estradas presentes em áreas urbanas, que podem diferenciar-se muito das estradas presentes em uma área de vegetação.
- **Classificadores por Padrões Automáticos de Classes:** Os classificadores por reconhecimento automático de classe são classificadores que requerem algum método de aprendizagem no seu processamento. A classificação por reconhecimento de padrões consiste em armazenar as informações obtidas da leitura das classes de amostras, de forma que esta leitura seja útil em futuras identificações, ou seja, é construído uma base de conhecimento tal que ao apresentar um determinado padrão a esse classificador, ele, por meio da correlação entre seu conhecimento e dessa amostra, saberá dizer em qual classe esta pertencerá.

Doucette et al. (2001) utilizaram o método K-médias e os Mapas Auto-Organizáveis para a classificação de imagens ópticas em alta resolução. O algoritmo intitulado SORM (*Self-Organizing Road Map*), possibilita a conexão dos pontos provenientes da classificação, de forma que seja aplicada uma análise topológica com métricas *fuzzy*, obtendo-se, com isso, os eixos centrais do conjunto de estradas.

Baumgartner et al. (1997) fazem um comparativo dos modelos de estradas reais com os presentes em imagens digitais e apresentam um classificador contextual para a extração de estradas. Ao considerar as características radiométricas, geométricas e topológicas de estradas, os autores definem as relações que estas estruturas podem ter com outros objetos na imagem, tais como árvores, carros, edifícios, entre outros. Tal relação é considerada como contexto local enquanto que, as informações quanto à região em que a estrada se encontra, tal como floresta ou área urbana, são

consideradas como contexto global.

Mena e Malpica (2005) apresentaram um método automático para a extração de estradas em imagens óticas de alta resolução, abordando o uso de classificadores estatísticos, em que o processo de classificação resulta em duas classes de padrões, i.e., uma imagem binária contendo as informações de estradas e não estradas. Haralick (1979) apresenta as diferentes abordagens de textura em uma imagem digital, e além disso, generaliza a abordagem estrutural-estatística para textura, agregando a este modelo, novas primitivas de texturização.

2.2.2 Transformada de Hough

A Transformada de Hough (TH) é um operador matemático capaz de detectar formas geométricas em imagens digitais, tais como retas, círculos e elipses. Desenvolvida em 1962 por Paul Hough, tornou-se uma contribuição relevante na área de visão computacional. O método consiste em aplicar na imagem original uma transformação, tal que todos os pontos pertencentes a um mesmo segmento de reta sejam mapeados em um único ponto no domínio da TH.

Por ser um operador muito eficiente na detecção de linhas, a TH é muito utilizada na extração de feições lineares. Dell'Acqua e Gamba (2001) apresentaram uma variação do algoritmo, de forma a serem consideradas diferentes larguras para as feições. Além de reduzir o tempo de processamento, o algoritmo fornece melhores resultados, detectando os pontos de início e término dos segmentos retos, ao invés de linhas completas. A utilização da TH pode ser aplicada também na detecção de curvas com baixo grau de curvatura.

Amberg et al. (2004) utilizaram, a princípio, a Programação Dinâmica para a extração dos eixos centrais das vias, e posteriormente, a TH é utilizada para a conexão dos pontos descontínuos provenientes da extração.

Hu et al. (2004) obtiveram bons resultados utilizando a TH como auxiliadora no processo de extração de estradas. O método apresentado pelos autores é usado em uma imagem óptica de alta resolução, em que a TH é aplicada iterativamente sobre uma imagem classificada, até que se obtenham todas as vias.

2.2.3 Morfologia Matemática

A morfologia matemática é utilizada há muito tempo no processamento de imagens digitais, e as aplicações realizadas com operadores morfológicos remetem à Teoria dos Conjuntos. Processos como união, interseção, dilatação, erosão, fechamento, abertura e afinamento são operações fundamentais na morfologia matemática. Os operadores são utilizados no problema de extração de estradas para a remoção de pequenas regiões indesejáveis, tais como buracos, sombras ou carros, permitindo também a suavização de formas e contornos. Na extração de estradas, a morfologia matemática é frequentemente utilizada como filtro durante as etapas de pré e pós-processamento (HAUPTFLEISCH, 2010).

Os processos de dilatação, erosão, abertura e afinamento formam a base da morfologia matemática (GONZALEZ; WOODS, 2008). Tais técnicas são frequentemente utilizadas como recurso na extração de estradas. Chanussot et al. (1999) apresentaram um método automático para a extração de feições lineares em imagens *SAR*, na qual utilizaram operadores morfológicos para a filtragem das estradas e remoção de regiões indesejáveis. A princípio, são definidos as características geométricas da via, tais como largura, contraste e comprimento. Na Figura 2.3 é ilustrado este processo em um recorte na imagem. Para tais filtragens, os autores utilizaram o operador morfológico de abertura (*opening*) para a remoção de pequenas regiões escuras e entorno claro, denominados vales Figura 2.3(b), em seguida é aplicado o mesmo operador, no entanto, direcional, removendo cristas descontínuas Figura 2.3(c). Após a remoção desses pontos, a imagem torna-se mais suave, aplicando-se o operador de fechamento (*closing*) as regiões de vales são removidas Figura 2.3(d). Por fim, é aplicado o operador cartola (*top-hat*) para a remoção de resíduos escuros, restantes do último processamento, resultado em uma imagem com feições claras realçadas Figura 2.3(e).

Idbraim et al. (2008) utilizaram o operador morfológico de abertura para a reconstrução de elementos lineares em imagens ópticas. A operação é realizada com um operador direcional de estrutura linear, sendo sua largura de cinco *pixels*. Esse processo é realizado em complemento na aplicação de filtros de realces, onde são destacadas as feições conforme sua orientação. Um operador muito utilizado para a obtenção do eixo de simetria de feições lineares, é o operador de afinamento, que tem como objetivo obter o esqueleto da forma.

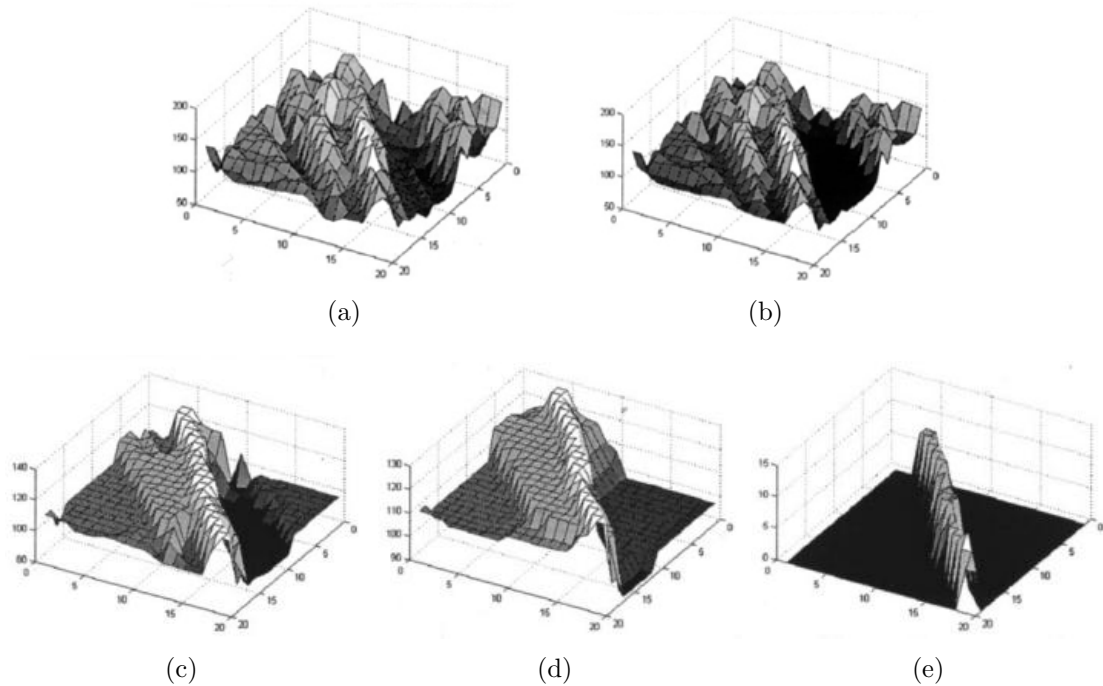


Figura 2.3 - Aplicação do operador morfológico de abertura para realce de estradas. (a) Imagem original. (b) Imagem após a aplicação do operador de abertura. (c) Imagem após a aplicação do operador de abertura direcional. (d) Imagem após a aplicação do operador de fechamento. (e) Imagem final, após a aplicação do operador *top-hat*.

Fonte: Adaptado de [Chanussot et al. \(1999\)](#).

[Zhang et al. \(1999\)](#) utilizaram o processo de afinamento para a obtenção dos eixos de simetria e estradas. O processo é realizado sobre uma imagem segmentada, onde são aplicados dois outros operadores: de abertura, para que sejam removidos pontos espúrios da detecção, e de fechamento, para que os extremos dos segmentos de estrada desconectados sejam conectados.

Os operadores morfológicos são, em geral, componentes importantes no processamento de imagens digitais e principalmente na extração de estradas. Sua aplicação é eficaz na remoção de regiões indesejadas e operações que envolvam o contexto local.

2.2.4 Segmentação

O processo de segmentação por regiões em imagens digitais é o processo de agrupamento de dadas regiões, com o objetivo de simplificar a representação da imagem e,

consequentemente, facilitar o seu processo e análise. Barrow e Tenenbaum (1978) definem como segmentação por regiões o processo de particionamento de uma imagem em regiões semanticamente interpretadas. O resultado gerado a partir deste processo consiste em um conjunto de regiões disjuntas, com interseção nula e cuja união do conjunto de regiões é a própria imagem, ou seja, considerando I uma imagem, a segmentação é o processo que particiona I em n sub-regiões R_i , para $i = 1, 2, \dots, n$, tal que:

- a) $\cup_{i=1}^n R_i = I$
- b) R_i é uma região conectada, $i = 1, 2, \dots, n$
- c) $R_i \cap R_j = \emptyset \forall i \neq j$
- d) $P(R_i) = \text{verdadeiro}$, $i = 1, 2, \dots, n$
- e) $P(R_i \cup R_j) = \text{falso}$, para $i \neq j$

em que $P(R_i)$ representa um predicado lógico sobre os pontos no conjunto R_i . A primeira condição indica que todos os *pixels* da imagem pertencem a uma determinada região. A condição **b** indica que todos os pontos de uma determinada região devem estar conectados. A condição **c** indica que as regiões devem ser disjuntas. A condição **d** determina a propriedade que deve ser satisfeita pelos *pixels* em uma determinada sub-região. Por último, a condição **e** indica que as sub-regiões R_i e R_j são distintas em relação à propriedade definida (GONZALEZ; WOODS, 2008). Os *pixels* pertencentes a uma dada região possuem características similares, e cada região difere significativamente de suas vizinhas.

Assim como a sementeação, a segmentação não é considerada um método de extração de estradas como listado nessa seção, embora seja uma operação muito útil e eficiente em grande parte dos métodos de extração. O resultado desta operação permite separar as diversas regiões contidas na imagem em regiões únicas e bem definidas, tornando possível, por exemplo, a seleção dos segmentos pertencentes às regiões de estradas. A segmentação tem um papel importante no processamento de imagens de radar. Mena e Malpica (2005) utilizaram a segmentação baseada na análise de textura da imagem, em que o processo é realizado sobre uma imagem óptica. Após concluir a etapa de segmentação, a imagem é submetida a um processo de esqueletização, fornecendo os eixos de simetria da rede de estradas. Amini et al. (2002)

utilizaram o método de divisão e junção (*split and merge*) para a segmentação. O método é baseado na estrutura “*quadtree*”² para a partição da imagem, que explora as regiões homogêneas (*split*) e em seguida as combinam caso suas vizinhas possuam características semelhantes. Ao fim, é criada uma imagem segmentada.

Grote et al. (2009) utilizaram o conceito de segmentação para a extração de estradas em imagens ópticas de alta resolução, focando a extração em áreas urbanas. Com o processo de segmentação, são geradas pequenas regiões na imagem cujas bordas coincidem com as bordas das estradas, e os extremos de cada região são então conectados aos extremos de outras regiões, de acordo com critérios de distância, direção e ângulo.

2.2.5 Modelos Deformáveis

O conceito tradicional denominado Modelo de Contorno Ativo ou simplesmente *Snares*, possui um grande número de trabalhos abordando-o como método extrator de estradas, principalmente, ao se tratar da obtenção dos eixos centrais das vias. Desenvolvido inicialmente por Kass et al. (1987), o método consiste em um processo iterativo e adaptativo para a identificação de contornos de objetos, de modo que uma curva poligonal seja convergida para as formas do objeto. O progresso da curva até a localização desejada é alcançada minimizando-se, a cada ciclo iterativo, uma determinada propriedade da curva, denominada energia. A Figura 2.4 ilustra este processo em sua representação tradicional.

A utilização de Modelos Deformáveis na extração de estradas em imagens digitais consiste em uma abordagem promissora, embora se trate de uma técnica complexa, sua utilização permite identificar com exatidão o contorno de objetos presentes na cena. No entanto, o modelo apresenta uma sensibilidade muito grande quanto a seleção de seus parâmetros. Na literatura, os trabalhos de extração de estradas abordando o uso de Modelos Deformáveis mostram-se eficazes em imagens ópticas. Por outro lado, na imagem de radar, a aplicação desse modelo possui limitações, justamente pela presença de ruído em grande quantidade. A área de Modelos Deformáveis possui inúmeras variantes, todas com o objetivo de reconhecer o contorno de objetos presentes na cena. Nesta seção, serão apresentados alguns trabalhos que utilizaram tal recurso para a extração de estradas, destacando, principalmente, o Modelo de

²Árvore estrutural de dados em que subdivide recursivamente a imagem em quadrantes de quatro, a fim de determinar a homogeneidade de regiões (FINKEL et al., 1974).

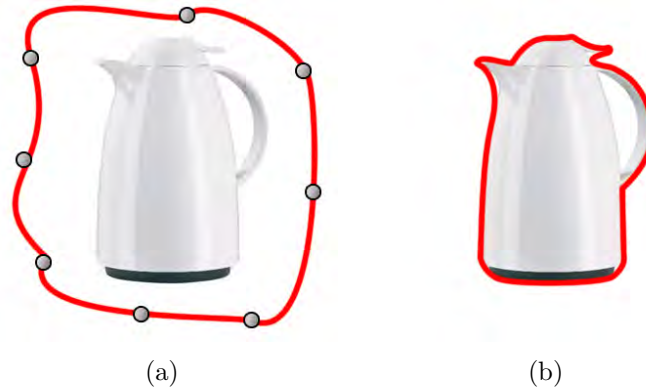


Figura 2.4 - Ilustração da aplicação de *Snakes* com curva fechada (método tradicional) para a identificação de objetos: (a) Interpolação inicial. (b) Evolução final da curva.

Contorno Ativo, que será melhor discutido na Seção [A.15](#).

Silva et al. (2004) introduz algumas das técnicas para a obtenção de contornos de objetos, tais como o Modelo Geométrico (*Level Set*) (OSHER; SETHIAN, 1988), cujo processo de identificação de contornos não necessita da parametrização da curva, embora basear-se também na otimização de uma função objetivo. O Modelo de Formas Ativas (*Active Shape Models*) (COOTES et al., 1999) não é uma abordagem paramétrica, ao invés disso, o modelo se baseia nas características do objeto a ser identificado, que após definido, aplica-se o processo de deformação, obtendo o contorno das formas de acordo com a minimização da distância entre os pontos correspondentes à forma e ao objeto a ser extraído. Os Modelos Probabilísticos (STAIB; DUNCAN, 1992) permitem incorporar funções de probabilidade tal que delimitem na imagem, as estruturas das classes a serem extraídas.

Outros métodos, como o modelo Balão (*Balloon*)(COHEN, 1991), possui características semelhantes comparado ao método tradicional, no entanto, a curva inicial deste é iniciada dentro do objeto a ser extraído, e evolui até que se alcance as fronteiras do objeto, comportando-se como um balão ao ser enchido. O método de Fluxo do Vetor Gradiente (*Gradient Vector Flow - GVF*) (XU; PRINCE, 1997) é uma outra alternativa para a aquisição de contornos de objetos. Trata-se também de uma abordagem paramétrica que agrega uma nova energia externa baseando-se no gradiente da imagem. O conceito apresenta benefícios, principalmente, pela não necessidade de pontos-sementes próximos ao contorno, além de convergir eficientemente para as

formas do objeto.

O método possibilitou uma significativa contribuição para a área de Visão Computacional. Como mencionado anteriormente, surgiram inúmeras variantes a partir desse trabalho, entre essas evoluções, o método *Snakes* deixou de ser modelado apenas como uma curva fechada, como em sua representação original, mas passou também a possuir representações abertas. A curva aberta possibilita o delineamento de segmentos retos, tais como as estradas na Figura 2.5.



Figura 2.5 - Ilustração da aplicação de *Snakes* com curva aberta para a identificação de estradas: (a) Interpolação inicial. (b) Evolução final da curva em um processo de obtenção do eixo de simetria.

Dada as suas características o método *Snakes* é frequentemente utilizado na extração de estradas. Gruen e Li (1997) utilizaram uma extensão ao método, cuja abordagem permite extrair os eixos centrais das vias em imagens ópticas. Nessa abordagem, o método *Snakes* é executado em conjunto com a Programação Dinâmica (BELLMAN, 1954), o qual é utilizada como método de otimização na evolução da curva. A abordagem apresentada pelos autores permite a extração de estradas, nas quais existam obstáculos, como sombras causadas por árvores, veículos, edificações.

Mayer et al. (1997) utilizaram o método *Ribbon-Snakes* em imagens ópticas de alta resolução. O método estende a abordagem tradicional de *Snakes*, em que acrescenta uma característica de largura w no modelo paramétrico. Desta forma, ao extrair o eixo de simetria da feição de interesse, são expandidos paralelamente ao ponto (x_i, y_i) , a distância w_i para cada um dos lados da curva, de modo a obter as fronteiras das estradas Figura 2.6. Em relação a abordagem tradicional, o método

Ribbon-Snakes pode apresentar algumas vantagens. Obter as bordas da estrada é um processo útil em regiões em que a largura das vias varia ou, até mesmo, por se obter a largura das vias mesmo em condições adversas, tal como a presença de sombras ou árvores (em imagens ópticas) que prejudicam a identificação real das fronteiras. O mesmo é considerado para a imagem de radar, embora os obstáculos na extração sejam outros.

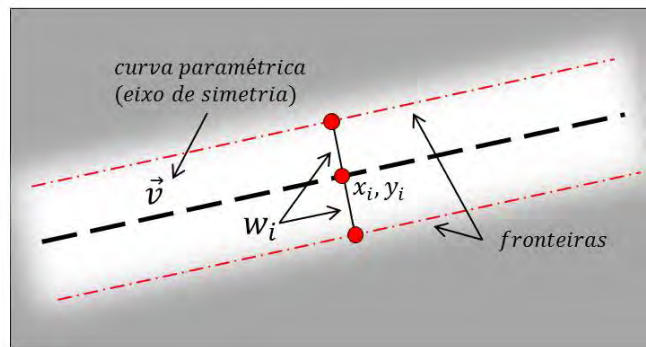


Figura 2.6 - Ilustração do conceito de expansão do método *Ribbon-Snakes*.
 Fonte: Adaptado de Mayer et al. (1997).

Bentabet et al. (2003) propuseram um método para a extração de estradas em imagens de radar, utilizando para tanto o método *Snakes*. Em uma etapa de pré-processamento, os autores utilizaram filtros redutores de ruído *speckle*, destacando os filtros de *Frost* (FROST et al., 1982) e *Lee* (LEE, 1980) como sendo filtros eficientes na minimização deste tipo de ruído. Após a etapa de suavização, é aplicado um detector de linhas, proposto por Ziou (2000) e, por fim, utilizou-se de *Snakes* para a extração das vias. Os parâmetros de rigidez e elasticidade (ver Seção A.15), utilizados pela *Snakes*, são estimados automaticamente no processo de extração apresentado.

Agouris et al. (2001) apresentaram uma extensão do método *Snakes* para a detecção de estradas em imagens ópticas, denominado Modelo de *Snakes* diferencial. Neste caso, o método original é acrescido de um novo termo, que é relacionado com as informações da imagem existentes em um banco de dados geográficos. Tal termo descreve a diferença entre a solução atual da curva com a informação pré-existente. Com isso, são analisadas as mudanças geoespaciais das estradas em relação ao dado armazenado e, se houver mudanças consideráveis, estas são atualizadas no banco.

Bentabet et al. (2003) classificaram os principais trabalhos envolvendo a extração de estradas em imagens de radar, organizando-os por complexidade. Na próxima seção, são apresentados alguns desses trabalhos, que não se enquadraram de forma específica nos tópicos apresentados neste capítulo, porém, possuem grande importância na literatura sobre a extração de estradas em imagens digitais.

2.2.6 Outras Abordagens

Muitos dos trabalhos envolvendo o problema de extração de estradas, não possuem ainda uma abordagem específica para a sua resolução, alguns por apresentarem conceitos híbridos, outros por apresentarem conceitos inovadores. Nesta seção, são descritos alguns dos importantes trabalhos envolvendo a extração de estradas em imagens de radar.

Tupin et al. (1998) propuseram um eficiente método automático para a extração de feições lineares em imagens de radar. Para tanto, os autores dividiram o processamento em duas etapas. A primeira é caracterizada pela detecção das estruturas lineares na imagem, onde são utilizados dois filtros: o detector de bordas por relação (BOVIK, 1988), utilizado para o realce de linhas na presença de ruído *speckle*, e um segundo filtro proposto pelos autores, que consiste em um detector de linhas por correlação cruzada, chamados pelos autores de D1 e D2, respectivamente. Os resultados de ambos detectores são combinados e repassados à segunda etapa do processamento, esta caracterizada pela conexão dos segmentos lineares e construção da rede de estradas. Isso é executado, criando-se um Campo Aleatório Markoviano (*Markov Random Field - MRF*), técnica melhor fundamentada em Spitzer (1971), em que, para cada extremo dos segmentos de reta, é agregada uma probabilidade de junção, probabilidade esta estabelecida de acordo com regras geométricas entre o par de pontos, indicando a possível conexão. Assim, é gerado um mapa de possíveis conexões, na qual somente os pares com probabilidades satisfeitas segundo um limiar são conectados.

Costa e Medeiros (2002) estenderam a abordagem dos detectores de Tupin et al. (1998) e aplicaram a TH sobre o resultado dos detectores proposto pelos autores. Lisini et al. (2006) utilizaram também os detectores propostos em Tupin et al. (1998). Nessa abordagem apresentada pelos autores, são aplicados os detectores de linhas e, posteriormente, ajusta-se os resultados com a imagem classificada. Com isso, é realizado um processo de reconstrução da rede de estradas, na qual são selecionados

os segmentos de reta que satisfazem os valores de largura (*pixels*) e orientação. Seleccionadas as vias, estas são submetidas ao operador morfológico de afinamento, que obtêm seus eixos de simetria. Finalmente, é aplicado um algoritmo de linearização, tornando-as mais alinhadas.

Zhou et al. (2011) combinaram a análise multiescala com informações polarimétricas para a extração de feições lineares em imagens de radar. São aplicados detectores de linhas em baixa resolução em diferentes polarizações da imagem. Em seguida, cada resultado é submetido a uma combinação. Em uma segunda etapa, é realizada a detecção de linhas em alta resolução, em que são aplicados detectores *fuzzy* em uma imagem filtrada.

Baumgartner et al. (1999) propuseram um método automático para a extração de estradas em imagens ópticas, utilizando o conceito multiescala combinado à análise do contexto global e local da imagem. Inicialmente, o processo de extração é realizado sobre áreas rurais, e depois estendido às áreas urbanas e de florestas. Esta ordem de processamento é justificada já que nas áreas rurais possuem interferências menores de objetos como edifícios, automóveis ou árvores, e assim, o processo realizado nas áreas rurais pode ser utilizado como modelo para as etapas seguintes. Esse processo é realizado em diferentes escalas da imagem, onde, em baixas resoluções são aplicados filtros detectores de borda. A partir desta análise, são obtidas as hipóteses dos segmentos de estradas, os quais são posteriormente combinados, formando a rede de estradas. A abordagem se mostrou útil não só pelo resultado da extração, mas também por apresentar eficiência nos resultados onde houve intersecções de vias.

Xiao et al. (2004) propuseram uma nova forma de extrair estradas em imagens de radar. A abordagem tratada pelos autores remete à Inteligência Computacional, com o uso de algoritmo genético (*Genetic Algorithm - GA*). A princípio, a imagem é submetida a uma etapa de pré-processamento, na qual é realizada a classificação da mesma em classes, como vegetação, áreas de construções e estradas. A classificação é realizada por um operador *fuzzy*, denominado FCM (*Fuzzy C Means*). Após a classificação, é atribuído um conjunto de pontos sementes para cada segmento de estrada, em que, para cada conjunto, é estabelecida uma curva passando por seus respectivos pontos. O GA implica na busca de soluções inspirada na genética humana de evolução.

Entre as inúmeras técnicas para a extração de estradas tanto em imagens óticas

como de radar, nesta seção foram destacadas as de maior importância na literatura. O levantamento de soluções para o problema é vasto, e fica evidente que outros métodos adequariam-se perfeitamente ao conteúdo deste trabalho. Um levantamento mais completo envolvendo a extração de estradas é apresentado por [Mena \(2003\)](#).

Grande parte dos trabalhos na área utilizam operadores manuais ou semiautomáticos para a sementeção. Um método completamente automático, tal como apresentado por [Tupin et al. \(1998\)](#), é construído, geralmente, sobre muita pesquisa e cautela. Os detalhes envolvendo a implementação automática são, quase sempre, constituídos por modelos complexos de extração de estradas, o que acaba por ultrapassar o escopo desta pesquisa. O propósito deste trabalho é apresentar um operador que realize a extração dos eixos de simetria das estradas e caminhos presentes em imagens de radar *SAR* aerotransportadas, de forma que este processo seja realizado de maneira semiautomática. Para tanto, será apresentada uma abordagem híbrida, utilizando dois conceitos descritos anteriormente: o método de sementeção para a identificação de pontos que caracterizam a presença de uma estrada, conjugado ao método *Snakes*.

O método de sementeção apresentado é fundamentado em técnicas de Inteligência Computacional, mais precisamente nos Mapas Auto-Organizáveis, técnica que permite o reconhecimento de formas em imagens digitais, de acordo com padrões de treinamento. O método de sementeção possui autonomia em parte de sua execução, portanto, é denominado um método semiautomático. Os pontos-sementes identificados pelo semeador são interpolados conforme o segmento de reta identificado, a curva inicial gerada é aplicada ao método *Snakes* que, por fim, obtêm os eixos centrais das estradas. Este trabalho não busca uma solução ótima para o problema, mas sim a pesquisa acerca da utilização de métodos específicos em processamento de imagem de radar de abertura sintética de alta resolução espacial.

No capítulo seguinte, será apresentado o detalhamento do processo de extração de estradas. A primeira parte da seção é destinada à apresentação dos dados utilizados para o conjunto de testes do método. Na segunda parte, são descritos todas as etapas de desenvolvimento do método, iniciando-se pelo detalhamento do método de sementeção e, posteriormente, o método *Snakes* para a identificação de estradas. Por fim, os resultados são analisados e discutidos, seguindo critérios específicos de avaliação.

3 CONCEITOS E TECNOLOGIA

Neste capítulo, são apresentados conceitos envolvendo a implementação do método de extração de estradas. A princípio, são apresentados os fundamentos do imageamento por radar, em que são mostrados os conceitos básicos envolvendo a aquisição de uma imagem SAR. Em seguida, são apresentadas as características matemáticas da estrada, objeto a ser extraído na imagem. Por fim, descreve-se o procedimento para a extração de estradas utilizando os métodos de Semeação e Modelo de Contorno Ativo.

3.1 Fundamentos do imageamento por radar

A geometria básica da formação de uma imagem SAR aerotransportada pode ser ilustrada pela Figura 3.1, em que V determina a velocidade segundo a qual a plataforma se move em uma determinada direção, conhecida também por direção de azimute. O parâmetro H representa a altitude em que ela se encontra. Esta plataforma transporta uma antena com observação lateral, que ilumina a superfície em estudo com pulsos de radiação eletromagnética. A distância da plataforma até a área iluminada é conhecida como *slant range* R_0 , e a distância da superfície da Terra sobre a qual se encontra a plataforma (nadir) até a área iluminada é conhecida como *ground range*, expresso por R_g na Figura 3.1 (OLIVER; QUEGAN, 1998).

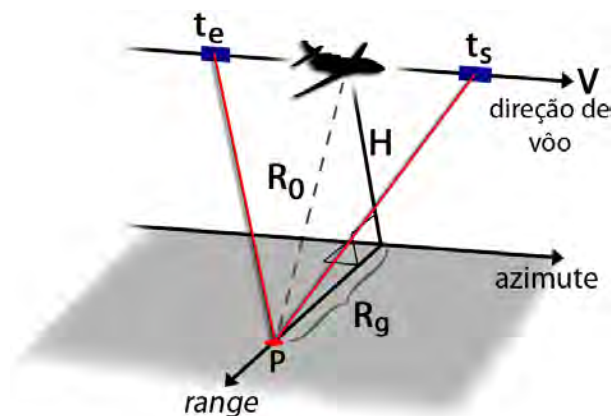


Figura 3.1 - Geometria de imageamento de radares aerotransportados de visada lateral.
Fonte: Adaptado de Oliver e Quegan (1998).

Os sistemas de radar são formados por um transmissor, receptor, antena, processador de sinais e uma unidade de visualização. Uma sequência de pulsos é emitida em intervalos de tempos regulares, sendo que esses sinais se propagam em uma área iluminada pela antena. O objeto iluminado absorve uma fração da energia incidente e espalha o restante de acordo com os parâmetros do sistema e as propriedades físicas do objeto. Parte da energia espalhada é retornada para o sensor, caracterizando o sinal recebido (CARRARA et al., 1995).

Os sensores a bordo de aeronaves permitem obter imagens com maior resolução (1 a 20 metros) quando comparados aos sensores a bordo de satélites, porém sua área iluminada é menor. Os satélites podem visualizar uma área maior da superfície da Terra. Estando continuamente em órbita terrestre, é relativamente fácil coletar imagens da mesma área de modo sistemático, a fim de monitorar mudanças na biosfera.

As imagens de radar são geradas a partir da emissão de pulsos de largura T_p em intervalos de T segundos. Utilizando o modelo de ponto fixo P , mostrado na Figura 3.1, o sistema de imageamento *SAR* possui um intervalo de tempo $(t_s - t_e)$, onde t_e representa o ponto onde é iniciado a transmissão dos pulsos pelo sensor, e t_s o ponto final da transmissão. Neste intervalo, o radar envia N pulsos e coleta N pulsos resultantes do retorno do ponto P . As amostras recebidas são retidas na memória. Durante este intervalo, a plataforma se desloca $(t_s - t_e)$ metros com velocidade V . Este intervalo é conhecido como comprimento de abertura sintética. Os pulsos recebidos sofrem variações de frequência em virtude da velocidade, consistindo no efeito *Doppler* (OLIVER; QUEGAN, 1998).

Os sensores de radar operam na faixa de micro-ondas, na qual são estabelecidas as bandas para cada intervalo de comprimento de onda e frequência, como mostrado na Tabela 3.1. Em vista da frequência e do comprimento de onda de cada banda, o comportamento do sinal em relação ao objeto pode variar consideravelmente. Quanto maior a frequência do radar, maior o nível de potência requerida e maior sua penetração na superfície da Terra, o que influencia sobremaneira na formação da imagem final. Alguns dos parâmetros do sistema que influenciam no retroespalhamento são: frequência, polarização, ângulo de incidência e resolução. A frequência é o número de períodos ou ciclos completos por segundo, cada ciclo completo em um segundo é denominado um *Hertz* (Hz). A polarização de uma onda eletromagnética, em geral, é definida pela figura geométrica que o vetor campo elétrico descreve no espaço. Os

parâmetros do alvo são fatores que podem também influenciar no retroespalhamento do sinal, tais como: rugosidade, constante dielétrica e geometria.

Tabela 3.1 - Principais bandas do espectro de frequência em micro-ondas.

Banda	Frequência(GHz)	Comprimento de onda(cm)
P	0,3 <	100 a 90
L	0,3 a 1,5	90 a 25
S	1,5 a 3,9	24 a 9
C	4 a 6	9 a 3,75
X	6 a 11	3,75 a 2,5
K	11 a 32	2,5 a 0,8

Fonte: Adaptado de Curlander e McDonough (1991).

Com o fim de apresentar um método de extração de estradas em imagens de radar, cuja solução aborde o método de sementeação semiautomático e Modelo de Contorno Ativo (*Snakes*), é necessário que se definam, a princípio, as características dos objetos a serem identificados no processo de extração. Suas dimensões, comprimento e níveis de intensidade são requeridas na fase de treinamento da rede neural artificial (ver seção 3.3.1).

3.2 Definições de Estrada em uma Imagem Digital

Segundo Gruen e Li (1997), para a construção de um método de extração de estradas é necessário que se defina a *priori* um modelo genérico de estrada, na qual descreva matematicamente sua aparência em uma imagem digital. Considerando ζ uma curva em uma imagem digital e uma estrada no espaço objeto, são definidas as seguintes propriedades de uma estrada genérica:

- A curva ζ pode ser representada por uma função vetor $f(s)$, em que mapeia o comprimento de arco s para os pontos (x, y) na imagem.
- A curva ζ possui derivadas contínuas, e um vetor unitário $n(s)$ é normal a ζ .
- A imagem digital pré-processada é representada por uma função bidimensional $G(x, y)$, mensurável, de energia finita e derivada contínua.

As propriedades genéricas de estrada são apresentadas acima como sendo suposições, baseando-se apenas no conhecimento sobre o objeto estrada. As formulações matemáticas equivalentes são apresentadas a seguir.

- As estradas são identificadas por regiões estreitas e contínuas de alta intensidade e entorno de baixa intensidade. A equação que descreve estas características é dada por:

$$E_p = \int [G(f(s))]^2 ds \Rightarrow \max , \quad (3.1)$$

em que E_p denota o quadrado da soma dos níveis de cinza ao longo de uma curva ζ , que deverá ser máximo, ou seja, uma estrada de perfil claro, o que nem sempre ocorre no imageamento por radar, uma vez que as estradas deste tipo podem apresentar tonalidades mais escuras que seu entorno.

- Os valores de níveis de cinza ao longo da estrada devem permanecer constantes em curtas distâncias, o que equivale à:

$$E_z = \sum_i \int \Delta_{si} [G(f(s)) - G_m(\Delta_{si})]^2 ds \Rightarrow \min , \quad (3.2)$$

em que i é o índice de um determinado ponto na curva, Δ_{si} é um pequeno trecho da curva ζ , e $G_m(\Delta_{si})$ é o valor médio de G em Δ_{si} , expresso por:

$$G_m(\Delta_{si}) = \frac{\int_{\Delta_{si}} G(f(s)) ds}{|\Delta_{si}|} , \quad (3.3)$$

em que $|\cdot|$ corresponde ao tamanho do trecho Δ_{si} .

- As propriedades apresentadas acima descrevem as estradas como feições lineares claras. Tais propriedades podem ser generalizadas em uma única equação E_{pz} , dada por:

$$E_{pz} = \int w(d(s)) [G(f(s) + d(s)n(s))]^2 ds \Rightarrow \max , \quad (3.4)$$

em que $d(s)$ é a distância entre o ponto s em ζ e a feição linear próxima a ela; $w(d(s))$ é uma função de pesagem gaussiana que decresce quando a distância $d(s)$ aumenta.

- A maioria das estradas consistem de segmentos de retas conectadas com curvas suaves, representadas por:

$$E_g = \int |f''(s)|^2 ds \Rightarrow \min , \quad (3.5)$$

em que $|\cdot|$ representa o tamanho do vetor normal $f''(s)$ no ponto s em ζ , cuja variação, E_g , ao longo desta, seja mínima. Isto implica que os vetores tangente ($f'(s)$) e normal ($f''(s)$), são contínuos em ζ . Sendo assim, ζ pode ser representada por uma *spline* cúbica.

- A curvatura em um ponto s de uma estrada deve possuir limite superior, ou a mudança local de direção ascendentemente limitada. Que significa dizer que o vetor normal em um determinado ponto deverá permanecer abaixo de uma limiar:

$$C_g = |f''(s)| < T_1 , \quad (3.6)$$

em que T_1 é o limiar que especifica o valor máximo de curvatura permitido para um determinado ponto s de ζ .

- A largura da estrada não se altera consideravelmente. Portanto, não é dada uma representação matemática para esta propriedade.

Na Figura 3.2 são ilustradas algumas das propriedades matemáticas apresentadas acima, tal que a curva ζ representa a curva inicial sobre a estrada, adquirida com a interpolação de seus vértices, ilustrados na imagem como pontos pretos sobre a curva.

3.3 Metodologia

A sequência de apresentação desta seção segue a lógica de processamento do método de extração de estradas apresentado. Cada um dos conceitos utilizados são dispostos em quatro seções, detalhadas de acordo com o fluxograma na Figura 3.3. Primeiramente, emprega-se o método de Semeação; na segunda seção, o método de poda de pontos espúrios e, na sequência, *Snakes*. Por fim, na seção Métricas de Qualidade são apresentadas as medidas utilizadas para a avaliação dos resultados.

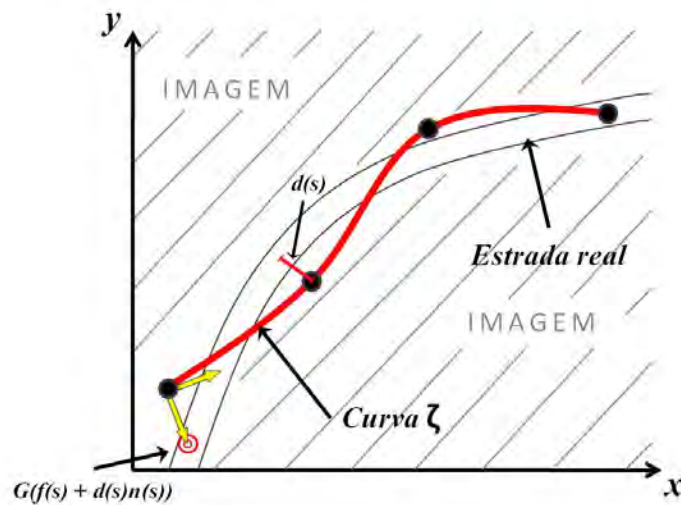


Figura 3.2 - Ilustração da curva ζ sobre estrada em uma imagem digital, segundo propriedades matemáticas definidas por Gruen e Li (1997).

3.3.1 Semeação

O método de semeação tem um papel importante no processo de extração de estradas. O procedimento consiste basicamente em identificar pontos que caracterizam a passagem de uma estrada na imagem. Como foi mencionado na seção anterior, o método de semeação não é um método de extração de estradas por si só, embora sua utilização tenha uma parcela importante no processo. Os pontos-sementes identificados nesta etapa interferem diretamente no método *Snakes*, pois uma vez que os pontos identificados estejam próximos à feição de interesse, o método de *Snakes* tende a convergir rapidamente e eficientemente.

Para que a semeação tenha autonomia na identificação das estradas, é necessário atribuir a ela heurísticas ou modelos que reconheçam na imagem as características de sua forma. Para tanto, será utilizado um método de reconhecimento de padrões utilizando métodos em Inteligência Computacional, caracterizadas por simular, computacionalmente, os comportamentos e a aprendizagem humana.

O processo de reconhecimento das formas presentes na imagem inicia-se apresentando-se à rede neural artificial todas os padrões a serem reconhecidos, além daqueles que não caracterizam as feições de interesse. Neste trabalho, os padrões de treinamento são definidos a partir da observação de perfis ideais de estrada em uma

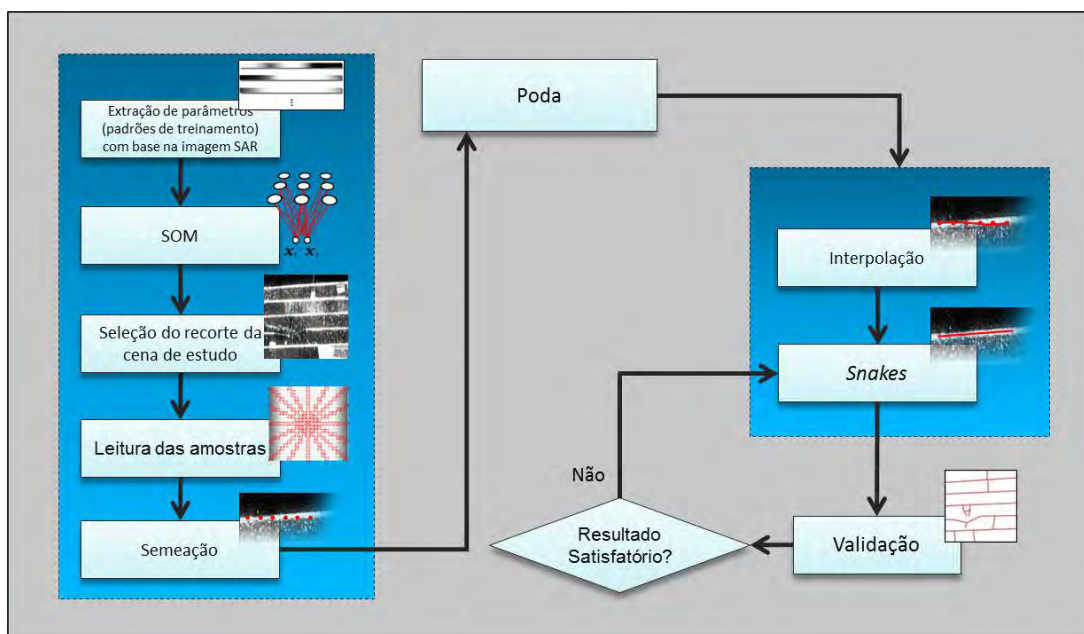


Figura 3.3 - Fluxograma do método de extração de estradas.

imagem SAR, em que são adquiridos os níveis de intensidade de diferentes perfis. Após a extração dos parâmetros referentes ao conjunto de padrões, é iniciada a etapa de treinamento da rede, processo detalhado na próxima seção.

3.3.1.1 *Self-Organizing Maps - SOM*

Um dos primeiros modelos matemáticos a simular a aprendizagem humana foi desenvolvido em 1943 por Warren McCulloch e Walter Pitts (MCCULLOCH; PITTS, 1943). Constituíam-se de um conjunto de unidades processadoras de informação, denominados neurônios ou nós, interligados entre si, simulando uma rede neural. Mais tarde, em 1958, tal modelo foi estendido por Rosenblatt (ROSENBLATT, 1958; ROSENBLATT, 1962) e chamado de rede Perceptron, constituída por duas camadas de neurônios, uma camada de entrada e outra de saída, de forma que cada neurônio de uma camada era interligado com os da outra. Devido a esta estrutura, sua utilização era limitada a resolver apenas problemas linearmente separáveis. Diante de tais limitações, fez necessário a criação de uma nova rede Perceptron com uma ou mais camadas intermediárias de neurônios, o que possibilitaria a resolução de problemas mais complexos. A rede foi chamada de Perceptron de múltiplas camadas, ou simplesmente rede MLP (*Multi-Layer Perceptron*). Este modelo possibilitou a resolução de problemas mais complexos.

Mais tarde, em 1982, surgiram os Mapas Auto-Organizáveis (*Self-Organizing Maps* - *SOM*), baseados no aprendizado não supervisionado, em que não é necessário apresentar à rede uma entrada e saída desejada para que ocorra o aprendizado. A ativação dos neurônios é realizada em um processo de “competição” entre os mesmos, cujo lema é “o vencedor leva tudo”, de modo que o neurônio com menor estímulo a uma determinada entrada é nomeado vencedor, tendo como recompensa o ajuste de seus pesos.

As redes SOM são também chamadas de Mapas de Kohonen, em referência ao seu criador Teuvo Kohonen (KOHONEN, 1982). Sua arquitetura é constituída por apenas uma camada de entrada e uma de saída, sendo que esta última representa o mapa de neurônios (Figura 3.4). A ideia do Mapa de Kohonen remete ao funcionamento do cérebro humano, onde cada região é responsável por sentidos específicos do corpo humano. Da mesma forma, os Mapas Auto-Organizáveis são caracterizados pela formação de um mapa topográfico de padrões de entrada, em que a localização espacial do neurônio na grade corresponde às características intrínsecas dos padrões de entrada, justificando, assim, o nome do método (HAYKIN, 1994).

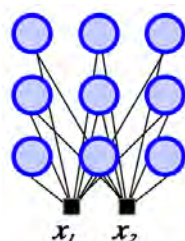


Figura 3.4 - Exemplo de arquitetura de uma rede SOM, com mapa de saída bidimensional de dimensão 3×3 neurônios e camada de entrada com dois neurônios (x_1 e x_2).

Segundo Haykin (1994), o processo de aprendizagem da rede SOM pode ser categorizada em três etapas essenciais: competitivo, cooperativo e adaptativo. O primeiro diz respeito ao processo que define o neurônio vencedor ou a determinação do neurônio que apresentou maior semelhança em relação ao vetor de entrada. No processo cooperativo, são definidos quais neurônios, além do vencedor, terão seus pesos ajustados. Uma função de vizinhança topológica é definida, de modo que satisfaça a simetria entre o ponto máximo definido pelo neurônio vencedor, sendo que a va-

riável de correção, deve decair conforme aumenta a distância do ponto central. As funções comumente utilizadas são a Chapéu Mexicano e a Gaussiana. Por fim, o processo adaptativo consiste no ajuste dos pesos em relação as entradas apresentadas. Estes três processos compõem o princípio básico do treinamento da rede SOM. Neste trabalho, adotou-se o processo cooperativo de aprendizagem, cuja função Gaussiana é utilizada para a determinação dos neurônios vizinhos e posterior ajuste.

A eficiência dos resultados na etapa de sementeação reflete diretamente no resultado do método *Snakes*. Sendo assim, o processo de treinamento da RNA é considerado o mais importante nesta etapa, pois a partir dos valores obtidos aqui são gerados os pontos-sementes ao final da etapa. Os pontos devem ser marcados, pelo menos, próximos às feições que devem ser extraídas. No treinamento da RNA, são agregadas todas as formas que a rede deverá reconhecer, conforme as definições de estradas apresentadas na Seção 3.2. Em cada padrão de treinamento são agregadas as características específicas das estradas a serem reconhecidas, tais como largura, comprimento e níveis de intensidade.

Na definição dos padrões de treinamento todas as informações dos objetos a serem identificados devem ser levados em consideração, tais como as características de largura, orientação da estrada, níveis de intensidade e comprimentos. Essas definições são importantes justamente pelas estradas aparecerem em diferentes formas na imagem. Levando em conta que o reconhecimento é aplicado sobre uma imagem de radar, as características que diferem uma estrada de um caminho devem ser levadas em conta, uma vez que tanto estradas como caminhos podem apresentar tonalidades claras ou escuras. Sendo assim, neste trabalho, adotou-se o total de oito padrões de treinamento; sendo quatro deles tipos específicos de estradas, em que são consideradas duas larguras a serem reconhecidas: três *pixels* e quinze *pixels*, além de outros padrões, idênticos aos anteriores, porém em níveis de intensidade opostos, como forma de representação das estradas com tonalidades escuras. Os quatro últimos padrões representam o que não é estrada na imagem, ou seja, padrões nulos, utilizados para a classificação das amostras inválidas no reconhecimento (processo detalhado na Seção 3.3.1.3).

Como mencionado anteriormente, as RNAs do tipo Kohonen geralmente possuem somente uma camada de entrada e uma camada de saída, esta última podendo ser unidimensional, bidimensional ou até mesmo tridimensional. Neste trabalho, a camada de saída foi definida como um mapa bidimensional, onde cada elemento

do mapa é conectado com os elementos de entrada X_p , em que p é o índice do padrão de treinamento. Cada uma dessas conexões são representadas por um valor de pesagem, de forma que, cada elemento do padrão de treinamento X_p possua sua matriz de pesos W correspondente. Desta forma, W é uma matriz tridimensional W_e , tal que a dimensão de W_e é determinada pelo usuário. Na Figura 3.5 são ilustrados a arquitetura da rede SOM e suas respectivas variáveis.

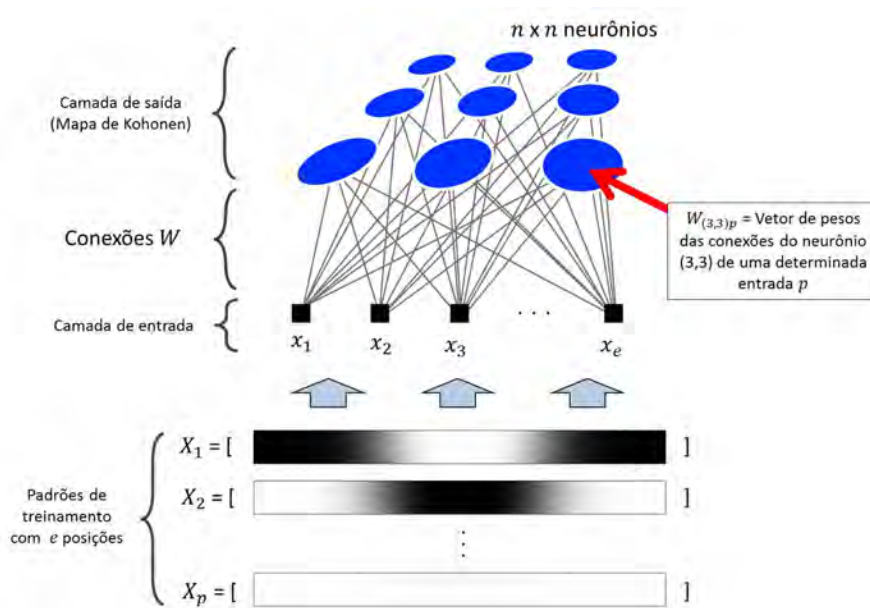


Figura 3.5 - Arquitetura da rede SOM empregada no trabalho, em que e é igual a 21, n igual a 6 e p igual a 8.

Após carregar os padrões de treinamento, a matriz de pesos W é então iniciada. Geralmente, os valores de conexões W são iniciados aleatoriamente. É definido um valor máximo de iterações, em que, a cada ciclo, os valores de W são ajustados para que o mapa não seja formatado com uma tendência inicial de se organizar. Esse processo de ajuste é denominado treinamento, em que, a cada ciclo de iteração, é obtida a localização do neurônio de saída com menor distância euclidiana em relação à entrada apresentada. O cálculo dessa distância é dado pela equação:

$$i(X) = \underset{(j,k)}{\operatorname{arg\,min}} \|W_{(j,k)} - X_p\|, \quad \begin{cases} j = 1, 2, \dots, n \\ k = 1, 2, \dots, n \end{cases} \quad (3.7)$$

em que $\|\cdot\|$ denota a norma euclidiana entre os valores de entrada X_p e $W_{(j,k)}$, correspondentes aos pesos de cada neurônio (j, k) . $i(X)$ descreve a posição do neurônio vencedor no mapa, dado por $arg_{(j,k)}$, e n a raiz quadrada do número de neurônios na camada de saída (HAYKIN, 1994). Após obter a localização do neurônio vencedor, ou seja, o neurônio que entre todos mais se assemelhou à entrada apresentada, são realizadas as correções da matriz de pesos $W_{(j,k)}$. Nesta etapa são corrigidos os valores do neurônio vencedor e seus vizinhos.

Por meio da Equação 3.8 são realizadas as correções dos pesos, em que $\eta(t)$ é a taxa de aprendizagem da rede, a qual decai a cada iteração (t) . A variável σ denota a influência do neurônio (j, k) analisado, em relação ao vencedor, obedecendo um determinado raio de vizinhança. Na Figura 3.6(a) é ilustrado o mapa de neurônios, na qual o círculo envolvendo os neurônios vizinhos ao neurônio vencedor, representa a vizinhança pela qual ocorrerão os ajustes. Da mesma forma, na Figura 3.6(b) são demonstrados como seriam os valores de σ para os neurônios vizinhos, determinando a magnitude da correção dos neurônios vizinhos; quanto maior o valor, maior a proximidade. Se o neurônio não estiver próximo o suficiente do neurônio vencedor, sua influência será nula, e conseqüentemente, não será aplicado o ajuste.

$$W(t+1)_{(j,k)} = W(t)_{(j,k)} + \eta(t) * \sigma_{(j,k)} * [X_p - W(t)_{(j,k)}] . \quad (3.8)$$

Os valores de σ para um determinado neurônio (j, k) é dado pela propriedade gaussiana:

$$\sigma_{(j,k)} = e^{-\left(\frac{dist}{2*raio^2}\right)} , \quad (3.9)$$

em que $dist$ representa a distância euclidiana entre o neurônio vencedor e o neurônio em questão, e $raio$, o raio de vizinhança.

A taxa de aprendizagem decai a cada ciclo do treinamento. A função que descreve o seu decaimento é dada por:

$$\eta(t) = \eta_0 [e^{-\left(\frac{t}{T}\right)}] , \quad (3.10)$$

em que η_0 é a taxa de aprendizagem inicial, e T é o número máximo de iterações.

O processo de treinamento se encerra quando são concluídas todas as iterações T .

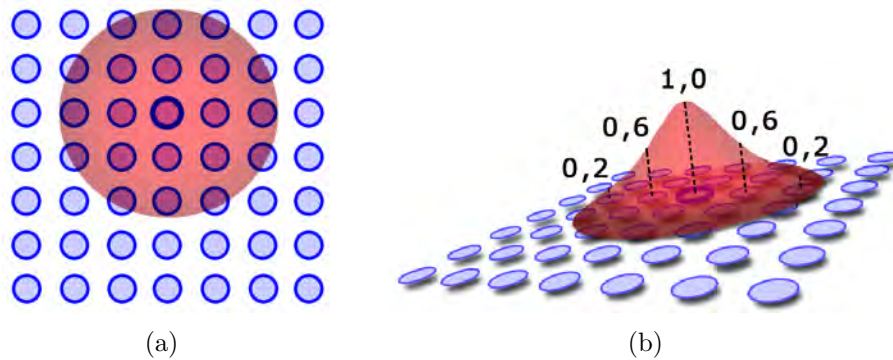


Figura 3.6 - Rede SOM: (a) Raio de vizinhança de um neurônio vencedor. (b) Valores de σ dos neurônios vizinhos.
 Fonte: Adaptado de Haykin (1994).

Todo o processo de ajuste dos pesos de conexões faz com que o mapa de Kohonen crie regiões ou aglomerados (*clusters*), sensíveis a padrões específicos, por meio das quais o treinamento foi realizado. Isto significa que o centro de cada uma dessas regiões é sensível a um tipo específico de informação, que diz respeito ao padrão exato que foi apresentado. Sendo assim, seu entorno representa as formas que se assemelham a esse padrão.

Como exemplo, considere o círculo tracejado na Figura 3.7 como sendo o aglomerado que corresponde a um padrão de treinamento de perfil claro e entorno escuro, e largura aproximada de 6 *pixels*. Ao apresentar à rede SOM amostras cujo valores se assemelhem com o respectivo padrão de treinamento, o neurônio vencedor tende a se localizar dentro desse círculo, quanto maior a semelhança, maior a proximidade do vencedor em relação ao centro do aglomerado.

No domínio de Kohonen podem existir áreas pouco desenvolvidas no processo de treinamento, principalmente quando a dimensão do mapa for relativamente grande. Neste caso, não há garantias de que o vencedor irá se localizar dentro de um determinado aglomerado. Desta forma, é aconselhável que a dimensão do mapa de Kohonen seja definida como a menor possível, evitando a geração de regiões de incertezas.

O mapa com as regiões é dado pelos valores da matriz W , de forma que seus valores são unificados, caracterizando-se na Matriz Unificada ou Matriz U , representada por W_{norm} . O cálculo para a obtenção de W_{norm} é dado pela norma euclidiana de cada um dos pontos $W_{(j,k)}$ em relação a cada elemento de entrada e de um padrão

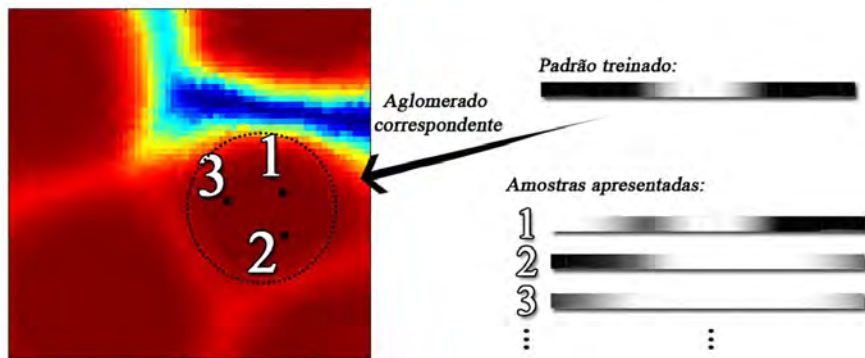


Figura 3.7 - Processo de classificação das estradas.

de treinamento p . A Equação 3.11 expressa o cálculo de W_norm , em que E é o número total de elementos de um determinado padrão de treinamento.

$$W_norm = \sqrt{\sum_{e=1}^E (W_e)^2} . \quad (3.11)$$

A Figura 3.8 ilustra esse processo, no qual obtém-se ao final o mapa da norma dos pesos (Matriz U).

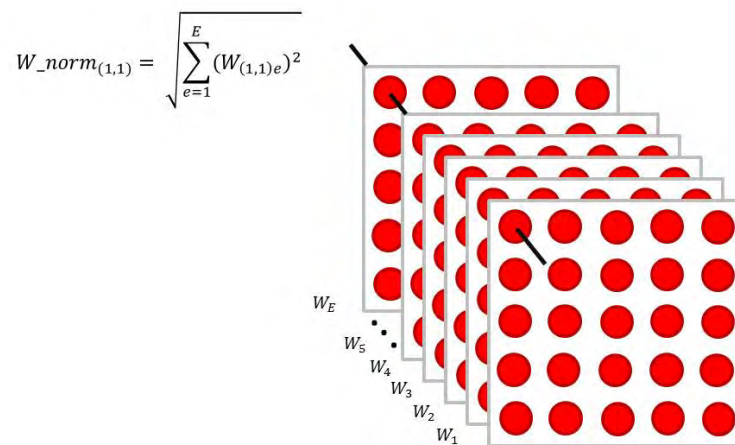


Figura 3.8 - Pesos utilizados no cálculo da norma para a montagem da Matriz U.

Após o treinamento da rede SOM, o usuário seleciona a área de estudo a serem

extraídas as estradas. Nessa área, é realizado o processo de leitura das amostras, processo detalhado na próxima seção. Nesse etapa, o usuário pré-determina o tamanho do salto da janela de leitura, sua dimensão, e a quantidade de perfis a serem lidos em cada uma das amostras.

3.3.1.2 Leitura das Amostras

O processo de leitura das amostras consiste em varrer a imagem em saltos pré-estabelecidos, de forma que, a cada salto uma janela de dimensão fixa especifica o conjunto de *pixels* que correspondem à respectiva amostra. A dimensão da janela é fixada de acordo com os perfis de treinamento apresentados anteriormente à rede SOM. Neste caso, os perfis de treinamento possuem 21 elementos, consequentemente a dimensão da janela é de 21×21 *pixels*. A cada salto, são lidos os valores de diferentes orientações, como ilustrado na Figura 3.9. Em cada uma dessas orientações, são obtidos os perfis (vetor de valores), que, posteriormente, são utilizados na classificação ou identificação das estradas.

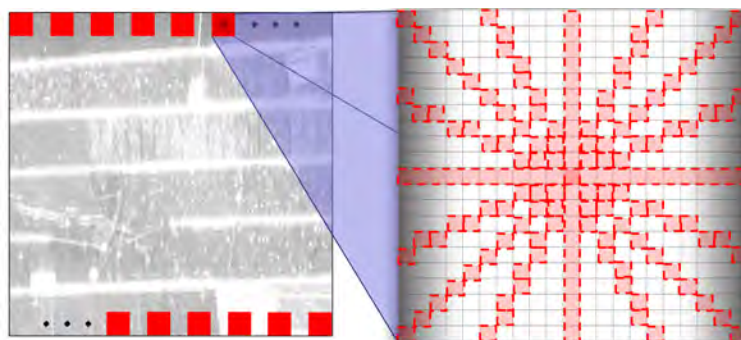


Figura 3.9 - Varredura da imagem para a leitura do conjunto f de amostras, em que cada amostra, são obtidos k perfis de estradas (Na ilustração, 8 perfis).

Para a leitura das diferentes orientações na amostra, é necessário uma heurística para a determinação dos *pixels* em determinadas inclinações de reta. O algoritmo do Ponto Médio, conhecido também por algoritmo de Bresenham (1965), determina justamente quais *pixels* são renderizados, a fim de formar uma aproximação de uma linha reta entre dois pontos.

O algoritmo desenvolvido em 1965 por Jack E. Bresenham (BRESENHAM, 1965),

consiste em um método simples para a obtenção de uma reta em um dispositivo matricial, tal como televisores e monitores. Dados dois pontos, (x_0, y_0) e (x_1, y_1) , o primeiro representa o par de coordenadas do ponto inicial, e o segundo, o ponto final. O algoritmo se inicia a partir da determinação da projeção da reta, verificando especificamente as diferenças entre as linhas final e inicial $(x_1 - x_0)$, e colunas $(y_1 - y_0)$, e com isso, determinar a inclinação da reta. Para melhor compreensão, na Figura 3.10, é ilustrado o resultado da execução do método, em que em cada coluna entre x_0 e x_1 há exatamente um *pixel* computado, enquanto que para cada linha entre y_0 e y_1 pode haver múltiplos *pixels*. O algoritmo de Bresenham é utilizado neste trabalho

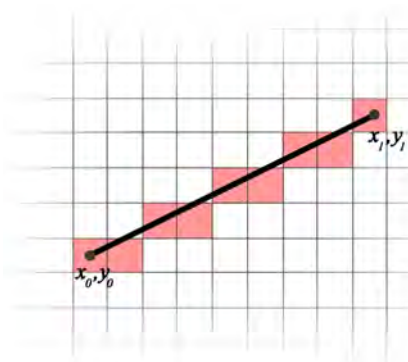


Figura 3.10 - Exemplo da determinação dos *pixels* entre dois pontos em um dispositivo matricial (Algoritmo de Bresenham).

para a leitura dos perfis em diferentes orientações (Figura 3.9), aumentando a chance de identificar a passagem de uma estrada, bem como a sua orientação na imagem.

O vetor de valores de cada uma das orientações possui, obrigatoriamente, a mesma quantidade de elementos que foram apresentados à rede. Para que a RNA aponte uma amostra como sendo uma estrada, pelo menos uma das orientações esteja próxima o suficiente de algum centro que corresponde a uma estrada. Isto é, pelo menos uma orientação deve estar no raio de algum dos aglomerados que representam o padrão de estrada. Se pelo menos uma das orientações satisfizer essa condição, a amostra é identificada como sendo uma estrada. O cálculo para a obtenção da

orientação com menor valor é representado por:

$$c(X) = \underset{orient}{arg\ min} \|W_{(j,k)} - X_{orient}\|, \quad \begin{cases} orient & = 1, \dots, l \\ j & = 1, 2, \dots, n \\ k & = 1, 2, \dots, n \end{cases} \quad (3.12)$$

em que X representa o conjunto dos valores para cada orientação, $orient$ é o índice da orientação, e $c(X)$ é o índice do vetor correspondente à orientação que mais se aproximou de um padrão de estrada, e portanto, a orientação escolhida para a determinação do ponto semente no processo de classificação.

A seção seguinte apresenta o método de decisão aplicado no reconhecimento das estradas, utilizando, essencialmente, a matriz de pesos obtida na etapa de treinamento, ou seja, as amostras lidas nessa etapa são apresentadas à RNA já treinada, que, por sua vez, determina ou não a presença de estradas em seu conteúdo.

3.3.1.3 Classificação e Identificação

Considerando que o treinamento da RNA tenha sido realizado adequadamente e as amostras da imagem lidas, o processo de reconhecimento ou identificação das estradas pode então ser realizado. Cada amostra é apresentada à RNA treinada, para que esta faça o reconhecimento daquelas estradas que apresentarem padrões semelhantes aos utilizados no treinamento. Como exposto na seção anterior, ao apresentar uma determinada amostra para reconhecimento, seus valores são associados a uma posição no mapa de neurônios, o que caracteriza o neurônio vencedor para esse vetor de dados.

No treinamento da rede SOM neste trabalho, foram utilizados oito padrões de treinamento. Os dois primeiros padrões correspondem ao perfil de estrada larga (15 *pixels*), sendo que o primeiro representa a estrada clara e entorno escuro, e o segundo, o inverso. Os dois padrões seguintes correspondem a um perfil mais estreito de estrada (3 *pixels*) e, assim como no caso dos dois primeiros padrões, também são definidos com intensidades inversas. Por fim, os últimos quatro padrões representam alvos que não correspondem a estradas.

A determinação de uma amostra possuir ou não a passagem de uma estrada é dada pela apresentação de $c(X)$ (Equação 3.12) à rede SOM, em que é adquirido um

neurônio vencedor para a respectiva entrada. Caso esta esteja presente em um dos quatro aglomerados que dizem respeito ao padrão estrada, a amostra é considerada como ponto-semente, caso contrário, ignorada. Os índices das amostras que forem selecionadas pelo método de sementeção são armazenados. Tais índices são utilizados posteriormente em um processo de fragmentação, em que são selecionados os pontos-sementes pertencentes a um mesmo segmento de reta. Ao final do processo, é obtido uma matriz, tal que cada linha dessa corresponde aos índices das amostras pertencentes a um mesmo segmento de reta. A Figura 3.11 ilustra as amostras identificadas como estrada no processo de classificação. O vetor não fragmentado

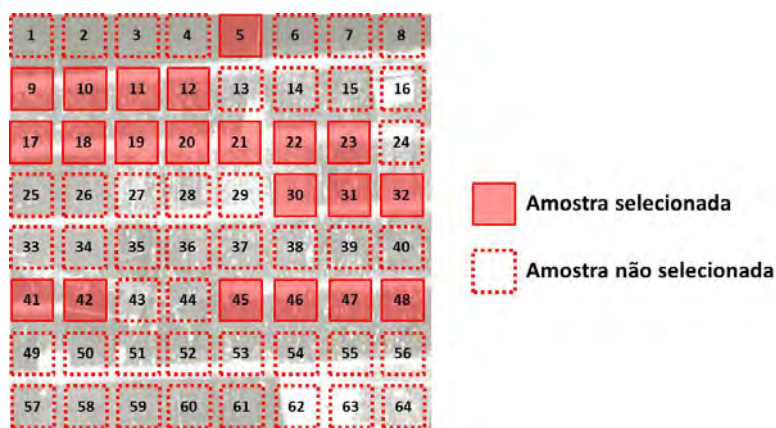


Figura 3.11 - Exemplo do conjunto de amostras selecionadas pelo método de sementeção.

armazena sequencialmente os índices de cada amostra. Ao fragmentar este vetor, os índices ficam dispostos em uma matriz, de forma que cada linha caracteriza um conjunto de pontos-sementes pertencentes a um mesmo segmento de reta. Desta forma, o vetor fragmentado correspondente a Figura 3.11 é dado por:

$$pontos = \begin{bmatrix} 9 & 10 & 11 & 12 & 0 & 0 & 0 \\ 17 & 18 & 19 & 20 & 21 & 22 & 23 \\ 45 & 46 & 47 & 48 & 0 & 0 & 0 \end{bmatrix}$$

Na Seção 3.3.2 são apresentados os detalhes do processo de fragmentação e poda dos pontos-sementes.

3.3.2 Processo de poda dos pontos-sementes

Muitos dos pontos-sementes são marcados incorretamente, grande parte, em função do ruído presente na imagem. Como estratégia para a eliminação destes pontos-sementes incorretos, foi desenvolvido um algoritmo de poda com o objetivo de separar todos os pontos que formam segmentos retos e contínuos, obedecendo algumas regras. Portanto, um conjunto de pontos-sementes é considerado como pertencente a um segmento de reta se as seguintes inequações se verificam:

- $(f + 1) - (f) \leq \text{LMD}$.
- $\text{QPC} \geq \text{LMC}$.

Em que f representa o índice das amostras identificadas, representando o ponto-semente propriamente dito. E os limiares LMC, LMD e QPC representam respectivamente:

- **Limite Mínimo de Continuidade:** Limiar que define a quantidade mínima de pontos-sementes subsequentes para que sejam considerados como pertencentes a um mesmo segmento de reta.
- **Limite Máximo de Descontinuidade:** Limiar que define a quantidade máxima de descontinuidade entre os pontos-sementes para que sejam considerados como pertencentes a um mesmo segmento de reta.
- **Quantidade de Pontos Contínuos:** Índice com o qual se verifica a continuidade dos pontos-sementes, caso este valor seja menor que LMC, a sequência é ignorada.

A Figura 3.12 ilustra um conjunto de pontos-sementes sobre uma estrada. Os pontos tracejados correspondem aos pontos-sementes não identificados na classificação, gerando as descontinuidades. Considerando o fator LMC igual a 4 e LMD igual a 2, somente os pontos: 6, 7, 8, 10 e 11 pertencem a um mesmo e único segmento de reta, pois entre os pontos 8 e 10 há somente um ponto de descontinuidade, o que satisfaz a primeira condição, e o total de pontos-sementes é igual a 5 (QPC), o que satisfaz a segunda condição. Portanto, o conjunto de pontos-sementes é considerado como sendo um segmento de reta presente na imagem. Os pontos-sementes 1 e 2 satisfazem a primeira inequação, mas não a segunda, excluindo-se da classificação.

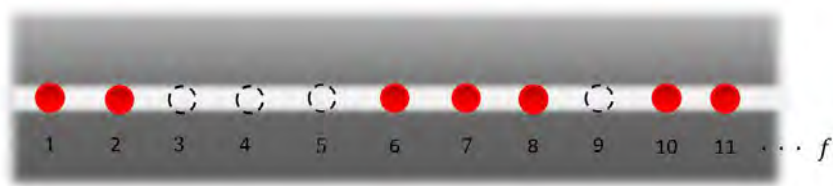


Figura 3.12 - Descontinuidades da classificação dos pontos-sementes.

Após este processo, tanto os pontos isolados apontados como uma estrada pelo método quanto os segmentos de reta pequenos, são excluídos da identificação.

Nesta etapa do processo, os pontos-sementes estão separados em grupos de segmentos de reta, dispostos em uma matriz. Nesta matriz, cada linha caracteriza um segmento de reta, e cada coluna dessas linhas, o índice da amostra (Figura 3.12). Assim, para cada linha é aplicado o método *Snakes*, de forma que todos os elementos da linha são interpolados e aplicados ao método. Ao término das linhas, é obtido então a rede de estradas.

A sementeação realizada neste trabalho foi aplicada sobre a imagem de radar sem pré-processamento, justamente para que se analisasse o comportamento do método com a presença do ruído (imagem original), apresentada no Capítulo 4.

Nesta seção 3.3.1, foi apresentado como o processo de sementeação é realizado, sua fundamentação e importância para o método de extração de estradas. Na próxima seção, serão apresentados detalhes do processo de obtenção do eixo de simetria das estradas pelo método *Snakes*.

3.3.3 Modelo de Contorno Ativo (*Snakes*)

Embora a técnica tenha sido empregada no final da década de 1980 por Kass et al. (1987), o termo Modelos Deformáveis surgiu em 1998 em um trabalho apresentado por Singh et al. (1998). A técnica consiste em obter formas (ou contornos) dos objetos presentes em uma imagem digital, permitindo que um contorno inicial se deforme e evolua para as bordas do objeto. Evolução esta que pára sob determinadas condições de conversão, pré-estabelecidas no modelo, e assim, sejam identificados os objetos presentes na imagem.

Os modelos deformáveis são, em geral, considerados robustos pela sua capacidade em

adaptar-se e identificar os objetos em cena com eficiência representando, portanto, uma contribuição considerável para a Visão Computacional. Após o surgimento da técnica, foram realizados diversos trabalhos que exploraram a maneira como os objetos eram reconhecidos (COHEN, 1991; OSHER; SETHIAN, 1988; COOTES et al., 1999; STAIB; DUNCAN, 1992). Basicamente, as soluções apresentadas estendiam a maneira como a curva evoluía em direção ao objeto, e essa evolução pode ser traduzida como a minimização do funcional de energia da curva. Muitas soluções se diferenciaram do método variacional de Euler-Lagrange, apresentado por Kass et al. (1987) para a minimização da energia, e apresentaram técnicas igualmente boas, como, por exemplo, a utilização da Programação Dinâmica, apresentada por Amini et al. (1990), para a extração de estradas utilizando *Snakes*, ou o método de Fluxo do Vetor Gradiente (*Gradient Vector Flow - GVF*) (XU; PRINCE, 1997), que, embora utilize o mesmo termo variacional, apresenta uma melhora considerável nos resultados quando as curvas iniciais são apresentadas distantes das feições a serem extraídas, o que é uma desvantagem no modelo tradicional.

A fim de que a curva venha a convergir para o contorno do objeto de maneira eficaz, é necessário que sua localização se encontre próxima à feição de interesse. Por este fato, na maioria das vezes, o método compreende um pré-processamento manual, no qual um operador humano fornece pontos que se localizam próximos à feição a ser extraída. Esses pontos são então interpolados, obtendo-se uma curva inicial. Após a obtenção da curva inicial, essa é submetida a um processo iterativo para o cálculo de sua nova localização. Ao longo dessas iterações, a curva se comporta de maneira semelhante a uma cobra (*snake*), o que explica a razão do nome (KASS et al., 1987).

Ao longo dos tempos, foram criadas diferentes variações do método *Snakes*. Silva et al. (2004) relacionam as principais delas, tais como o método *Level Set*, que consiste em um modelo não-paramétrico, cujo processo de identificar o contorno dos objetos não necessita de parametrização da curva (OSHER; SETHIAN, 1988). Outra extensão frequentemente abordada em trabalhos de reconhecimento de formas é o método *Balloon Snake* (COHEN, 1991), no qual a curva evolui a partir do centro do objeto de interesse, imitando o comportamento de um balão de ar, ao contrário do que acontece no método convencional, onde a curva inicial é formada ao redor do objeto.

Na Figura 3.3 é ilustrada a metodologia usada na extração de estradas em imagens digitais. A partir de um conjunto de pontos provenientes do processo de Semeação, os pontos-ementes, e a fragmentação desses em segmentos de reta, aplica-se o método

Snakes para que seja realizada a extração da estrada, de forma que os pontos de um determinado segmento de reta são interpolados, obtendo-se, então, uma curva inicial ζ , tal como exemplificado na Figura 3.2.

Na seção seguinte, é apresentada a fundamentação matemática da Teoria de *Snakes*. As definições e equações apresentadas a seguir, são baseadas na teoria tradicional do método, apresentada por Kass et al. (1987). Todas as adaptações realizadas para a extração de estradas, são citadas juntamente com suas definições.

3.3.3.1 Energia da *Snake*

Conforme exposto anteriormente, o problema do método *Snakes* se resume ao processo de minimização de um funcional de energia. O modelo pode ser considerado como sendo uma *spline* cúbica, controlada sobre a influência de suas energias interna e externa.

A curva ζ pode ser representada parametricamente por $v(s) = (x(s), y(s))$, em que s é relacionado ao comprimento de arco, com valores crescentes ao longo da curva, $s \in [0, 1]$, que se move no domínio espacial da imagem. O contorno ativo funciona pela minimização da energia total da curva, E_{snake} , que é composta pela soma das energias interna, E_{int} , da imagem, E_{image} e a energia E_{con} , imposta pelo usuário (Equação 3.13).

$$\begin{aligned} E_{snake} &= \int_0^1 E_{snake}(v(s)) ds \\ &= \int_0^1 [E_{int}(v(s)) + E_{ext}(v(s))] ds \\ &= \int_0^1 [E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))] ds . \end{aligned} \quad (3.13)$$

Cada energia possui uma responsabilidade específica no método e, sendo assim, as seções que se seguem destinam-se à apresentação das funções e definições de cada uma delas, exemplificando sua importância em cada etapa da iteração.

3.3.3.2 Energia Interna

A força interna é responsável por manter a relação de suavidade da curva, enquanto que a energia externa a impulsiona na direção de linhas, bordas e contornos subje-

tivos (KASS et al., 1987). A energia interna pode ser descrita por:

$$E_{int} = \frac{1}{2} \int_0^1 \alpha(s) \left| \frac{\partial v(s)}{\partial s} \right|^2 + \beta(s) \left| \frac{\partial^2 v(s)}{\partial s^2} \right|^2 ds, \quad (3.14)$$

a qual é composta por dois termos: o termo de primeira ordem $\partial v(s)/\partial s$, controlado por $\alpha(s)$; e de segunda ordem $\partial^2 v(s)/\partial s^2$, controlado por $\beta(s)$. O termo de primeira ordem faz com que a curva se comporte como uma membrana, representando sua elasticidade. O segundo termo diz respeito à rigidez, agindo como uma placa fina, permitindo que, durante o processo de minimização da energia, ela não se dobre (KASS et al., 1987). As variáveis α e β podem ser consideradas como constantes, que funcionam como um fator de ponderação, possibilitando o ajuste de acordo com as características da forma a ser identificada. Seus valores são essenciais para a eficiência do método, ao passo que variações pequenas podem mudar significativamente o resultado (BENTABET et al., 2003). Agouris et al. (2001) mencionam que a escolha dos parâmetros α e β pode ser realizada empiricamente, definindo-os de acordo com as características da imagem ou feições a serem extraídas. Bentabet et al. (2003) utilizaram valores de β estimados automaticamente, em que inicialmente são verificadas as características predominantes das feições por meio de uma base de dados com a verdade terrestre das mesmas, e então, obtém-se o valor médio da curvatura dessas feições. A partir dessa informação, estima-se o valor de β .

As funções $\partial v(s)/\partial s$ e $\partial^2 v(s)/\partial s^2$, que compõem cada um dos termos, são as derivadas de primeira e segunda ordem de $v(s)$, respectivamente. A primeira fornece o coeficiente angular tangente à curva no ponto s , e a segunda, o vetor normal a este ponto. A Equação 3.14 não admite qualquer solução analítica, fazendo com que seja requerido um processo iterativo para resolvê-la. O mais comum dos processos é o método de Euler-Lagrange, em que a Equação 3.14 é reescrita na equação de Euler, correspondendo assim a uma equação de equilíbrio ou de minimização (BENTABET et al., 2003). O processo de minimização é apresentado na Seção 3.3.3.4.

3.3.3.3 Energia externa

A soma das energias E_{image} e E_{con} é denominada energia externa E_{ext} . A energia da imagem, E_{image} , é composta pela soma do produto de três outras energias.

$$E_{image} = w_{line} E_{line} + w_{edge} E_{edge} + w_{term} E_{term}. \quad (3.15)$$

Os valores de w_{line} , w_{edge} e w_{term} são constantes que funcionam como ajustes para suas respectivas energias. E_{line} especifica as características do contorno quanto aos níveis de intensidade da forma, ou seja, dependendo do valor de w_{line} , a curva será atraída por linhas claras ou escuras. Tal energia refere-se simplesmente aos níveis de intensidade da imagem. Na abordagem original apresentada por Kass et al. (1987), são apresentadas duas representações do funcional de linha. Na Equação 3.16, a primeira apresentada pelos autores, a curva é atraída por contornos com gradientes altos de imagem, em que a curva tende a convergir na borda do objeto.

$$E_{line} = I(x, y) . \quad (3.16)$$

A segunda representação é dada por:

$$E_{line} = -(G_{\sigma} * \nabla^2 I)^2 , \quad (3.17)$$

em que G_{σ} representa uma Gaussiana de desvio padrão σ , e $\nabla^2 I$ o gradiente de segunda ordem da imagem. Ambas permitem adquirir o contorno do objeto sendo que, a segunda representação se difere pelo comportamento da curva em ser atraída às regiões de passagem-por-zero (*zero-crossing*), isto é, regiões com mudanças bruscas de intensidade.

O funcional de borda, E_{edge} , pode ser interpretado também como um operador para a identificação de bordas na imagem, e é dado por:

$$E_{edge} = -|\nabla I|^2 , \quad (3.18)$$

em que ∇I é o gradiente da imagem I e, E_{term} é definido por Kass et al. (1987) como o funcional de limite que, tem como responsabilidade de encontrar terminações e cantos dos segmentos de reta. Para tanto, utiliza-se o nível de curvatura das linhas da imagem após suavização, dada por um filtro gaussiano:

$$C(x, y) = G_{\sigma}(x, y) * I(x, y) , \quad (3.19)$$

O nível de curvatura pode ser obtido pelo cálculo da variação do ângulo gradiente

em relação ao vetor perpendicular ao gradiente:

$$E_{term} = \frac{\partial \theta}{\partial n_{\perp}} \quad (3.20)$$

$$= \frac{\partial^2 C / \partial n_{\perp}^2}{\partial C / \partial n} \quad (3.21)$$

$$= \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{\frac{3}{2}}}, \quad (3.22)$$

em que $\theta = \tan^{-1}(C_y/C_x)$ é o ângulo do gradiente, $n = (\cos\theta, \sin\theta)$ e $n_{\perp} = (-\sin\theta, \cos\theta)$ são vetores unitários na direção do gradiente e perpendicular a este, respectivamente. A Equação 3.22 representa a solução discreta do cálculo dos níveis de curvatura das linhas na imagem, que em termos práticos, C_x , C_y e C_{xy} representam as diferenças discretas avançadas de primeira ordem da imagem suavizada $C(x, y)$ em relação aos respectivos eixos x e y , que caracterizam-se em uma imagem com realces de borda do tipo degrau com orientação em x , do tipo degrau com orientação em y e do tipo degrau com orientação nos dois eixos, respectivamente. Os termos C_{xx} e C_{yy} correspondem às diferenças discretas avançadas de segunda ordem, obtendo uma imagem com realces de borda do tipo cartola (*top-hat*) com orientação em x e y , respectivamente (HOFFMAN, 2001). A combinação do funcional de terminação E_{term} , com o funcional de borda E_{edge} , faz com que a curva evolua não só para os contornos do objeto, como também para regiões onde não há contornos reais, mas sim fictícios. Na Figura 3.13, é ilustrado a evolução da curva para as bordas do objeto, que, apesar de não existirem de fato, são reconhecidas pelo método.

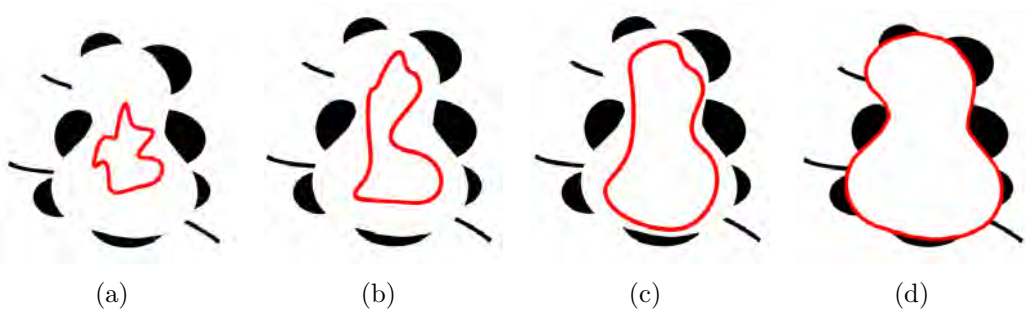


Figura 3.13 - Exemplo da obtenção de contornos criados pela *Snakes*: (a) Imagem original. (b) Curva inicial. (c) Evolução intermediária da curva. (d) Curva final.
Fonte: Adaptado de Kass et al. (1987).

Por fim, o parâmetro E_{con} , na Equação 3.13, é definido como sendo uma energia de restrição imposta pelo usuário. Muitas vezes, esse termo não é utilizado no processo de extração de estradas, como apresentado em Bentabet et al. (2003), Agouris et al. (2001) e Youn e Bethel (2004). Assim como a energia de restrição E_{con} , outros trabalhos negligenciam o uso das energias E_{edge} e E_{term} justamente por não serem úteis, face às características do objeto a ser extraído. A aplicação de *Snakes* apresentada neste trabalho dispensa o uso dos termos E_{con} , E_{term} e E_{edge} . O último termo é desprezado, uma vez que o objetivo é obter o centro das vias e não suas bordas. Portanto, a Equação 3.13 é reescrita:

$$\begin{aligned} E_{snake} &= E_{int} + E_{image} \\ &= E_{int} + w_{line}E_{line} \end{aligned} \tag{3.23}$$

O processo de obtenção dos eixos de simetria das vias é definido como sendo um procedimento de minimização da energia total da curva (Equação 3.13). Existem diferentes maneiras de se minimizar tal energia. Na seção seguinte, serão apresentadas as principais técnicas utilizadas para a resolução deste problema.

3.3.3.4 Minimização da Energia Total

Nas seções anteriores, foram apresentadas as características gerais da curva, entre as quais a representação no domínio espacial, definida como uma curva parametrizada por um conjunto de pontos no plano (x, y) da imagem, denotada por $v(s)$. De forma que a obtenção do contorno de um determinado objeto é dado pela minimização de $v(s)$, que corresponde à busca do mínimo local da soma de suas energias, representada pela Equação 3.3.3.1, a qual não admite solução analítica, o que equivale dizer que, para o problema da minimização da curva, a equação anterior deve ser reescrita de forma que num esquema iterativo possa-se encontrar soluções para ela.

Segundo Bentabet et al. (2003), métodos como Algoritmos Gananciosos (*Greeding Algorithm*) e Otimização por Arrefecimento Simulado (*Simulated Annealing*) são tipos requeridos para a solução do problema de minimização. Esses algoritmos são utilizados na busca de mínimos globais que representam pontos de estabilidades ou de equilíbrio em um determinado problema de otimização, tal como a minimização do funcional de energia, E_{snake} . A Programação Dinâmica (BELLMAN; DREYFUS, 1962) é outra alternativa para a obtenção do mínimo de E_{snake} (AMINI et al., 1990;

GRUEN; LI, 1997; LI, 1997; POZ et al., 2006).

O método mais comum de minimização do funcional de energia, é reescrito como uma equação de *Euler-Lagrange*, correspondendo à uma função de equilíbrio (KASS et al., 1987; BENTABET et al., 2003; LAPTEV et al., 2000; HAUPTFLEISCH, 2010), tal que a minimização do funcional de energia na Equação 3.14 dê origem a duas novas equações de Euler independentes.

$$\alpha(s) \frac{\partial^2 x(s)}{\partial s^2} + \beta(s) \frac{\partial^4 x(s)}{\partial s^4} + \frac{\partial E_{ext}}{\partial x} = 0 , \quad (3.24)$$

$$\alpha(s) \frac{\partial^2 y(s)}{\partial s^2} + \beta(s) \frac{\partial^4 y(s)}{\partial s^4} + \frac{\partial E_{ext}}{\partial y} = 0 , \quad (3.25)$$

Para Laptev et al. (2000), os equilíbrios entre a energia interna e a energia da imagem são importantes no processo de minimização. Se uma delas tiver domínio sobre a outra, anulará a relação de suavidade da curva ou a forçará a ignorar a energia da imagem, sendo que ambos efeitos são indesejados. Desde que a energia interna possa ser controlada, o equilíbrio pode ser alcançado com o ajuste preciso de $\alpha(s)$ e $\beta(s)$. Sendo assim, Laptev et al. (2000) utilizaram a idéia apresentada por Fua e Leclerc (1990), os quais criaram uma maneira de automatizar a definição dos valores de $\alpha(s)$ e $\beta(s)$, em que se substituem as funções por uma constante λ .

Resolver as Equações 3.24 e 3.25 computacionalmente implica dizer que a evolução da curva pode ser realizada somente em período discreto de tempo. Assim, é necessária a discretização da representação da *snake* a fim de aproximar sua posição e derivadas (LAPTEV et al., 2000). Uma abordagem que oferece a discretização precisa de derivadas é dada pelo método de diferenças finitas (BENTABET et al., 2003). Na resolução apresentada por Kass et al. (1987), aproximando-as pelo método de diferenças finitas, considera-se:

$$v_i = (x_i, y_i) = (x_i(s, t), y_i(s, t)) , \quad i = 1, \dots, n ; \quad (3.26)$$

em que s denota o parâmetro do comprimento de arco, t , a variável temporal e i o índice do ponto na curva. Neste caso, a expansão da energia interna do i -ésimo vértice é dada por:

$$E_{int}(i) = \frac{\alpha_i |v_i - v_{i-1}|^2}{2s^2} + \frac{\beta_i |v_{i-1} - 2v_i + v_{i+1}|^2}{2s^4} . \quad (3.27)$$

Considerando as derivadas parciais da energia externa em relação a x , $f_x(i) = \partial E_{ext}/\partial x_i$, e em relação a y , $f_y(i) = \partial E_{ext}/\partial y_i$, em cada vértice v_i , as equações de Euler Lagrange, de acordo com o cálculo variacional (COURANT; HILBERT, 1989), podem ser assim apresentadas, já considerando o método de diferenças finitas:

$$\begin{aligned} & \alpha_i(v_i - v_{i-1}) - \alpha_{i+1}(v_{i+1} - v_i) \\ & \quad + \beta_{i-1}[v_{i-2} - 2v_{i-1} + v_i] \\ & \quad - 2\beta_i[v_{i-1} - 2v_i + v_{i+1}] \\ & \quad + \beta_{i+1}[v_i - 2v_{i+1} + v_{i+2}] \\ & \quad + (f_x(i), f_y(i)) = 0 . \end{aligned} \tag{3.28}$$

A Equação 3.28 pode ser reescrita na forma matricial:

$$Ax + f_x(x, y) = 0 , \tag{3.29}$$

$$Ay + f_y(x, y) = 0 , \tag{3.30}$$

em que A é uma matriz penta-diagonal simétrica $n \times n$, dada pelos coeficientes da Equação 3.28. Devido as derivadas parciais da energia externa utilizada, são necessárias mudanças em A a cada iteração. Por simplicidade, admita-se que f_x e f_y são constantes durante um intervalo de tempo, o que origina o método explícito de Euler em relação à energia externa. A energia interna, entretanto, é especificada pela matriz A , assim, é possível validar a derivada temporal no tempo t , ao invés de $t - 1$ e, conseqüentemente, obter um passo implícito de Euler para as energias internas, resultando em:

$$Ax^{(t)} + f_x(x^{(t-1)}, y^{(t-1)}) = -\gamma(x^{(t)} - x^{(t-1)}) , \tag{3.31}$$

$$Ay^{(t)} + f_y(x^{(t-1)}, y^{(t-1)}) = -\gamma(y^{(t)} - y^{(t-1)}) , \tag{3.32}$$

em que γ é o tamanho do passo, $x^{(t)}$ e $y^{(t)}$ são vetores colunas $n \times 1$, correspondendo às localizações dos pontos v no tempo t e o termo $f(\cdot)$ denota as derivadas parciais dos pontos no tempo $t - 1$ que, por sua vez, corresponde também a um vetor coluna $n \times 1$. Isolando-se as variáveis x e y tem-se:

$$x^{(t)} = (A + \gamma I)^{-1}(\gamma x^{(t-1)} - f_x(x^{(t-1)}, y^{(t-1)})) , \tag{3.33}$$

$$y^{(t)} = (A + \gamma I)^{-1}(\gamma y^{(t-1)} - f_y(x^{(t-1)}, y^{(t-1)})) , \tag{3.34}$$

em que I é uma matriz identidade $n \times n$, assim, a matriz $A + \gamma I$ é uma matriz penta-diagonal, podendo simplesmente ser invertida pelo método de decomposição LU .

3.3.3.5 Representação Aberta da Curva *Snake*

Existem duas importantes configurações da curva a serem consideradas: as de caráter fechado e as de caráter aberto. Até então, foram apresentadas as definições para a curva fechada (laço), assim como a definição em sua base original. Tratando-se da extração de feições lineares, fica evidente que a configuração da curva necessita ser aberta. Seria impraticável utilizar a base tradicional do método, que apresenta as definições para a curva fechada, como um extrator de estradas, e ainda, obter um bom resultado. Encontram-se na literatura muitos trabalhos abordando a extração de estradas com o uso da curva aberta, porém, o grande obstáculo é identificar como esse tipo de configuração é executado em relação à reparametrização da curva.

A princípio, o método desenvolvido por Kass et al. (1987), visava à obtenção de contornos em objetos presentes em imagens ópticas. Baseando-se nessa implementação tradicional, foram identificados duas possíveis modificações no método para o problema de extração de estradas. A primeira delas foi a adequação do método para a representação da curva aberta. Os trabalhos que aplicam a curva aberta simplesmente ignoram a necessidade de eventuais alterações, para que este tipo de curva se comporte adequadamente como em sua representação fechada. Após a pesquisa sobre como executar a extração de estradas utilizando a curva aberta, surgiram as alternativas de, por um lado, se fixar os pontos extremos e aplicar o processo somente nos demais pontos da curva; e, por outro, de se agregar conceitos de forma a permitir que os extremos sejam atraídos pelos contornos, expandindo-se e possibilitando que, a curva se estique ao longo do objeto. Devido à complexidade de implementação da segunda alternativa, adotou-se a opção de se fixar os extremos de uma determinada curva e aplicar o processamento de minimização sobre os pontos internos. A Figura 3.14 ilustra o avanço da curva conforme o processo de minimização sobre esses pontos, em que a cada iteração é obtido uma nova localização somente para os pontos internos.

Em alguns casos, os pontos extremos da curva pode localizar-se distantes da região de interesse, ocasionando o efeito ilustrado na Figura 3.14(d), em que o posicionamento da aresta entre os vértices extremo e o interno mais próximo pode tornar-se

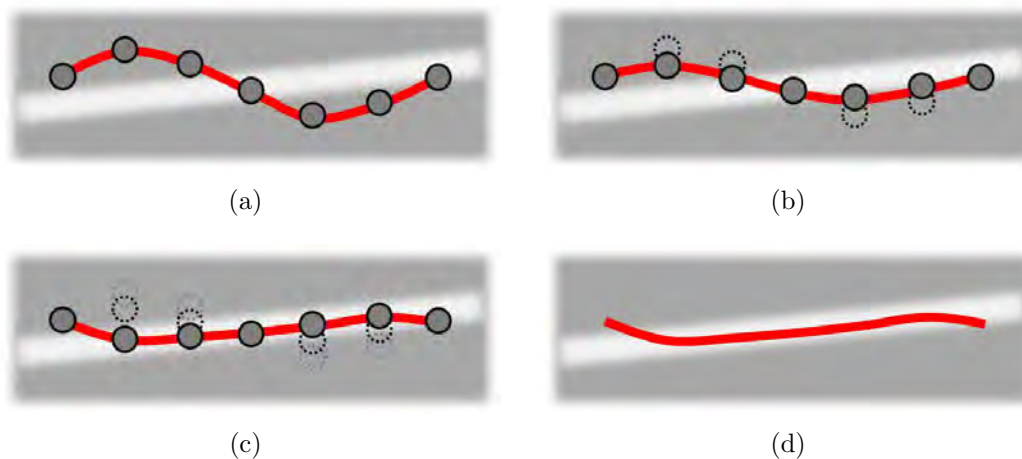


Figura 3.14 - Avanço da *snake* com os pontos extremos fixos:(a) Curva inicial. (b) e (c) Avanços intermediários da curva. (d) Curva final.

incoerente com a real posição da estrada. Desta forma, após a aplicação do método, optou-se por excluir as arestas de ligação com os extremos fixados no processo *Snakes* (Figura 3.15).

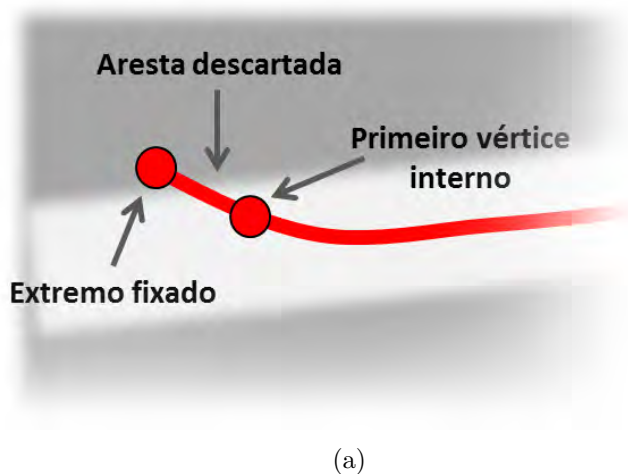


Figura 3.15 - Exclusão da aresta de ligação entre o extremo fixado e o vértice interno.

Na representação tradicional, a curva tende a ser atraída para o contorno das formas, conforme as energias E_{edge} e E_{term} . Sabendo-se que o operador deve obter os eixos de simetria, essas energias são inutilizadas, assim como o termo imposto pelo usuário

E_{con} . Desta forma, o termo de energia externa E_{ext} , composta pelos fatores E_{line} , E_{edge} , E_{term} e E_{con} , passa a ser composto somente pelo termo E_{line} , que consiste do produto entre a imagem suavizada e o fator de ponderação w_{line} , o qual dependendo do sinal, definirá a atração da curva: negativo para perfis claros, e positivo para perfis escuros. Nessa definição, a curva será atraída não para as fronteiras da estrada, mas sim para o seu centro.

Por ser uma imagem de radar, as estradas podem aparecer tanto em níveis de cinza claros como escuros, e assim, o operador de extração deve ser capaz de reconhecer as duas características de tonalidade. O termo E_{line} , especifica qual característica que a curva deverá ser atraída, ou seja, dependendo da sua sinalização, negativo ou positivo, a curva será atraída para feições escuras ou claras, respectivamente.

Na Figura 3.3.3.5, são apresentados exemplos da aplicação do operador de extração em um recorte na imagem de Paragominas-PA (ver Seção 4.2), cujas estradas aparecem em ambas representações, clara e escura. Na Figura 3.16(a) o termo de energia E_{ext} possui sinal positivo, enquanto que na Figura 3.16(b), o resultado apresentado possui o termo com sinal negativo.

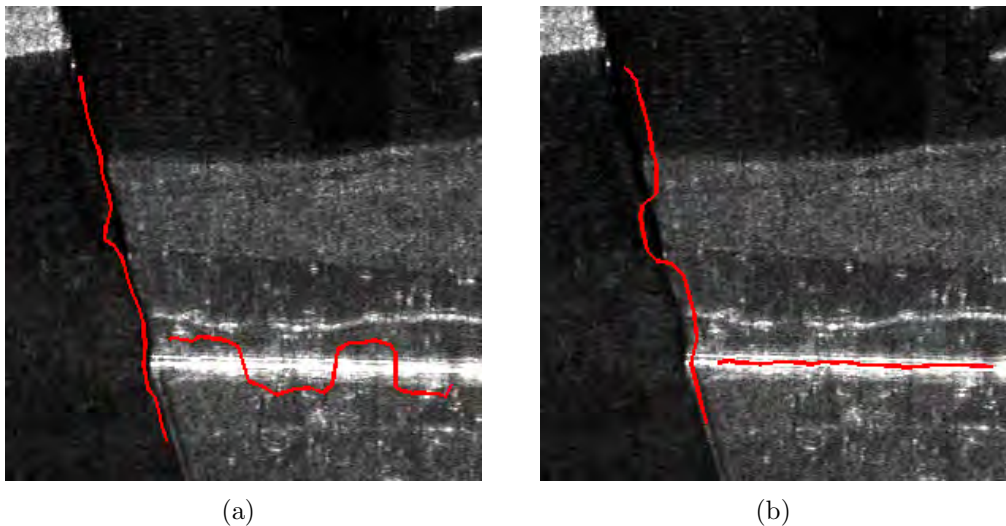


Figura 3.16 - Resultados da extração para diferentes perfis de tonalidades: (a) Resultados obtidos com o termo de energia externa positivo e (b) Resultados obtidos com o termo de energia externa negativo.

Nesta seção, foram apresentados os conceitos envolvendo a utilização do Modelo de Contorno Ativo na extração de estradas em imagens digitais. Com o objetivo de avaliar a eficiência do método, serão apresentadas na seção seguinte a métrica de validação utilizada em muitos dos trabalhos envolvendo a extração de estradas.

3.3.4 Métricas de Qualidade

É essencial que, após as etapas de processamento e aquisição dos resultados, seja medida a qualidade do método aplicado. Isto permite visualizar se realmente o método apresentado atingiu as expectativas para a resolução do problema. Geralmente, esta quantificação se baseia na comparação entre o conjunto de estradas extraídas pelo método e o conjunto-referência (verdade terrestre). A qualidade dos resultados é medida, respondendo-se a duas indagações: (i) Quão completa é a rede de estradas extraídas?; e (ii) Quão correta ela é? (WIEDEMANN et al., 1998).

As medidas de qualidade não são destinadas a avaliar a extração e os resultados correspondentes de uma forma absoluta. Em vez disso, elas são utilizadas para comparar os resultados dos diferentes algoritmos (HEIPKE et al., 1997). Harvey (1999), Wiedemann et al. (1998), Heipke et al. (1997) e Hauptfleisch (2010) aplicaram técnicas semelhantes para a análise das feições extraídas pelo método em questão. O conceito no qual se basearam é apresentado a seguir.

Essencialmente, são definidos quatro fatores que representam cada resultado possível que o extrator de estrada pode alcançar:

- **TP (*True Positive*):** O verdadeiro positivo representa as regiões de estradas extraídas pelo método que, por sua vez, pertence ao conjunto-referência.
- **TN (*True Negative*):** O verdadeiro negativo representa as regiões de não-estrada não extraídas pelo método, que, por sua vez, pertence ao conjunto-referência.
- **FP (*False Positive*):** O falso positivo representa as regiões de estradas extraídas pelo método que, por sua vez, não pertence ao conjunto-referência.
- **FN (*False Negative*):** Por fim, o falso negativo representa as regiões de estradas não extraídas pelo método e que pertence ao conjunto-referência.

Com esses quatro fatores definidos, índices como correção, perfeição, qualidade e redundância, podem ser obtidos e, com isso, pode-se analisar a eficiência do método. A seguir, descreve-se cada um desses índices e como são obtidos por meio dos fatores listados anteriormente.

- Correção (*Correctness*): O índice de correção representa a porcentagem de estradas extraídas corretamente, ou seja, a porcentagem de estradas que condiz com o conjunto-referência. Wiedemann et al. (1998) consideram um desvio-padrão quando esses resultados são comparados. A Figura 3.17 ilustra esse princípio. A Equação 3.35 expressa o índice de correção.

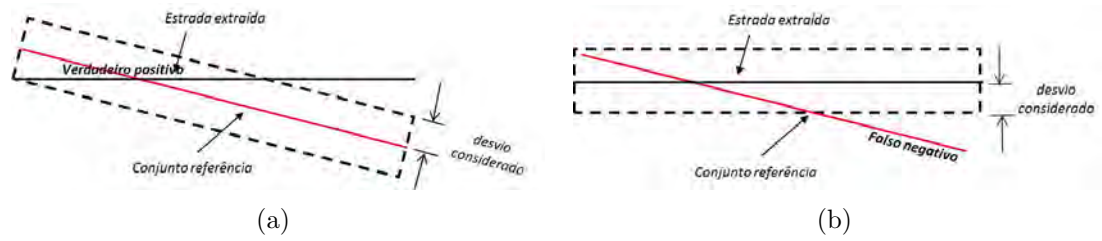


Figura 3.17 - Comparação dos resultados extraídos considerando um desvio-padrão de erro: (a) Verdadeiro positivo. (b) Falso negativo.

Fonte: Adaptado de Wiedemann et al. (1998).

$$\begin{aligned}
 corr &= \frac{exC}{ex} , \\
 corr &\approx \frac{TP}{TP + FP} , \\
 corr &\in [0; 1] ,
 \end{aligned}
 \tag{3.35}$$

em que exC é comprimento total dos segmentos extraídos que coincidiram com a referência, e ex é o comprimento total dos segmentos extraídos. O valor ótimo para o índice de correção é 1.

- Perfeição (*Completeness*): O índice de perfeição representa a porcentagem em que os dados de referência coincidem com os dados de estradas extraídas.

dos. Dado pela equação:

$$\begin{aligned} perf &= \frac{rC}{r} , \\ perf &\approx \frac{TP}{TP + FN} , \\ perf &\in [0; 1] , \end{aligned} \tag{3.36}$$

em que rC denota o comprimento total dos segmentos de referência que coincidiram com a extração, e r o comprimento total dos segmentos de referência. O valor ótimo para o índice de perfeição é 1.

- **Qualidade:** A qualidade leva em conta tanto o índice de perfeição dos dados extraídos, como o índice de correção. É a medida mais geral entre todas elas e é expressa por:

$$\begin{aligned} qualidade &= \frac{rC}{qq} , \\ qualidade &\approx \frac{TP}{TP + FP + FN} , \\ qualidade &\in [0; 1] , \end{aligned} \tag{3.37}$$

em que qq denota a soma entre o comprimento total dos segmentos extraídos que coincidiram com a referência e o comprimento total dos segmentos de referência que não coincidiram com a extração. O valor ótimo para o índice de qualidade é 1.

- **Redundância:** A redundância representa a porcentagem em que foram extraídos segmentos de reta redundantemente. A equação que calcula fator de redundância é dada:

$$\begin{aligned} Redundancia &= \frac{rr}{ex} , \\ Redundancia &\in [0; +\infty] , \end{aligned} \tag{3.38}$$

em que rr denota a diferença entre o comprimento total dos segmentos extraídos que coincidiram com a referência e o comprimento total dos segmentos de referência que coincidiram com a extração. O valor ótimo para o fator de redundância é 0.

Assim como [Wiedemann et al. \(1998\)](#) consideraram o desvio tanto para o conjunto referência como para o conjunto extraído, ilustrado na Figura 3.17, sendo este desvio um valor fixo para ambos. O desvio utilizado por [Wiedemann et al. \(1998\)](#) não é adotado na validação do método de extração deste trabalho. Ao invés disso, o conjunto referência utilizado constitui-se dos eixos centrais de largura de 1 *pixel*, construídos a partir da referência cartográfica. Tais eixos são dilatados de acordo com a largura de estrada predominante na cena. O segmento de reta extraído, por sua vez, é também caracterizado por uma linha de largura de 1 *pixel*, correspondendo ao eixo de simetria da estrada, que, ao ser extraída, é a única característica conhecida. No entanto, na validação dos resultados é essencial que os segmentos de reta extraídos possuam o mesmo desvio agregado aos respectivos segmentos de reta no conjunto referência. Como a largura da estrada no conjunto extraído é desconhecida, o desvio agregado ao conjunto extraído é a largura predominante das estradas presentes na cena, a mesma agregada na dilatação do conjunto referência.

Este capítulo apresentou os conceitos e a estratégia envolvendo a solução do problema de extração de estradas em imagens digitais, bem como as medidas de desempenho do método. A validação dos resultados seguem as medidas apresentadas na última seção deste capítulo, tanto para dados sintéticos como para imagens SAR.

4 RESULTADOS E DISCUSSÃO

Neste capítulo, são apresentados os resultados dos testes realizados para avaliar o desempenho do método de extração de estradas em imagens digitais desenvolvido e apresentado no capítulo anterior. As seções foram categorizadas de acordo com os diferentes experimentos realizados. Em razão da qualidade das imagens e também para efeitos de análise, os experimentos são realizados primeiramente sobre dados sintéticos e, posteriormente, sobre dados de radar. Sendo assim, a primeira parte deste capítulo consiste em apresentar os dados utilizados nos experimentos: imagens sintéticas (Seção 4.1) e imagens reais de radar (Seção 4.2). Na Seção 4.3 são apresentadas a avaliação e discussão envolvendo os resultados, as medidas obtidas, tempos de processamento e complexidade do método.

4.1 Dados Sintéticos

As imagens sintéticas empregadas neste trabalho possuem tamanhos pré-determinados de 574×574 *pixels* (linhas \times colunas) e foram simuladas visando o problema da extração de estradas. O processo de obtenção das imagens sintéticas envolve três etapas: criação, classificação das regiões e modelagem estatística (SILVA et al., 2012). O resultado das duas etapas iniciais denomina-se imagem *phantom* (imagem idealizada de classes de homogeneidade) (SCOFIELD et al., 2007). Serão utilizadas três imagens sintéticas para os experimentos. As imagens, todas em alto contraste, são diferenciadas apenas em suas propriedades polarimétricas, deste modo, as características espectrais das estradas presentes nas diferentes imagens são modificadas.

Na seção seguinte, são apresentados os detalhes da obtenção da imagem *phantom*, imagem simulada a partir de dados reais.

4.1.1 Obtenção do *Phantom*

A imagem *phantom* possui dimensões de 574×574 *pixels*, representando uma região contendo estradas (áreas claras) orientadas em direções distintas. As estradas presentes na cena possuem largura de 3 e 15 *pixels*, respectivamente. Portanto, a imagem *phantom* que será utilizada como imagem de verdade terrestre (imagem referência), possui apenas 2 classes: estrada (regiões de estrada na imagem) e fundo (regiões que não são estradas).

As imagens sintéticas geradas são polarimétricas, nas polarizações HH, HV e VV,

e em uma visada (*1-look*). As matrizes de covariância das duas classes da imagem utilizadas para a simulação das imagens sintéticas foram extraídas a partir de uma imagem SAR polarimétrica, banda L (1,27GHz), adquirida em outubro de 2005 pelo sensor R99B, da Força Aérea Brasileira da região do município de Paulínea, no estado de São Paulo (Figura 4.1.1). As amostras possuem dimensões de 20×20 *pixels*. Para

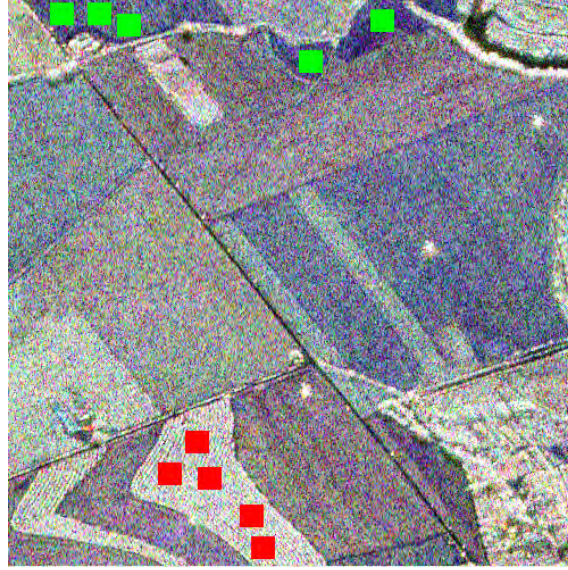


Figura 4.1 - Imagem SAR da região de Paulínea-SP e respectivas amostras para as classes estrada e fundo. Imagem em composição colorida HH(R), HV(G) e VV(B).

cada classe (estrada e fundo) foram extraídas 5 amostras, utilizadas para estimar a matriz de covariância de cada uma das classes. As matrizes de covariância complexas das classes estrada e fundo é dada, respectivamente, por:

$$\sum_{estrada} = \begin{pmatrix} (0,0129; 0,0000) & (0,0012; -0,0007) & (0,0039; 0,0019) \\ (0,0012; 0,0007) & (0,0337; 0,0000) & (-0,0008; -0,0012) \\ (0,0039; -0,0019) & (-0,0008; 0,0012) & (0,0154; 0,0000) \end{pmatrix}$$

$$\sum_{fundo} = \begin{pmatrix} (0,0219; 0,0000) & (-0,0066; -0,0009) & (0,0031; -0,0048) \\ (-0,0066; 0,0009) & (0,0020; 0,0000) & (-0,0007; 0,0016) \\ (0,0031; 0,0048) & (-0,0007; -0,0016) & (0,0015; 0,0000) \end{pmatrix}$$

Os elementos são baseados na notação (\Re, \Im) , em que \Re representa a parte real do número complexo e \Im a parte imaginária. A metodologia empregada para a obtenção da imagem *phantom* (dado simulado) é detalhada por Silva et al. (2012). A imagem

phantom empregada bem como as imagens simuladas, são apresentadas na Figura 4.2.

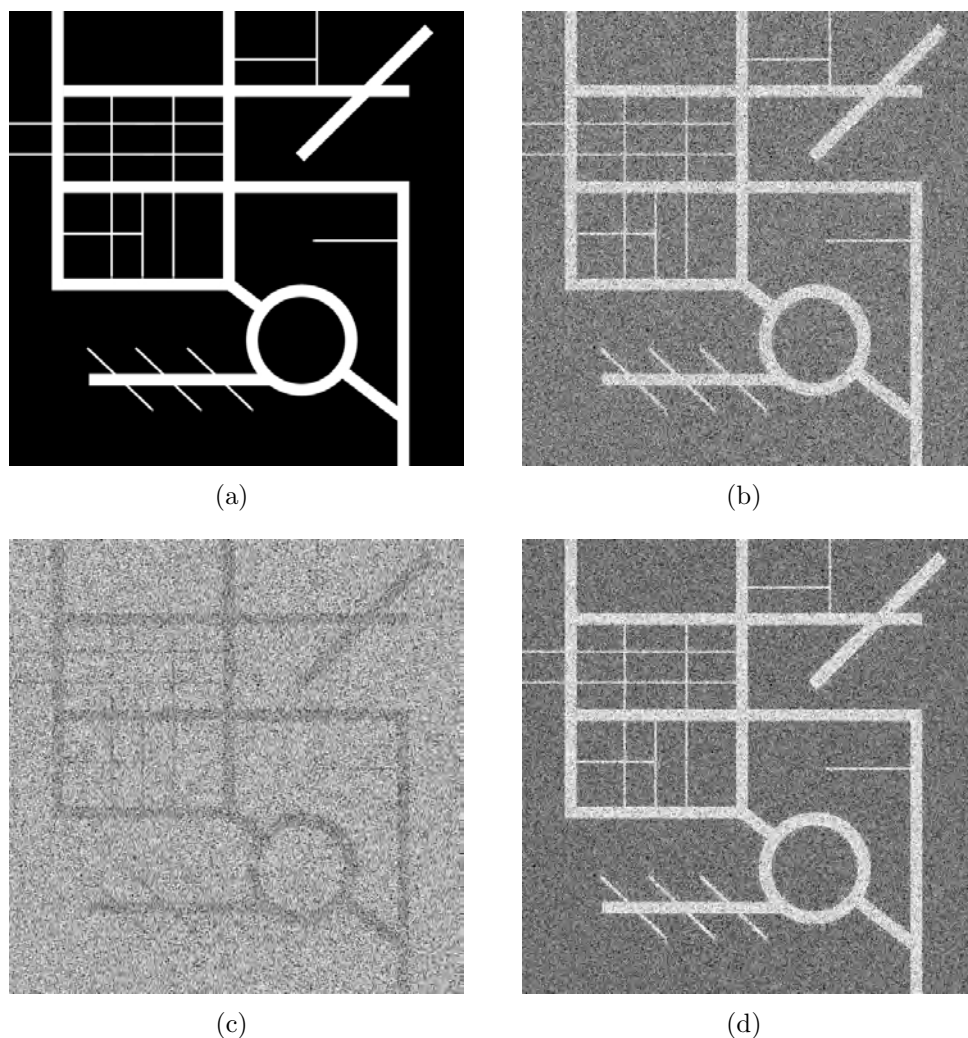


Figura 4.2 - Imagens sintéticas de radar: (a) *Phantom* (b) Polarização VV. (c) Polarização HH. (d) Polarização HV.

4.2 Dados de SAR

Para a realização dos experimentos foram adotados cinco diferentes recortes de uma imagem SAR com dimensões 513×513 , 434×573 , 247×707 , 460×649 e 357×357 *pixels*, respectivamente. Os recortes recobrem a região de Paragominas no estado do Pará, com resolução radiométrica de 8 *bits* e espacial de 2,5 metros na banda P e polarização HH. A imagem foi adquirida com o sensor OrbiSAR da empresa Orbisat

da Amazônia Ind. e Aerolevantamento S.A. no período de aerolevantamento entre 11 de fevereiro de 2007 e 13 de março de 2007, com altitude de voo em torno de 11.000 metros. Esta imagem é mostrada na Figura 4.3, em que são ressaltadas as cinco regiões que representam os recortes extraídos para a realização dos experimentos.

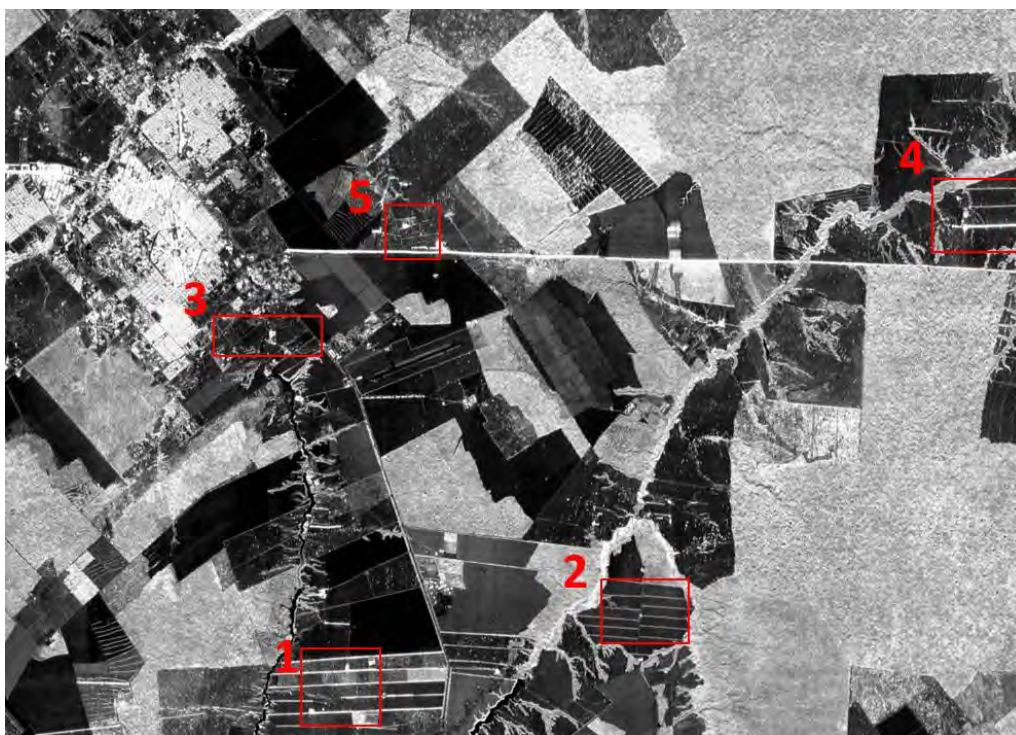


Figura 4.3 - Imagem SAR da região de Paragominas-PA e as localizações das áreas de estudo.

Todos os recortes são baseados em regiões não urbanizadas. Na Figura 4.4 é ilustrado o recorte da região 1 (recorte 1) e a sua respectiva referência cartográfica, a qual será utilizada como verdade terrestre na localização das estradas. As estradas na Figura 4.4(b) estão representadas por linhas tracejadas. Pode-se observar, ainda, que há um predomínio das estradas orientadas na direção horizontal, as quais aparecem na imagem (Figura 4.4(a)) em níveis de cinza mais claros que o nível de cinza do fundo e também apresentam largura maior que as estradas orientadas em outras direções. A classe fundo por apresentar alta variabilidade de níveis de cinza pode dificultar a tarefa de extração de estradas.

O recorte 2, mostrado na Figura 4.5(a), apresenta estradas em níveis de cinza cla-

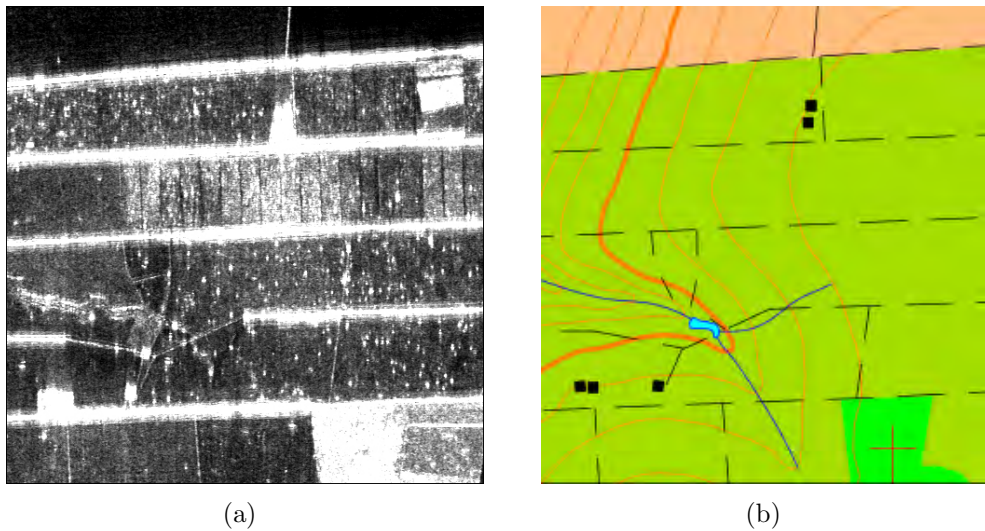


Figura 4.4 - Recorte 1: (a) Imagem SAR (513×513 pixels). (b) Referência cartográfica. Fonte: Orbisat (2010).

ros, estreitas (aproximadamente 7 pixels) e entorno escuro. As linhas tracejadas e contínuas vermelhas, expressas na referência cartográfica (Figura 4.5(b)), representam as estradas na imagem. Assim como no recorte 1 (Figura 4.4), as estradas com orientação horizontal são predominantes no recorte 2, sendo essas interceptadas por somente uma estrada de orientação próxima à vertical, na qual é caracterizada por níveis de cinza mais escuros em relação ao seu entorno. A classe fundo, entretanto, é aparentemente mais homogênea que a classe fundo apresentada no recorte 1, o que pode favorecer o método de extração. No entanto, as estradas no recorte 2 apresentam larguras menores, devido essa característica, o método pode confundir-las com seu entorno.

O terceiro recorte, ilustrado na Figura 4.6, apresenta uma complexidade maior em relação aos anteriores. As linhas tracejadas e contínuas vermelhas presentes na Figura 4.6(b), representam as estradas e caminhos a serem extraídos pelo método de extração. Todas as estradas presentes na cena são orientadas na direção diagonal, as quais aparecem na Figura 4.6(b) em níveis de cinza claro e largura variando entre 1 a 3 pixels. A classe fundo apresenta alta variabilidade de níveis de cinza. A presença de edificações, lagos e rios pode tornar confuso a classificação, principalmente no processo de sementeação. As características de superfície de rio ou lago fazem com que, quase sempre, o retorno do sinal transmitido tenha baixa intensidade, tornando estas

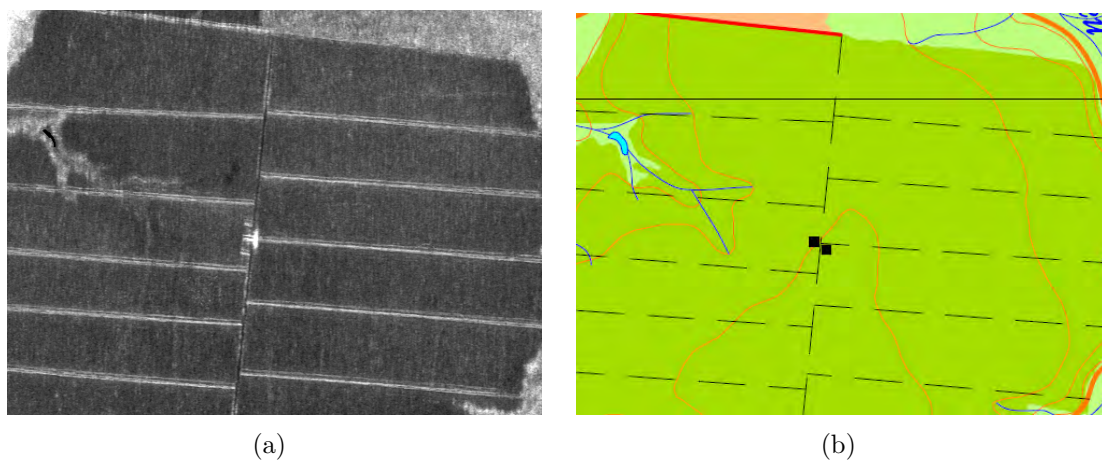


Figura 4.5 - Recorte 2: (a) Imagem SAR (434×573 pixels). (b) Referência cartográfica.
 Fonte: Orbisat (2010).

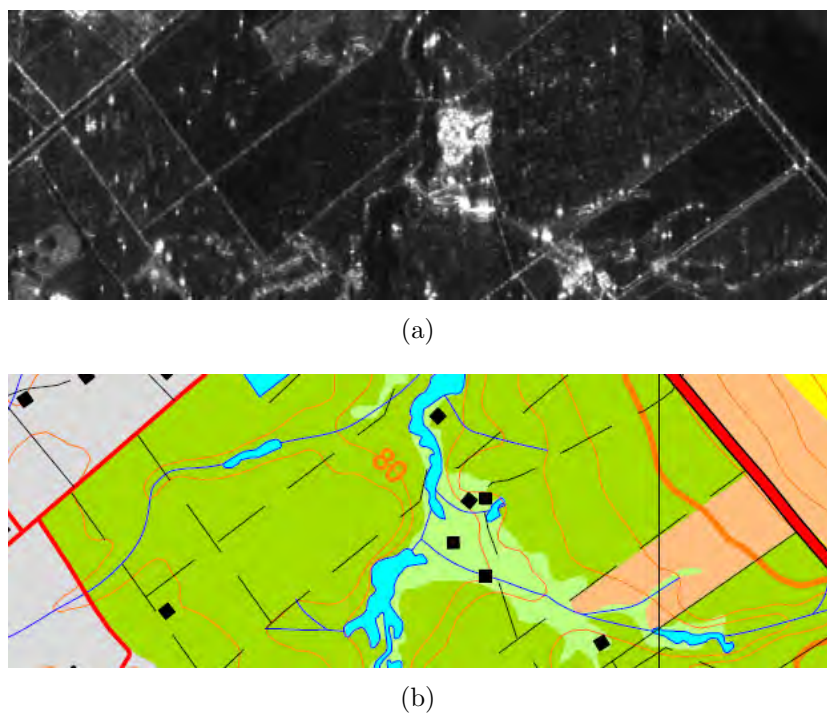


Figura 4.6 - Recorte 3: (a) Imagem SAR (247×707 pixels). (b) Referência cartográfica.
 Fonte: Orbisat (2010).

regiões escuras. Devido à não-construção de uma heurística que certifique que tal região não é uma estrada e considerando que algumas estradas possuem características

tais que tornam sua representação mais escura na imagem de radar, a probabilidade do método de extração reconhecer um lago como uma estrada é grande.

Na Figura 4.7 é mostrada a região 4 (recorte 4), em que grande parte das estradas presentes na cena aparecem em níveis de cinza claros, caracterizadas pelas linhas tracejadas e contínuas vermelhas na Figura 4.7(b). Assim como na Figura 4.6, a classe fundo apresenta também um grande variabilidade de níveis de cinza, causado pela presença de edificações, lagos e regiões de mata ciliar. As estradas são orientadas tanto na direção horizontal como na vertical, em larguras que variam entre 3 a 15 *pixels*.

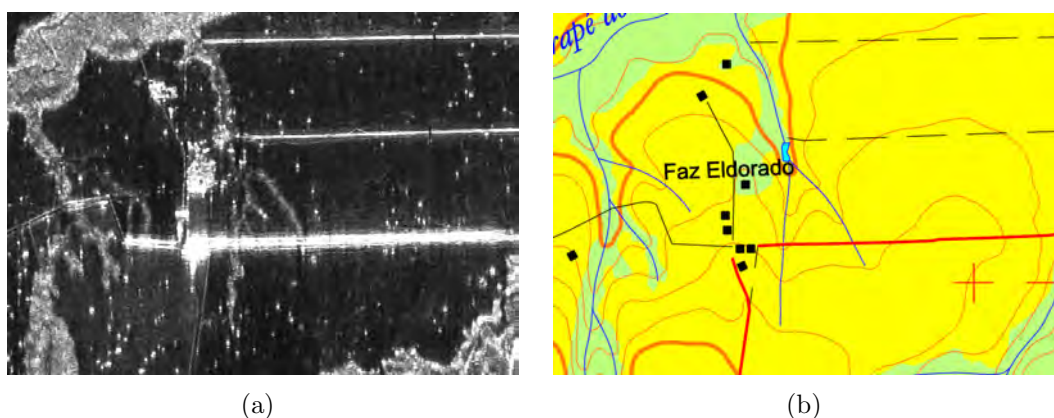


Figura 4.7 - Recorte 4: (a) Imagem SAR (460×649 *pixels*). (b) Referência cartográfica. Fonte: Orbisat (2010).

Por fim, na Figura 4.8, é ilustrado o recorte 5. As estradas neste recorte aparecem em níveis de cinza mais claros que seu entorno, grande parte delas, aparecendo em larguras entre 1 e 2 *pixels*. As linhas tracejadas identificam as estradas com larguras inferiores a 2 *pixels*, e a linha contínua vermelha a estrada com largura aproximada de 12 *pixels*. Nesta região, a variação de níveis de cinza da classe fundo pode ser classificada como moderada, contudo, os perfis de estrada com larguras inferiores podem dificultar o processo de extração.

As regiões 1 e 2 (Figuras 4.4 e 4.5) representam regiões menos complexas para o método de extração. Tendo em vista a ausência de áreas heterogêneas, tais recortes apresentam estradas mais evidentes, sem grande interferência de ruído ou alta vari-

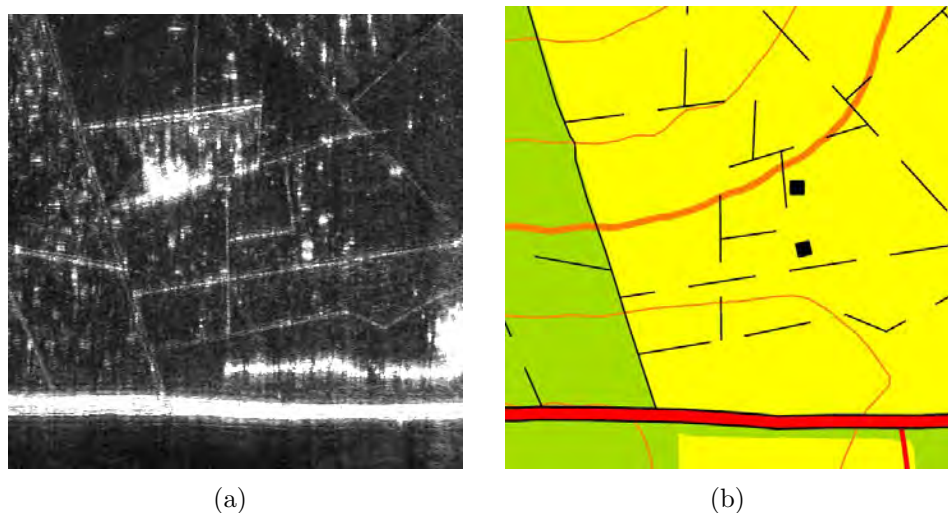


Figura 4.8 - Recorte 5: (a) Imagem SAR (357×357 pixels). (b) Referência cartográfica. Fonte: Orbisat (2010).

abilidade de níveis de cinza em seu entorno. No entanto, as demais regiões podem apresentar um grau elevado de complexidade, tal como aquelas regiões cuja estradas não aparecem nitidamente, ou aparecem em larguras muito inferiores, o que pode aumentar a margem de erro do método de sementeação. A presença de outros objetos não pertencentes à classe estrada, tais como rios, lagos, matas e edificações, podem contribuir consideravelmente com o mau desempenho do método.

Na próxima seção, são apresentados os resultados obtidos, bem como a análise de desempenho do método utilizando as imagens apresentadas nesta seção.

4.3 Resultados e Validação

A fim de avaliar a eficiência do método *Snakes*, os pontos-sementes nas imagens sintéticas são selecionados manualmente, de forma que a localização desses pontos seja realizada adequadamente próximos às feições e, assim, a análise do extrator é voltada apenas para o comportamento do método *Snakes*. As medidas de desempenho são baseadas no método desenvolvido por Wiedemann et al. (1998).

O primeiro experimento é realizado sobre a imagem sintética 1 de polarização VV, apresentada na Figura 4.2(b). As feições presentes na imagem, aparecem em larguras de 3 e 15 pixels, em níveis de cinza mais claro que a classe fundo. Em razão das

estradas neste experimento serem mais evidentes e os pontos-sementes serem selecionados adequadamente por um método, a aplicação do método *Snakes* foi favorecida. Os resultados da aplicação do método de extração são mostrados na Figura 4.9.

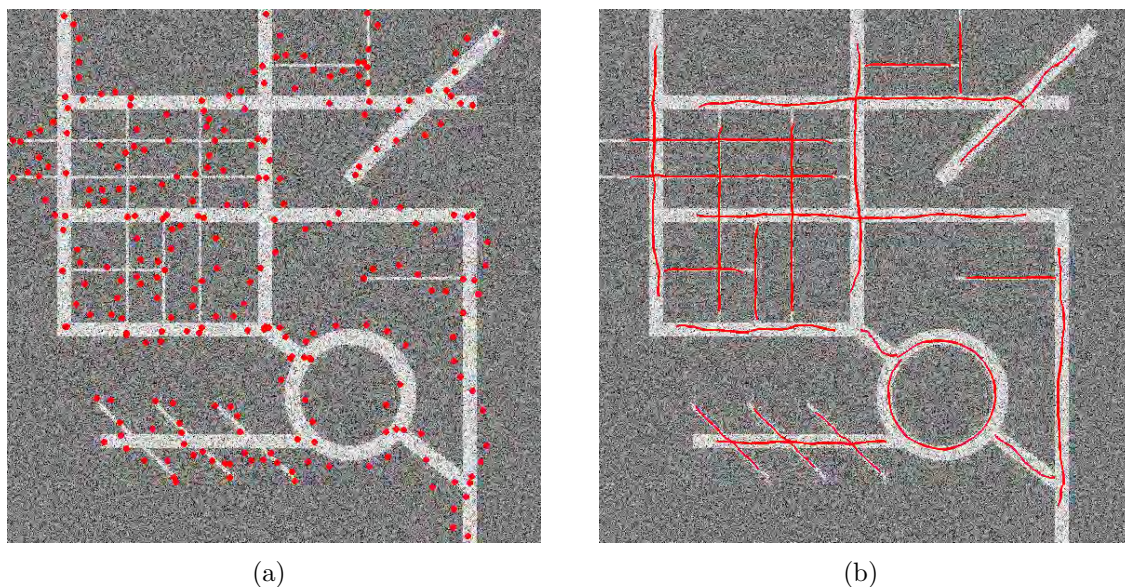


Figura 4.9 - Resultados obtidos para a imagem sintética 1 (Figura 4.2(b)): (a) Pontos-sementes selecionados manualmente e (b) Resultado final após a aplicação do método *Snakes*.

Em alguns trechos da curva não ocorre a conversão para o eixo central da via, isso ocorre devido ao número de iterações não serem suficiente para o processo de minimização. Contudo, a eficiência do método para a extração do eixo de simetria em ambas as larguras de estrada (3 e 15 *pixels*), mostrou-se satisfatória. Na Tabela 4.1 são apresentados os valores adquiridos com a validação dos resultados apresentados na Figura 4.9.

Tabela 4.1 - Medidas de desempenho do método sobre a Figura 4.2(b).

	Perfeição	Correção	Qualidade	Redundância
Sintética 1	0,77	0,63	0,71	0,02

Uma vez que a sementeação é realizada manualmente e os pontos-sementes selecionados sempre próximos às feições, é esperado que os fatores de perfeição, correção e redundância mantenha-se elevados, pois, a medida de perfeição diz respeito à porcentagem que conjunto extraído sobrepôs o conjunto referência, a correção, a porcentagem que o conjunto referência sobrepôs o conjunto extraído, e a medida de redundância, a porcentagem que os segmentos de reta pertencentes ao conjunto extraído sobrepôs uns aos outros. Sabendo que a sementeação é realizada próxima às feições, e considerando a conversão esperada do método *Snakes*, o conjunto extraído se manterá próximo dos eixos centrais, ou seja, medida de perfeição elevada, e conseqüentemente, medida de correção também elevada, pois nenhum ponto-semente foi selecionado distante das feições. O fator redundância é, portanto, também elevado, pois para uma única feição é semeada um único conjunto de pontos, o que, quase sempre, exclui a possibilidade de sobreposições.

No segundo experimento, o método *Snakes* é aplicado sobre a imagem sintética 2 de polarização HH (Figura 4.2(c)). As estradas, principalmente as de largura inferior, confundem-se quase que inteiramente com a classe fundo, tornando difícil a tarefa de reconhecimento. No entanto, as curvas convergiram com exatidão para o centro das vias em muito dos trechos dos segmentos de reta. O sinal do parâmetro w_{line} , que considera o tipo de perfil de estrada que a curva deve ser atraída, deve ser alterado. Considerando que as estradas na Figura 4.2(c) apresentam tonalidades mais escura que a classe fundo, é agregado o sinal positivo ao parâmetro w_{line} , que possibilita à curva atrair-se para o centro de feições escuras.

Os experimentos realizados sobre a imagem sintética da Figura 4.2(c) são apresentados na Figura 4.10.

A aplicação do método *Snakes* mostrou-se eficiente ao identificar, em grande parte, os eixos centrais das estradas. Na Tabela 4.2, são apresentados os resultados obtidos na aplicação do método.

Tabela 4.2 - Medidas de desempenho do método sobre a Figura 4.2(c).

	Perfeição	Correção	Qualidade	Redundância
Sintética 2	0,71	0,57	0,64	0,01

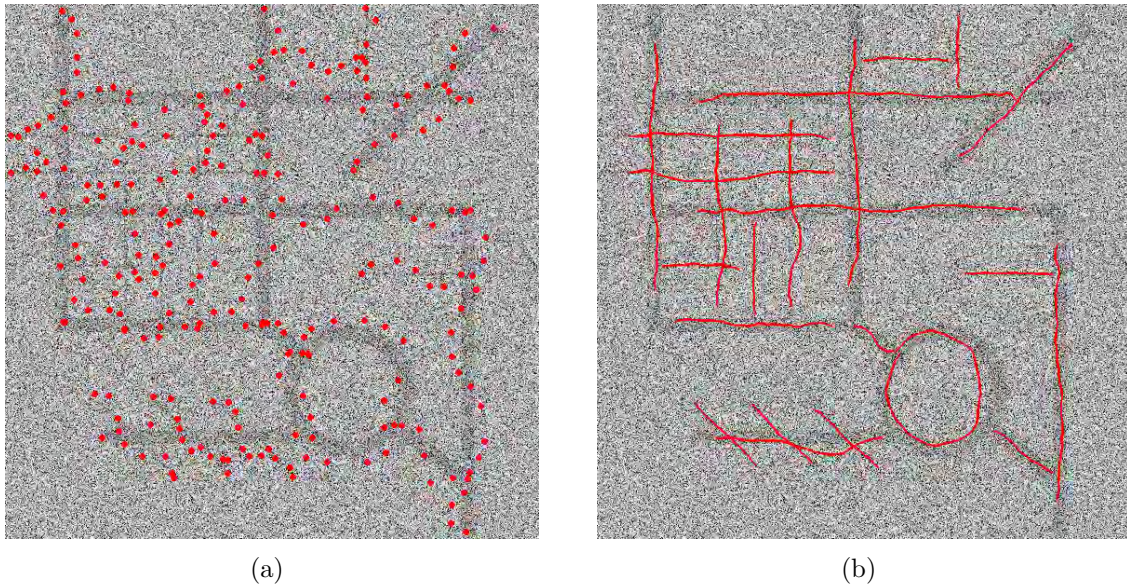


Figura 4.10 - Resultados obtidos para a imagem sintética 2 (Figura 4.2(c)): (a) Pontos-ementes selecionados manualmente e (b) Resultado final após a aplicação do método *Snakes*.

Apesar dos valores na Tabela 4.2 aparentarem mau desempenho do método, sua eficiência é clara quando visualizada na Figura 4.10(b). Pouco dos trechos com a presença de estradas, as curvas não convergiram adequadamente até o centro das feições, o que impediu de alcançar qualidade superior. É válido salientar que a qualidade da extração nesse experimento é dada principalmente pela seleção dos pontos-ementes próximos às estradas.

No terceiro e último experimento, a imagem sintética de polarização HV apresentou resultados semelhantes aos obtidos na Figura 4.10. Assim como no primeiro experimento, em alguns trechos a curva não converge completamente para o eixo central. Os resultados da aplicação do método para a imagem sintética 3 são apresentados na Figura 4.11.

Embora o reconhecimento do eixo central não tenha sido completo, as áreas não reconhecidas foram mínimas, justificando as medidas de desempenho apresentadas na Tabela 4.3.

Nesses três primeiros experimentos com imagens sintéticas, é testada a eficiência do método *Snakes* sobre a presença de ruído na imagem, em que é utilizada a semea-

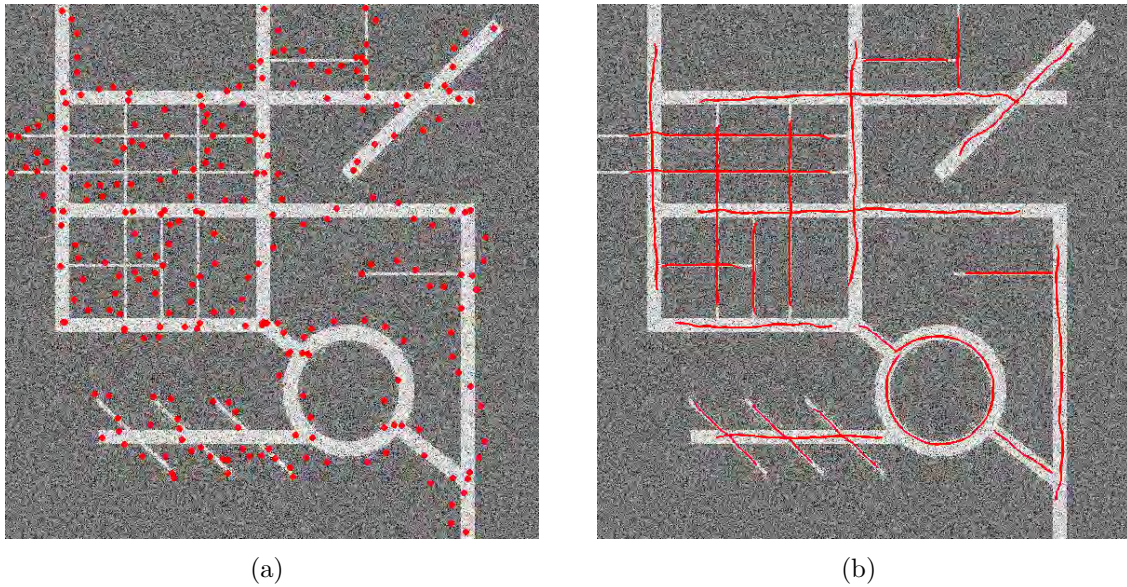


Figura 4.11 - Resultados obtidos para a imagem sintética 3 (Figura 4.2(d)): (a) Pontos-sementes selecionados manualmente e (b) Resultado final após a aplicação do método *Snakes*.

Tabela 4.3 - Medidas de desempenho do método sobre a Figura 4.2(d).

	Perfeição	Correção	Qualidade	Redundância
Sintética 3	0,77	0,64	0,71	0,03

ção manual dos pontos-sementes. A seguir, a sementeção é realizada em experimentos utilizando imagens SAR reais, de modo que a sementeção é aplicada de forma semi-automática, utilizando para tanto, o auxílio da rede SOM, apresentada no Capítulo 3.

A princípio, são selecionados alguns parâmetros para a alimentação do método de sementeção, tais como a taxa de aprendizagem da rede SOM, a dimensão do mapa de Kohonen, a quantidade máxima de épocas de treinamento, tamanho do salto da janela de leitura e quantidade de perfis a serem lidos por amostra. Esses fatores são definidos empiricamente por um operador humano, baseando-se nas características da imagem. O parâmetro referente ao tamanho do salto, por exemplo, influencia consideravelmente na marcação dos pontos-sementes. Considerando uma imagem com largura predominante de estradas de 3 *pixels*, ao se definir um salto relativamente grande, a probabilidade de a janela de leitura “saltar” sobre uma estrada,

descartando-a, é grande, e assim, independentemente da largura das estradas presentes na cena, é recomendado um fator de salto pequeno. A dimensão do mapa de Kohonen também deve possuir dimensão pequena. Isto se deve ao surgimento de regiões de incerteza no domínio de Kohonen, ou seja, as regiões tendem a não se especializar quando o mapa possui um grande número de neurônios, de forma que sejam criadas pequenas áreas que induzem ao erro. Desta forma, a dimensão do mapa é definida como sendo a menor possível para a aplicação.

Tabela 4.4 - Parâmetros adotados no método de extração para os recortes apresentados.

Imagem SAR	Largura Predominante (n° <i>pixels</i>)	Fator de salto (n° <i>pixels</i>)	Perfis por amostras
Recorte 1	15	12	10
Recorte 2	7	10	10
Recorte 3	3	10	10
Recorte 4	10	10	10
Recorte 5	2	10	10

A Tabela 4.5 são apresentados os níveis de intensidade utilizados para cada padrão de treinamento. Os padrões foram definidos em um perfil de 21 *pixels*, tal que os valores de intensidade são estabelecidos a partir da observação das imagens reais utilizadas neste trabalho. Os quatro primeiros padrões da tabela dizem respeito aos padrões a serem reconhecidos, e os quatro últimos, os padrões a serem ignorados.

Tabela 4.5 - Níveis de cinza utilizados nos padrões de treinamento.

Padrões	Níveis de cinza																				
	158	125	87	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	158	166	184
X_1	158	125	87	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	158	166	184
X_2	186	163	168	191	199	186	176	204	0	0	0	184	212	204	201	194	204	212	201	212	207
X_3	158	125	87	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	158	166	184
X_4	186	163	168	191	199	186	176	204	255	255	255	184	212	204	201	194	204	212	201	212	207
X_5	186	186	199	173	176	168	171	207	196	201	181	186	176	194	196	189	201	186	209	196	186
X_6	245	247	237	250	255	245	237	230	235	247	237	232	232	240	250	247	245	250	242	250	237
X_7	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
X_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

No processo de poda, foram considerados (para todos os recortes) os limiares de 4 e 2 para os fatores de Limite Mínimo de Continuidade (LMC) e Limite Máximo de Descontinuidade (LMD), respectivamente. Da mesma forma, os valores de taxa de aprendizagem, dimensão do mapa de Kohonen, dimensão das amostras, raio de vizinhança e quantidade máxima de épocas, foram mantidas as mesmas, sendo 0,05, 6×6 , 21×21 , 3 e 2.000, respectivamente.

Os valores de perfeição e correção denotam o quanto foi extraído dos dados de referência e o quão corretamente as estradas foram extraídas, respectivamente. A qualidade é calculada a partir dessas duas medidas. Os resultados para as imagens de radar são apresentados conforme a sequência em que foram descritas na seção anterior. Para cada imagem, são exibidos os resultados (i) da sementeação, os pontos-sementes obtidos antes e após a aplicação do algoritmo de poda, e (ii) o resultado final obtido após a aplicação do algoritmo de *Snakes*. Nas figuras a seguir, são apresentados os resultados da sementeação aplicados a Figura 4.4 (recorte 1).

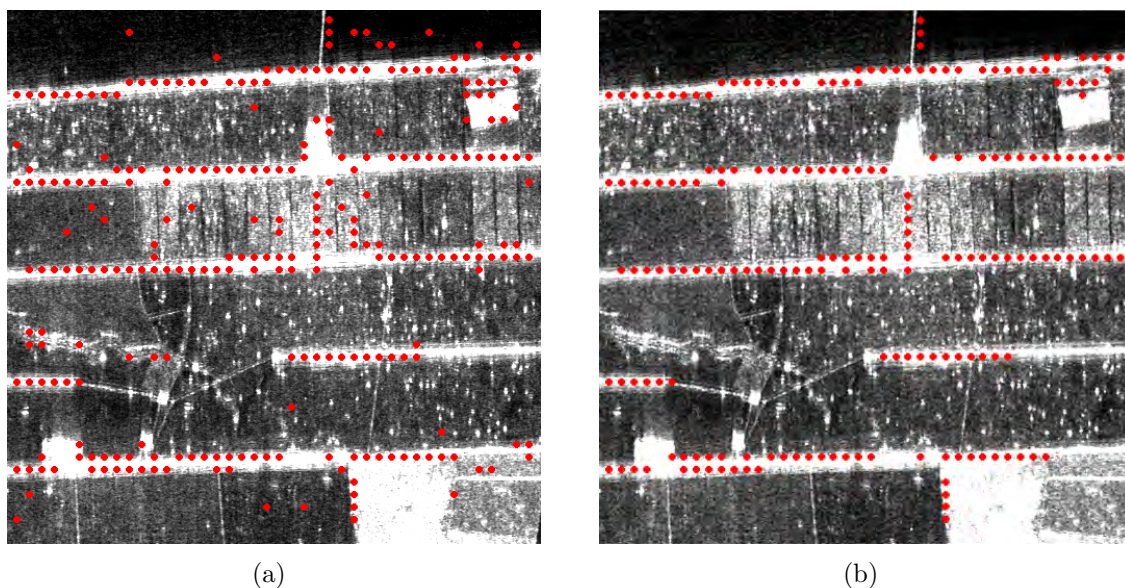


Figura 4.12 - Resultados obtidos para o recorte 1 (Figura 4.4): (a) Sementeação sem a realização da poda. (b) Sementeação após a poda.

Nota-se, pelo resultado da sementeação, que o método de poda foi eficaz na exclusão dos pontos espúrios. Torna-se evidente que alguns pontos, mesmo sendo marcados

corretamente pelo método, são excluídos na poda. Isso ocorre devido à quantidade de pontos ser inferior ao limiar LMC. O número de pontos-sementes em sequência deve atingir esse limite para que um determinado segmento de reta seja considerado. Após a sementeação, os pontos-sementes são repassados ao método *Snakes*, que por sua vez os interpola e aplica o processo de minimização da curva para cada um dos segmentos do conjunto.

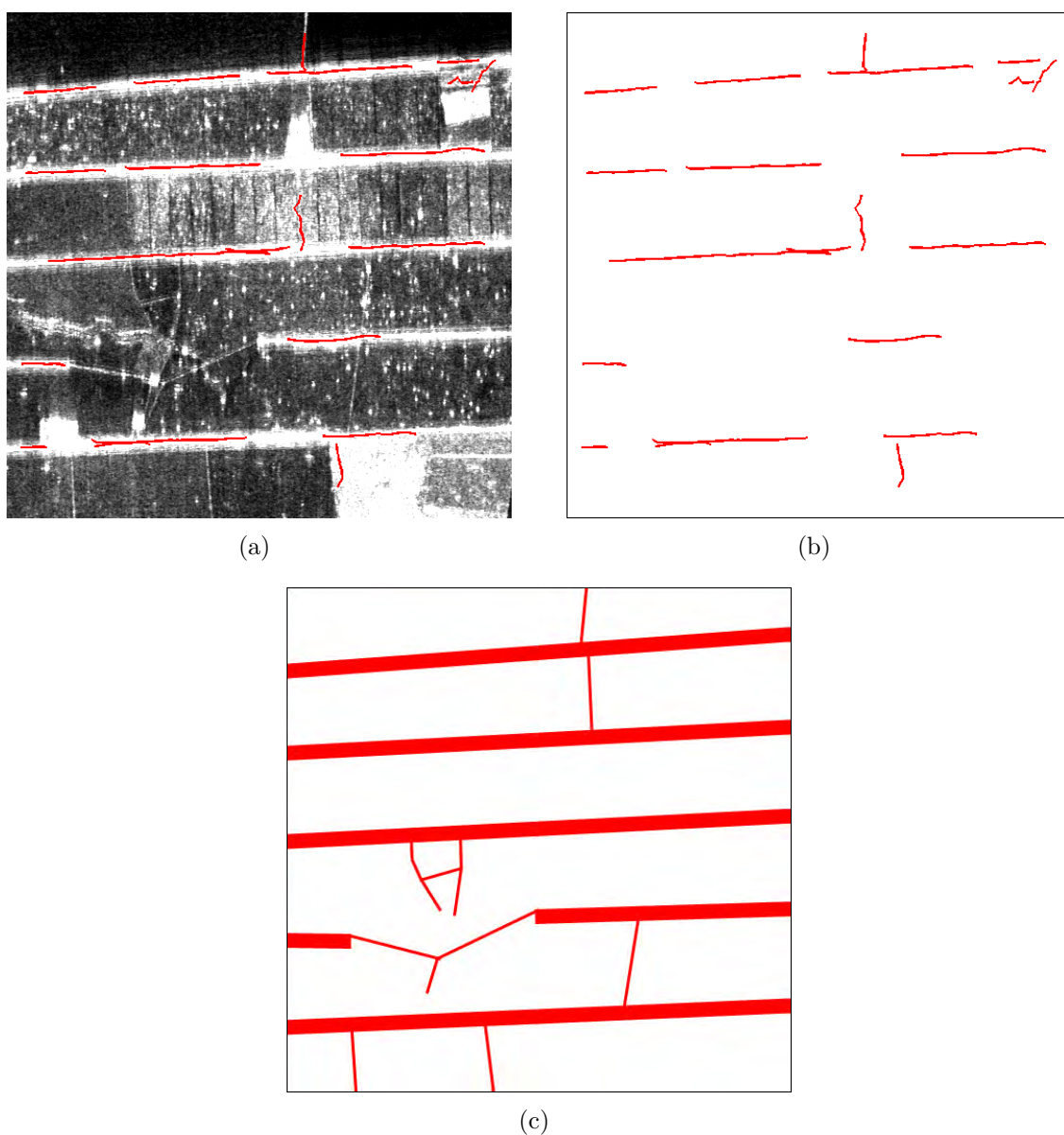


Figura 4.13 - Resultados obtidos para o recorte 1 (Figura 4.4): (a) Resultado sobre a imagem, após a aplicação do método *Snakes*. (b) Resultado final sem a imagem. (c) Redes de estradas plotadas manualmente (referência).

O resultado final é mostrado na Figura 4.13, em que se pode notar a eficiência do método na extração das estradas com largura aproximada de 15 *pixels*. As estradas estreitas (entre 2 a 3 *pixels*), no entanto, não é identificada pelo método de sementeação que, conseqüentemente, não é extraída pelo método *Snakes*. Na Tabela 4.6 são apresentadas as medidas de desempenho obtidas a partir dos resultados na Figura 4.13.

Tabela 4.6 - Medidas de desempenho do método sobre o recorte 1.

	Perfeição	Correção	Qualidade	Redundância
Recorte 1	0,47	0,68	0,58	0,03

Ainda que as estradas com largura inferior não tenham sido identificadas pelo método de sementeação, a qualidade dos resultados mostrou-se satisfatória em razão de as estradas largas serem predominantes nesse recorte. Os trechos marcados incorretamente foram mínimos, contribuindo para a qualidade da extração.

No recorte 2 (Figura 4.5), a classe fundo apresenta uma região mais homogênea quando comparada ao recorte 1. Embora essa característica seja favorável ao método de extração, o número de pontos espúrios é maior em relação aos apresentados na Figura 4.12. No entanto, a qualidade da sementeação mostrou-se equivalente. As estradas neste recorte são caracterizadas pela largura aproximada de 3 *pixels* e perfis de tonalidades claras, e em alguns trechos, essa tonalidade se mistura com a classe fundo. Esses fatores fazem com que o método, no processo de leitura das amostras, se confunda e não determine de forma precisa a presença de uma estrada. Como a largura predominante neste recorte é menor em relação ao primeiro, o salto da janela de leitura é também definido com um valor menor.

Em algumas regiões da imagem, há a presença de lagos, matas e edificações, que prejudicam o método na identificação de estradas. Esta deficiência pode ser visualizada na Figura 4.14, em que as transições de níveis de cinza causadas pelas fronteiras de vegetação são apontadas pelo método como sendo estradas.

Após a eliminação dos pontos espúrios no processo de poda, o método *Snakes* é aplicado em cada um dos segmentos de reta. O resultado é apresentado na Figura 4.15 que, assim como no recorte 1, mostrou-se satisfatório. Uma vez que a localização

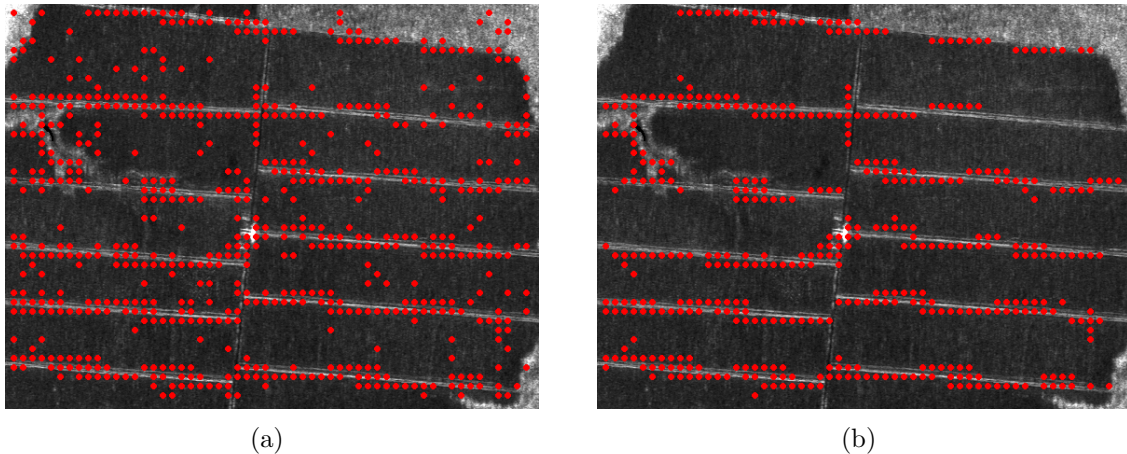


Figura 4.14 - Resultados obtidos para o recorte 2 (Figura 4.5): (a) Semeação sem a realização da poda. (b) Semeação após a poda.

dos pontos-sementes estejam próximos às estradas, a maioria das curvas convergiram para o eixo central da via. A Tabela 4.7 apresenta as medidas de desempenho do método para o recorte 2.

Tabela 4.7 - Medidas de desempenho do método sobre o recorte 2.

	Perfeição	Correção	Qualidade	Redundância
Recorte 2	0,69	0,71	0,70	0,25

Os segmentos de retas incorretos estão presentes em maior quantidade neste recorte, porém, a medida de correção foi elevada em decorrência da quantidade de acertos. A identificação correta das estradas é representada pelo fator perfeição. Ao analisar esse fator no recorte 2, nota-se que as estradas com feições claras e orientação próxima a horizontal foram identificadas de maneira correta. Alguns trechos das estradas presentes na cena são extraídos em duplicidade, o que justifica a elevação do fator redundância no recorte 2.

O terceiro recorte (Figura 4.6) é caracterizado por uma imagem com ruídos em excesso e estradas com largura predominante de 3 *pixels*. Este cenário representa maior complexidade para o método em relação aos recortes 1 e 2. As estradas aparecem em orientações distintas, largura inferior a 3 *pixels* e a presença de outros objetos não

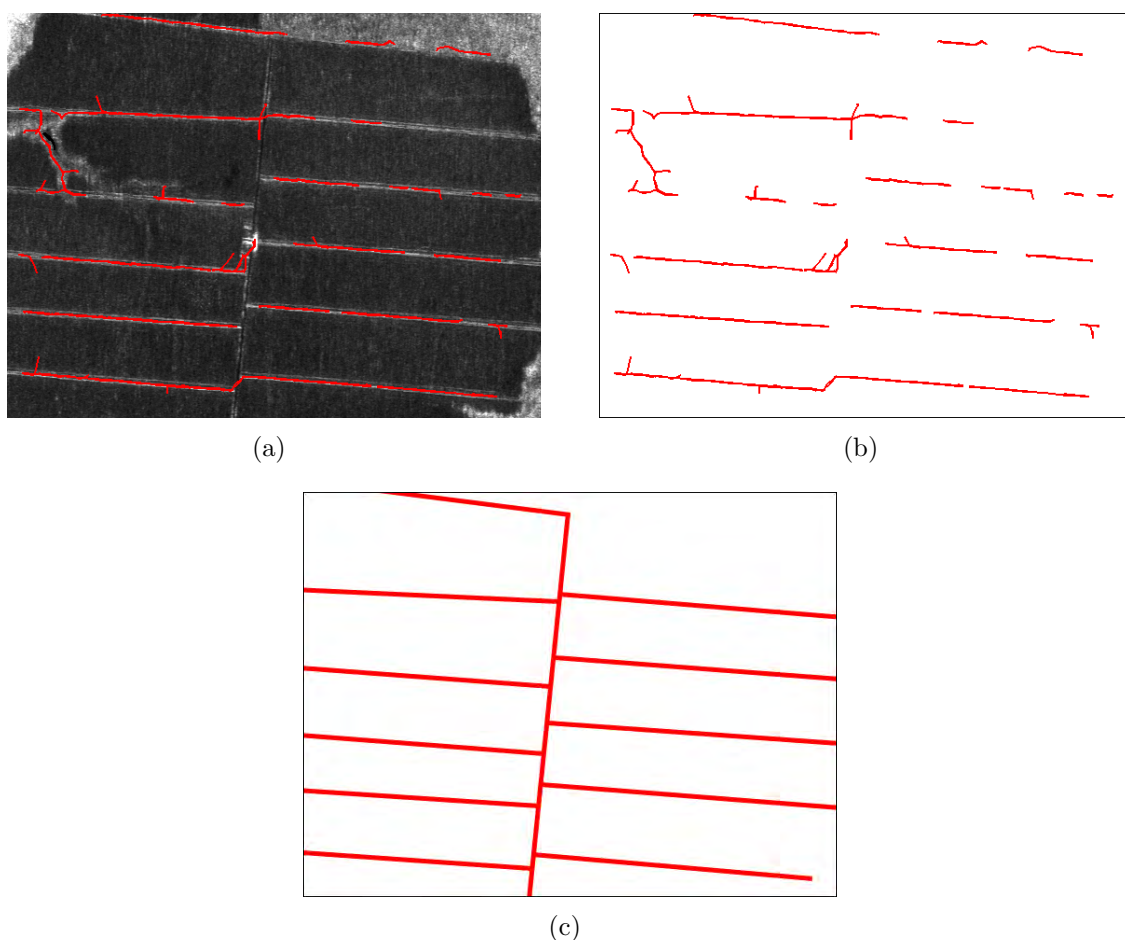
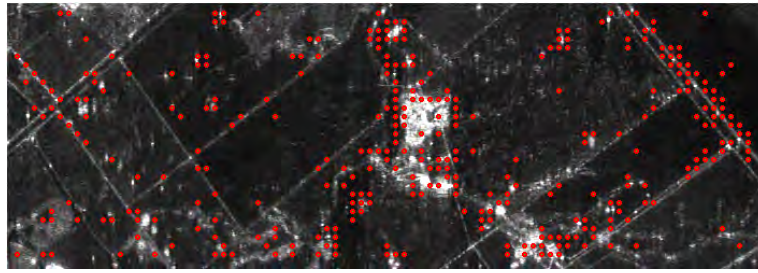


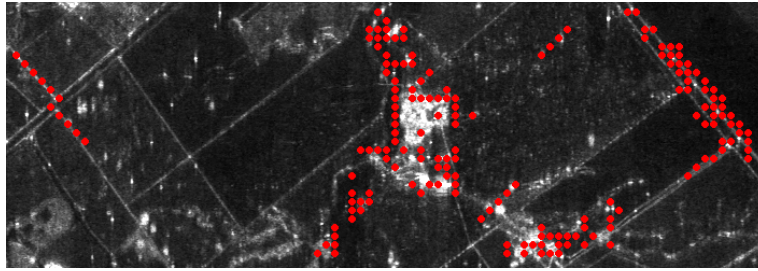
Figura 4.15 - Resultados obtidos para o recorte 2 (Figura 4.5): (a) Resultado sobre a imagem, após a aplicação do método *Snakes*. (b) Resultado final sem a imagem. (c) Redes de estradas plotadas manualmente (referência).

pertencentes a classe estrada, tais como lagos, edificações e mata ciliar, faz com que o método as identifique como sendo o padrão de interesse. A Figura 4.16 representa o resultado do processo de sementeação sem a aplicação da poda e após a poda, respectivamente.

Analisando a Figura 4.16, nota-se que as interferências presentes neste recorte contribuíram notoriamente na performance do método. Muitos pontos são sinalizados em edificações, lagos e outros alvos. Em virtude desses obstáculos e da largura da estrada, o método apresentou resultados ruins, porém, um resultado esperado, tendo em vista a presença de ruído em grande quantidade. Na Figura 4.17 é apresentado o recorte 3 após a aplicação do método *Snakes*, cujas curvas foram obtidas em pequena



(a)



(b)

Figura 4.16 - Resultados obtidos para o recorte 3 (Figura 4.6): (a) Semeação sem a realização da poda. (b) Semeação após a poda.

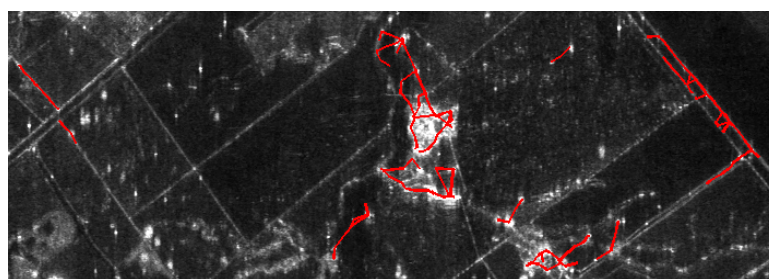
quantidade, evidenciando baixa qualidade do resultado. A Tabela 4.8 são mostrados os valores obtidos na validação do método sobre a aplicação no recorte 3.

Tabela 4.8 - Medidas de desempenho do método sobre o recorte 3.

	Perfeição	Correção	Qualidade	Redundância
Recorte 3	0,13	0,26	0,21	0,35

De fato, os valores apresentados na Tabela 4.8 tem a ver com o baixo rendimento do método de extração que, embora ruins, é esperado, justamente pelas características da imagem e a maneira em que as estradas estão dispostas neste recorte.

No recorte 4, mostrado na Figura 4.7, as estradas aparecem na cena em larguras de 3 e 15 *pixels*, aproximadamente, sendo predominante as estradas com largura inferior. Neste recorte, a classe fundo possui um número maior de elementos, que dificultou a tarefa de reconhecimento do método. A grande variabilidade de níveis de cinza presentes nas regiões de mata ciliar e rios fizeram com que o método de semeação reconhecesse estas áreas como estradas, justamente pelas transições de



(a)



(b)



(c)

Figura 4.17 - Resultados obtidos para o recorte 3 (Figura 4.6): (a) Resultado sobre a imagem, após a aplicação do método *Snakes*. (b) Resultado final sem a imagem. (c) Redes de estradas plotadas manualmente (referência).

níveis de cinza serem similares aos padrões de treinamento.

Novamente, no recorte 4 pode-se notar a ineficiência do método de semente na identificação das estradas com largura inferior a 3 *pixels*. As feições com largura entre 4 e 6 *pixels* são identificadas, ainda que parcialmente. Na Figura 4.7 é apresentado o recorte 4 após a aplicação do método *Snakes*.

Os eixos de simetria obtidos para o recorte 4 são considerados, no geral, satisfatórios, face às características da imagem. A Tabela 4.9 são mostrados os valores obtidos na validação do método sobre a aplicação no recorte 4.

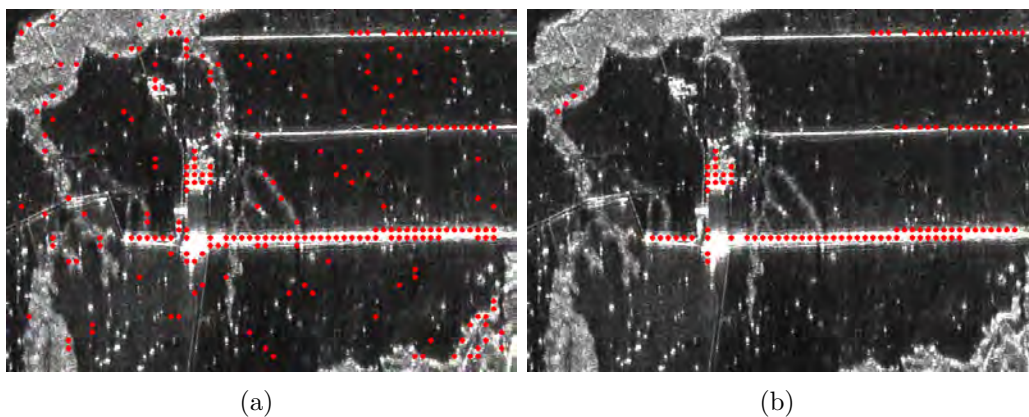


Figura 4.18 - Resultados obtidos para o recorte 4 (Figura 4.7): (a) Semeação sem a realização da poda. (b) Semeação após a poda.

Tabela 4.9 - Medidas de desempenho do método sobre o recorte 4.

	Perfeição	Correção	Qualidade	Redundância
Recorte 4	0,31	0,79	0,60	0,00

Os valores apresentados na Tabela 4.9 são coerentes ao serem contrastados com a Figura 4.19. A porcentagem maior do conjunto de estradas presentes na cena não são identificadas pelo método, reduzindo o índice de perfeição. Ambos os fatores de correção e redundância mantiveram-se com valores mínimos, demonstrando a não detecção de segmentos incorretos e a baixa sobreposição entre os segmentos extraídos, respectivamente.

No recorte 5 (Figura 4.8), as estradas com largura de 1 *pixel* são predominantes. A semeação é eficiente somente na estrada de largura aproximada de 12 *pixels*, única na cena. Os demais pontos-sementes marcados na imagem constituem-se de pontos espúrios, que são eliminados pelo método de poda (Figura 4.20).

Embora o salto da janela de varredura tenha sido definido com valor menor, o reconhecimento das estradas não ocorre. Entre os experimentos realizados, grande parte dos pontos-sementes identificados em regiões de mata ciliar, edificações e lagos, são excluídos de maneira eficaz pelo método de poda, o que permite ao processo de extração ser aplicado somente em segmentos de reta cuja a continuidade dos pontos-sementes seja pertinente. Na Figura 4.21 são mostrados os resultados da aplicação do método *Snakes* para os pontos-sementes obtidos no recorte 5.

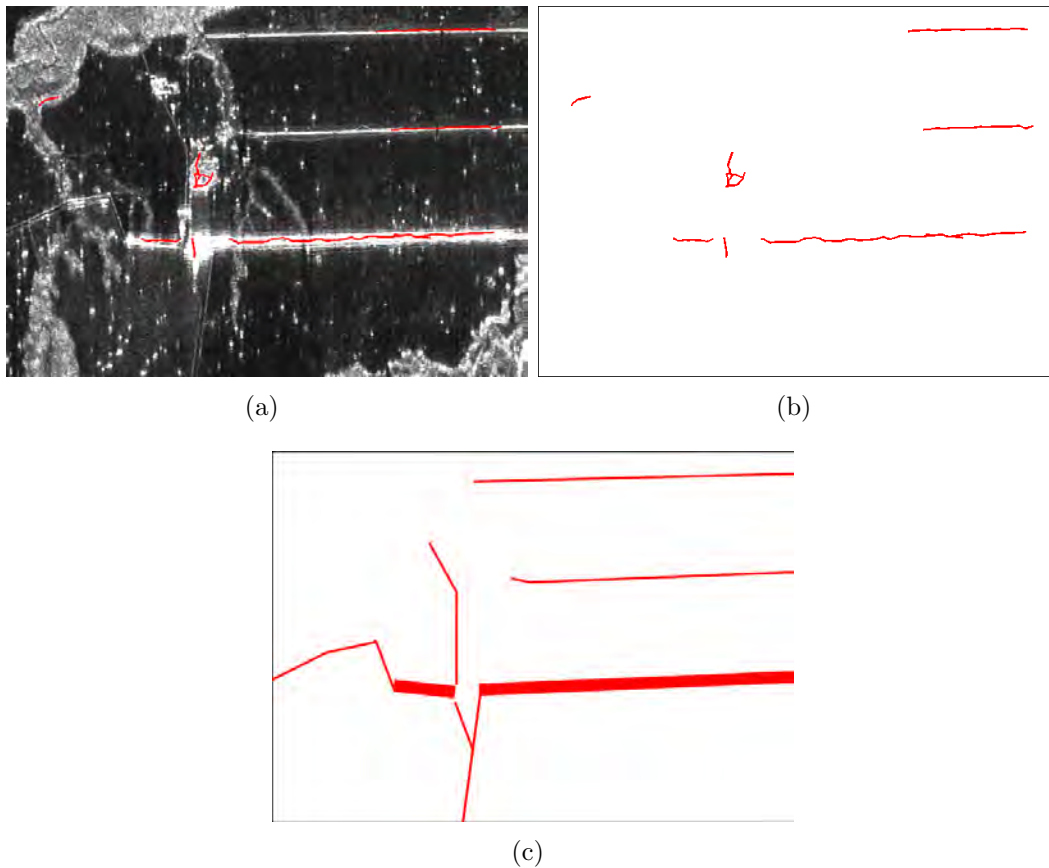


Figura 4.19 - Resultados obtidos para o recorte 4 (Figura 4.7): (a) Resultado sobre a imagem, após a aplicação do método *Snakes*. (b) Resultado final sem a imagem. (c) Redes de estradas plotadas manualmente (referência).

Uma vez os pontos-sementes identificados próximos às estradas, a obtenção do eixo de simetria é quase sempre certa, tal como mostrados nos demais experimentos. Na Tabela 4.10 são mostrados os valores obtidos na validação do método sobre a aplicação no recorte 5.

Tabela 4.10 - Medidas de desempenho do método sobre o recorte 5.

	Perfeição	Correção	Qualidade	Redundância
Recorte 5	0,09	0,66	0,47	0,00

A porcentagem de acerto dos resultados, mostrada na Tabela 4.10, mostrou-se pertinente devido ao não reconhecimento da maioria das estradas presentes na cena.

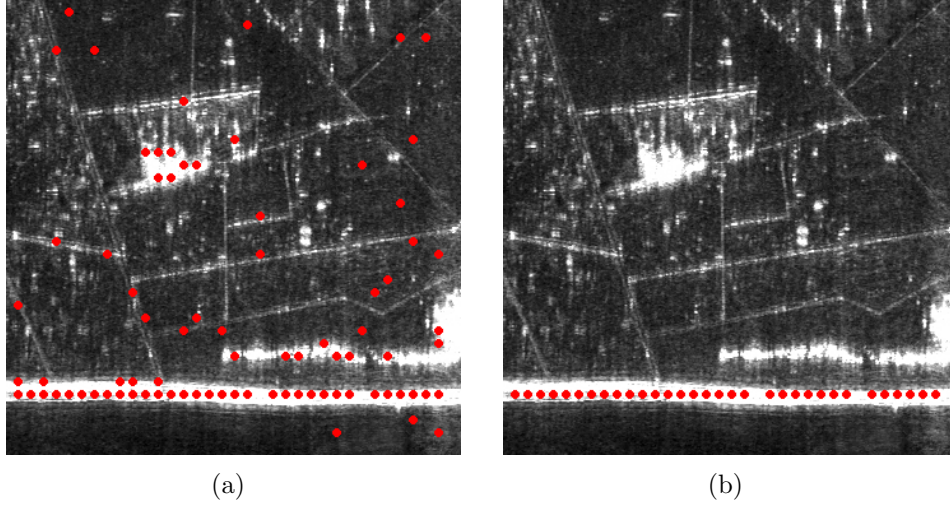


Figura 4.20 - Resultados obtidos para o recorte 5 (Figura 4.8): (a) Semeação sem a realização da poda. (b) Semeação após a poda.

Nenhum segmento de reta foi extraído de forma incorreta, o que justifica o fator de correção elevado. Por ter extraído um único elemento, o fator perfeição apresentou valor inferior e, redundância, valor ótimo.

As medidas de desempenho obtidas para cada experimento variam entre 0 e 1, em que o valor 0 representa o ajuste ótimo para a medida de Redundância, e 1 representa o ajuste ótimo para as medidas de Perfeição, Correção e Qualidade. Um fator de desempenho global que considera as quatro medidas pode ser calculado pela distância euclidiana, em que é calculada a distância das medidas em relação ao valores ótimos $P = [1, 1, 1, 0]$:

$$d_l = \sqrt{(perf - 1)^2 + (corr - 1)^2 + (qual - 1)^2 + (redu - 0)^2} . \quad (4.1)$$

A variação de d_l é entre 0 e 2, em que admi-se como melhor resultado aquele que produzir a menor distância d_l . Na Tabela 4.11 são apresentados os valores de d_l para cada experimento (SCOFIELD et al., 2007).

Os resultados correspondentes à aplicação semiautomática do método de semeação sobre as imagens de radar, embora tenham apresentado bons resultados, necessitam de melhoramentos, principalmente no método de semeação, pois, conforme mostraram os experimentos, à medida em que as estradas se estreitam, e pelo o aumento de ruído, o método torna-se ineficiente.

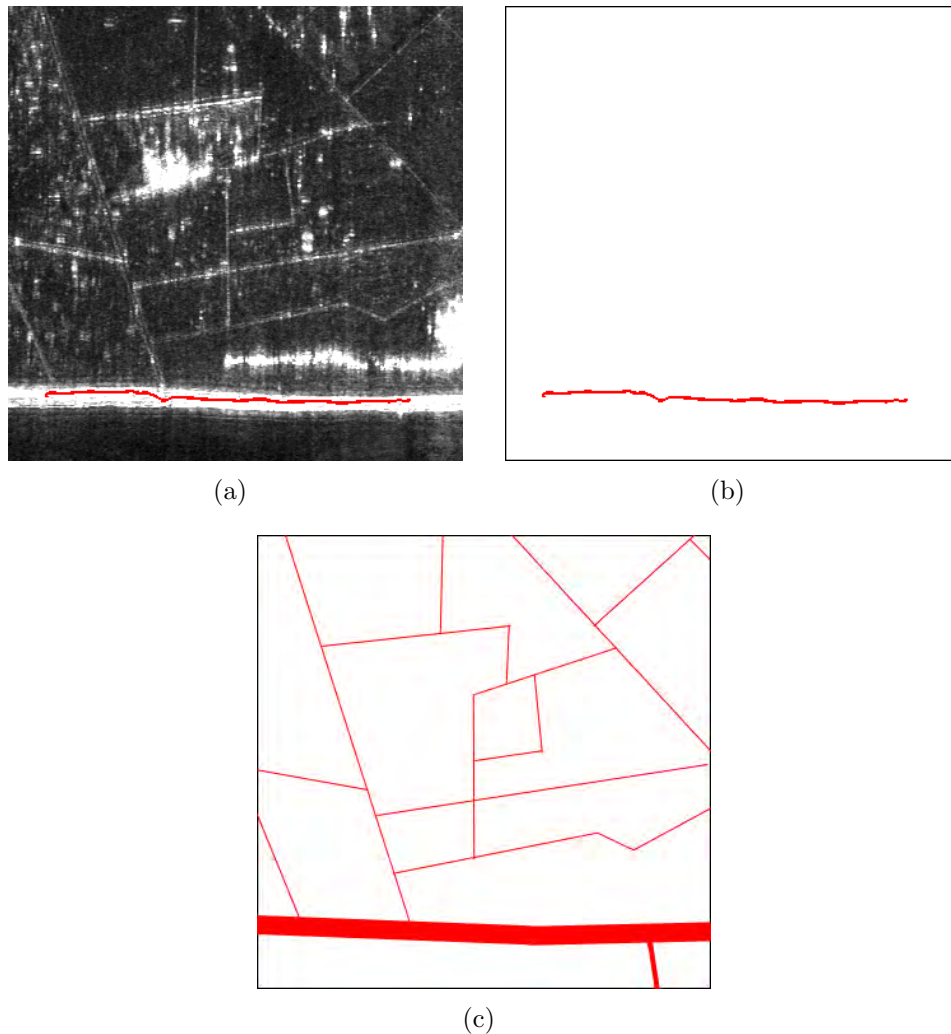


Figura 4.21 - Resultados obtidos para o recorte 5 (Figura 4.8): (a) Resultado sobre a imagem, após a aplicação do método *Snakes*. (b) Resultado final sem a imagem. (c) Redes de estradas plotadas manualmente (referência).

A fim de medir o tempo e complexidade do algoritmo, são apresentados a seguir o tempo e complexidade do método de extração de estradas divididos entre as etapas: Treinamento da RNA, Semeação e *Snakes*. O tempo de processamento da segunda etapa depende diretamente da quantidade de segmentos de reta obtidos pelo método de Semeação. Assim, o valor apresentado diz respeito ao tempo médio de processamento do método *Snakes* para cada uma das estradas obtidas, em que $s:S$ denotam a unidade em segundos e centésimos, respectivamente, e n refere-se a entrada correspondente para cada uma das operações, assim, no treinamento da rede SOM o n corresponde a dimensão do mapa de Kohonen, na Semeação à dimensão da imagem

Tabela 4.11 - Medidas de desempenho do método de extração de estradas para os recortes apresentados.

Imagem	Perfeição	Correção	Qualidade	Redundância	Unificação (d_l)
Sintética 1	0,77	0,63	0,71	0,02	0,52
Sintética 2	0,71	0,57	0,64	0,01	0,62
Sintética 3	0,77	0,64	0,71	0,03	0,52
Recorte 1	0,47	0,68	0,58	0,03	0,74
Recorte 2	0,69	0,71	0,70	0,25	0,57
Recorte 3	0,13	0,26	0,21	0,35	1,43
Recorte 4	0,31	0,79	0,60	0,01	0,82
Recorte 5	0,09	0,66	0,47	0,00	1,10

Tabela 4.12 - Tempo de processamento e complexidade dos algoritmos.

	Tempo (s:S)	Variância (s:S)	Complexidade
SOM	00 : 72	00 : 0001	$O(n^2)$
Semeação	04 : 73	00 : 1836	$O(n^2)$
Snakes	04 : 17	00 : 0097	$O(n^3)$
Total	05 : 45 + (04 : 17 * n)	--	$O(n^3)$

e no método *Snakes* ao número de pontos em um segmento de reta. O tempo de processamento apresenta variações entre os experimentos. Para medir o tempo médio de processamento do extrator, foram realizados cinco diferentes experimentos. Ao final, é obtido o tempo médio e a variação entre eles. Qualquer alteração entre os processamentos é identificada na medida de variância que, neste caso, deve ser mínima. Sendo assim, os valores na Tabela 4.12 dizem respeito não só ao tempo médio de processamento dos cinco experimentos, como também às suas respectivas variâncias.

Não foram considerados quaisquer tipos de otimizações no processo, tanto no método de Semeação quanto de *Snakes*. Os valores apresentados na Tabela 4.12 denotam apenas o estado atual da implementação. Enfatiza-se a necessidade de refatorações futuras para a melhora de desempenho e qualidade. Os valores apresentados acima foram obtidos no *software* MATLAB versão 7.8.0 com o método *tic/toc*, em execução sobre uma máquina virtual de sistema operacional Linux Ubuntu 64 *Bits*, versão 10.04 LTS, com a seguinte arquitetura: 60Gb de capacidade de armazenamento, 4Gb de memória *RAM*, virtualizada com Oracle® Virtual Box, versão 4.1.6, em uma máquina hospedeira de sistema operacional Windows 64 *Bits*, versão 7, com

processador Intel® Core™ i7 2.93GHz e memória *RAM* de 8Gb.

5 CONCLUSÕES

Esse trabalho apresentou uma nova metodologia para a extração de estradas em imagens SAR aerotransportadas, utilizando Mapas Auto-Organizáveis como abordagem na sementeação semiautomática, em que os pontos-sementes adquiridos são submetidos a um processo de poda, também proposto neste trabalho, que exclui do processo de extração os pontos-sementes isolados ou indesejáveis. A obtenção dos eixos de simetria das estradas é executada pelo método *Open Snakes*, que corresponde a uma variação do modelo tradicional apresentado por Kass et al. (1987), de modo que as adaptações realizadas permitam a aplicação do método em uma curva aberta e que esta evolua para o centro da estrada, e não para suas fronteiras.

Os resultados indicam eficiência parcial na utilização da rede SOM como ferramenta para o modelo classificador de dados, e portanto, um bom método na sementeação semiautomática de estradas em imagens de radar. O processo de identificação foi realizado sem qualquer pré-processamento na imagem, tais como filtros suavizadores e filtros de realce.

Grande parte dos trabalhos utilizando a sementeação de pontos aborda o método como um processo manual de sementeação, no qual é necessário que o usuário forneça a localização dos pontos próximos à feição de interesse. O processo de sementeação apresentado neste trabalho é realizado de modo semiautomático, de forma que seja necessária a interferência do usuário em parte do processo, na qual sejam fornecidos alguns parâmetros, como os valores de dimensão da janela de varredura, os padrões a serem identificados, o tamanho do salto da janela de leitura, entre outros. Dados esses parâmetros, os pontos-sementes são identificados de forma automática com o auxílio da RNA.

Nota-se pelos resultados finais, que a sementeação influencia diretamente na qualidade final da extração das estradas. Devido à natureza da imagem, a identificação de um determinado padrão torna-se uma tarefa difícil, e a sementeação, quase sempre, produz erros. O método de sementeação apresentado mostrou-se eficiente no caso de estradas com largura de cerca de 10 *pixels*. Em estradas com larguras inferiores a 5 *pixels*, o método não foi capaz de reconhecê-las de maneira eficiente.

O processo de treinamento ótimo de uma RNA pode ser algo incerto. Ora a rede pode aprender a reconhecer os padrões de forma eficiente, ora não. Isso ocorre em

vista dos que inibem a rede de avançar no processo de aprendizado, que, neste caso, diz respeito à mudança dos valores de peso. A eficiência de reconhecimento de uma RNA depende também da quantidade de padrões de treinamento que são apresentados, ou seja, da quantidade de conhecimento que é repassado a ela para o reconhecimento dos mesmos. Embora a RNA tenha sido igualmente treinada para o reconhecimento de estradas com larguras inferiores, nos experimentos realizados nota-se o baixo rendimento na identificação das estradas com esta característica. O decaimento pode ser dado não pela característica do padrão ou pelo aumento do ruído entre as imagens, mas possivelmente pela quantidade reduzida de padrões de treinamento apresentados.

Ao se utilizar dos recursos de Inteligência Artificial, o método de semente torna-se dependente dos padrões inicialmente apresentados. Sua robustez é dada pela quantidade de referências que são lhe apresentadas. Assim, a limitação dos padrões de treinamento apresentados à RNA minimizou sua capacidade em reconhecer as diferentes variações dos perfis de estradas.

Mesmo havendo a presença de estradas, o método de semente, muitas vezes, não as identifica em alguns trechos, que são denominados como *gaps* de processamento. Em consequência disso, as curvas iniciais geradas posteriormente pelo método *Snakes* são interpoladas de forma independente, de modo que várias curvas sejam convergidas para uma mesma estrada em trechos diferentes, tal como mostraram os resultados. Uma solução para a junção desses segmentos de reta é sugerida na seção seguinte.

Em face das características da imagem SAR utilizada nos experimentos, foram identificados muitos pontos espúrios, ocasionando erros de comissão em grande quantidade. Uma solução para isto foi a criação do algoritmo de poda, que viabilizou a escolha dos melhores pontos entre aqueles identificados, eliminando os demais. Com isso, muitos dos pontos espúrios são eliminados e, conseqüentemente, são selecionados hipóteses mais concisas de segmentos retos.

Em se tratando de um método robusto para a identificação de eixos de simetria de estradas, o método *Snakes* apresenta, contudo, algumas limitações. Uma das mais importantes é a falta de flexibilidade e a impossibilidade de alteração de sua topologia quanto da inicialização. O êxito na obtenção do eixo de simetria de uma determinada estrada é dado pela localização da curva inicial próxima à região de interesse. Este requisito é indispensável para que o método seja bem sucedido. Devido à presença

de ruídos não pertencentes às estradas, poderá ocorrer a inibição da convergência da curva para um mínimo local de energia, o que ocorre no experimento 2 (Figura 4.10).

Há inúmeros trabalhos na literatura que abordam o uso do método *Snakes* para a extração de estradas. Muitos deles apresentam a teoria quanto ao uso do método tradicional (curva fechada), sendo que os detalhes envolvendo a implementação do método de curva aberta, quase sempre, são omitidos. Desta forma, a escassez de informação envolvendo o uso do método *Snakes* para esta solução específica foi um obstáculo a ser superado. De maneira geral, nota-se na literatura envolvendo a extração de estradas que grande parte dos trabalhos que apresentou o fator de qualidade elevado consistia em aplicações com imagens óticas. A extração de estradas em imagens de radar carece de métodos automáticos e que apresentem, ao mesmo tempo, qualidade satisfatória. Embora no imageamento por radar haja muitas pesquisas sobre a extração de estradas, são poucos os trabalhos que utilizaram ambos os conceitos utilizados nesta pesquisa.

A abordagem semiautomática proposta neste trabalho, mostrou-se parcialmente eficiente na extração de estradas em imagens SAR. A largura da estrada e o ruído inerente ao imageamento por radar, contribuíram diretamente no desempenho do método de sementeção. A aplicação do método *Snakes* nas imagens sintéticas na Seção 4.3, mostrou que, ao fornecer pontos-sementes cuja as localidades sejam próximas às regiões de interesse, a conversão da curva para um mínimo local cuja localização seja o eixo de simetria da via, é quase sempre certo. A presença de rios, lagos ou feições com perfis de larguras semelhantes às de uma estrada fizeram com que o método os identificassem como tal. Embora tenha apresentado esta deficiência, o método, de modo geral apresentou-se satisfatório.

Na próxima seção, são apontadas possíveis implementações que possam, futuramente, tornar o método de extração ainda mais robusto e eficaz quanto à sementeção, remoção de pontos espúrios, generalização das regiões de reconhecimento e combinação de segmentos descontínuos.

5.1 Trabalhos futuros

É desejável que a pesquisa realizada neste trabalho tenha prosseguimento e que as atividades executadas até então possam ser refinadas, de modo a produzir resultados

com acurácia superior em relação às apresentadas na seção anterior. Desta forma, nesta seção, serão apresentadas algumas dessas possíveis melhorias.

Os segmentos finais obtidos após o processamento do método *Snakes* apresentaram descontinuidades, justamente em vista da forma como os pontos-sementes foram interpolados. Os pontos-sementes são identificados sequencialmente, de acordo os índices das amostras provenientes da leitura na imagem, tal como ilustrado na Figura 3.9. A título de exemplo, ter-se-iam dois segmentos de reta, cujos os índices de pontos-sementes seriam 1 a 5, e 13 a 17, respectivamente. Em uma imagem com nove amostras por linha, estes segmentos de reta seriam praticamente paralelos, porém, identificados de forma independente e aplicados no método *Snakes*. Assim, quando esses segmentos não estivessem necessariamente paralelos, o resultado da extração tenderia a apresentar tais descontinuidades. A utilização do método *Perceptual Grouping* seria uma estratégia viável para melhorar a caracterização da rede de estradas, tais como a exclusão de segmentos isolados, a combinação de segmentos descontínuos ou o alinhamento destes (DELL'ACQUA et al., 2002; GAMBA et al., 2006; MOHAN; NEVATIA, 1992; HU; TAO, 2007; AMINI et al., 2002).

As cenas analisadas nos recortes de Paragominas-PA, referem-se à regiões não-urbanizadas, cujas as feições lineares são relativamente fáceis de serem identificadas aos olhos humanos, porém, difíceis de serem representadas computacionalmente, considerando que todas as características geométricas, espectrais e possíveis ruídos devem ser levados em conta. Por outro lado, é desejável que o método seja aplicado também em áreas urbanas, em que as feições aparecem mais estreitas e alinhadas, tornando o procedimento um pouco mais complexo. Em se tratando de imagem da banda P, as estradas são mais evidentes do que em uma imagem na banda X, por exemplo. A generalização do método para a identificação de estradas em qualquer área ou região é uma tarefa que exige cautela e um estudo apurado sobre as características da imagem e, portanto, não foi inserida no escopo deste trabalho. Como implementações futuras, é desejável a investigação desse tópico.

Fica de responsabilidade do usuário a mudança do sinal em w_{line} , de modo que a curva no processo de minimização da energia seja atraída para os centros de feições claras ou escuras, tal como detalhado na Seção 3.3.3.5. Desta forma, é desejável a extensão do modelo matemático que considera tanto a evolução da curva para perfis claros quanto escuros, sem a necessidade da preinterpretação da imagem por método humano.

REFERÊNCIAS BIBLIOGRÁFICAS

AGOURIS, P.; DOUCETTE, P.; STEFANIDIS, A. **Automation and digital photogrammetric workstations**. Manual of photogrammetry: Fifth. [S.l.], 2004. Chapter 12. 9

AGOURIS, P.; STEFANIDIS, A.; GYFTAKIS, S. Differential snakes for change detection in road segments. **Photogrammetric Engineering and Remote Sensing**, v. 67, n. 12, p. 1391–1399, dez. 2001. ISSN 00991112. 2, 20, 46, 49

AMBERG, V.; COULON, M.; MARTON, P.; SPIGAI, M. Structure extraction from high resolution SAR data on urban areas. In: **Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International**. [S.l.: s.n.], 2004. v. 3, p. 1784–1787 vol.3. 13

AMINI, A.; WEYMOUTH, T.; JAIN, R. Using dynamic programming for solving variational problems in vision. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 12, n. 9, p. 855–867, sep 1990. ISSN 0162-8828. 4, 44, 49, 50

AMINI, J.; SARADJIAN, M.; BLAIS, J.; LUCAS, C.; AZIZI, A. Automatic road-side extraction from large scale imagemaps. **International Journal of Applied Earth Observation and Geoinformation**, v. 4, n. 2, p. 95–107, 2002. ISSN 0303-2434. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0303243402000041>>. 16, 88

BAJCSY, R.; TAVAKOLI, M. Computer recognition of roads from satellite pictures. **IEEE Systems, Man, and Cybernetics Society**, p. 623–637, set. 1976. ISSN 0018-9472. 2

BARROW, H. G.; TENENBAUM, J. M. **Recovering intrinsic scene characteristics from images**. [S.l.]: Academic Press, 1978. 3–26 p. 16

BAUMGARTNER, A.; HINZ, S.; WIEDEMANN, C. Efficient methods and interfaces for road tracking. **International Archives of Photogrammetry and Remote Sensing**, v. 34, p. 28–31, 2002. 3

BAUMGARTNER, A.; STEGER, C.; MAYER, H.; ECKSTEIN, W. Multi-resolution, semantic objects, and context for road extraction. In: **In**

Semantic Modeling for the Acquisition of Topographic Information from Images and Maps. [S.l.]: Birkhauser Verlag, 1997. p. 140–156. 12

BAUMGARTNER, A.; STEGER, C.; MAYER, H.; ECKSTEIN, W.; EBNER, H. Automatic road extraction based on multi-scale, grouping, and context. **Photogrammetric Engineering and Remote Sensing**, v. 55, n. 7, p. 777–785, jul. 1999. 22

BELLMAN, R. **The theory of dynamic programming.** [S.l.]: The RAND Corporation, 1954. 19

BELLMAN, R.; DREYFUS, S. **Applied dynamic programming.** Santa Monica: The RAND Corporation, 1962. R-352-PR. 4, 49

BENTABET, L.; JODOUIN, S.; ZIOU, D.; VAILLANCOURT, J. Road vectors update using sar imagery: a snake-based method. **Geoscience and Remote Sensing, IEEE Transactions on**, v. 41, n. 8, p. 1785–1803, aug. 2003. ISSN 0196-2892. 20, 21, 46, 49, 50

BOVIK, A. On detecting edges in speckle imagery. **Acoustics, Speech and Signal Processing, IEEE Transactions on**, v. 36, n. 10, p. 1618–1627, oct 1988. ISSN 0096-3518. 21

BRESENHAM, J. E. Algorithm for computer control of a digital plotter. **IBM Systems Journal**, v. 4, n. 1, p. 25–30, 1965. 38, 132

CARRARA, W. G.; GOODMAN, R. S.; MAJEWSKI, R. M. **Spotlight Synthetic Aperture Radar: Signal Processing Algorithms.** [S.l.]: Artech House Publishers, 1995. ISBN 0890067287. 26

CHANUSSOT, J.; MAURIS, G.; LAMBERT, P. Fuzzy fusion techniques for linear features detection in multitemporal sar images. **Geoscience and Remote Sensing, IEEE Transactions on**, v. 37, n. 3, p. 1292–1305, may 1999. ISSN 0196-2892. 14, 15

COHEN, L. D. On active contour models and balloons. **CVGIP: Image Underst.**, Academic Press, Inc., Orlando, FL, USA, v. 53, n. 2, p. 211–218, mar. 1991. ISSN 1049-9660. Disponível em:
<[http://dx.doi.org/10.1016/1049-9660\(91\)90028-N](http://dx.doi.org/10.1016/1049-9660(91)90028-N)>. 18, 44

- COOTES, T. F.; EDWARDS, G.; TAYLOR, C. Comparing active shape models with active appearance models. In: **in Proc. British Machine Vision Conf.** [S.l.]: BMVA Press, 1999. p. 173–182. 18, 44
- COSTA, R.; MEDEIROS, F. Linear features detection in SAR images for urban analysis. In: **Computer Graphics and Image Processing, 2002. Proceedings. XV Brazilian Symposium on.** [S.l.: s.n.], 2002. p. 401. ISSN 1530-1834. 21
- COURANT, R.; HILBERT, D. **Methods of Mathematical Physics.** [S.l.]: Wiley, 1989. 51
- CURLANDER, J. C.; MCDONOUGH, R. N. **Synthetic Aperture Radar: Systems and Signal Processing.** [S.l.]: Wiley-Interscience, 1991. ISBN 047185770X. 27
- DELL'ACQUA, F.; GAMBA, P. Detection of urban structures in SAR images by robust fuzzy clustering algorithms: the example of street tracking. **Geoscience and Remote Sensing, IEEE Transactions on**, v. 39, n. 10, p. 2287–2297, Oct 2001. ISSN 0196-2892. 13
- DELL'ACQUA, F.; GAMBA, P.; LISINI, G. Road extraction aided by adaptive directional filtering and template matching. **Proc. of IEEE Int. Geoscience and Remote Sensing Symposium**, 2002. 88
- DOUCETTE, P.; AGOURIS, P.; STEFANIDIS, A. Automated road extraction from high resolution multispectral imagery. **Photogrammetric Engineering and Remote Sensing**, v. 70, n. 12, p. 1405–1416, Dec 2004. 2, 3, 7, 8, 9
- DOUCETTE, P.; AGOURIS, P.; STEFANIDIS, A.; MUSAVI, M. Self-organised clustering for road extraction in classified imagery. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 55, p. 347–358, Feb 2001. 12
- FAZAN, A. J.; MENDES, T. S. G.; POZ, A. P. D. Extração semi-automática de feições em imagens digitais usando a estratégia de teste ativo. **Congresso Brasileiro de Cartografia**, XXI, 2003. 10
- FINKEL, R. A.; BENTLEY, J. L.; FINKEL, R. A.; BENTLEY, J. L. Quad trees a data structure for retrieval on composite keys. **Acta Informatica**, Springer Berlin / Heidelberg, v. 4, n. 1, p. 1–9, mar. 1974. ISSN 0001-5903. Disponível em: <<http://dx.doi.org/10.1007/BF00288933>>. 17

FROST, V. S.; STILES, J. A.; SHANMUGAN, K. S.; HOLTZMAN, J. C. A model for radar images and its application to adaptive digital filtering of multiplicative noise. **Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-4**, n. 2, p. 157–166, march 1982. ISSN 0162-8828. 20

FUA, P.; LECLERC, Y. Model driven edge detection. **Machine Vision and Applications, III**, p. 45–56, 1990. 50

GAMBA, P.; DELL'ACQUA, F.; LISINI, G. Improving urban road extraction in high-resolution images exploiting directional filtering, perceptual grouping, and simple topological concepts. **IEEE Geoscience and Remote Sensing Letters**, v. 3, n. 3, p. 387–391, jul. 2006. 88

GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2008. ISBN 013168728X. 14, 16

GROTE, A.; HEIPKE, C.; ROTTENSTEINER, F.; MEYER, H. Road extraction in suburban areas by region-based road subgraph extraction and evaluation. In: **Urban Remote Sensing Event, 2009 Joint**. [S.l.: s.n.], 2009. p. 1–6. 17

GRUEN, A.; LI, H. Linear feature extraction with LSB-snakes from multiple images. **International Archives of Photogrammetry and Remote Sensing**, v. 31, p. 266–272, 1996. 10

_____. Semi-automatic linear feature extraction by dynamic programming and LSB-snakes. **Photogrammetric Engineering and Remote Sensing**, v. 63, n. 8, p. 985–995, 1997. ISSN 00991112. xv, 3, 7, 19, 27, 30, 49, 50

HARALICK, R. Statistical and structural approaches to texture. **Proceedings of the IEEE**, v. 67, n. 5, p. 786–804, may 1979. ISSN 0018-9219. 13

HARVEY, W. A. Performance evaluation for road extraction. In: **Proc. International Workshop on 3D Geospatial Data Production: Meeting Application requirements, ISPRS WG II/6**. Paris, France: [s.n.], 1999. p. 175–185. Disponível em: <<http://www.cs.cmu.edu/~wah/pubs/isprs99.pdf>>. 4, 8, 55

HAUPTFLEISCH, A. C. **Automatic road network extraction from high resolution satellite imagery using spectral classification methods**. Dissertação (Mestrado) — University of Pretoria, fev. 2010. 2, 11, 14, 50, 55

HAYKIN, S. **Neural Network**: a comprehensive foundation. [S.l.]: Macmillan College Publishing Company, 1994. ISBN 002352761-7. 32, 35, 36

HEIPKE, C.; MAYER, H.; WIEDEMANN, C.; JAMET, O. Evaluation of automatic road extraction. In: **In: International Archives of Photogrammetry and Remote Sensing**. [S.l.: s.n.], 1997. p. 47–56. 55

HOFFMAN, J. D. **Numerical Methods for Engineers and Scientists**. 2. ed. [S.l.]: CRC Press, 2001. ISBN 0824704436. 48

HU, J.; RAZDAN, A.; FEMIANI, J.; CUI, M.; WONKA, P. Road network extraction and intersection detection from aerial images by tracking road footprints. **Geoscience and Remote Sensing, IEEE Transactions on**, v. 45, n. 12, p. 4144–4157, dec. 2007. ISSN 0196-2892. 3, 8

HU, X.; TAO, C. V.; HU, Y. Automatic road extraction from dense urban area by integrated processing of high resolution imagery and LIDAR data. **International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences**, v. 35 (Part B3), p. 288–292, 2004. 13

HU, X.; TAO, V. Automatic extraction of main road centerlines from high resolution satellite imagery using hierarchical grouping. **Photogrammetric Engineering & Remote Sensing**, v. 73, n. 9, p. 1049–1056, set. 2007. 88

IDBRAIM, S.; MAMMASS, D.; ABOUTAJDINE, D.; DUCROT, D. An automatic system for urban road extraction from satellite and aerial images. **WSEAS Transaction Signal Processing**, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, v. 4, n. 10, p. 563–572, Oct 2008. ISSN 1790-5022. Disponível em: <<http://dl.acm.org/citation.cfm?id=1486759.1486760>>. 14

KASS, M.; WITKIN, A.; TERZOPOULOS, D. Snakes: Active contour models. **International Journal of Computer Vision**, v. 1, n. 4, p. 321–331, 1987. 3, 4, 17, 43, 44, 45, 46, 47, 48, 50, 52, 85

KOHONEN, T. Self-organized formation of topologically correct feature maps. **Biological Cybernetics**, v. 43, p. 59–69, 1982. 4, 32

LAPTEV, I.; MAYER, H.; LINDBERG, T.; ECKSTEIN, W.; STEGER, C.; BAUMGARTNER, A. Automatic extraction of roads from aerial images based on

scale space and snakes. **Machine Vision and Application**, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 12, n. 1, p. 23–31, jul. 2000. ISSN 0932-8092.

Disponível em: <<http://dx.doi.org/10.1007/s001380050004>>. 3, 50

LEE, J.-S. Digital image enhancement and noise filtering by use of local statistics. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, PAMI-2, n. 2, p. 165–168, march 1980. ISSN 0162-8828. 20

LI, H. **Semi-automatic road extraction from satellite and aerial images**. Tese (Doutorado) — Institute of Geodesy and Photogrammetry, ETH-Zurich, Switzerland, 1997. 4, 49, 50

LISINI, G.; TISON, C.; TUPIN, F.; GAMBÀ, P. Feature fusion to improve road network extraction in high-resolution SAR images. **Geoscience and Remote Sensing Letters, IEEE**, v. 3, n. 2, p. 217–221, Apr 2006. ISSN 1545-598X. 21

MAYER, H.; LAPTEV, I.; BAUMGARTNER, A.; STEGER, C. Automatic road extraction based on multi-scale modeling, context, and snakes. **International Archives of Photogrammetry and Remote Sensing**, v. 32, p. 106–113, 1997. 3, 19, 20

MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 115–133, 1943. 31

MENA, J. State of the art on automatic road extraction for GIS update: a novel classification. **Pattern Recognition Letters**, v. 24, p. 3037–3058, 2003. ISSN 0167-8655. 23

MENA, J. B.; MALPICA, J. A. An automatic method for road extraction in rural and semi-urban areas starting from high resolution multi-spectral satellite imagery. **Pattern Recognition Letters**, v. 26, p. 1201–1220, 2005. 13, 16

MOHAN, R.; NEVATIA, R. Perceptual organization for scene segmentation and description. **IEEE Trans. Pattern Anal. and Machine Intel.**, v. 14, n. 6, p. 616–635, jun. 1992. 88

OLIVER, C.; QUEGAN, S. **Understanding Synthetic Aperture Radar Images**. Boston: Artech House Publishers, 1998. ISBN 089006850X. 25, 26

ORBISAT. Orbisat da amazônia ind. e aerolevantamento s.a. Imagem de radar de abertura sintética aerotransportada da cidade de Paragominas (PA) e entorno 2007. Sensor OrbiSAR. Aquisição entre 11 fev. 2007 e 13 mar. 2007. 2010. 63, 64, 65, 66

OSHER, S.; SETHIAN, J. A. Fronts propagation with curvature dependent speed: algorithms based on hamilton-jacobi formulations. **Journal of Computational Physics**, v. 79, p. 12–49, 1988. 18, 44

POZ, A. P. D.; AGOURIS, P. Um algoritmo de otimização global para a extração de rodovias em imagens digitais. **Revista Brasileira de Cartografia**, n. 53, p. 65–72, Dec 2001. 2, 9

POZ, A. P. D.; GALLIS, R. A.; SILVA, J. F. C. da. Semi-automatic road extraction by Dynamic Programming optimisation in the object space: single image case. In: **Symposium of ISPRS Commission III - Photogrammetric Computer Vision**. [S.l.]: The International Archives the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2006. v. 36, n. XXX, p. 215–220. 49, 50

POZ, A. P. D.; SILVA, M. A. O. Active testing and edge analysis for road centerline extraction. In: **ISPRS - Symposium on Photogrammetric Computer Vision**. Áustria: [s.n.], 2002. v. 34, p. 44–47. 10

QUAM, L. Road tracking and anomaly detection in aerial imagery. In: . [S.l.]: Image Understanding Workshop, 1978. p. 51–55. 2

ROSENBLATT, F. The Perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, p. 386–408, 1958. 31

_____. **Principles of Neurodynamics**. Washington, DC: Spartan Books, 1962. 31

SCOFIELD, G. B.; SANT'ANNA, S. J. S.; FREITAS, C. C.; DUTRA, L. V. Avaliação quantitativa do SegSAR através de medidas de borda e regiões em imagens ópticas sintéticas. In: **Anais XIII Simpósio Brasileiro de Sensoriamento Remoto**. [S.l.: s.n.], 2007. p. 6167–6174. 59, 81

SILVA, J. S.; SANTOS, B. S.; SILVA, A.; MADEIRA, J. Modelos deformáveis na segmentação de imagens médicas: uma introdução. **Revista do DETUA**, v. 4, n. 3, Sep 2004. 18, 44

- SILVA, W. B.; FREITAS, C. C.; SANT'ANNA, S. J. S.; FRERY, A. C. PolSAR region classification by stochastic distance minimization between wishart distributions. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, 2012. 59, 60
- SINGH, A.; GOLDFOF, D.; TERZOPOULOS, D. (Ed.). **Deformable models in medical image analysis**. Alamitos, CA: IEEE Press, 1998. 43
- SPITZER, F. **Random fields and interacting particle systems**. M. a. a. summer seminar notes. [S.l.: s.n.], 1971. 21
- STAIB, L. H.; DUNCAN, J. S. **Boundary Finding with Parametrically Deformable Models**. 1992. 18, 44
- TUPIN, F.; MAÎTRE, H.; MANGIN, J.-F.; NICOLAS, J.-M.; PECHERSKY, E. Detection of linear features in sar images: Application to road network extraction. **IEEE Transactions on Geoscience and Remote Sensing**, v. 36, n. 2, p. 434–453, Mar 1998. 7, 21, 23
- WIEDEMANN, C.; HEIPKE, C.; MAYER, H.; JAMET, O. Empirical evaluation of automatically extracted road axes. In: **Empirical Evaluation Techniques in Computer Vision**. [S.l.: s.n.], 1998. p. 172–187. 55, 56, 58, 66
- XIAO, Z. Q.; BAO, G. S.; JIANG, X. Q. Road network extraction in classified sar images using genetic algorithm. **Journal of Central South University of Technology**, Central South University, co-published with Springer, v. 11, p. 180–184, 2004. ISSN 1005-9784. Disponível em: <<http://dx.doi.org/10.1007/s11771-004-0038-x>>. 22
- XU, C.; PRINCE, J. Gradient vector flow: a new external force for snakes. In: **Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on**. [S.l.: s.n.], 1997. p. 66–71. ISSN 1063-6919. 18, 44
- YOUN, J.; BETHEL, J. S. Adaptive snakes for urban road extraction. In: **In: International Archives of Photogrammetry and Remote Sensing**. [S.l.: s.n.], 2004. p. 465–470. 49
- ZHANG, C.; MURAI, S.; BALTSAVIAS, E. Road network detection by mathematical morphology. In: **Proc. ISPRS Workshop “3D Geospatial Data**

Production: Meeting Application Requirements". [S.l.: s.n.], 1999. p. 185–200. 15

ZHAO, H.; KUMAGAI, J.; NAKAGAWA, M.; SHIBASAKI, R. Semi-automatic road extraction from high-resolution satellite image. **International Society for Photogrammetry and Remote Sensing**, XXXIV, n. 3, p. A–406 ff, set. 2002. 3, 9, 10

ZHOU, G.; CUI, Y.; CHEN, Y.; YANG, J.; RASHVAND, H.; YAMAGUCHI, Y. Linear feature detection in polarimetric SAR images. **IEEE Transactions on Geoscience and Remote Sensing**, v. 49, n. 4, p. 1453–1463, Apr 2011. ISSN 0196-2892. 22

ZIOU, D. Finding lines in grey-level images. **In Processing of International Conference Pattern Recognition**, p. 534–537, 2000. 20

APÊNDICE A

Neste capítulo são apresentadas todas as rotinas de implementação construídas para a realização deste trabalho. Essas foram realizadas no ambiente de desenvolvimento MATLAB versão 7.8.0.347, e os experimentos realizados sobre imagens sintéticas de radar e recortes da região de Paragominas-PA com extensões *tiff*, e os resultados finais mostrados no padrão *jpg*.

A.1 Fluxo principal de execução do código

Os códigos-fonte são apresentados de acordo com o fluxo de execução do Código-Fonte A.1. As demais rotinas de implementação, utilizadas para o auxílio de operações básicas, tais como: conversão de imagem, algoritmo de Bresenham e validação, são apresentadas na Seção A.2.

A.1.1 Principal

É por meio do Código-Fonte A.1 que o processo de extração se inicia, nesse são inseridas as chamadas para cada um dos processos ilustrados na Figura 3.3. Ao final, é obtida a matriz com a localização das estradas presentes na imagem, sendo que cada segmento de reta é agregado a uma linha dessa matriz.

Código-Fonte A.1 - Código de execução principal do método de extração de estradas

```
1 function SSS(saidas , taxa , epocas , opcao , loadFile)
2
3     % – TREINAMENTO DA RNA
4
5     % Carrega os padroes de treinamento
6     % arg: 0 – Nao carrega dados do sistema de arquivo
7     %       1 – Carrega dado do sistema de arquivo (raiz do projeto)
8     % retorno: Vetores de entradas X bruta e normalizada ,
9     %           respectivamente. [matriz – Cada linha um padrao de
10            treinamento]
11
12    [xTrain xTrain_norm] = carregaPadroes(0);
13
14    NumEntradas = size(xTrain_norm,1);
15    JanelaVarredura = NumEntradas^2;
16
17    % Inicializa os pesos de treinamento
18    % arg: 0 – Nao carrega dados do sistema de arquivo
```

```

17 % 1 - Carrega dado do sistema de arquivo (raiz do projeto)
18 % retorno: Matrizes de pesos W bruta e normalizada [matriz]
19 [wk wk_norm] = carregaPesos(loadFile , NumEntradas, saidas);
20
21 % Treina a rede neural
22 % arg 1: Matriz de entrada normalizada [matriz]
23 % arg 2: Matriz de pesos normalizada [matriz]
24 % arg 3: Dimensao do mapa de neuronios [escalar]
25 % arg 4: Taxa de aprendizagem da rede [escalar]
26 % arg 5: Numero maximo de epocas [escalar]
27 % retorno: Mapa final de pesos apos o treinamento [matriz]
28 [wk_final] = treinaSOM(xTrain_norm, wk_norm, saidas, taxa, epocas);
29
30 % Obtem a matriz unitaria (norma dos pesos)
31 % arg: Matriz W, matriz final de pesos [matriz]
32 % retorno: Matriz U [matriz]
33 [matrizU] = matrizUnitaria(wk_final, pathResult);
34
35 % - SEMEACAO

```

```

36
37 % Carrega a imagem
38 % retorno 1: Imagem em niveis de cinza [matriz]
39 % retorno 2: Caminho da imagem
40 [I file] = carregaImagem;
41
42 % Realiza a leitura das amostras
43 % arg 1: Imagem [matriz]
44 % arg 2: Dimensao da janela de leitura [escalar]
45 % arg 3: Tamanho do salto [escalar]
46 % retorno: Todas as amostras lidas, dispostas em vetores de dados
    e
47 % os dados normalizados correspondentes [matrizes]
48 [xAmostra xAmostra_norm] = carregaAmostrasEsp(I, JanelaVarredura,
    esp);
49
50 % Localiza os centros dos clusters
51 % arg 1: Padroes de treinamento [matriz]
52 % arg 2: Matriz de pesos finais [matriz]
53 % arg 3: Dimensao do mapa de neuronios [escalar]
54 % arg 4: Matriz unitaria dos pesos [escalar]
55 % retorno: Localizacao do centro de cada cluster [matriz]
56 [centros] = localizaCluster(xTrain_norm, wk_final, saidas, matrizU)

```

```

57
58 % Realiza a Semeacao
59 % arg 1: Vetores de amostras normalizados [matriz]
60 % arg 2: Matriz de pesos normalizados [matriz]
61 % arg 3: Matriz contendo as informacoes de localizacao, raio e
        raio
62 %           de pertinencia de cada aglomerado [matriz]
63 % arg 4: Dimensao do mapa de neuronios [escalar]
64 % retorno: Vetor de indices referentes as amostras que foram
        apontadas
65 %           como ponto semente [vetor]
66 [pontosSementes] = semearImagem(xAmostra_norm, wk_final, centros,
        saidas);
67
68 % Modifica extensao .tif para .jpg
69 file = regexprep(file, '.tif', '.jpg', 'ignorecase');
70
71 fprintf('Pontos identificados: %d\n', size(nonzeros(pontosSementes)
        ));
72 fprintf('Pintando os pontos na imagem...\n');
73
74 % Pinta os pontos na imagem
75 % arg 1: Caminho da imagem
76 % arg 2: Vetor contendo os indices dos padroes da imagem
77 % arg 3: Dimensao da janela de leitura dos padroes
78 % arg 4: Se a opcao igual a 0, o resultado nao e exibido com imagem
        no
79 %           fundo. Caso contrario, sim
80 % arg 5: Sequencia da figura a ser plotada
81 % arg 6: Caminho caso seja necessario armazenar a imagem
82 % arg 7: Espacamento da janela de leitura a cada iteracao da
        varredura
83 % retorno: Quantidade de padroes por linha
84 [pad_lin] = pintaPontosImagemEsp(file, pontosSementes, NumEntradas,
        0, 2, pathResult, esp);
85
86 fprintf('Obtendo os segmentos - (Prunning)...\n');
87
88 % Poda dos pontos espurios na imagem
89 % arg 1: Indices das amostras que foram marcadas [vetor]
90 % arg 2: Quantidade de padroes por linha [escalar]
91 % arg 3: Parametro de pertinencia para a continuidade dos pontos

```

```

92     % arg 4: Parametro de pertinencia para a descontinuidade dos pontos
93     % retorno: Variavel pontosSementes filtrada [vetor]
94     [pontosSementesPrunning] = segmenta_estradas(pontosSementes ,
          pad_lin , 4, 1);
95
96     % Pinta os pontos na imagem, com a poda ja realizada.
97     % arg 1: Caminho da imagem [string]
98     % arg 2: Pontos sementes filtrado [vetor]
99     % arg 3: Dimensao da janela de leitura dos padroes [escalar]
100    % arg 4: Se a opcao igual a 0, o resultado nao e exibido com imagem
          no
101    %         fundo. Caso contrario, sim
102    % arg 5: Sequencia da figura a ser plotada
103    % arg 6: Caminho caso seja necessario armazenar a imagem [string]
104    % arg 7: Tamanho do salto [escalar]
105    % retorno: Matriz com a localizacao dos pixels referentes aos
          pontos
106    %         marcados [matriz]
107    [xyPontosSementes] = pintaPontosImagemEsp_seg(file ,
          pontosSementesPrunning , NumEntradas , 0, 3, pathResult , esp);
108
109    % - SNAKES

```

```

110
111    % Aplica o metodo Snakes em cada um dos segmentos de reta
112    % arg 1: Matriz com a localizacao dos pixels referentes aos pontos
113    %         marcados [matriz]
114    % arg 2: Caminho da imagem [string]
115    % arg 3: Sigma - Fator de suavizacao gaussiana [escalar]
116    % arg 4: Alpha - Especifica a elasticidade da Snake [escalar]
117    % arg 5: Beta - Especifica a rigidez do contorno [escalar]
118    % arg 6: Gamma - Especifica o tamanho do passo [escalar]
119    % arg 7: Kappa - Age como um fator de escala para o termo de
          energia
120    % arg 8: wl - Fator de pesagem para intensidade baseada no termo de
121    %         potencial [escalar]
122    % arg 9: we - Fator de pesagem para borda baseada no termo de
123    %         potencial [escalar]
124    % arg 10: wt - Fator de pesagem para terminacao baseada no termo de
125    %         potencial [escalar]
126    % arg 11: Numero de iteracao em cada processo de minimizacao [
          escalar]
127    snakes(xyPontosSementes , file , 2.0, 0.05, 0, 1.00, 0.3, 0.3, 0.4,

```



```

        0.7, 500, 0);
128
129     % Fim do processo
130     fprintf('Processo Concluido em %0.2f minutos!\n', (toc(t3)/60));
131     fprintf('-----\n');
132
133 end

```

A.1.2 Carregamento dos padrões de treinamento

O Código-Fonte A.2 possui como principal função o carregamento dos oito padrões de treinamento. Cada um dos padrões constituem-se de valores em níveis de cinza (entre 0 a 255), utilizados para o treinamento da RNA.

Código-Fonte A.2 - Código que obtém a matriz de padrões de treinamento

```

1 function [xTrain xTrain_norm] = carregaPadroes(loadFile)
2
3     if(loadFile == 0)
4         xTrain(:,1) = [158 125 87 1 1 1 1 1 1 1 1 1 1 1 1 1 1 158 166
5                        184];
6         xTrain(:,2) = [186 163 168 191 199 186 176 204 1 1 1 184 212
7                        204 201 194 204 212 201 212 207];
8         xTrain(:,3) = [158 125 87 255 255 255 255 255 255 255 255 255
9                        255 255 255 255 255 158 166 184];
10        xTrain(:,4) = [186 163 168 191 199 186 176 204 255 255 255 184
11                       212 204 201 194 204 212 201 212 207];
12        xTrain(:,5) = [186 186 199 173 176 168 171 207 196 201 181 186
13                       176 194 196 189 201 186 209 196 186];
14        xTrain(:,6) = [245 247 237 250 255 245 237 230 235 247 237 232
15                       232 240 250 247 245 250 242 250 237];
16        xTrain(:,7) = [255 255 255 255 255 255 255 255 255 255 255 255
17                       255 255 255 255 255 255 255 255];
18        xTrain(:,8) = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
19
20        % Normalizado Dados de Treinamento
21        xTrain_norm = xTrain/norm(xTrain);
22    else
23        fprintf('Carregando xTrain...\n');
24        load 'xTrain.mat'
25        load 'xTrain_norm.mat'
26    end

```

```
21     return
22 end
```

A.1.3 Carregamento dos pesos da RNA

A rotina responsável por gerar os valores iniciais dos pesos de conexão é descrita no Código-Fonte A.3, em que, inicialmente, são gerados valores aleatórios que são posteriormente normalizados.

Código-Fonte A.3 - Código que carrega a matriz de pesos de conexões da RNA

```
1 function [wk_final wk_norm] = carregaPesos(loadFile , saidas , saidas)
2
3     if(loadFile == 0)
4         wk_final = rand(saidas ,saidas ,entradas);
5     else
6         fprintf('Carregando pesos da rede SOM...\n');
7         load 'wk.mat';
8     end
9
10    SomaVets = zeros(saidas);
11    wk_norm = wk_final;
12
13    for i=1:entradas
14        SomaVets = SomaVets + wk_norm(:, :, i);
15    end
16
17    for i=1:entradas
18        wk_norm(:, :, i) = wk_norm(:, :, i) ./ SomaVets;
19    end
20
21    return
22
23 end
```

A.1.4 Treinamento da RNA

Todas as rotinas envolvendo o treinamento da RNA do tipo SOM é apresentado no código abaixo, sendo apenas o ajuste dos pesos realizado pelo Código-Fonte A.5. Ao final do Código-Fonte A.4 é obtido a matriz de pesos com os seus valores já ajustados, ou seja, a rede está pronta para o reconhecimento dos padrões apresentados.

Código-Fonte A.4 - Código que realiza o treinamento da RNA do tipo SOM

```

1 function [wk_norm] = treinaSOM(xTrain_norm, wk_norm, saidas, taxa,
    epocas)
2
3     fprintf('Treinando a rede SOM...\n');
4     t = tic;
5
6     % Definir o mapa topologico
7     raioMap0 = saidas - (saidas * 0.3);
8     raioMap = raioMap0;
9     eta = taxa;
10    eta0 = eta;
11    NumEpMax = epocas;
12    lambda = NumEpMax/log(raioMap);
13
14    for ep = 1:NumEpMax
15        p = ceil(rand(1) * size(xTrain_norm,2));
16        raioViz = floor(raioMap * exp(-(ep/lambda)));
17
18        % Calcular as distancias euclidianas entre os vetores de
19        % entrada e
20        % de pesos para cada neuronio
21        d = zeros(saidas);
22
23        for x=1:saidas
24            for y=1:saidas
25                %err(x,y) = sum(wk_norm(x,y,:).^2);
26                a = wk_norm(x,y,:);
27                b = xTrain_norm(:,p);
28                d(x,y) = sum((a(:) - b(:)).^2);
29            end
30        end
31
32        % Encontrar o vencedor
33        [vec_l vec_c] = find(d == min(min(d)));
34        l = vec_l(1);
35        c = vec_c(1);
36
37        % Ajusta os pesos do vencedor e dos vizinhos
38        wk_norm = ajustarPesos(xTrain_norm, wk_norm, raioViz, l, c, eta
39        , saidas, p);
40
41        % Ajustar taxa de aprendizagem e influencia da Gaussiana

```

```

40         eta = eta0 * exp(-(ep/NumEpMax));
41
42     end
43
44     fprintf('Treinamento concluido em %0.2f minutos!\n', (toc(t)/60));
45     return
46
47 end

```

A.1.4.1 Ajuste dos pesos da RNA

O Código-Fonte A.5 ajusta os valores dos pesos conforme os parâmetros de localização do neurônio vencedor, taxa de ajuste, influência de seus vizinhos e o raio de abrangência desses vizinhos.

Código-Fonte A.5 - Código que realiza o ajuste dos pesos da RNA

```

1 function [wk] = ajustarPesos(X,wk,viz,l,c,eta,NumSaidas,p)
2
3     WidthSq = viz^2;
4
5     %Inicia ajuste dos pesos do vencedor e dos vizinhos
6     for i=1:NumSaidas
7         for j=1:NumSaidas
8             % Distancia de cada no para o vencedor
9             distNo = ((1-i)^2) + ((c-j)^2);
10            % Se estiver dentro da vizinhanca: ajusta pesos
11            if distNo < WidthSq
12                % Calculo de quanto o peso deve ser ajustado
13                influen = exp(-((distNo)/(2*WidthSq)));
14                for t=1:size(X,1)
15                    wk(i,j,t) = wk(i,j,t) + eta * influen * (X(t,p) - wk
16                        (i,j,t));
17                end
18            end % Fim do ajuste os pesos do vencedor e dos vizinhos
19        end
20
21    return
22
23 end

```

A.1.5 Obtenção da matriz unitária

A matriz unitária é obtida por meio do cálculo da norma dos pesos em cada ponto do mapa de Kohonen, em que cada um destes corresponde a uma determinada entrada (Figura 3.8). O Código-Fonte A.6 é responsável por este procedimento.

Código-Fonte A.6 - Código que obtém a matriz unitária dos pesos finais

```
1 function [matrizU] = matrizUnitaria(wk_final , pathResult)
2
3     matrizU = zeros(size(wk_final ,1));
4
5     for i=1:size(wk_final ,1)
6         for j=1:size(wk_final ,1)
7             matrizU(i ,j) = sqrt(sum(wk_final(i ,j ,:) .^2));
8         end
9     end
10
11     % Apresenta os pesos em imagem colorida com base na norma dos dados
12     .
13     h = figure(1);
14     imagesc(matrizU);
15     title('Norma dos vetores de pesos do mapa de Kohonen');
16     print(h, '-djpeg', strcat(pathResult , '/norma'));
17
18     return
19 end
```

A.1.6 Leitura da Imagem

Após o cálculo da matriz unitária, é carregada a imagem de entrada. O Código-Fonte A.7, responsável por essa tarefa, verifica a composição da imagem, convertendo-a em níveis de cinza, caso essa possua outra composição.

Código-Fonte A.7 - Código que carrega uma imagem do tipo *.tiff*

```
1 function [I file] = carregaImagem
2
3     % Filtra somente arquivos .tif
4     [filename , pathname] = uigetfile({'*.tif'}, 'Selecione uma imagem');
5     file = strcat(pathname, filename);
6     Im = imread(file);
7
8     % Se tiver 3 bandas, converte para tons de cinza
```

```

9     if (size(Im,3) == 3)
10         I = rgb2gray(Im);
11     end
12
13     I = double(Im);
14
15     return
16
17 end

```

A.1.7 Leitura das Amostras

Uma vez a imagem carregada, é realizada a leitura das amostras. Nesta etapa, são considerados os parâmetros de tamanho do salto da janela de varredura e quantidade de orientações por amostra.

Código-Fonte A.8 - Código que realiza a varredura na imagem para o carregamento da matriz de amostras

```

1 function [X X_norm] = carregaAmostrasEsp(Imagem, NumEntradas, esp)
2
3     lin = size(Imagem,1); % Quantidade de linhas da imagem
4     col = size(Imagem,2); % Quantidade de colunas da imagem
5     mult = sqrt(NumEntradas); % Dimensao quadratica da janela de
6         varredura
7     l_inicio = 1; c_inicio = 1;
8     l_fim = 1; c_fim = 1;
9
10    % ——— Inicializa variaveis ———
11    padrao=1;
12    inclinationStep = 4;
13
14    indicePadraoColuna = 1;
15    indicePadraoLinha = 1;
16
17    % Varre a imagem e agrega cada valor do pixel a um valor de entrada
18    % do padrao de entrada correspondente.
19    while ((l_inicio + (mult-1)) < lin)
20
21        c_fim = c_inicio + (mult-1);
22        l_fim = l_inicio + (mult-1);
23        c_cont = 0;
24        orient = 1;

```

```

24     auxCoord = zeros(ceil(mult/inclinationStep),mult,2);
25
26     for i=c_inicio:inclinationStep:c_fim
27         % Orientacoes de 135 a 45 graus
28         auxCoord(orient,:,i) = bresenham(l_inicio,i,l_fim,c_fim-(
29             c_cont));
30         c_cont = c_cont + inclinationStep;
31
32     if(orient ~= 1)
33         if (orient+(ceil(mult/inclinationStep)-1) <= floor((
34             mult*2)/inclinationStep))
35
36             orientOpost = orient+(ceil(mult/inclinationStep)-1)
37                 ;
38             c2_cont = c_fim;
39
40             % Amostras na diagonal
41             if(indicePadraoLinha == indicePadraoColuna)
42                 for h=1:mult
43                     auxCoord(orientOpost,h,1) = auxCoord(orient
44                         ,h,2);
45                     auxCoord(orientOpost,h,2) = c2_cont;
46                     c2_cont = c2_cont - 1;
47                 end
48             end
49
50             % Amostras acima da diagonal
51             if(indicePadraoLinha < indicePadraoColuna)
52                 aux = (esp*(indicePadraoColuna-
53                     indicePadraoLinha));
54                 for h=1:mult
55                     auxCoord(orientOpost,h,1) = auxCoord(orient
56                         ,h,2) - aux;
57                     auxCoord(orientOpost,h,2) = c2_cont;
58                     c2_cont = c2_cont - 1;
59                 end
60             end
61
62             % Amostras abaixo da diagonal
63             if(indicePadraoLinha > indicePadraoColuna)
64                 aux = (esp*(indicePadraoLinha-
65                     indicePadraoColuna));
66                 for h=1:mult

```

```

60         auxCoord(orientOpost ,h,1) = auxCoord(orient
61             ,h,2) + aux;
62         auxCoord(orientOpost ,h,2) = c2_cont;
63         c2_cont = c2_cont - 1;
64     end
65
66     end
67
68     orient = orient + 1;
69 end
70
71 % Popula o vetor de entrada X, com as orientacoes da amostra
72     lida
73 for slope=1:size(auxCoord,1)
74     for input=1:mult
75         if (Imagem(auxCoord(slope ,input ,1) ,auxCoord(slope ,input
76             ,2)) == 0)
77             X(slope ,input ,padrao) = 1;
78         else
79             X(slope ,input ,padrao) = Imagem(auxCoord(slope ,input
80                 ,1) ,auxCoord(slope ,input ,2));
81         end
82     end
83 end
84
85     c_inicio = c_inicio + esp;
86     indicePadraoColuna = indicePadraoColuna + 1;
87     padrao = padrao + 1;
88
89 % Se exceder o incremento de colunas: pule de linha
90 if (c_inicio + (mult-1)) > col
91     l_inicio = l_inicio + esp;
92     c_inicio = 1;
93     indicePadraoLinha = indicePadraoLinha + 1;
94     indicePadraoColuna = 1;
95 end
96
97 end
98
99 % Normaliza dados de treinamento
100 X_norm = X;
101
102 for m=1:size(X,3)

```



```

99         u = X(:, :, m);
100        X_norm(:, :, m) = u/norm(u);
101    end
102
103    return
104
105 end

```

A.1.8 Localização dos centros no domínio de Kohonen

O Código-Fonte A.9 é responsável por obter a localização dos centros de cada aglomerado, isto é, cada centro caracteriza a forma exata pela qual a rede foi treinada a reconhecer. Esta informação é útil, pois é por meio desses pontos que o operador determina a presença de estradas ou não, verificando a distância entre o ponto de uma determinada amostra e cada um dos centros.

Código-Fonte A.9 - Código que obtém a localização dos centros dos aglomerados

```

1 function [ind] = localizaCluster(xTrain_norm, wk_norm, NumSaidas,
2     w_norma)
3
4     NumPadroes = size(xTrain_norm, 2);
5
6     for p=1:NumPadroes
7
8         d = zeros(NumSaidas);
9
10        for x=1:NumSaidas
11            for y=1:NumSaidas
12                a = wk_norm(x, y, :);
13                b = xTrain_norm(:, p);
14                d(x, y) = sum((a(:) - b(:)).^2);
15            end
16        end
17
18        % Encontrar o vencedor
19        [vec_l vec_c] = find(d == min(min(d)));
20        l = vec_l(1);
21        c = vec_c(1);
22
23        ind(p, 1) = l;
24        ind(p, 2) = c;

```

```

25     end
26
27     return
28
29 end

```

A.1.9 Semeação

Sendo a localização dos centros dos aglomerados uma informação conhecida, inicia-se o processo de semeação. O Código-Fonte A.10 apresenta as rotinas correspondentes a esse processo. Cada amostra é comparada com cada um dos centros, determinando se são pertencentes ou não aos padrões referentes a estrada. Ao final, é adquirido um vetor cujos elementos caracterizam o índice de cada amostra reconhecida como estrada.

Código-Fonte A.10 - Código que realiza a semeação

```

1  function [pontosSementes] = semearImagem(xAmostra_norm, wk_norm,
      centros, saidas)
2
3      cont = 1;
4      NumPadroes = size(xAmostra_norm,3);
5      pontosSementes = zeros(1,NumPadroes);
6
7      for p=1:NumPadroes
8          dist_Estrada = zeros(1,size(centros,1));
9
10         % Calcular as distancias euclidianas entre os vetores de
            entrada e
11         % de pesos para cada neuronio
12         d = zeros(saidas);
13         minOrient = zeros(1,size(xAmostra_norm,1));
14
15         for x=1:saidas
16             for y=1:saidas
17                 a = wk_norm(x,y);
18                 % Qual orientacao tem mais semelhanca com algum dos
                    padroes
19                 for orient=1:size(xAmostra_norm,1)
20                     b = xAmostra_norm(orient,:,p);
21                     d(x,y) = sum((a - b(:)).^2);
22                     minOrient(1,orient) = d(x,y);
23             end

```

```

24         [menorValor indiceOrientacao] = min(minOrient);
25
26         b = xAmostra_norm(indiceOrientacao ,: ,p);
27         d(x,y) = sum((a(:) - b(:)).^2);
28     end
29 end
30
31 % Encontrar o vencedor
32 [vec_l vec_c] = find(d == min(min(d)));
33 l = vec_l(1);
34 c = vec_c(1);
35
36 % Verifica se ha algum outro centro com distancia menor para o
37 % padrao "p".
38 % Se tiver , entao o padrao e relacionado como sendo uma estrada
39
40 for i=1:size(centros,1)
41     dist_Estrada(1,i) = sqrt((l - centros(i,1))^2 + (c -
42         centros(i,2))^2);
43 end
44 [distMin padraoMin] = min(dist_Estrada);
45
46 if padraoMin >= (size(centros,1)/2)
47     pontosSementes(cont) = p;
48     cont = cont + 1;
49 end
50 end
51 return
52 end

```

A.1.10 Pinta pontos-sementes

O Código-Fonte A.11 converte cada valor no vetor de saída do Código-Fonte A.10 em coordenadas (x, y) da imagem, em seguida, desenha o ponto-semente.

Código-Fonte A.11 - Código que pinta os pontos-sementes sobre a imagem

```

1 function [padroes_por_linha] = pintaPontosImagemEsp(file , estradas ,
2     mult , opcao , fig , pathResult , esp)
3     ponto = [0 255 0];

```

```

4
5   Im_final = imread(file);
6
7   lin = size(Im_final,1);
8   col = size(Im_final,2);
9
10  if opcao == 1
11      % Exibe pontos com a imagem no fundo
12      if size(Im_final,3)==1
13          Im_final = repmat(double(Im_final)./255,[1 1 3]);
14      end
15  else
16      % Exibe pontos sem a imagem no fundo
17      Im_final = ones(lin , col ,3);
18      Im_final = Im_final * 255;
19  end
20
21  Im_final = double(Im_final);
22  padroes_por_linha = floor((col+(esp-mult))/esp);
23  indices_imagem = zeros(padroes_por_linha ,2);
24
25  % Transforma os indices dos padroes em coordenadas na imagem
26  for i=1:size(estradas ,2)
27      if estradas(i) ~= 0
28          linha_do_padrao = ceil(estradas(i)/padroes_por_linha);
29          linha = ((linha_do_padrao-1)*esp) + floor(mult/2);
30          coluna_do_padrao = estradas(i) - ((linha_do_padrao - 1) *
31              padroes_por_linha);
32          coluna = ((coluna_do_padrao-1)*esp) + floor(mult/2);
33          indices_imagem(i,1) = linha;
34          indices_imagem(i,2) = coluna;
35      end
36  end
37  for i=1:size(indices_imagem ,1)
38      if indices_imagem(i,1)~=0
39          Im_final(indices_imagem(i,1),indices_imagem(i,2),:) = ponto
40              ;
41      else
42          break;
43      end
44  end

```

```

45     h = figure(fig);
46     imshow(Im_final);
47     print(h, '-djpeg', strcat(pathResult, '/noPruning_', date));
48
49 end

```

A.1.11 Fragmentação dos segmentos de reta

Até o momento, os pontos-sementes estão dispostos em um vetor, sem qualquer conhecimento de quais são os segmentos de reta e quais são os pontos espúrios no conjunto. O Código-Fonte A.12 realiza a tarefa de separar cada um dos segmentos retos, podar os pontos de descontinuidades e isolados.

Código-Fonte A.12 - Código que separa cada segmento de reta identificado em cada linha de uma matriz, realizando a poda em conjunto

```

1  function [segmentosFinais] = segmenta_estradas(estr, pad_lin, con, desc)
2
3      qtd_segmentos = 1;
4      estradasH = zeros(size(estr, 2));
5      ponto = 1;
6      flag = 0;
7
8      % Limiares de continuidades e descontinuidades, respectivamente
9      continuo = con;
10     descontinuo = desc;
11     v_continuo = 1;
12
13     % Controla os pontos de cada segmento
14     cont = 1;
15
16     % Separa somente os segmentos HORIZONTAIS
17     while flag == 0
18         if ((estr(cont+1)-estr(cont)) > descontinuo) && (v_continuo >=
19             continuo)
20             estradasH(qtd_segmentos, ponto) = estr(cont);
21             v_continuo = 1;
22             qtd_segmentos = qtd_segmentos + 1;
23             ponto = 1;
24             cont = cont + 1;
25         elseif ((estr(cont+1)-estr(cont)) > descontinuo) && (v_continuo
26             < continuo)
27             estradasH(qtd_segmentos, :) = 0;

```

```

26         ponto = 1;
27         v_continuo = 1;
28         cont = cont + 1;
29         elseif (mod(estr(cont)/pad_lin,1) == 0) && (v_continuo >=
continuo)
30             estradasH(qtd_segmentos , ponto) = estr(cont);
31             v_continuo = 1;
32             qtd_segmentos = qtd_segmentos + 1;
33             ponto = 1;
34             cont = cont + 1;
35         elseif mod(estr(cont)/pad_lin,1) == 0
36             estradasH(qtd_segmentos ,:) = 0;
37             v_continuo = 1;
38             ponto = 1;
39             cont = cont + 1;
40         else
41             estradasH(qtd_segmentos , ponto) = estr(cont);
42             ponto = ponto + 1;
43             v_continuo = v_continuo + 1;
44             cont = cont + 1;
45         end
46
47         if cont == size(estr,2)
48             if (v_continuo >= continuo)
49                 estradasH(qtd_segmentos , ponto) = estr(cont);
50             end
51             qtd_segmentos = qtd_segmentos + 1;
52             flag = 1;
53         end
54     end
55
56     flag = 0;
57     ponto = 1;
58     v_continuo = 1;
59     cont = 1;
60
61     % Reordena padroes na variavel estradas
62     % estradas = [13 20 23 30 33 37 40 43 47 50 53 55 60 65 75 85 87
93];
63     % estradas = [13 23 33 43 53 93 20 30 40 50 60 37 47 87 55 65 75
85];
64     estradas = 0;
65     estradas = ordena(estr , pad_lin , 1);

```

```

66
67 % Separa somente os segmentos VERTICAIS
68 while flag == 0
69     if (estradas(cont+1) < estradas(cont)) || ((estradas(cont+1) -
70         estradas(cont)) < pad_lin)
71         estradasH(qtd_segmentos , ponto) = estradas(cont);
72         if (v_continuo >= continuo)
73             qtd_segmentos = qtd_segmentos + 1;
74         else
75             estradasH(qtd_segmentos , :) = 0;
76         end
77         v_continuo = 1;
78         ponto = 1;
79         cont = cont + 1;
80     elseif (((estradas(cont+1) - estradas(cont)) - pad_lin)/
81         pad_lin) > descontinuo) && (v_continuo >= continuo)
82         estradasH(qtd_segmentos , ponto) = estradas(cont);
83         v_continuo = 1;
84         qtd_segmentos = qtd_segmentos + 1;
85         ponto = 1;
86         cont = cont + 1;
87     elseif (((estradas(cont+1) - estradas(cont)) - pad_lin)/
88         pad_lin) > descontinuo) && (v_continuo < continuo)
89         estradasH(qtd_segmentos , :) = 0;
90         ponto = 1;
91         v_continuo = 1;
92         cont = cont + 1;
93     elseif (mod(estradas(cont)/pad_lin , pad_lin) > (pad_lin - 1)) &&
94         (v_continuo >= continuo)
95         estradasH(qtd_segmentos , ponto) = estradas(cont);
96         v_continuo = 1;
97         qtd_segmentos = qtd_segmentos + 1;
98         ponto = 1;
99         cont = cont + 1;
100     elseif mod(estradas(cont)/pad_lin , pad_lin) > (pad_lin - 1)
101         estradasH(qtd_segmentos , :) = 0;
102         v_continuo = 1;
103         ponto = 1;
104         cont = cont + 1;
105     else
106         estradasH(qtd_segmentos , ponto) = estradas(cont);
107         ponto = ponto + 1;
108         v_continuo = v_continuo + 1;

```

```

105         cont = cont + 1;
106     end
107
108     if cont == size(estradas,2)
109         if (v_continuo >= continuo) && (estradas(cont) < estradas(
110             cont-1)) || ((estradas(cont) - estradas(cont-1)) <
111                 pad_lin)
112             estradasH(qtd_segmentos, ponto) = estradas(cont);
113         end
114         qtd_segmentos = qtd_segmentos + 1;
115         flag = 1;
116     end
117 end
118
119 ponto = 1;
120 flag = 0;
121 v_continuo = 1;
122 cont = 1;
123
124 % Reordena padroes na variavel estradas
125 estradas = 0;
126 estradas = ordena(estr, pad_lin, 2);
127
128 % Separa somente os segmentos DIAGONAIS -45
129 while flag == 0
130     if (estradas(cont+1) < estradas(cont)) || ((estradas(cont+1) -
131         estradas(cont)) < pad_lin+1)
132         estradasH(qtd_segmentos, ponto) = estradas(cont);
133         if (v_continuo >= continuo)
134             qtd_segmentos = qtd_segmentos + 1;
135         else
136             estradasH(qtd_segmentos, :) = 0;
137         end
138         v_continuo = 1;
139         ponto = 1;
140         cont = cont + 1;
141     elseif (((estradas(cont+1) - estradas(cont)) - (pad_lin+1))/(
142         pad_lin+1)) > descontinuo) && (v_continuo >= continuo)
143         estradasH(qtd_segmentos, ponto) = estradas(cont);
144         v_continuo = 1;
145         qtd_segmentos = qtd_segmentos + 1;
146         ponto = 1;
147         cont = cont + 1;

```



```

144     elseif (((estradas(cont+1) - estradas(cont)) - (pad_lin+1))/(
145             pad_lin+1)) > descontínuo) && (v_continuo < contínuo)
146         estradasH(qtd_segmentos,:) = 0;
147         ponto = 1;
148         v_continuo = 1;
149         cont = cont + 1;
150     elseif (mod(estradas(cont)/pad_lin, pad_lin) > (pad_lin - 1))
151             && (v_continuo >= contínuo)
152         estradasH(qtd_segmentos,ponto) = estradas(cont);
153         v_continuo = 1;
154         qtd_segmentos = qtd_segmentos + 1;
155         ponto = 1;
156         cont = cont + 1;
157     elseif mod(estradas(cont)/pad_lin, pad_lin) > (pad_lin - 1)
158         estradasH(qtd_segmentos,:) = 0;
159         v_continuo = 1;
160         ponto = 1;
161         cont = cont + 1;
162     else
163         estradasH(qtd_segmentos,ponto) = estradas(cont);
164         ponto = ponto + 1;
165         v_continuo = v_continuo + 1;
166         cont = cont + 1;
167     end
168     if cont == size(estradas,2)
169         if (v_continuo >= contínuo) && ((estradas(cont) < estradas
170             (cont-1)) || ((estradas(cont) - estradas(cont-1)) <
171                 pad_lin+1))
172             estradasH(qtd_segmentos,ponto) = estradas(cont);
173         end
174         qtd_segmentos = qtd_segmentos + 1;
175         flag = 1;
176     end
177 end
178 ponto = 1;
179 flag = 0;
180 v_continuo = 1;
181 cont = 1;
182 % Reordena padroes na variavel estradas
183 estradas = 0;

```

```

183     estradas = ordena(estr , pad_lin , 3) ;
184
185     % Separa somente os segmentos DIAGONAIS 45
186     while flag == 0
187         if (estradas(cont+1) < estradas(cont)) || ((estradas(cont+1) -
188             estradas(cont)) < pad_lin - 1)
189             estradasH(qtd_segmentos , ponto) = estradas(cont) ;
190             if (v_continuo >= continuo)
191                 qtd_segmentos = qtd_segmentos + 1 ;
192             else
193                 estradasH(qtd_segmentos , :) = 0 ;
194             end
195             v_continuo = 1 ;
196             ponto = 1 ;
197             cont = cont + 1 ;
198         elseif (((estradas(cont+1) - estradas(cont)) - (pad_lin - 1)) / (
199             pad_lin - 1)) > descontinuo) && (v_continuo >= continuo)
200             estradasH(qtd_segmentos , ponto) = estradas(cont) ;
201             v_continuo = 1 ;
202             qtd_segmentos = qtd_segmentos + 1 ;
203             ponto = 1 ;
204             cont = cont + 1 ;
205         elseif (((estradas(cont+1) - estradas(cont)) - (pad_lin - 1)) / (
206             pad_lin - 1)) > descontinuo) && (v_continuo < continuo)
207             estradasH(qtd_segmentos , :) = 0 ;
208             ponto = 1 ;
209             v_continuo = 1 ;
210             cont = cont + 1 ;
211         elseif (mod(estradas(cont)/pad_lin , pad_lin) > (pad_lin - 1))
212             && (v_continuo >= continuo)
213             estradasH(qtd_segmentos , ponto) = estradas(cont) ;
214             v_continuo = 1 ;
215             qtd_segmentos = qtd_segmentos + 1 ;
216             ponto = 1 ;
217             cont = cont + 1 ;
218         elseif mod(estradas(cont)/pad_lin , pad_lin) > (pad_lin - 1)
219             estradasH(qtd_segmentos , :) = 0 ;
220             v_continuo = 1 ;
221             ponto = 1 ;
222             cont = cont + 1 ;
223         else
224             estradasH(qtd_segmentos , ponto) = estradas(cont) ;
225             ponto = ponto + 1 ;

```

```

222         v_continuo = v_continuo + 1;
223         cont = cont + 1;
224     end
225
226     if cont == size(estradas,2)
227         if (v_continuo >= continuo) && ((estradas(cont) < estradas
228             (cont-1)) || ((estradas(cont) - estradas(cont-1)) <
229                 pad_lin-1))
230             estradasH(qtd_segmentos,ponto) = estradas(cont);
231         end
232         flag = 1;
233     end
234 end
235
236 counterLine = 1;
237 segmentosFinais = zeros(size(nonzeros(estradasH(:,1)),1),size(
238     estradasH,2));
239
240 for i=1:size(estradasH,1)
241     if (size(nonzeros(estradasH(i,:)),1) >= 1)
242         for j=1:size(estradasH,2)
243             segmentosFinais(counterLine,j) = estradasH(i,j);
244         end
245         counterLine = counterLine + 1;
246     else
247         continue;
248     end
249 end
250 return
251 end

```

A.1.11.1 Ordena os índices para a fragmentação

Para a determinação se um ponto é ou não pertencente à um mesmo segmento de reta, é analisado a sua sequência em relação ao próximo ponto do vetor. Desta forma, ao fragmentar segmentos de reta na horizontal, os pontos devem ser analisados na sequência comum (e.g.: 1, 2, 3,...). No entanto, se forem fragmentados na vertical, por exemplo, os índices de amostras nesse vetor devem ser alterados, de forma que a análise seja realizada da mesma forma que a orientação horizontal. O Código-Fonte

A.13 realiza a ordenação do vetor (ver Seção A.1.9) conforme a orientação em que se deseja analisar.

Código-Fonte A.13 - Código que ordena os índices para a fragmentação dos segmentos de reta conforme sua orientação

```

1 function [estradasY] = ordena(estradas , pad_lin , opcao)
2
3     estradasY = zeros(size(estradas));
4     pivo = 1;
5     contad = 2;
6     flag = 0;
7     estradasY(1) = estradas(1);
8
9     switch(opcao)
10        case 1 % VERTICAL
11            while flag == 0
12                for i=pivo+1:size(estradas,2)
13                    vf = mod(estradas(pivo)/pad_lin,1);
14                    vr = mod(estradas(i)/pad_lin,1);
15                    if (vf-vr)^2 < (1*10^(-5))
16                        estradasY(contad) = estradas(i);
17                        contad = contad + 1;
18                    end
19                end
20
21                pivo = pivo + 1;
22                flag2 = 0;
23
24                % Obtem o proximo pivo
25                % Neste caso os pivos devem ser: 13, 20, 37 e 55;
26                while flag2 == 0
27                    possui_num = 0;
28                    for l=1:contad-1
29                        v1 = mod(estradas(pivo)/pad_lin,1);
30                        v2 = mod(estradasY(l)/pad_lin,1);
31                        if (v1-v2)^2 < (1*10^(-5))
32                            possui_num = 1;
33                            break;
34                        end
35                    end
36                    if possui_num == 1 && pivo < size(estradas,2)
37                        pivo = pivo + 1;
38                    else

```

```

39         flag2 = 1;
40     end
41 end
42
43 if contad == size(estradas,2) || pivo == size(estradas
44     ,2)
45     estradasY(contad) = estradas(pivo);
46     flag = 1;
47 else
48     estradasY(contad) = estradas(pivo);
49     contad = contad + 1;
50 end
51 case 2 % DIAGONAL (-45 - DECAIMENTO DA ESQUERDA PARA DIREITA)
52     % pad_lin = 18
53     % [3 13 17 23 29 39 43 49 51 53 59 63 65 79]
54     % [3 13 23 43 53 63 17 29 39 49 59 79 51 65]
55 while flag == 0
56     for i=pivo+1:size(estradas,2)
57         Q_var = ceil(estradas(i)/pad_lin);
58         Q_var = Q_var - (ceil(estradas(pivo)/pad_lin));
59         vf = mod(estradas(pivo)/pad_lin,1);
60         vr = mod((estradas(i) - Q_var)/pad_lin,1);
61         if (vf-vr)^2 < (1*10^(-5))
62             estradasY(contad) = estradas(i);
63             contad = contad + 1;
64         end
65     end
66
67     pivo = pivo + 1;
68     flag2 = 0;
69
70     % Obtem o proximo pivo
71     % Neste caso os pivos devem ser: 3, 17, 29, 51 e 65;
72 while flag2 == 0
73     possui_num = 0;
74     for l=1:contad-1
75         Q_var2 = ceil(estradasY(l)/pad_lin);
76         Q_var2 = Q_var2 - (ceil(estradas(pivo)/pad_lin)
77             );
78         v1 = mod(estradas(pivo)/pad_lin,1);
79         v2 = mod((estradasY(l) - Q_var2)/pad_lin,1);
80         if (v1-v2)^2 < (1*10^(-5))

```

```

80             possui_num = 1;
81             break;
82         end
83     end
84     if possui_num == 1 && pivo < size(estradas,2)
85         pivo = pivo + 1;
86     else
87         flag2 = 1;
88     end
89 end
90
91 if contad == size(estradas,2) || pivo == size(estradas
92     ,2)
93     flag = 1;
94 else
95     estradasY(contad) = estradas(pivo);
96     contad = contad + 1;
97 end
98 case 3 % DIAGONAL (45 - DECAIMENTO DA DIREITA PARA ESQUERDA)
99     % pad_lin = 18
100    % [49 97 141 158 162 175 192 196 209 213 226 230 256
101       260];
102    % [49 97 141 158 175 192 209 226 260 162 196 213 230
103       256];
104 while flag == 0
105     for i=pivo+1:size(estradas,2)
106         Q_var = ceil(estradas(i)/pad_lin);
107         Q_var = Q_var - (ceil(estradas(pivo)/pad_lin));
108         vf = mod(estradas(pivo)/pad_lin,1);
109         vr = mod((estradas(i) + Q_var)/pad_lin,1);
110         if (vf-vr)^2 < (1*10^(-5))
111             estradasY(contad) = estradas(i);
112             contad = contad + 1;
113         end
114     end
115     pivo = pivo + 1;
116     flag2 = 0;
117
118     % Obtem o proximo pivo
119     % Neste caso os pivos devem ser: 3, 17, 29, 51 e 65;
120     while flag2 == 0

```

```

120         possui_num = 0;
121         for l=1:contad-1
122             Q_var2 = ceil(estradasY(l)/pad_lin);
123             Q_var2 = Q_var2 - (ceil(estradas(pivo)/pad_lin)
124                 );
125             v1 = mod(estradas(pivo)/pad_lin,1);
126             v2 = mod((estradasY(l) + Q_var2)/pad_lin,1);
127             if (v1-v2)^2 < (1*10^(-5))
128                 possui_num = 1;
129                 break;
130             end
131         if possui_num == 1 && pivo < size(estradas,2)
132             pivo = pivo + 1;
133         else
134             flag2 = 1;
135         end
136     end
137
138     if contad == size(estradas,2) || pivo == size(estradas
139         ,2)
140         estradasY(contad) = estradas(pivo);
141         flag = 1;
142     else
143         estradasY(contad) = estradas(pivo);
144         contad = contad + 1;
145     end
146     end
147     otherwise fprintf('Opcao de ordenacao invalida!\n');
148 end
149 return
150
151 end

```

A.1.12 Pinta pontos-sementes (Após a poda)

Após a fragmentação e o processo de poda, os pontos-sementes são desenhados novamente, porém, com o decréscimo dos pontos espúrios e os segmentos descontínuos.

Código-Fonte A.14 - Código que pinta os pontos-sementes sobre a imagem após a realização da poda

```
1 function [indices_imagem] = pintaPontosImagemEsp_seg(file , estradas ,
2     mult , opcao , fig , pathResult , esp , it)
3     ponto = [255 0 0];
4
5     Im_final = imread(file);
6
7     lin = size(Im_final,1);
8     col = size(Im_final,2);
9
10    if opcao == 1
11        % Exibe pontos com a imagem no fundo
12        if size(Im_final,3)==1
13            Im_final = repmat(double(Im_final)./255,[1 1 3]);
14        end
15    else
16        % Exibe pontos sem a imagem no fundo
17        Im_final = ones(lin , col ,3);
18        Im_final = Im_final * 255;
19    end
20
21    padroes_por_linha = floor((col+(esp-mult))/esp);
22    indices_imagem = zeros(padroes_por_linha , padroes_por_linha ,2);
23
24    % Transforma os indices dos padroes em coordenadas na imagem
25    for j=1:size(estradas,1)
26        for i=1:size(estradas,2)
27            if estradas(j,i)~=0
28                linha_do_padrao = ceil(estradas(j,i)/padroes_por_linha)
29                ;
30                linha = ((linha_do_padrao-1)*esp) + floor(mult/2);
31                coluna_do_padrao = estradas(j,i) - ((linha_do_padrao -
32                1) * padroes_por_linha);
33                coluna = ((coluna_do_padrao-1)*esp) + floor(mult/2);
34                indices_imagem(j,i,1) = linha;
35                indices_imagem(j,i,2) = coluna;
36            else
37                break
38            end
39        end
40    end
41 end
```



```

39
40     for j=1:size(indices_imagem,1)
41         for i=1:size(indices_imagem,2)
42             if indices_imagem(j,i,1)~=0
43                 Im_final(indices_imagem(j,i,1),indices_imagem(j,i,2),:)
44                     = ponto;
45             else
46                 break
47             end
48         end
49     end
50
51     h = figure(fig);
52     imshow(Im_final);
53     print(h, '-djpeg', strcat(pathResult, '/Pruning_', date));
54
55     return
56 end

```

A.1.13 Execução do método *Open-Snakes*

A obtenção dos eixos de simetria se inicia nesta etapa. Inicialmente, o Código-Fonte A.15 realiza a interpolação entre os pontos de cada segmento de reta, obtendo-se as respectivas curvas iniciais. Em seguida, é realizado o processo iterativo para a minimização da energia de cada um dos segmentos de reta. Ao final, é adquirido duas matrizes de coordenadas x e y , correspondentes a localização espacial de cada curva.

Código-Fonte A.15 - Código que obtém o eixos de simetria das estradas

```

1 function [countSegm finalSnakeX finalSnakeY row col] = snakes(indice ,
2     file , sigma , alpha , beta , gamma , kappa , wl , max_iter , param)
3
4     initialSnakeX = zeros(size(indice,1),150);
5     initialSnakeY = zeros(size(indice,1),150);
6
7     % Interpola os pontos
8     for i=1:size(indice,1)
9
10         if(size(nonzeros(indice(i,:,1)),1) <= 3)
11             indice(i,:,1) = 0;
12             indice(i,:,2) = 0;

```

```

12         continue;
13     end
14
15     n = size(nonzeros(indice(i,:,1))',2);
16     indices = [nonzeros(indice(i,:,1))';nonzeros(indice(i,:,2))'];
17
18     points = 1:n; pointsInterp = 1:0.3:n;
19     cordXY = spline(points,indices,pointsInterp);
20
21     initialSnakeX(i,1:size(cordXY,2)) = cordXY(2,:);
22     initialSnakeY(i,1:size(cordXY,2)) = cordXY(1,:);
23
24 end
25
26 countSegm = size(initialSnakeX,1);
27 finalSnakeX = zeros(size(initialSnakeX,1),size(initialSnakeX,2));
28 finalSnakeY = zeros(size(initialSnakeY,1),size(initialSnakeY,2));
29
30 im = imread(file);
31
32 % se tiver 3 bandas, converte para tons de cinza
33 if (size(im,3) == 3)
34     im = rgb2gray(im);
35 end
36
37 im = double(im);
38 [row col] = size(im);
39
40 smask = fspecial('gaussian',ceil(3*sigma),sigma);
41 im_smth = filter2(smask,im,'same');
42
43 % Obtem a energia da linha, que e a propria imagem
44 e_line = im_smth;
45
46 % A energia externa e entao calculada, sendo esta a soma das
47 energias
48 e_ext = -wl * e_line;
49
50 % Obtem o gradiente da enerngia externa
51 [Fx, Fy] = gradient(e_ext);
52
53 h = figure(6);

```

```

54     if (param == 1)
55         im = ones(row, col);
56         im = im*255; % Fundo branco
57     end
58
59     imshow(im, []);
60     hold on
61
62     % Aplica o processo para cada segmento da estrada
63     for i=1:size(initialSnakeX,1)
64         % Se nao tiver pontos suficientes, analisa proximo segmento de
           reta
65         if (size(nonzeros(initialSnakeX(i,:)),1) <= 4)
66             countSegm = countSegm - 1;
67             continue;
68         end
69
70         cordX_temp = nonzeros(initialSnakeX(i,:));
71         cordY_temp = nonzeros(initialSnakeY(i,:));
72
73         pointsOnCurve = size(cordX_temp,1);
74
75         alphaVt = alpha*ones(1,pointsOnCurve);
76         betaVt = beta*ones(1,pointsOnCurve);
77
78         % Produz a matriz pentadiagonal
79         alphap1 = [alphaVt(pointsOnCurve) alphaVt(1:pointsOnCurve-1)];
80         betam1 = [betaVt(2:pointsOnCurve) betaVt(1)];
81         betap1 = [betaVt(pointsOnCurve) betaVt(1:pointsOnCurve-1)];
82
83         a = betam1;
84         b = -alphaVt - 2*betaVt - 2*betam1;
85         c = alphaVt + alphap1 + betam1 + 4*betaVt + betap1;
86         d = -alphap1 - 2*betaVt - 2*betap1;
87         e = betap1;
88
89         % Gera os parametros da matriz
90         A = diag(a(1:pointsOnCurve-2),-2) + diag(a(pointsOnCurve-1:
           pointsOnCurve),pointsOnCurve-2);
91         A = A + diag(b(1:pointsOnCurve-1),-1) + diag(b(pointsOnCurve),
           pointsOnCurve-1);
92         A = A + diag(c);
93         A = A + diag(d(1:pointsOnCurve-1),1) + diag(d(pointsOnCurve),-(

```

```

    pointsOnCurve-1));
94  A = A + diag(e(1:pointsOnCurve-2),2) + diag(e(pointsOnCurve-1:
    pointsOnCurve),-(pointsOnCurve-2));
95
96  A_inv = inv(A + gamma * diag(ones(1,pointsOnCurve)));
97
98  ssx = zeros(pointsOnCurve,1);
99  ssy = zeros(pointsOnCurve,1);
100
101  motionX(1) = cordX_temp(1); ssx(pointsOnCurve) = cordX_temp(
    pointsOnCurve);
102  motionY(1) = cordY_temp(1); ssy(pointsOnCurve) = cordY_temp(
    pointsOnCurve);
103
104  % A cada iteracao, obtem uma nova posicao para a curva
105  for v=1:max_iter
106
107      motionX(2:pointsOnCurve-1) = gamma * cordX_temp(2:
        pointsOnCurve-1) - (kappa * interp2(fx, cordX_temp(2:
        pointsOnCurve-1), cordY_temp(2:pointsOnCurve-1)));
108      motionY(2:pointsOnCurve-1) = gamma * cordY_temp(2:
        pointsOnCurve-1) - (kappa * interp2(fy, cordX_temp(2:
        pointsOnCurve-1), cordY_temp(2:pointsOnCurve-1)));
109
110      % Calcula a nova posicao da snake
111      cordX_temp = A_inv * motionX;
112      cordY_temp = A_inv * motionY;
113
114  end
115
116  finalSnakeX(i,2:size(cordX_temp,1)-1) = cordX_temp(2:
    pointsOnCurve-1)';
117  finalSnakeY(i,2:size(cordY_temp,1)-1) = cordY_temp(2:
    pointsOnCurve-1)';
118
119  plot(nonzeros(finalSnakeX(i,:))', nonzeros(finalSnakeY(i,:))',
    '-r', 'LineWidth', 2);
120
121  end
122
123  im = ones(row,col);
124  im = im*255; % Fundo branco
125

```

```

126     h = figure(7);
127     imshow(im, []);
128     hold on
129     for j=1:size(finalSnakeX,1)
130         plot(nonzeros(finalSnakeX(j,:))', nonzeros(finalSnakeY(j,:))',
131             '-r', 'LineWidth', 2);
132
133     return
134 end

```

A.2 Códigos-Fonte auxiliares no processo de extração

As rotinas apresentadas abaixo, são utilizadas como rotinas complementares durante o processo de extração de estradas.

A.2.1 Converte a imagem de composição RGB para níveis de cinza

O Código-Fonte A.16 verifica a composição da imagem selecionada pelo usuário, caso esta não possuir as características de níveis de cinza, a imagem é convertida.

Código-Fonte A.16 - Código que realiza a conversão da imagem de composição RGB para níveis de cinza

```

1 function [IGray] = rgbtorgay(File, I)
2
3     [lin, col] = size(File);
4     verExt = strfind(File, '.jpg');
5     [lin2, col2] = size(verExt);
6
7     if col2 ~= 0
8         if (col-3) == verExt(col2)
9             IGray = rgb2gray(I);
10        end
11    end
12
13    return
14
15 end

```

A.2.2 Algoritmo de Bresenham

O algoritmo de Bresenham (BRESENHAM, 1965), conhecido também por algoritmo de Ponto Médio, determina em uma matriz (Imagem), os *pixels* que devem ser destacados entre dois pontos na imagem, tal que entre eles formem uma reta. O Código-Fonte A.17 realiza esta operação para a aquisição de diferentes orientações nas amostras.

Código-Fonte A.17 - Código de Bresenham para a obtenção dos *pixels* entre dois pontos em uma imagem

```
1 function [coord] = bresenham(x1, y1, x2, y2)
2
3     slope = 0;
4     dx = 0;
5     dy = 0;
6     incE = 0;
7     incNE = 0;
8     d = 0;
9     x = 0;
10    y = 0;
11    coord = zeros((y2-y1)+1,2);
12
13    % Onde inverte a linha x1 > x2
14    if (x1 > x2)
15        bresenham(x2, y2, x1, y1);
16        return;
17    end
18
19    dx = x2 - x1;
20    dy = y2 - y1;
21
22    if (dy < 0)
23        slope = -1;
24        dy = -dy;
25    else
26        slope = 1;
27    end
28
29    % Constante de Bresenham
30    incE = 2 * dy;
31    incNE = 2 * dy - 2 * dx;
32    d = 2 * dy - dx;
```

```

33     y = y1;
34     cont = 1;
35
36     for x=x1:x2
37         coord(cont,1) = x;
38         coord(cont,2) = y;
39         cont = cont + 1;
40         if(d <= 0)
41             d = d + incE;
42         else
43             d = d + incNE;
44             y = y + slope;
45         end
46
47     end
48
49 end

```

A.2.3 Validação dos resultados

O algoritmo de validação é baseado nas métricas de validação apresentadas na Seção 3.3.4. Os valores são obtidos a partir da sobreposição dos conjuntos referência e extraído. Os fatores de acerto e erro são determinados a partir da verificação de cada um dos *pixels* entre as conjuntos.

Código-Fonte A.18 - Código que realiza a validação dos resultados

```

1 function [cpt, cor, qlt, rdn, unf] = evaluation(finalSnakeX, finalSnakeY,
2     row, col, largPred)
3
4     % Completeness => cpt
5     % Correctness => cor
6     % Quality => qlt
7     % Redundancy => rdn
8     % Unification => sqrt((cpt-1)^2 + (cor-1)^2 + (qlt-1)^2 + (rdn-0)
9     ^0)
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

9     cpt = 0; cor = 0; qlt = 0; rdn = 0; unf = 0;
10
11     [filename, pathname] = uigetfile({'*.jpg'}, 'Selecione uma imagem de
12     referencia');

```

```

12     file = strcat(pathname, filename);
13     reference = imread(file);
14
15     [filename, pathname] = uigetfile({'*.jpg'}, 'Selecione o resultado
        adquirido');
16     file = strcat(pathname, filename);
17     result = imread(file);
18
19     ref = limiariza(reference);
20     result = limiariza(result);
21     result = dilata(largPred, result);
22
23     if((size(ref,1) ~= row) || (size(ref,2) ~= col) || (size(ref,1) ~=
        size(result,1)) || (size(ref,2) ~= size(result,2)))
24         fprintf('Dimensoes das imagens selecionadas estao incoerentes.
        Verifique!!\n');
25     return
26 else
27     [lengthReference lengthExtracted lengthReferenceMatched
        lengthExtractedMatched] = counter(ref, result, finalSnakeX,
        finalSnakeY, row, col, largPred);
28
29     cpt = lengthReferenceMatched/lengthReference;
30     cor = lengthExtractedMatched/lengthExtracted;
31     qlt = lengthExtractedMatched/(lengthExtracted + (
        lengthReference - lengthReferenceMatched));
32     rdn = (lengthExtractedMatched - lengthReferenceMatched)/
        lengthExtractedMatched;
33     unf = sqrt(((cpt-1)^2) + ((cor-1)^2) + ((qlt-1)^2) + (rdn^2));
34     end
35
36     return
37
38 end

```

A.2.3.1 Contagem de acerto e erro

O Código-fonte A.19 realiza a contagem dos índices de acerto e erro, em que a imagem referência e os resultados obtidos são sobrepostos para a verificação *pixel a pixel* de cada elemento que coincidiu uns aos outros.

Código-Fonte A.19 - Código que realiza a contagem dos índices de acerto e erro.

```

1 function [lengthReference lengthExtracted lengthReferenceMatched
   lengthExtractedMatched] = counter(ref, result, finalSnakeX,
   finalSnakeY, row, col, largPred)
2
3     lengthReference = 0;
4     lengthExtracted = 0;
5     lengthReferenceMatched = 0;
6     lengthExtractedMatched = 0;
7
8     im = ones(row, col)*255;
9     file = 'C:\counterTemp.jpg';
10
11    for i=1:size(ref,1)
12        for j=1:size(ref,2)
13            if (ref(i,j) == 1)
14                lengthReference = lengthReference + 1;
15            end
16            if ((ref(i,j) == 1) && (result(i,j) == 1))
17                lengthReferenceMatched = lengthReferenceMatched + 1;
18            end
19        end
20    end
21
22    for i=1:size(nonzeros(finalSnakeX(:,2)), 1)
23        imshow(im, []);
24        [r c] = size(im);
25        set(gca, 'Units', 'normalized', 'Position', [0 0 1 1]);
26        set(gcf, 'Units', 'pixels', 'Position', [200 200 c r]);
27        hold on
28        plot(nonzeros(finalSnakeX(i,:))', nonzeros(finalSnakeY(i,:))',
   '-r', 'LineWidth', 1);
29
30        h = getframe(gcf);
31        imwrite(h.cdata, file);
32
33        tpItem = imread(file);
34        imagem = limiariza(tpItem);
35        imagem = dilata(largPred, imagem);
36
37        for lin=1:row
38            for col=1:col
39                if (imagem(lin, col) == 1)
40                    lengthExtracted = lengthExtracted + 1;

```

```

41         end
42         if ((imagem(lin , col) == 1) && (ref(lin , col) == 1))
43             lengthExtractedMatched = lengthExtractedMatched +
44                 1;
45         end
46     end
47     im = ones(row , col) * 255;
48 end
49
50 return
51
52 end

```

A.2.3.2 Dilatação dos segmentos de reta extraídos

O Código-fonte A.20 realiza a dilatação dos segmentos de reta extraídos, de modo que a comparação *pixel a pixel* entre o conjunto extraído e a imagem referência seja realizada de forma adequada.

Código-Fonte A.20 - Código que realiza a dilatação de uma imagem.

```

1 function [result] = dilata(largPred , result)
2     dilatacao = floor(largPred/2);
3     se = strel('ball', dilatacao , 0);
4     result = imdilate(result , se);
5
6     return
7 end

```
