



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m19/2012/03.16.13.15-TDI

**PLANEJAMENTO DE TRAJETÓRIAS PARA
VEÍCULOS AÉREOS NÃO TRIPULADOS USANDO
MODELAGEM COMPUTACIONAL DE AMBIENTES DE
NAVEGAÇÃO ATRAVÉS DE GRAFOS DE
VISIBILIDADE E MODELOS DIGITAIS DE ELEVAÇÃO**

Felipe Leonardo Lôbo Medeiros

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Drs. José Demisio Simões da Silva, e Horacio Hideki Yanasse, aprovada em 03 de fevereiro de 2012.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3BH8SRL>>

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-m19/2012/03.16.13.15-TDI

**PLANEJAMENTO DE TRAJETÓRIAS PARA
VEÍCULOS AÉREOS NÃO TRIPULADOS USANDO
MODELAGEM COMPUTACIONAL DE AMBIENTES DE
NAVEGAÇÃO ATRAVÉS DE GRAFOS DE
VISIBILIDADE E MODELOS DIGITAIS DE ELEVAÇÃO**

Felipe Leonardo Lôbo Medeiros

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Drs. José Demisio Simões da Silva, e Horacio Hideki Yanasse, aprovada em 03 de fevereiro de 2012.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3BH8SRL>>

INPE
São José dos Campos
2012

Medeiros, Felipe Leonardo Lôbo.

M467p Planejamento de trajetórias para veículos aéreos não tripulados usando modelagem computacional de ambientes de navegação através de grafos de visibilidade e modelos digitais de elevação / Felipe Leonardo Lôbo Medeiros. – São José dos Campos : INPE, 2012.

xxxviii + 198 p. ; (sid.inpe.br/mtc-m19/2012/03.16.13.15-TDI)

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

Orientadores : Drs. José Demisio Simões da Silva, e Horacio Hideki Yanasse.

1. veículos aéreos não tripulados. 2. planejamento automático de trajetórias de navegação. 3. modelagem de ambientes de navegação. 4. gráficos de visibilidade. 5. modelos digitais de elevação. I. Título.

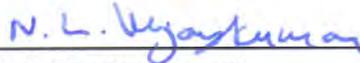
CDU 004.832.25

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.

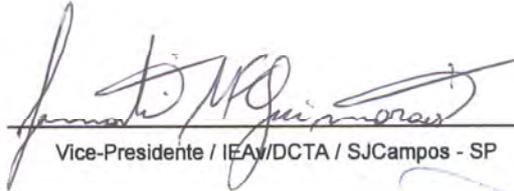
Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de Doutor(a) em
Computação Aplicada

Dr. Nandamudi Lankalapalli Vijaykumar



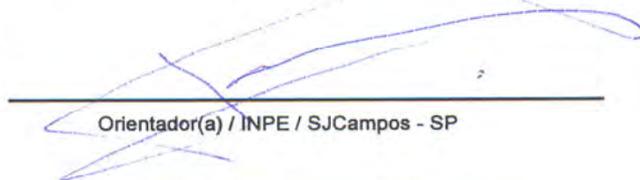
Presidente / INPE / SJC Campos - SP

Dr. Lamartine Nogueira Frutuoso
Guimarães



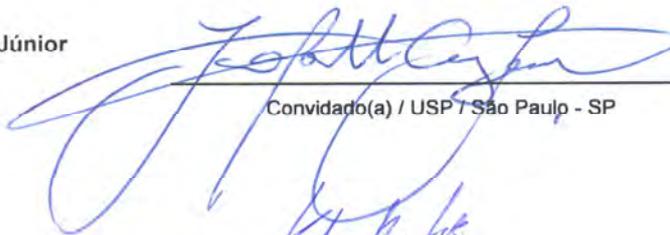
Vice-Presidente / IEAv/DCTA / SJC Campos - SP

Dr. Horacio Hideki Yanasse



Orientador(a) / INPE / SJC Campos - SP

Dr. João Batista Camargo Júnior



Convidado(a) / USP / São Paulo - SP

Dr. Karl Heinz Kienitz



Convidado(a) / ITA / SJC Campos - SP

Este trabalho foi aprovado por:

() maioria simples

unanimidade

Aluno (a): Felipe Leonardo Lôbo Medeiros

São José dos Campos, 03 de fevereiro de 2012

“O rio atinge seus objetivos porque aprendeu a contornar obstáculos”.

Lao Tsé

Eu dedico este trabalho a meus pais, Manoel Medeiros Neto e Maria Magna Lôbo Medeiros, fundamentais em minha vida. Eu faço uma dedicatória especial ao professor José Demisio Simões da Silva, por toda orientação, motivação e amizade, as quais nunca serão esquecidas.

AGRADECIMENTOS

Agradeço a Deus pela luz e proteção em toda minha vida.

Agradeço a meus pais e a meus irmãos Fernando Henrique Medeiros e Nathália Magna Medeiros pelo carinhoso apoio e pela confiança transmitida, principalmente, diante das dificuldades que ocorreram ao longo desta jornada.

Agradeço ao professor José Demisio Simões da Silva por toda dedicação e orientação do meu trabalho. Uma pessoa que possuía um refinado conhecimento científico na área de inteligência artificial e uma capacidade incrível de motivação. Você sempre será lembrado por seus alunos.

Agradeço ao professor Horacio Hideki Yanasse por ter assumido a orientação do meu doutorado em um momento delicado, e por ter procurado sempre contribuir com meu trabalho.

Agradeço aos amigos Elcio Hideiti Shiguemori e Marco Antônio Pizani Domiciano, companheiros de projeto no Instituto de Estudos Avançados (IEAv), pela imensa ajuda para que eu pudesse realizar o doutorado e, ao mesmo tempo, cumprir minhas atividades no projeto.

Agradeço aos amigos do IEAv e do INPE que também contribuíram para a conclusão deste trabalho: Osvaldo Catsumi Imamura, Monica Maria De Marchi, Maria Helena de Faria, Maria Cristina Peloggia de Araújo, Maria José Pinto Lamosa, Orlando Zaloti Junior, João Camilo da Silva, Flávio Petersen Junior, Claudia Carraro e Caroline Aparecida Lima da Silva.

Agradeço aos amigos do cafezinho no trabalho pelos rápidos, porém, agradáveis momentos.

Agradeço aos meus amigos de república pela convivência tranquila e pelo espírito de fraternidade.

RESUMO

Este trabalho aborda o problema de planejamento automático de trajetórias bidimensionais para Veículos Aéreos Não Tripulados (VANTs) usando modelagem computacional de ambientes de navegação através de grafos de visibilidade, modelos digitais de elevação e distribuições espaciais de análises de efeitos causados pela queda destes veículos. Grafos de visibilidade construídos através de modelos digitais de elevação podem apresentar um elevado número de nós e de arestas. O primeiro objetivo deste trabalho foi o estudo da simplificação de tais grafos. Inicialmente, foi abordada a simplificação pela redução do número de vértices intermediários convexos dos obstáculos poligonais que definem os nós dos grafos de visibilidade. Foram identificados dois problemas com a aplicação de métodos convencionais para a simplificação de obstáculos poligonais: a ocultação de obstáculos; e a criação de áreas de invisibilidade, causando perda de área navegável. Visando à solução de tais problemas, foi proposto um algoritmo para simplificação de polígonos baseado na área de invisibilidade entre vértices intermediários convexos. Através de análises dos resultados obtidos, verificou-se que o algoritmo proposto permite uma redução significativa do número de vértices intermediários convexos, limitando a criação de áreas de invisibilidade e evitando a ocultação de obstáculos. Outro objetivo foi avaliar a possibilidade de simplificação direta dos grafos de visibilidade, através da eliminação de nós ou de arestas. Neste trabalho foi proposto o Algoritmo Dijkstra baseado em Elevações de terreno (ADE) para o planejamento de trajetórias livres de colisão para VANTs através de representações de ambientes de navegação definidos por grafos de visibilidade e modelos digitais de elevação. O ADE utiliza restrições cinemáticas de VANTs para o planejamento. Uma trajetória de navegação deve ser livre de colisão e dinamicamente viável, mas também é importante que a trajetória seja planejada para reduzir o risco à população em acidentes com queda de VANTs. Deste modo, um método foi proposto para modelar ambientes de navegação definidos através da avaliação dos efeitos da queda de VANTs e através de modelos digitais de elevação. Através dos resultados obtidos, pôde-se observar que os grafos de visibilidade construídos através dos ambientes modelados pelo método proposto permitem o planejamento de rotas e trajetórias para VANTs baseado em uma estimativa do número esperado de fatalidades em possíveis acidentes com queda destes veículos.

TRAJECTORY PLANNING FOR UNMANNED AERIAL VEHICLES USING COMPUTATIONAL MODELING OF NAVIGATION ENVIRONMENTS THROUGH VISIBILITY GRAPHS AND DIGITAL ELEVATION MODELS

ABSTRACT

This work focuses on the problem of automatic planning of two-dimensional trajectories for Unmanned Aerial Vehicles (UAVs) using computational modeling of navigation environments through visibility graphs, digital elevation models and spatial distributions of analysis of the effects caused by the fall of these vehicles. Visibility graphs constructed through digital elevation models can present a high number of nodes and edges. The first objective of this work was the study of the simplification of such graphs. Initially, a simplification was considered by decreasing the number of convex intermediate vertices of the polygonal obstacles that define the nodes of a visibility graph. Two problems were identified with the application of conventional methods for the simplification of polygonal obstacles: the occultation of obstacles; and the creation of invisibility areas, causing loss of navigable area. Aiming at the solution of such problems, an algorithm was proposed for the simplification of polygons based on the invisibility area among convex intermediate vertices. Analyzing the results obtained, it was verified that the proposed algorithm allows a significant reduction of the number of vertices, limiting the creation of invisibility areas and avoiding the occultation of obstacles. Another objective was to evaluate the possibility of simplifying the visibility graphs directly by eliminating nodes or edges. In this work, the Dijkstra Algorithm based on terrain Elevation (DAE) was proposed for the planning of free-collision trajectories for UAVs through the representations of navigation environments defined by visibility graphs and by digital elevation models. ADE uses kinematic constraints of UAVs for the planning. A navigation trajectory must be dynamically feasible and collision-free, but it is also important to plan the trajectory to reduce the risk to the population in accidents with falls of UAVs. This way, a method was proposed to model navigation environments defined by the evaluation of the effects of UAVs falls and through digital models of elevation. With the results obtained, it could be observed that the visibility graphs constructed by the environments modeled of the proposed method allow the planning of paths and trajectories for UAVs based on an estimation of the expected number of fatalities in possible accidents with falls of these vehicles.

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 1.1 - Exemplos de uma estação de controle e de VANTs: (a) estação de controle de um VANT MQ-1 Predator; (b) MQ-1 Predator; (c) Eagle Eye; (d) CryoWing; (e) Fire Scout; (f) Pathfinder Plus; (g) RMAX; e (h) Global Hawk.....	2
Figura 1.2 - Aumento de autonomia de VANTs.....	3
Figura 1.3 - Exemplo de uma trajetória de navegação.....	5
Figura 1.4 - Exemplo simplificado da arquitetura de um sistema de navegação com planejamento automático de trajetórias.....	7
Figura 2.1 - Antigas representações de referências e de ambientes de navegação	13
Figura 2.2 - Exemplo de um grafo de visibilidade.....	15
Figura 2.3 - Exemplo de uma (a) rota e de uma (b) trajetória planejadas para um VANT através de um grafo de visibilidade	16
Figura 2.4 - Exemplo de ambiente de navegação obtido de um modelo digital de elevação pela especificação de uma altitude de navegação e de uma altura de segurança	17
Figura 2.5 - Exemplo de determinação de um nó do grafo de visibilidade	18
Figura 2.6 - Exemplo de verificação de situação de colisão na aresta que une dois nós.....	21
Figura 2.7 - Exemplo do problema de verificação de situação de colisão	22
Figura 2.8 - Modelos digitais de elevação utilizados neste trabalho.....	24
Figura 2.9 - Ambientes de navegação obtidos com o primeiro modelo digital de elevação, considerando: $a_s = 300$ m; (a) $a_n = 1100$ m; (b) $a_n = 1300$ m; (c) $a_n = 1550$ m; e (d) $a_n = 1800$ m.....	26
Figura 2.10 - Ambientes de navegação obtidos com o segundo modelo digital de elevação, considerando: $a_s = 300$ m; (a) $a_n = 1300$ m; (b) $a_n = 1500$ m; (c) $a_n = 1900$ m; e (d) $a_n = 2200$ m.....	27
Figura 2.11 - Conjuntos de grafos de visibilidade construídos para os ambientes de navegação com: (a) $a_n = 1100$ m; (b) $a_n = 1300$ m; (c) $a_n = 1550$ m; e (d) $a_n = 1800$ m	28
Figura 2.12 - Conjuntos de grafos de visibilidade construídos para os ambientes de navegação com: (a) $a_n = 1300$ m; (b) $a_n = 1500$ m; (c) $a_n = 1900$ m; e (d) $a_n = 2200$ m	29
Figura 3.1 - Método para modelagem de ambientes de navegação através de modelos digitais de elevação e de avaliações dos efeitos da queda de VANTs.....	36
Figura 3.2 - Fase de criação da matriz de áreas urbanas	38
Figura 3.3 - Amostras de treinamento das oito classes do processo de classificação.....	40

Figura 3.4 - Resultados dos processos de criação da matriz de áreas urbanas: (a) imagem I ; (b) classificação C ; (c) representação binária U ; e (d) a matriz U	41
Figura 3.5 - Representação poligonal dos municípios da superfície do modelo digital de elevação analisado	42
Figura 3.6 - Exemplo da determinação do código m_{lc} através de cinco pontos da região representada por b_{lc}	43
Figura 3.7 - Distribuição espacial de t_m para o ambiente de navegação considerado neste trabalho.....	45
Figura 3.8 - Grades binárias criadas, considerando: (a) $a_n = 1100$ m e $l_{t_m} = 104$ h; (b) $a_n = 1300$ m e $l_{t_m} = 104$ h; (c) $a_n = 1100$ m e $l_{t_m} = 20000$ h; e (d) $a_n = 1300$ m e $l_{t_m} = 20000$ h	48
Figura 3.9 - Grafos de visibilidade construídos, considerando: (a) $d_s = 100$ m; (b) $d_s = 500$ m; (c) $d_s = 1000$ m; e $d_s = 2000$ m.....	50
Figura 3.10 - Grafos de visibilidade construídos, considerando: (a) $a_n = 2200$ m e $d_s = 100$ m; (b) $a_n = 1600$ m e $d_s = 500$ m; (c) $a_n = 1800$ m e $d_s = 1000$ m; e (d) $a_n = 2000$ m e $d_s = 2000$ m	51
Figura 3.11 - Rotas planejadas com grafos de visibilidade construídos através de: (a) grade binária obtida com a métrica t_m ; e (b) grade binária obtida sem a métrica t_m	54
Figura 3.12 - Exemplo de uma trajetória de queda de um VANT	57
Figura 3.13 - Esquema para definição de l_{d_s}	58
Figura 3.14 - Rotas planejadas através de grafos de visibilidade construídos, considerando as especificações do VANT RQ-4 Global Hawk, $a_n = 2000$ m, $a_s = 300$ m, $l_{t_m} = 200000$ h, (a) $d_s = 2000$ m e (b) $d_s = 500$ m.....	59
Figura 4.1 - Exemplo de grafo de visibilidade construído através de um modelo digital de elevação	61
Figura 4.2 - Exemplo de cálculo de envoltória convexa	63
Figura 4.3 - (a) Modelo digital de elevação e respectivo (b) ambiente de navegação considerando uma altitude de navegação de 1100 metros e uma altura de segurança de 300 metros.....	64
Figura 4.4 - (a) Modelo digital de elevação e respectivo (b) ambiente de navegação considerando uma altitude de navegação de 1500 metros e uma altura de segurança de 300 metros.....	64
Figura 4.5 - (a) Representações poligonais originais do primeiro ambiente de navegação e (b) envoltórias convexas obtidas com a aplicação do algoritmo de Graham a tais representações	65
Figura 4.6 - (a) Representações poligonais originais do segundo ambiente de navegação e (b) envoltórias convexas obtidas com a aplicação do algoritmo de Graham a tais representações	66
Figura 4.7 - Exemplo do problema de ocultação de obstáculos no planejamento de rotas	69
Figura 4.8 - (a) Representações poligonais originais do primeiro ambiente de navegação e (b) Simplificação destas representações pela	

	aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 200$ m	72
Figura 4.9 -	(a) Representações poligonais originais do primeiro ambiente de navegação e (b) Simplificação destas representações pela aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 500$ m	73
Figura 4.10 -	(a) Representações poligonais originais do primeiro ambiente de navegação e (b) Simplificação destas representações pela aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 1000$ m	74
Figura 4.11 -	(a) Representações poligonais originais do segundo ambiente de navegação e (b) Simplificação destas representações pela aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 200$ m	75
Figura 4.12 -	(a) Representações poligonais originais do segundo ambiente de navegação e (b) Simplificação destas representações pela aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 500$ m	76
Figura 4.13 -	(a) Representações poligonais originais do segundo ambiente de navegação e (b) Simplificação destas representações pela aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 1000$ m	77
Figura 4.14 -	(a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 0.05$ km ²	79
Figura 4.15 -	(a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 0.5$ km ²	80
Figura 4.16 -	(a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 5$ km ²	81
Figura 4.17 -	(a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 0.05$ km ²	82
Figura 4.18 -	(a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 0.5$ km ²	83
Figura 4.19 -	(a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 5$ km ²	84
Figura 4.20 -	Exemplo de regiões de invisibilidade geradas por dois vértices intermediários convexos v_{ei} e v_{ej}	86
Figura 4.21 -	Exemplo do problema de invisibilidade entre os vértices v_1 e v_2	88
Figura 4.22 -	(a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 5$ km ²	92

Figura 4.23 - (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 10 \text{ km}^2$	93
Figura 4.24 - (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 20 \text{ km}^2$	94
Figura 4.25 - (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 50 \text{ km}^2$	95
Figura 4.26 - (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 100 \text{ km}^2$	96
Figura 4.27 - (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 50 \text{ km}^2$	97
Figura 4.28 - (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 100 \text{ km}^2$	98
Figura 4.29 - (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 200 \text{ km}^2$	99
Figura 4.30 - (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 300 \text{ km}^2$	100
Figura 4.31 - (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 400 \text{ km}^2$	101
Figura 4.32 - Grafos de visibilidade construídos com a (a) representação original do primeiro ambiente de navegação e com as representações simplificadas, considerando: (b) $a_{ij} = 5 \text{ km}^2$; (c) $a_{ij} = 10 \text{ km}^2$; (d) $a_{ij} = 20 \text{ km}^2$; (e) $a_{ij} = 50 \text{ km}^2$; e (f) $a_{ij} = 100 \text{ km}^2$	104
Figura 4.33 - Grafos de visibilidade construídos com a (a) representação original do segundo ambiente de navegação e com as representações simplificadas, considerando: (b) $a_{ij} = 50 \text{ km}^2$; (c) $a_{ij} = 100 \text{ km}^2$; (d) $a_{ij} = 200 \text{ km}^2$; (e) $a_{ij} = 300 \text{ km}^2$; e (f) $a_{ij} = 400 \text{ km}^2$	105
Figura 4.34 - Trajetória planejada com o grafo de visibilidade original do primeiro ambiente de navegação	108
Figura 4.35 - Trajetórias planejadas com (a) o grafo de visibilidade original do primeiro ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 5 \text{ km}^2$	109
Figura 4.36 - Trajetórias planejadas com (a) o grafo de visibilidade original do primeiro ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 10 \text{ km}^2$	110
Figura 4.37 - Trajetórias planejadas com (a) o grafo de visibilidade original do primeiro ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 20 \text{ km}^2$	111

Figura 4.38 - Trajetória planejada com o grafo de visibilidade original do segundo ambiente de navegação	112
Figura 4.39 - Trajetórias planejadas com (a) o grafo de visibilidade original do segundo ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 50 \text{ km}^2$	113
Figura 4.40 - Trajetórias planejadas com (a) o grafo de visibilidade original do segundo ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 100 \text{ km}^2$...	114
Figura 4.41 - Trajetórias planejadas com (a) o grafo de visibilidade original do segundo ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 200 \text{ km}^2$...	115
Figura 4.42 - Trajetórias planejadas com (a) o grafo de visibilidade original do segundo ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 300 \text{ km}^2$...	116
Figura 4.43 - Trajetórias planejadas com (a) o grafo de visibilidade original do segundo ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 400 \text{ km}^2$...	117
Figura 4.44 - (a) Grafo de visibilidade construído com a representação original do primeiro ambiente de navegação e grafos simplificados, considerando: (b) $k = 5$ grupos; (c) $k = 10$ grupos; (d) $k = 20$ grupos; (e) $k = 30$ grupos; e (f) $k = 40$ grupos.....	121
Figura 4.45 - (a) Grafo de visibilidade construído com a representação original do segundo ambiente de navegação e grafos simplificados, considerando: (b) $k = 10$ grupos; (c) $k = 40$ grupos; e (d) $k = 100$ grupos.....	122
Figura 4.46 - Trajetórias planejadas com (a) o grafo simplificado original do primeiro ambiente de navegação e com (b) o grafo simplificado, considerando $k = 10$ grupos.....	124
Figura 4.47 - Trajetórias planejadas com (a) o grafo simplificado original do segundo ambiente de navegação e com (b) o grafo simplificado, considerando $k = 100$ grupos.....	125
Figura 5.1 - Três principais abordagens para o problema de planejamento automático de trajetórias através de grafos de visibilidade.....	128
Figura 5.2 - Exemplos de tentativas de planejamento de (b) trajetórias através de suavizações de (a) rotas planejadas.....	129
Figura 5.3 - Exemplos de trajetórias planejadas com um GVM.....	130
Figura 5.4 - Exemplo de uma k -trajetória	133
Figura 5.5 - Exemplo de uma 1-trajetória	134
Figura 5.6 - Exemplo de parte de uma 1-trajetória planejada pelo ADE	136
Figura 5.7 - Comparação entre (a) trajetória planejada pelo ADM e (b) 1-trajetória planejada pelo ADE	137
Figura 5.8 - Exemplo de verificação de colisão entre o arco de trajetória do VANT e a matriz de elevações E	140
Figura 5.9 - (a) Verificação de colisão através de um polígono que contém a trajetória do VANT entre os pontos p_{j-1} e p_j com (b) busca local na matriz de elevações E	141

Figura 5.10 - Esquema para cálculo da área do pentágono que contém a trajetória do VANT entre os pontos p_{j-1} e p_j	143
Figura 5.11 - Projeção da envergadura de um VANT	145
Figura 5.12 - Ambientes de navegação obtidos com $a_s = 300$ m, $d_s = 100$ m, (a) $a_n = 1100$ m e (b) $a_n = 1700$ m.....	146
Figura 5.13 - Comparação entre a heurística baseada em discretização do arco de trajetória e a heurística baseada em avaliação do quadrilátero de células, considerando a variação do raio de curva r_c	148
Figura 5.14 - Comparação entre a heurística baseada em discretização do arco de trajetória e a heurística baseada em avaliação do quadrilátero de células, considerando a variação da resolução r_s	148
Figura 5.15 - Trajetória planejada pelo ADE entre as coordenadas $w'_o =$ (latitudo, longitude) = (-22.91, -45.95) e $w'_d =$ (latitudo, longitude) = (-23.0, -45.735) com um grafo de visibilidade de 1489 nós e 65241 arestas e com $r_c = 400$ m	151
Figura 5.16 - Trajetória planejada pelo ADE entre as coordenadas $w'_o =$ (latitudo, longitude) = (-22.91, -45.95) e $w'_d =$ (latitudo, longitude) = (-23.0, -45.735) com um grafo de visibilidade de 1489 nós e 65241 arestas e com $r_c = 2000$ m	153
Figura 5.17 - Trajetória planejada pelo ADE entre as coordenadas $w'_o =$ (latitudo, longitude) = (-23.075, -46.072) e $w'_d =$ (latitudo, longitude) = (-22.98, -45.87) com um grafo de visibilidade de 1489 nós e 65241 arestas e com $r_c = 400$ m	155
Figura 5.18 - Trajetória planejada pelo ADE entre as coordenadas $w'_o =$ (latitudo, longitude) = (-22.99, -45.01) e $w'_d =$ (latitudo, longitude) = (-22.05, -45.05) com um grafo de visibilidade de 3937 nós e 490707 arestas e com $r_c = 1500$ m	157
Figura 5.19 - Trajetória planejada pelo ADE entre as coordenadas $w'_o =$ (latitudo, longitude) = (-22.99, -45.01) e $w'_d =$ (latitudo, longitude) = (-22.05, -45.05) com um grafo de visibilidade de 3937 nós e 490707 arestas e com $r_c = 100$ m	159
Figura 5.20 - Trajetória planejada pelo ADE entre as coordenadas $w'_o =$ (latitudo, longitude) = (-22.5, -45.05) e $w'_d =$ (latitudo, longitude) = (-22.58, -45.482) com um grafo de visibilidade de 3937 nós e 490707 arestas e com $r_c = 200$ m	161
Figura 5.21 - Trajetória planejada pelo ADE entre as coordenadas $w'_o =$ (latitudo, longitude) = (-22.09, -45.97) e $w'_d =$ (latitudo, longitude) = (-22.335, -45.13) com um grafo de visibilidade de 3937 nós e 490707 arestas e com $r_c = 1000$ m	163
Figura 5.22 - Trajetória planejada pelo ADE entre as coordenadas $w'_o =$ (latitudo, longitude) = (-22.09, -45.97) e $w'_d =$ (latitudo, longitude) = (-22.335, -45.13) com um grafo de visibilidade de 3937 nós e 490707 arestas e com $r_c = 4000$ m	165
Figura 5.23 - Aeronave simulada: Piper J3 Cub.....	168
Figura B.1 - Exemplo de uma rota mais curta, que é poligonal e cujos vértices são vértices intermediários convexos de C_o	193

Figura C.1 - Exemplos das seis classes de trajetórias de Dubins..... 195

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 2.1 - Algoritmo para determinação dos nós de grafos de visibilidade através de modelos digitais de elevação	19
Tabela 2.2 - Algoritmo para determinação das arestas de grafos de visibilidade através de modelos digitais de elevação.....	21
Tabela 2.3 - Informações dos grafos de visibilidade construídos através do primeiro modelo digital de elevação	25
Tabela 2.4 - Informações dos grafos de visibilidade construídos através do segundo modelo digital de elevação.....	25
Tabela 3.1 - Algoritmo para criação da matriz de densidades populacionais D e da matriz de códigos de municípios M	43
Tabela 3.2 - Especificações dos VANTs utilizadas neste trabalho.....	44
Tabela 3.3 - Valores mínimos e máximos de T_m para cada VANT.....	46
Tabela 3.4 - Algoritmo Dijkstra para planejamento de rotas de menor extensão entre dois nós de um grafo	53
Tabela 3.5 - Análise da segurança das rotas planejadas.....	55
Tabela 4.1 - Comparações entre a representação poligonal do primeiro ambiente de navegação e a representação obtida após a simplificação por envoltória convexa.....	67
Tabela 4.2 - Comparações entre a representação poligonal do segundo ambiente de navegação e a representação obtida após a simplificação por envoltória convexa.....	67
Tabela 4.3 - Algoritmo Visvalingam-Whyatt	70
Tabela 4.4 - Algoritmo Ramer-Douglas-Peucker.....	70
Tabela 4.5 - Comparação entre a representação poligonal do primeiro ambiente de navegação e as representações simplificadas pelo algoritmo Ramer-Douglas-Peucker.....	71
Tabela 4.6 - Comparação entre a representação poligonal do segundo ambiente de navegação e as representações simplificadas pelo algoritmo Ramer-Douglas-Peucker	71
Tabela 4.7 - Comparação entre a representação poligonal do primeiro ambiente de navegação e as representações simplificadas pelo algoritmo Visvalingam-Whyatt	78
Tabela 4.8 - Comparação entre a representação poligonal do segundo ambiente de navegação e as representações simplificadas pelo algoritmo Visvalingam-Whyatt.....	78
Tabela 4.9 - Algoritmo para simplificação de polígonos baseado em áreas de invisibilidade entre vértices intermediários convexos.....	87
Tabela 4.10 - Algoritmo para determinação do vértice intermediário convexo v_2	88

Tabela 4.11 - Algoritmo para verificação da visibilidade entre os vértices v_1 e v_2	89
Tabela 4.12 - Algoritmo para cálculo da área de invisibilidade entre os vértices v_1 e v_2	89
Tabela 4.13 - Comparação entre a representação poligonal do primeiro ambiente de navegação e as representações simplificadas pelo algoritmo proposto	91
Tabela 4.14 - Comparação entre a representação poligonal do segundo ambiente de navegação e as representações simplificadas pelo algoritmo proposto	91
Tabela 4.15 - Comparação entre os grafos construídos considerando a representação original do primeiro ambiente de navegação e as representações simplificadas pelo algoritmo proposto.....	103
Tabela 4.16 - Comparação entre os grafos construídos considerando a representação original do segundo ambiente de navegação e as representações simplificadas pelo algoritmo proposto.....	103
Tabela 4.17 - Comparação entre as trajetórias planejadas considerando o grafo de visibilidade original do primeiro ambiente de navegação e os grafos simplificados pelo algoritmo proposto	107
Tabela 4.18 - Comparação entre as trajetórias planejadas considerando o grafo de visibilidade original do segundo ambiente de navegação e os grafos simplificados pelo algoritmo proposto	107
Tabela 4.19 - Algoritmo para simplificação de grafos de visibilidade baseado em agrupamento de nós	120
Tabela 4.20 - Comparação entre o grafo de visibilidade original do primeiro ambiente de navegação e os grafos simplificados pelo algoritmo baseado em agrupamento de nós.....	120
Tabela 4.21 - Comparação entre o grafo de visibilidade original do segundo ambiente de navegação e os grafos simplificados pelo algoritmo baseado em agrupamento de nós.....	120
Tabela 5.1 - Algoritmo Dijkstra baseado em elevações para planejamento de trajetórias de entre dois nós de um grafo.....	135
Tabela 5.2 - Algoritmo Dijkstra Modificado	135
Tabela 5.3 - Heurística para verificar se um arco de a e b até s é livre de colisão	142
Tabela 5.4 - Comparação entre a heurística baseada em discretização do arco de trajetória e o método usado no ADM.....	147
Tabela 5.5 - Comparação entre a heurística baseada em discretização do arco de trajetória e a heurística baseada em avaliação do quadrilátero de células, considerando a variação do raio de curva r_c	149
Tabela 5.6 - Comparação entre a heurística baseada em discretização do arco de trajetória e a heurística baseada em avaliação do quadrilátero de células, considerando a variação da resolução r_s	150
Tabela 5.7 - Informações referentes à 1-trajetória apresentada na Figura 5.15	152

Tabela 5.8 - Informações referentes à 1-trajetória apresentada na Figura 5.16	154
Tabela 5.9 - Informações referentes à 1-trajetória apresentada na Figura 5.17	156
Tabela 5.10 - Informações referentes à 1-trajetória apresentada na Figura 5.18	158
Tabela 5.11 - Informações referentes à 1-trajetória apresentada na Figura 5.19	160
Tabela 5.12 - Informações referentes à 1-trajetória apresentada na Figura 5.20	162
Tabela 5.13 - Informações referentes à 1-trajetória apresentada na Figura 5.21	164
Tabela 5.14 - Informações referentes à 1-trajetória apresentada na Figura 5.22	166
Tabela 5.15 - Especificações da aeronave simulada	167
Tabela 5.16 - Cálculo de $r_{cmín}$ para a aeronave Piper J3 Cub	168
Tabela D.1 - Informações referentes às implementações dos princípios algoritmos propostos ou utilizados	197

LISTA DE SIGLAS E ABREVIATURAS

INPE	Instituto Nacional de Pesquisas Espaciais
VANT	Veículo Aéreo Não Tripulado
UAV	<i>Unmanned Aerial Vehicle</i>
SANT	Sistema de Aeronave Não Tripulada
DoD	<i>Department of Defense</i>
FAA	<i>Federal Aviation Administration</i>
ADM	Algoritmo Dijkstra Modificado
SRTM	<i>Shuttle Radar Topography Mission</i>
NASA	<i>National Aeronautics and Space Administration</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
ADE	Algoritmo Dijkstra baseado em Elevações do terreno
GVM	Grafo de Visibilidade Modificado
PITER	Processamento de Imagens em TEmpo Real
IEAv	Instituto de Estudos Avançados
m	metro(s)
km	quilômetro(s)
m ²	metro(s) quadrado(s)
J	Joule(s)

LISTA DE SÍMBOLOS

B	grade regular binária, que representa o ambiente de navegação
l	número da linha que indica a posição de uma célula
c	número da coluna que indica a posição de uma célula
b_{lc}	célula da grade binária B
n_{lg}	número de linhas da grade binária B
n_{cg}	número de colunas da grade binária B
E	matriz de elevações do modelo digital de elevação do ambiente de navegação
a_n	altitude de navegação
a_s	altura de segurança
r_s	resolução do modelo digital de elevações
d_s	distância de segurança entre um nó de um grafo de visibilidade e a extremidade da região não navegável que o define
d_c	distância entre o centro de uma célula binária b_{lc} e a extremidade desta célula que define um nó do grafo de visibilidade
w_{lc}	posição central de uma região representada pela célula binária b_{lc}
w'_{i_n}	nó de índice i_n do grafo de visibilidade
n_n	número de nós de um grafo de visibilidade
n_a	número de arestas de um grafo de visibilidade
\vec{u}_1	vetor unitário definido por w_{lc} e pelo vértice intermediário convexo que define w'_{i_n}
q_i	ponto da discretização da aresta que une dois nós ou coordenadas w'_r e w'_s
Δp_1	distância entre dois pontos q_{i-1} e q_i
n_q	número de parcelas ou pontos q_i da discretização para verificação de colisão
$d(w'_r, w'_s)$	distância entre dois nós w'_r e w'_s
c_{inf}	primeiro ponto inferior à esquerda do ambiente de navegação

c_{sup}	último ponto superior à direita do ambiente de navegação
t_m	intervalo de tempo mínimo aceitável entre dois acidentes com queda de VANTs
n_{ef}	número esperado de fatalidades
n_{es}	nível equivalente de segurança baseado na navegação aérea tripulada
a_e	área da superfície exposta ao acidente
d_p	densidade demográfica da superfície exposta ao acidente
p_f	probabilidade de ocorrer uma fatalidade
e_v	envergadura ou diâmetro do Veículo Aéreo Não Tripulado (VANT)
c_v	comprimento do VANT
a_p	altura média de uma pessoa
α_v	ângulo de descida ou queda do VANT, definido pelo eixo longitudinal do veículo e a superfície de impacto
v_v	velocidade do VANT
m_v	massa do VANT
p_1	parâmetro de proteção que indica quão exposta está uma pessoa ao impacto
p_2	energia de impacto necessária para uma probabilidade de fatalidade igual a 50%, com p_1 igual a 0.5
p_3	limiar da energia de impacto necessária para causar uma fatalidade para p_1 tendendo a zero
g	aceleração da gravidade
D	matriz de densidades populacionais
U	matriz de áreas urbanas
$P_{um_{lc}}$	população urbana do município de código m_{lc} da superfície representada pela célula b_{lc}
$P_{rm_{lc}}$	população rural do município de código m_{lc} da superfície representada pela célula b_{lc}
$a_{um_{lc}}$	área urbana do município de código m_{lc} da superfície representada pela célula b_{lc}

$a_{r_{m_{lc}}}$	área rural do município de código m_{lc} da superfície representada pela célula b_{lc}
m_{lc}	código do município de maior densidade populacional da superfície representada por b_{lc}
C_m	conjunto de códigos de municípios brasileiros padronizados pelo Instituto Brasileiro de Geografia e Estatística (IBGE)
l_{t_m}	limite para a métrica t_m
$r_{p_{cm_i}}$	representação poligonal do município de código c_{mi}
n_m	número de municípios em C_m
I	imagem georeferenciada da superfície do ambiente de navegação
C	matriz que corresponde à classificação da imagem I
U'	representação binária da matriz de classificação C
ξ_1	classe cultura no processo supervisionado de classificação da imagem I
ξ_2	classe água no processo supervisionado de classificação da imagem I
ξ_3	classe floresta no processo supervisionado de classificação da imagem I
ξ_4	classe solo exposto no processo supervisionado de classificação da imagem I
ξ_5	classe sombra no processo supervisionado de classificação da imagem I
ξ_6	classe nuvem no processo supervisionado de classificação da imagem I
ξ_7	classe pastagem no processo supervisionado de classificação da imagem I
ξ_8	classe área urbana no processo supervisionado de classificação da imagem I
M	matriz de códigos de municípios
$x_{i_n j_n}$	sinal binário que indica se a aresta que conecta o nó w'_{i_n} ao nó

	w'_{j_n}	pertence ou não ao caminho avaliado no grafo
$y_{i_n j_n}$		custo da aresta que conecta o w'_{i_n} ao nó w'_{j_n} do grafo
w'_o		nó ou coordenada de origem da rota ou trajetória a ser planejada no grafo
w'_d		nó ou coordenada de destino da rota ou trajetória a ser planejada no grafo
$f(a, v)$		custo da rota entre o nó de origem w'_o e o nó v
$d'(a)$		custo entre o nó a e o nó de origem w'_o
$d(a, v)$		custo da aresta que conecta o nó a ao seu nó vizinho v
l_{d_s}		limite inferior para a distância de segurança d_s
Δ_1		distância entre a região com $t_m > l_{t_m}$ e o ponto de colisão na superfície
β_v		ângulo formado pela superfície de colisão e a projeção da direção da queda de um VANT
a_{limite}		área limite utilizada no algoritmo Visvalingam-Whyatt para eliminação de vértices
a_{i_v}		área de cada sequência de três vértices de um polígono no algoritmo Visvalingam-Whyatt
n_v		número de vértices de um polígono
v_i		vértice inicial de controle utilizado no algoritmo Ramer-Douglas-Peucker
v_f		vértice final de controle utilizado no algoritmo Ramer-Douglas-Peucker
$d_{máx}$		maior distância ortogonal entre vértices v_k e o segmento de reta $\overline{v_i v_f}$, $\forall i < k < f$, utilizada no algoritmo Ramer-Douglas-Peucker
$i_{máx}$		índice do vértice com $d_{máx}$
d_{limite}		distância limite para eliminação de vértices no algoritmo Ramer-Douglas-Peucker
V_o		lista original de vértices utilizada no algoritmo baseado em áreas

	de invisibilidade entre vértices convexos
V_e	lista de vértices ainda não eliminados utilizada no algoritmo baseado em áreas de invisibilidade entre vértices convexos
v_{ei}	vértice inicial de controle utilizado no processo de eliminação do algoritmo baseado em áreas de invisibilidade entre vértices convexos
v_{ej}	vértice final de controle utilizado no processo de eliminação do algoritmo baseado em áreas de invisibilidade entre vértices convexos
a_i	área de invisibilidade entre os vértices v_{ei} e v_{ej} no algoritmo baseado em áreas de invisibilidade entre vértices convexos
a_{ti}	área total de invisibilidade formada pela eliminação de vértices no algoritmo baseado em áreas de invisibilidade entre vértices convexos
a_{il}	área de invisibilidade limitante para a eliminação de vértices no algoritmo baseado em áreas de invisibilidade entre vértices convexos
$\text{primeiro}(V_e)$	primeiro vértice da lista V_e no algoritmo baseado em áreas de invisibilidade entre vértices convexos
$\text{último}(V_e)$	último vértice da lista V_e no algoritmo baseado em áreas de invisibilidade entre vértices convexos
$\text{anterior}(v_{ek})$	vértice anterior ao vértice v_{ek} da lista V_e
$\text{próximo}(v_{ek})$	vértice posterior ao vértice v_{ek} da lista V_e
(x_v, y_v)	posição de um VANT que, neste trabalho, é uma coordenada de navegação
θ_v	ângulo formado pelo eixo x e o eixo longitudinal do VANT
ω_v	velocidade angular de um VANT
r_c	raio de curva da trajetória de um VANT
n_o	número de obstáculos de um ambiente de navegação
h_{cs}	heurística para verificação de situações de colisão no arco de

	trajetória entre as coordenadas a_e e a_s , utilizando a matriz de elevações E
a_e	coordenada de entrada em um arco definido pelo nó avaliado a no algoritmo ADE
a_s	coordenada de saída em um arco definido pelo nó avaliado a no algoritmo ADE
n_p	número de passos de verificação da heurística h_{cs}
$p_r(a)$	nó anterior ao nó avaliado a no algoritmo ADE
R	rota obtida pelo algoritmo Dijkstra original
T	1-trajetória planejada pelo algoritmo ADE
Q	conjunto de nós de um grafo
w_i	coordenada de índice i de uma 1-trajetória planejada pelo algoritmo ADE
n_t	número de coordenadas de uma 1-trajetória planejada pelo algoritmo ADE
$p(k)$	ponto que define uma k -trajetória
β	ângulo formado pelos segmentos de reta $\overline{p_r(a)a}$ e \overline{av} no algoritmo ADE
p_j	ponto da discretização de um arco de trajetória efetuada pela heurística h_{cs} , para j variando de 1 até n_p
Δp	distância entre dois pontos p_{j-1} e p_j da discretização efetuada pela heurística h_{cs}
α	ângulo utilizado para o cálculos dos pontos p_j no processo de discretização de um arco de trajetória
ψ	ângulo pertencente ao intervalo $(0, 180^\circ)$ e formado pelos segmentos de reta $\overline{cp_0}$ e $\overline{cp_{n_p}}$
θ	ângulo utilizado para indicar o sentido de discretização de um arco de trajetória
c	centro da circunferência utilizada para a definição do arco de uma 1-trajetória, considerando o nó avaliado a no algoritmo ADE

	sinal binário que indica se a conexão do nó w'_{i_n} ao nó w'_{j_n} ,
$x'_{i_n j_n}$	incluindo arcos e segmentos de reta, pertence ou não ao caminho avaliado no grafo
$y'_{i_n j_n}$	comprimento da trajetória que conecta o w'_{i_n} ao nó w'_{j_n} do grafo
A_C	área de uma célula
A_t	área do trapézio definido pelos pontos p_{j-1} , p_j , p_{j1} e $p_{(j-1)1}$ do pentágono utilizado para a verificação de situações de colisão
A_p	área do pentágono definido pelos pontos p'_{j1} , p_{j2} , p_j , p_{j-1} e $p_{(j-1)2}$ do pentágono utilizado para a verificação de situações de colisão
λ_v	ângulo de inclinação de um VANT
n_C	número de células avaliadas pela heurística h_{CS}
n_{VC}	número de verificações de colisão da heurística baseada em avaliação um quadrilátero de células
n_{Cq}	número de células de um quadrilátero
$r_{cmín}$	raio de curva mínimo da trajetória de um VANT
$v_{r_{cmín}}$	velocidade do VANT para a navegação com o $r_{cmín}$
$a_{m_{lc}}$	área total de um município de código m_{lc}
S_h	sentido horário ou anti-horário de uma sequência de três pontos
$n_{um_{lc}}$	número de células urbanas do município de código m_{lc}

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	1
2 CONSTRUÇÃO DE GRAFOS DE VISIBILIDADE ATRAVÉS DE MODELOS DIGITAIS DE ELEVAÇÃO	13
2.1. Grafos de visibilidade	14
2.2. Revisão bibliográfica e motivação	16
2.3. Algoritmos elaborados para a construção de grafos de visibilidade através de modelos digitais de elevação.....	18
3 CONSTRUÇÃO DE GRAFOS DE VISIBILIDADE COM AVALIAÇÃO DOS EFEITOS DA QUEDA DE VANTS	31
3.1. Revisão bibliográfica e motivação	31
3.2. Método para modelagem de ambientes de navegação através de modelos digitais de elevação e de análises dos efeitos da queda de VANTS..	32
3.2.1. Avaliação dos efeitos de acidentes com queda de VANTS	33
3.2.2. Modelagem do ambiente de navegação.....	35
3.2.3. Aquisição de informações geográficas (fase 1)	37
3.2.4. Criação da matriz de áreas urbanas (fase 2)	37
3.2.5. Criação da matriz de densidades populacionais (fase 3)	42
3.2.6. Aquisição de especificações do VANT (fase 4)	44
3.2.7. Aquisição de especificações de voo (fase 5).....	44
3.2.8. Criação da matriz da métrica t_m (fase 6).....	45
3.2.9. Criação da matriz de elevações (fase 7)	46
3.2.10. Criação da grade binária (fase 8)	47
3.3. Construção de grafos de visibilidade através da métrica t_m	49
3.4. Planejamento de rotas com grafos de visibilidade construídos por meio da métrica t_m	52
3.4.1. Análise das rotas planejadas.....	54
3.4.2. Limite para a distância de segurança.....	56
4 SIMPLIFICAÇÃO DE GRAFOS DE VISIBILIDADE	61
4.1. Simplificação por redução de vértices de polígonos.....	62
4.1.1. Métodos para o cálculo de envoltórias convexas	62
4.1.2. Métodos baseados em métricas para eliminação de vértices	68

4.1.3. Algoritmo para a simplificação de polígonos baseado na área de invisibilidade entre vértices intermediários convexos	85
4.2. Simplificação por eliminação de nós ou de arestas de grafos.....	118
5 PLANEJAMENTO AUTOMÁTICO DE TRAJETÓRIAS DE NAVEGAÇÃO PARA VANTS	127
5.1. Revisão bibliográfica e motivação	127
5.2. Algoritmo Dijkstra baseado em elevações de terreno	132
5.3. Heurística de busca local baseada em grades regulares	139
5.4. Experimentos com o ADE	145
5.5. Simulações de Navegações por Trajetórias Planejadas pelo ADE	167
6 CONCLUSÕES	171
REFERÊNCIAS BIBLIOGRÁFICAS.....	175
APÊNDICE A - DEFINIÇÃO FORMAL DE UMA TRAJETÓRIA DE NAVEGAÇÃO DINAMICAMENTE VIÁVEL.....	189
APÊNDICE B - PROVA DA PRINCIPAL PROPRIEDADE DE GRAFOS DE VISIBILIDADE.....	191
APÊNDICE C - DEFINIÇÃO FORMAL DE UMA TRAJETÓRIA DE DUBINS	195
APÊNDICE D - INFORMAÇÕES SOBRE IMPLEMENTAÇÕES COMPUTACIONAIS DE ALGORITMOS	197

1 INTRODUÇÃO

Veículos Aéreos Não Tripulados (VANTs) têm sido utilizados com êxito em diversas aplicações civis e militares, como: no reconhecimento em regiões de conflito armado, destacando-se o RQ-4 Global Hawk, o MQ-1 Predator e o RQ-2 Pioneer; na agricultura de precisão, destacando-se o RMAX; no monitoramento ambiental, destacando-se o Aerosonde; nas telecomunicações; na exploração mineral; no sensoriamento remoto, destacando-se o CryoWing e o Pathfinder Plus; na divulgação e cobertura de eventos; na vigilância de regiões; no apoio a atividades policiais; no apoio a atividades de resgate, destacando-se o Eagle Eye; na entrega de medicamentos em situações de emergência; etc. A Figura 1.1 apresenta um exemplo de estação de controle para VANT e alguns exemplos de VANTs citados anteriormente.

A utilização de VANTs justifica-se principalmente por fatores econômicos e de segurança. VANTs são utilizados na realização de tarefas que teriam um custo operacional mais elevado se fossem realizadas por aeronaves tripuladas: mapeamento de regiões de interesse; vigilância de rodovias e estradas; etc. O custo operacional corresponde ao custo do combustível consumido e ao custo de manutenção do veículo referentes à uma hora de voo. Outro fato importante, é que o custo de aquisição de um VANT é geralmente inferior ao custo de uma aeronave tripulada que pode executar a mesma tarefa. Com relação aos fatores de segurança, estes veículos também são destinados à realização de certas tarefas que colocariam em risco uma tripulação: monitoramento de áreas de conflito armado; auxílio ao resgate sob condições adversas ao voo; etc.

VANTs classificam-se, quanto ao nível de autonomia, em sistemas remotamente pilotados, remotamente operados, e parcialmente autônomos (DALAMAGKIDIS et al., 2008a). A primeira classe corresponde àqueles veículos controlados remotamente por pilotos com certificação. A segunda classe define os VANTs monitorados remotamente por operadores. Os veículos desta classe podem navegar autonomamente por uma rota pré-planejada. A

terceira classe é composta por veículos que possuem certo grau de autonomia para a realização das tarefas atribuídas a eles. Os veículos desta última classe podem: planejar suas próprias trajetórias de navegação; detectar e rastrear objetos; tomar decisões, caso seja detectada, por exemplo, a necessidade de reabastecimento ou possíveis falhas; cooperar com outros VANTs; etc.

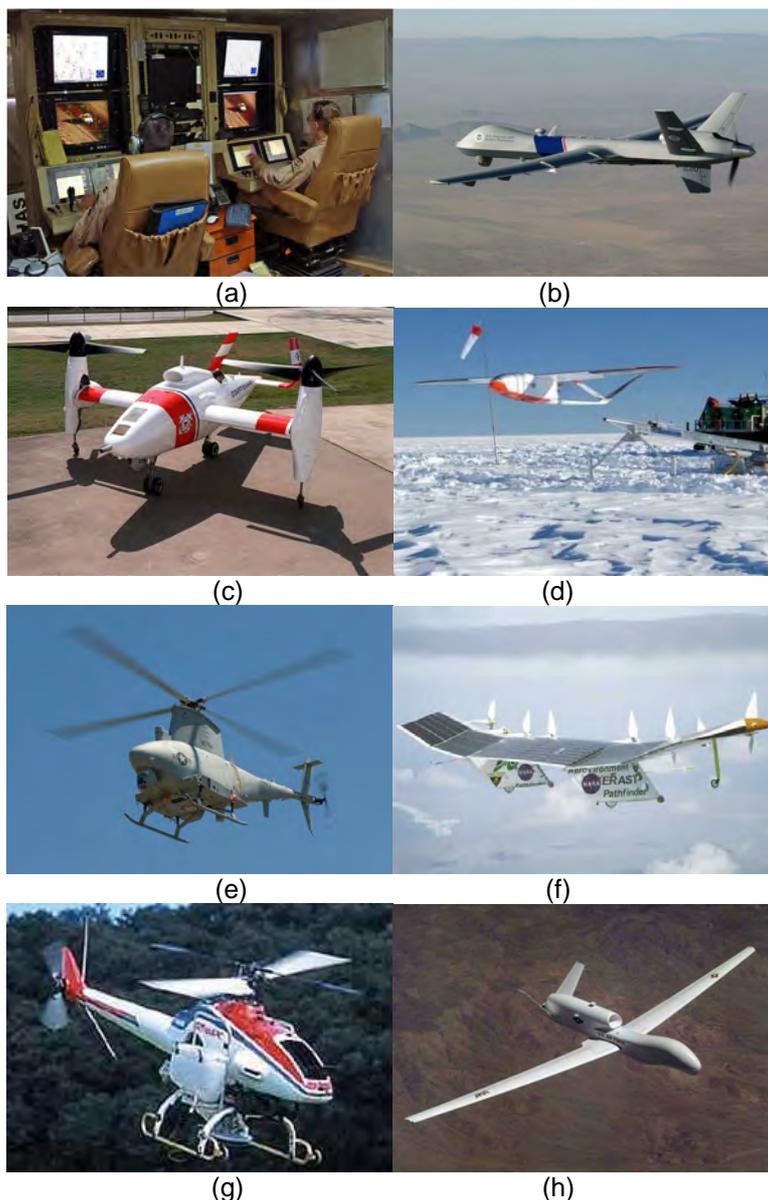


Figura 1.1 - Exemplos de uma estação de controle e de VANTs: (a) estação de controle de um VANT MQ-1 Predator; (b) MQ-1 Predator; (c) Eagle Eye; (d) CryoWing; (e) Fire Scout; (f) Pathfinder Plus; (g) RMAX; e (h) Global Hawk.

A denominação de Sistema de Aeronave Não Tripulada (SANT) foi introduzida pelo Department of Defense (DoD) dos Estados Unidos da América e adotada pela Federal Aviation Administration (FAA) para definir o conjunto composto por pelo menos: um VANT, que é a plataforma; uma estação de controle; um enlace de dados (*data link*); e outros equipamentos de suporte. Na estação de controle, trabalham técnicos ou pilotos que têm a função de controlar e ou monitorar o VANT. Um exemplo de estação de controle é apresentado na Figura 1.1a. O *data link* é o meio através do qual dados são transmitidos entre o VANT e a estação de controle.

Atualmente, um dos principais focos das pesquisas relacionadas a VANTs é o aumento de autonomia destes veículos (LUDINGTON et al., 2006; GROCHOLSKY et al., 2006). O aumento de autonomia consiste na diminuição do grau de dependência do VANT de operadores, transferindo parte do processo de tomada de decisão do operador para o próprio veículo, como apresentado no esquema da Figura 1.2.

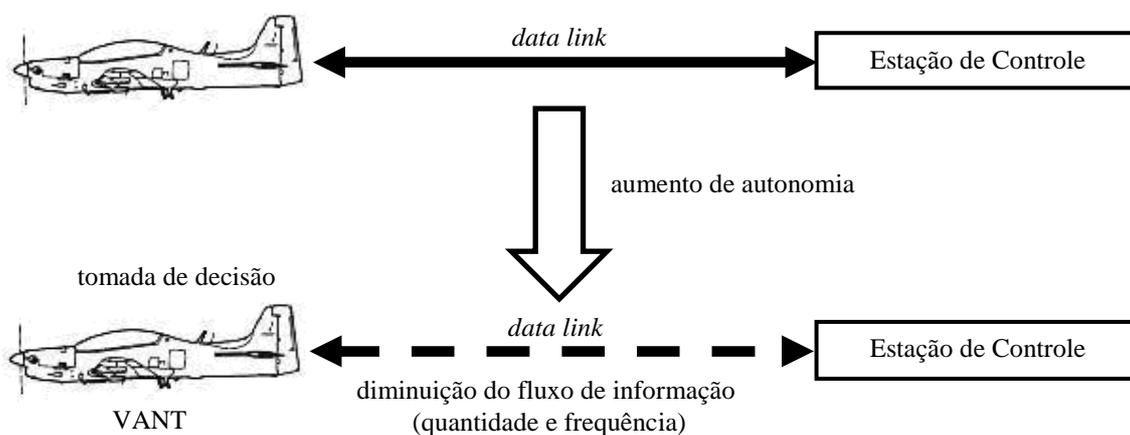


Figura 1.2 – Aumento de autonomia de VANTs.

O objetivo do aumento de autonomia é proporcionar novas capacidades aos VANTs como, por exemplo, a navegação baseada em visão computacional, a navegação à baixa altitude, o rastreamento e o reconhecimento visual de

objetos durante operações de vigilância, o reabastecimento autônomo, a comunicação e a realização de tarefas juntamente com outros VANTs, o combate aéreo, etc. A área de ação de um VANT é limitada pelo alcance do data link, portanto, observa-se que a transferência de uma parcela do processo de tomada de decisão para o VANT é uma solução para alguns dos problemas gerados pelo alcance do data link. Outro ganho importante com a diminuição do fluxo de comunicação entre a estação de controle e o VANT é a diminuição da exposição da informação trafegada no data link a uma possível interferência eletromagnética (WENZHUN et al., 2008).

O cálculo ou planejamento automático de trajetórias de navegação é essencial para a implementação da maioria das novas capacidades pretendidas com o aumento de autonomia de VANTs.

Este trabalho está relacionado com o planejamento de trajetórias para VANTs, considerando constantes o módulo da velocidade de navegação e a altitude de navegação do VANT. Outra consideração deste trabalho é que as trajetórias planejadas devem apresentar a menor extensão possível entre duas coordenadas do ambiente de navegação com obstáculos poligonais estáticos. A restrição de menor extensão visa economia de combustível e manutenção do veículo.

Devido às restrições dinâmicas e cinemáticas do veículo, o movimento ou trajetória de navegação para um VANT com módulo constante da velocidade é uma sequência de coordenadas de navegação (*waypoints*), segmentos de reta e arcos que permitem ao VANT navegar entre uma coordenada de origem e uma coordenada de destino do ambiente de navegação (ANDERSON et al., 2005; KUWATA e HOW, 2004). A trajetória deve ser segura e dinamicamente viável, isto é, a trajetória deve ser livre de colisão, isto é, ela deve permitir a navegação segura do VANT, evitando as regiões não navegáveis do ambiente que poderiam, por exemplo, causar situações de colisão do veículo com obstáculos e não deve violar as restrições cinemáticas do veículo (velocidade,

raio de curva, etc). O arco é definido pelo raio de curva da trajetória do veículo. Uma definição formal de uma trajetória dinamicamente viável foi proposta em (ANDERSON et al., 2005) e é apresentada no Apêndice A. Um exemplo de uma trajetória de navegação para um VANT é apresentado na Figura 1.3, considerando constantes a altitude de navegação e o raio de curva. Neste exemplo, a trajetória é a sequência de coordenadas representadas por círculos vermelhos e conectadas por segmentos de reta e arcos vermelhos. Os arcos vermelhos indicam as curvas que o veículo deve executar e são definidas pelo raio de curva. Os polígonos pretos indicam as regiões não navegáveis.

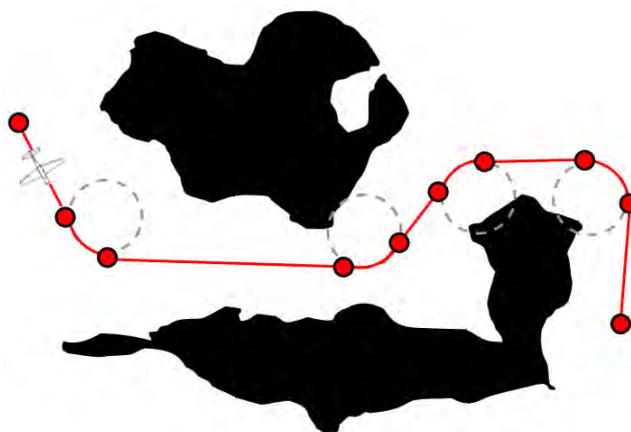


Figura 1.3 – Exemplo de uma trajetória de navegação.

Em um sistema de navegação autônoma de um VANT, uma trajetória é planejada e enviada para o sistema de controle do veículo, como uma referência para o controle do voo. Através de informações obtidas por sensores, o sistema de controle envia informações de controle para os atuadores do veículo, com o objetivo de fazer com que o VANT navegue pela trajetória planejada. A Figura 1.4 apresenta um exemplo simplificado da arquitetura de um sistema de planejamento de trajetórias acoplado a um sistema de controle de um VANT. Esta arquitetura, cujos sistemas são todos embarcados no VANT, foi baseada na arquitetura de planejamento de trajetórias apresentada em (GOERZEN et al., 2010). É importante ser

mencionado que a arquitetura de um sistema de navegação com planejamento de trajetórias pode variar de acordo com a aplicação do mesmo.

Neste trabalho, considera-se que representação computacional do ambiente de navegação é construída antes do voo, embora possa ser construída em tempo real através de informações obtidas por sensores. Portanto, o custo computacional para a construção da representação não é um fator crítico.

Entretanto, o planejamento de trajetórias de navegação deve ser realizado em tempo real, o que impõe a escolha de um método de planejamento com a menor complexidade computacional possível. Uma aplicação do planejamento em tempo real seria no rastreamento de alvos móveis. Durante o procedimento de rastreamento, trajetórias seriam planejadas entre a posição do VANT e um ponto de reabastecimento, e utilizadas para a estimação do momento de reabastecimento de combustível. Outra aplicação seria na correção da navegação devido a problemas de estimação da posição do VANT. Neste caso, a trajetória deve ser planejada no menor intervalo de tempo possível para evitar, por exemplo, uma colisão do veículo com algum obstáculo do ambiente de navegação.

Há diferentes abordagens, métodos e representações computacionais para o planejamento automático de uma trajetória de navegação. Revisões detalhadas destas abordagens, métodos e representações são apresentadas em (GOERZEN et al., 2010; LAVALLE, 2006; TOZOUR, 2003; LATOMBE, 1991).

A eficiência do método adotado para o planejamento automático de trajetórias depende principalmente da representação computacional adotada para o ambiente de navegação (TOZOUR, 2003). A representação computacional do ambiente de navegação é um modo de modelagem das regiões não navegáveis e um modo de discretização das regiões navegáveis do mesmo. São exemplos de representações computacionais de regiões navegáveis para planejamento de trajetórias: as grades regulares (COXETER, 1973; LATOMBE, 1991); as árvores de quatro regiões (quadrees) (FINKEL e BENTLEY, 1974);

as árvores aleatórias de rápida exploração, tradução de Rapidly-exploring Randon Trees (RRTs) (LAVALLE, 2006; FERGUSON e STENTZ, 2006); os diagramas de Voronoi (VORONOI, 1908; BHATTACHARYA e GAVRILOVA, 2008); os grafos de visibilidade (NILSSON, 1969; LOZANO-PÉREZ e WESLEY, 1979; TSOURDOS et al., 2011); as malhas de navegação (TOZOUR, 2003); os mapas de rotas probabilísticos (KAVRAKI et al., 1996; JUN e D'ANDREA, 2003); e os campos potenciais (KHATIB, 1986); etc. A escolha mais eficiente do modo de representação depende das considerações ou restrições da aplicação.

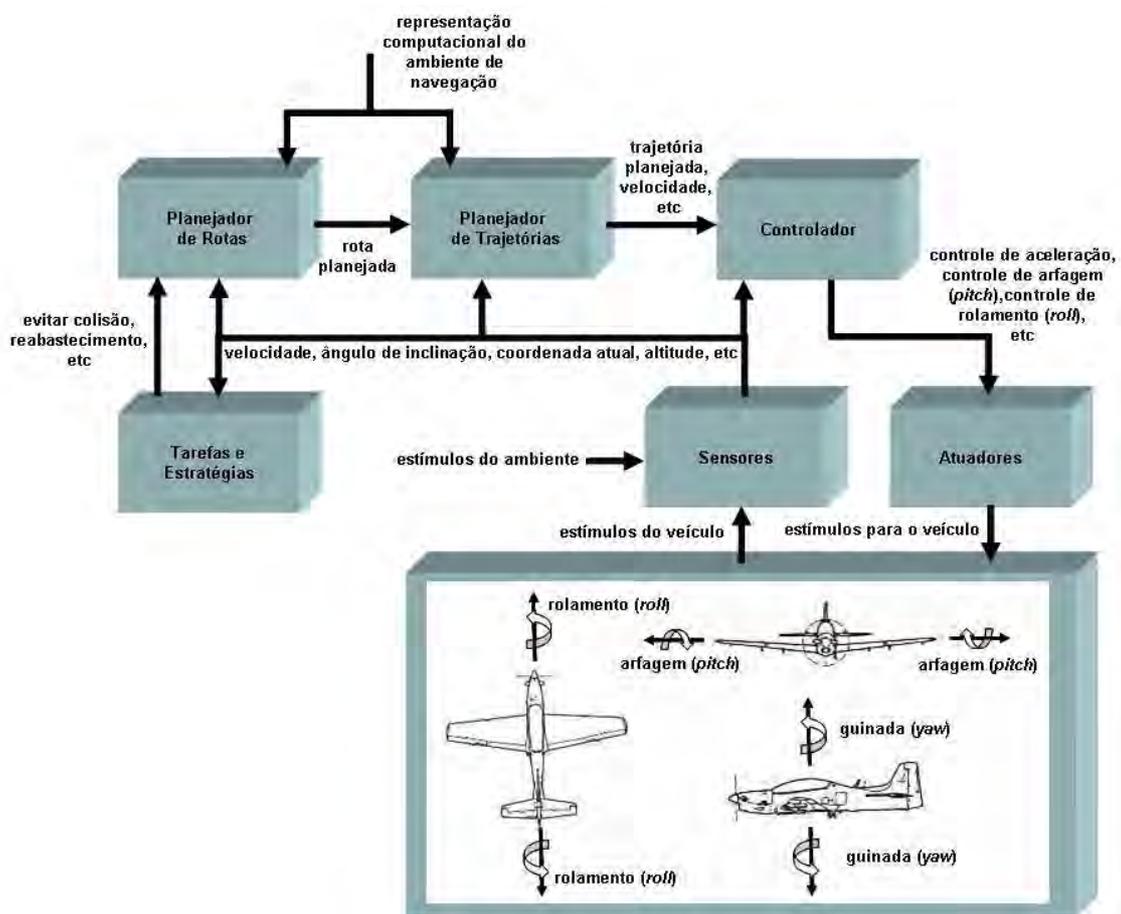


Figura 1.4 – Exemplo simplificado da arquitetura de um sistema de navegação com planejamento automático de trajetórias.

Este trabalho aborda o planejamento automático de trajetórias bidimensionais para VANTs, considerando o ambiente de navegação representado por grafos de visibilidade, modelos digitais de elevação e pela avaliação de riscos à população em acidentes com queda de VANTs.

Os grafos de visibilidade foram adotados por permitirem o planejamento de rotas de menor extensão possível entre quaisquer coordenadas navegáveis de um ambiente de navegação (DE BERG et al., 2000). Uma rota é definida neste trabalho como a estrutura através da qual uma trajetória é gerada com a associação de restrições cinemáticas do veículo. Neste trabalho, os grafos de visibilidade foram construídos inicialmente através de grades regulares definidas por modelos digitais de elevação do ambiente de navegação, que representam as regiões não navegáveis de acordo com a especificação de uma altitude de navegação para um VANT.

O primeiro problema verificado foi o elevado número de nós e de arestas dos grafos de visibilidade construídos (MEDEIROS e SILVA, 2008). Como os nós de um grafo de visibilidade são definidos pelos vértices das representações poligonais de regiões não navegáveis, o primeiro objetivo deste trabalho foi estudar a simplificação destas representações e avaliar também a possibilidade da simplificação dos grafos de visibilidade, através da eliminação direta de nós ou de arestas.

Assim como a representação do ambiente de navegação, a escolha do método de planejamento de trajetórias está relacionada com as restrições da aplicação (GOERZEN et al., 2010; SATHYARAJ et al., 2008). São exemplos de métodos usados para o planejamento de trajetórias para VANTs: o algoritmo Dijkstra clássico (DIJKSTRA, 1959) e algumas variações deste, como o algoritmo Dijkstra proposto em (KWATA e HOW, 2004); o algoritmo A* (HART et al., 1968) e suas variações (FERGUSON et al., 2005), como o algoritmo A* dinâmico (D*) (STENTZ, 1995); os algoritmos evolutivos (HOLLAND, 1975; JIA e VAGNERS, 2004; NIKOLOS et al., 2003; AHN e RAMAKRISHNA, 2002);

redes neurais artificiais (WANG et al., 2007; AHN e RAMAKRISHNA, 2001; ARAÚJO et al., 2001); colônias de formigas artificiais (DORIGO et al., 1996; DORIGO et al., 1999); adaptações do algoritmo Simplex (DANTZIG, 1949); etc.

O algoritmo Dijkstra clássico garante a solução ótima do problema de planejamento de rotas quando o ambiente de navegação é representado por grafos com arestas de custos não negativos (KONG e METTLER, 2011; GOERZEN et al., 2010; XIAO-WEI et al., 2010; LAVALLE, 2006; KWATA e HOW, 2004). O planejamento de trajetórias pelo algoritmo Dijkstra consiste na solução de um problema de menor caminho entre dois nós de um grafo. Deste modo, a aplicação do algoritmo Dijkstra clássico garante que uma rota planejada através de um grafo de visibilidade tenha a menor extensão.

Inicialmente, foi analisada a utilização do Algoritmo Dijkstra Modificado (ADM) proposto em (KWATA e HOW, 2004), que soluciona o problema de planejamento de trajetórias para VANTs baseado em grafos de visibilidade construídos através de representações retangulares de obstáculos, isto é, de regiões não navegáveis de um ambiente de navegação. Analisando o ADM, constatou-se que o mesmo poderia ser melhorado através da alteração do tipo de trajetória planejada e do modo de verificação de situações de colisão utilizado. Então, o terceiro objetivo deste trabalho foi propor um algoritmo que consiste em uma adaptação do algoritmo ADM para aplicação em ambientes de navegação que apresentam regiões não navegáveis definidas por modelos digitais de elevação.

A aplicação dos métodos de planejamento de trajetórias está focada em evitar situações de colisão do VANT com obstáculos do ambiente de navegação e em evitar situações de colisão entre VANTs dispostos em agrupamentos (GOERZEN et al., 2010; LAVALLE 2006). Porém, estes métodos não consideram a avaliação de riscos à população em acidentes com queda de VANTs (MEDEIROS e SILVA, 2010a). Portanto, o último objetivo deste trabalho foi unir o problema de planejamento de trajetórias com esta avaliação

de risco, visando à redução de possíveis fatalidades em acidentes com impacto de VANTs na superfície sobrevoada. Esta união foi alcançada por meio da modelagem de ambientes de navegação baseada em modelos digitais de elevação e na avaliação dos efeitos da queda de VANTs, e por meio da construção de grafos de visibilidade através dos resultados destas modelagens.

Assim, de modo resumido, este trabalho possui os seguintes objetivos:

- a) o estudo do problema de simplificação de grafos de visibilidade através da simplificação das representações poligonais dos obstáculos do ambiente de navegação ou através da eliminação direta de nós ou de arestas de tais grafos;
- b) a proposta de um novo algoritmo para planejamento de trajetórias para VANTs, que consiste em uma adaptação do algoritmo ADM para aplicação em ambientes de navegação que apresentam regiões não navegáveis definidas por modelos digitais de elevação; e
- c) a união do problema de planejamento de trajetórias com uma métrica para a avaliação de riscos à população em acidentes com queda de VANTs, visando à redução de possíveis fatalidades nestes tipos de acidentes.

As contribuições deste trabalho são descritas nos Capítulos 2 a 5. O Capítulo 2 deste trabalho apresenta o estudo da representação computacional de regiões navegáveis através de grafos de visibilidade construídos por meio de modelos digitais de elevação. No Capítulo 2 são descritas as seguintes contribuições: dois algoritmos para a construção de grafos de visibilidade para o planejamento de rotas e trajetórias para VANTs; e um lema para a verificação de situações de colisões de um VANT com elevações da superfície de um ambiente de navegação.

A contribuição descrita no Capítulo 3 é um método proposto para a modelagem de ambientes de navegação através da união de modelos digitais de elevação com um modo de análise dos efeitos causados pela queda de VANTs.

O Capítulo 4 apresenta o estudo do problema de simplificação de grafos de visibilidade. A contribuição descrita no Capítulo 4 é um algoritmo proposto para a simplificação de grafos de visibilidade baseado na área de invisibilidade gerada por vértices das representações poligonais das regiões não navegáveis de ambientes de navegação.

No Capítulo 5, é proposta uma adaptação do algoritmo Dijkstra para a solução do problema de planejamento automático de trajetórias para VANTs, considerando as regiões não navegáveis de um ambiente de navegação definidas por modelos digitais de elevação e as regiões navegáveis representadas por grafos de visibilidade. Outra contribuição do Capítulo 5 é um lema que garante que toda possível situação de colisão em uma trajetória curvilínea de um VANT pode ser verificada através de uma discretização de tal trajetória em seções, do encapsulamento destas seções em pentágonos e através da aplicação do lema proposto do Capítulo 2 para detecção de colisões nestas seções encapsuladas.

Conclusões finais baseadas no estudo desenvolvido e nos resultados obtidos são apresentadas no Capítulo 6.

2 CONSTRUÇÃO DE GRAFOS DE VISIBILIDADE ATRAVÉS DE MODELOS DIGITAIS DE ELEVAÇÃO

A Figura 2.1a apresenta uma ilustração de uma representação das estrelas criada por Tycho Brahe e datada do século XVI (BERGAMINI, 1970). A Figura 2.1b apresenta uma carta portuguesa para a navegação marítima elaborada pelo cartógrafo Pedro Reinel e datada de 1504 (WIKIPEDIA, 2011a). Representações de ambientes de navegação ou de referências para auxílio à navegação são desenvolvidas e utilizadas desde tempos remotos.

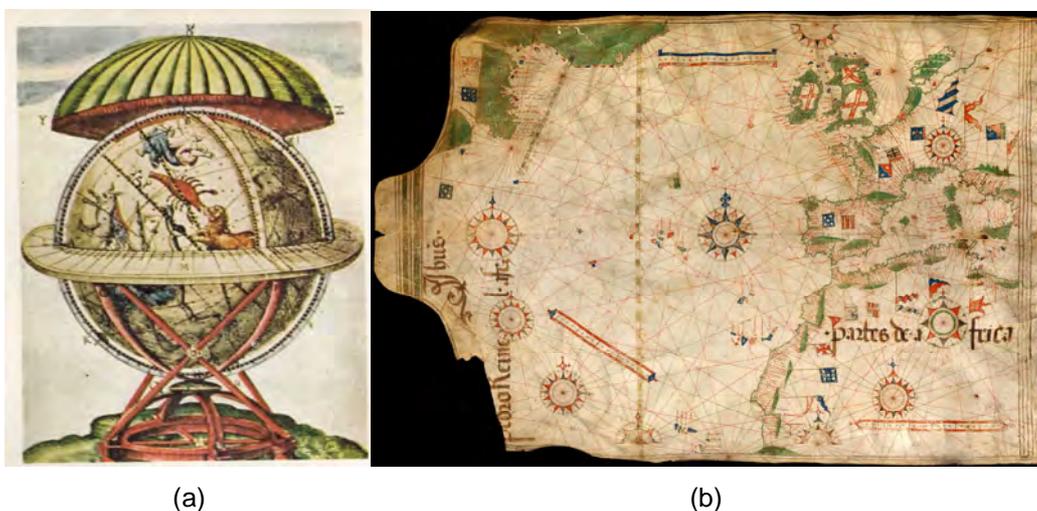


Figura 2.1 – Antigas representações de referências e de ambientes de navegação.

Fontes: Bergamini (1970) e Wikipedia (2011a)

Neste trabalho, grafos de visibilidade são utilizados como um modo de representação e de discretização das regiões navegáveis de ambientes de navegação aérea. Dois tipos de representações de regiões não navegáveis são abordados neste trabalho. Inicialmente, as regiões não navegáveis são definidas por modelos digitais de elevação de acordo com a especificação de uma altitude de navegação para um VANT. A segunda abordagem foi mesclar as informações da primeira abordagem com a avaliação dos riscos à população causados por acidentes com queda de VANTs, isto é, com o impacto de VANTs na superfície sobrevoada.

Neste capítulo, são apresentados métodos para a construção de grafos de visibilidade através de modelos digitais de elevação. A segunda abordagem é descrita no Capítulo 3.

2.1. Grafos de visibilidade

Um grafo de visibilidade é um conjunto de nós definidos pelos vértices intermediários convexos de polígonos e conectados por arestas que não interceptam tais polígonos. Assim, dois nós são visíveis se há uma aresta que os conecta e não intercepta tais polígonos (NILSSON, 1969). Supondo um polígono com vértices numerados em sentido horário, neste trabalho, um vértice intermediário convexo é definido como o vértice v_{i_v} da sequência $v_{i_v-1}v_{i_v}v_{i_v+1}$ que está no sentido horário. O cálculo do sentido horário ou anti-horário (s_h) entre três pontos pode ser obtido através da seguinte equação apresentada em (BOURKE, 1987)

$$s_h = (y_{i_v} - y_{i_v-1})(x_{i_v+1} - x_{i_v-1}) - (x_{i_v} - x_{i_v-1})(y_{i_v+1} - y_{i_v-1}) \quad (2.1)$$

Em que: (x_{i_v-1}, y_{i_v-1}) , (x_{i_v}, y_{i_v}) e (x_{i_v+1}, y_{i_v+1}) são as componentes dos respectivos vértices v_{i_v-1} , v_{i_v} e v_{i_v+1} ; $s_h > 0$ indica que a sequência $v_{i_v-1}v_{i_v}v_{i_v+1}$ está no sentido horário; $s_h < 0$ indica que a sequência $v_{i_v-1}v_{i_v}v_{i_v+1}$ está no sentido anti-horário; e $s_h = 0$ indica que os três vértices são colineares.

Em problemas de planejamento de rotas e trajetórias, grafos de visibilidade são usados para representar as regiões navegáveis de ambientes de navegação. Estes grafos possuem a propriedade de permitir o planejamento de rotas de menor extensão entre quaisquer duas coordenadas navegáveis de um ambiente de navegação com obstáculos poligonais (DE BERG et al., 2000). Uma rota de navegação é definida como uma sequência de coordenadas de navegação conectadas por segmentos de reta que não interceptam regiões não navegáveis. Portanto, considerando as regiões navegáveis representadas por um grafo de visibilidade, uma rota é o menor caminho possível entre dois

nós deste grafo. Uma prova desta propriedade é apresentada no Apêndice B. Para esta classe de problemas, os nós de um grafo de visibilidade são coordenadas de navegação e os polígonos são os obstáculos ou regiões não navegáveis do ambiente de navegação. Os nós são definidos pelos vértices intermediários convexos das regiões não navegáveis, considerando uma distância de segurança (d_s) destes vértices. A Figura 2.2 apresenta um exemplo de grafo de visibilidade. Nesta Figura, os polígonos pretos representam as regiões não navegáveis, os nós e arestas do grafo de visibilidade são representados na cor cinza.

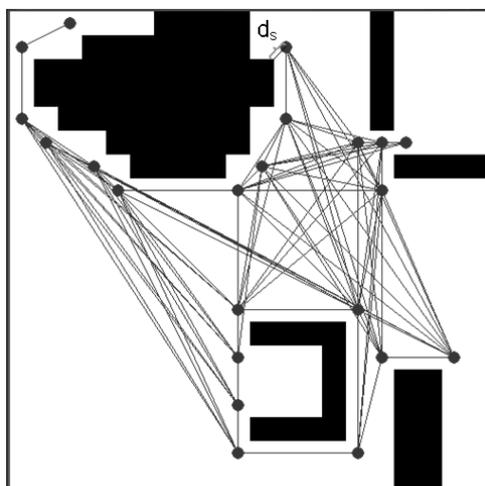


Figura 2.2 – Exemplo de um grafo de visibilidade.

Fonte: Medeiros e Silva (2010a)

A Figura 2.3a apresenta um exemplo de rota planejada através de um grafo de visibilidade. A rota é representada na cor azul. Embora uma rota seja livre de colisão, ela não está associada com as restrições cinemáticas do VANT. A rota é a estrutura com a qual a trajetória pode ser planejada, associando as restrições cinemáticas de um VANT. Um exemplo de trajetória planejada através de uma rota para um VANT é apresentado na Figura 2.3b. A trajetória é representada na cor vermelha. Uma explicação mais detalhada do processo de planejamento de trajetórias através de grafos de visibilidade é apresentada no Capítulo 3.

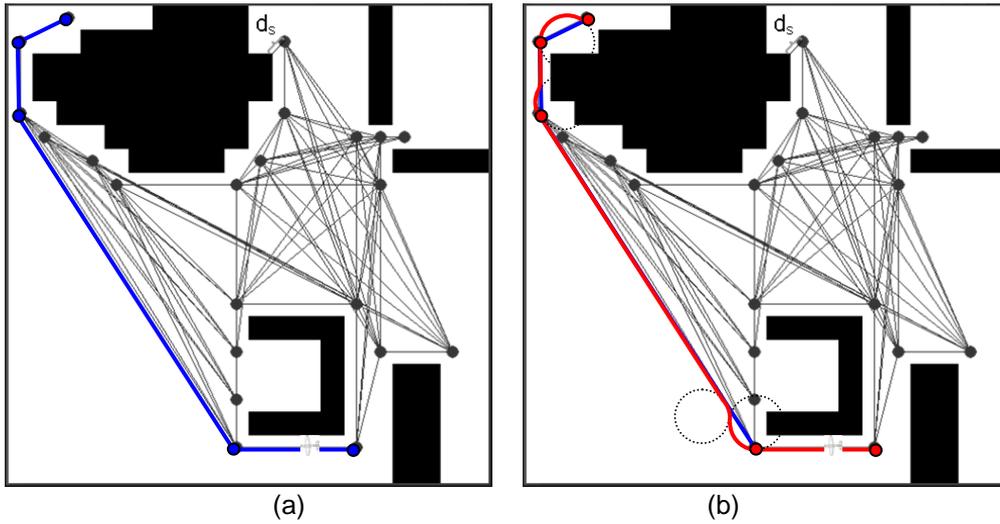


Figura 2.3 – Exemplo de uma (a) rota e de uma (b) trajetória planejadas para um VANT através de um grafo de visibilidade.

2.2. Revisão bibliográfica e motivação

Neste trabalho, um ambiente de navegação pode ser compreendido como uma superfície com altitude constante e representado inicialmente por uma grade regular binária B . Cada região b_{lc} corresponde a uma célula e_{lc} da matriz de elevações E do respectivo modelo digital de elevações. As regiões não navegáveis b_{lc} são definidas pelas células e_{lc} com elevação superior ou igual à subtração de uma altitude de navegação (a_n) por uma altura de segurança (a_s), como pode ser verificado na Figura 2.4. Assim, o valor de uma célula b_{lc} é definido por

$$b_{lc} = \begin{cases} 1, & \text{para } e_{lc} \geq (a_n - a_s) \\ 0, & \text{para } e_{lc} < (a_n - a_s) \end{cases} \quad (2.2)$$

Em que: l é o índice de linhas da grade, para l variando de 1 a n_{lg} ; e c é o índice de uma coluna da grade, para c variando de 1 até n_{cg} .

A Figura 2.4 apresenta um exemplo de ambiente de navegação obtido a partir das especificações de $a_n = 1100$ metros (m) e de $a_s = 300$ m.

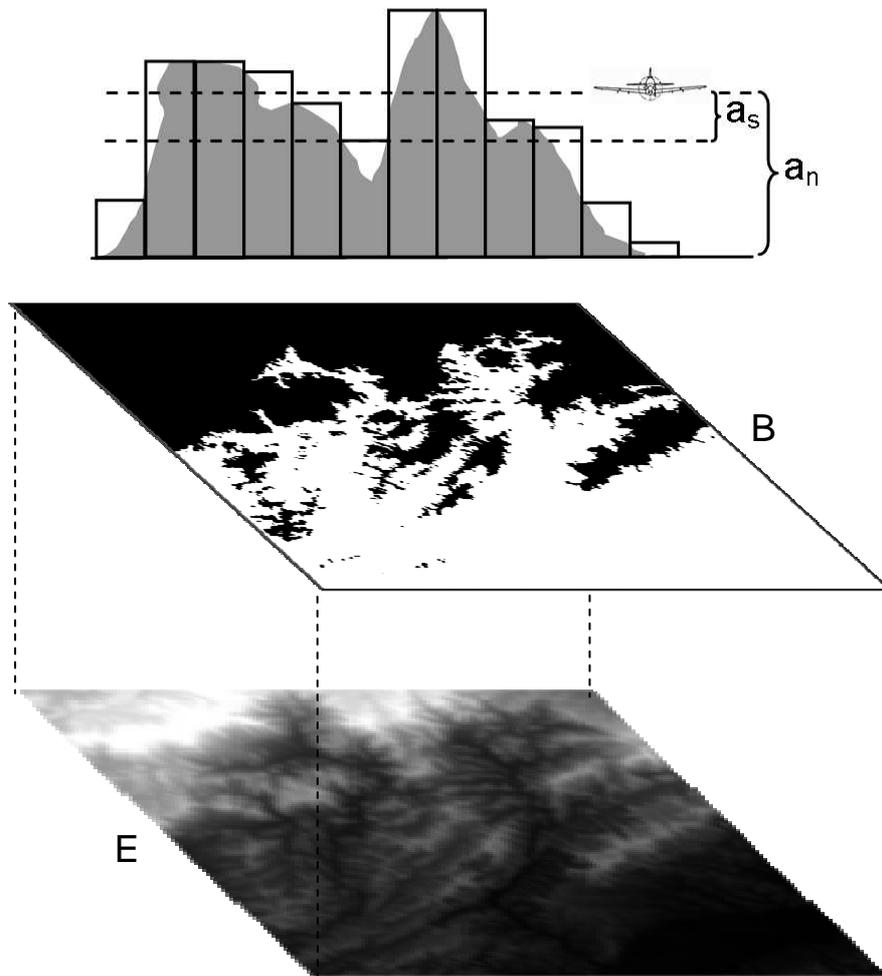


Figura 2.4 – Exemplo de ambiente de navegação obtido de um modelo digital de elevação pela especificação de uma altitude de navegação e de uma altura de segurança.

A fim de aumentar a segurança das rotas e trajetórias planejadas para os ambientes de navegação obtidos pela aplicação da Equação 2.2, uma envoltória de segurança é criada nas regiões não navegáveis destes ambientes. Esta envoltória é uma camada de uma única célula que encobre todas as regiões não navegáveis e também é considerada não navegável. Um exemplo desta envoltória é representado na Figura 2.6 por células cinza.

As regiões navegáveis de um único ambiente de navegação são representadas neste trabalho por um conjunto de grafos de visibilidade. Em

(GHOSH e GOSWAMI, 2009) é apresentada uma revisão de algoritmos para a construção destes grafos. A aplicação dos algoritmos mencionados em (GHOSH e GOSWAMI, 2009) requer a representação poligonal dos obstáculos da grade B ou o processamento destas representações poligonais, como a triangulação, para a determinação das arestas dos grafos de visibilidade. Porém, em (DE FLORIANI e MAGILLO, 2003) são apresentados algoritmos para cálculo da visibilidade entre dois pontos de um modelo digital de elevação. Assim, constatou-se que seria possível determinar arestas livres de colisão, verificando somente as células de B interceptadas por tais arestas associadas à envergadura de um VANT. Isto motivou a elaboração de dois algoritmos para a construção de grafos de visibilidade diretamente de modelos digitais de elevação. Estes dois algoritmos foram publicados em (MEDEIROS e SILVA, 2008).

2.3. Algoritmos elaborados para a construção de grafos de visibilidade através de modelos digitais de elevação

O primeiro algoritmo elaborado determina os nós de grafos de visibilidade, que são coordenadas de navegação definidas por uma coordenada geográfica latitude e por uma coordenada geográfica longitude, que pertencem ao eixo que intercepta a extremidade convexa e o centro de uma região não navegável b_{lc} . Como mencionado anteriormente, a distância entre o nó e o vértice intermediário convexo que o define, é a distância d_s . Um esquema de determinação de um nó w'_{i_n} é apresentado na Figura 2.5.

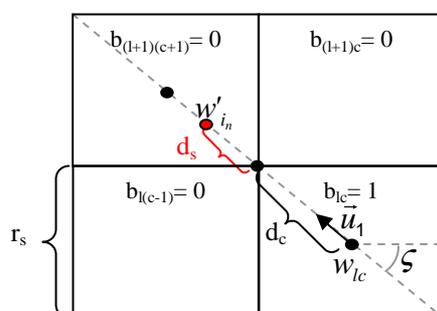


Figura 2.5 – Exemplo de determinação de um nó do grafo de visibilidade

O nó w'_{i_n} é definido por

$$w'_{i_n} = w_{l_c} + \left(\frac{r_s}{2\text{sen}(\zeta)} + d_s \right) \vec{u}_1 \quad (2.3)$$

Em que: w_{l_c} é a coordenada central de b_{l_c} ; r_s é a resolução de uma grade regular que, neste trabalho, é definida como o comprimento do menor lado de uma célula regular; e \vec{u}_1 é um vetor unitário definido pelo centro de b_{l_c} e pelo vértice intermediário convexo que define w'_{i_n} .

Tabela 2.1 – Algoritmo para determinação dos nós de grafos de visibilidade através de modelos digitais de elevação.

Índice	Algoritmo para Determinação de Nós de Grafos de Visibilidade
1	$i_n \leftarrow 0$
2	para l variando de 1 até n_{lg} faça
3	para c variando de 1 até n_{cg} faça
4	se $(l+1) \leq n_{lg}$ e $(c-1) > 0$ faça
5	se $\begin{bmatrix} b_{(l+1)(c-1)} & b_{(l+1)c} \\ b_{l(c-1)} & b_{l_c} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ faça
6	$i_n \leftarrow i_n + 1$
7	calcular o nó w'_{i_n} baseado nos centros de $b_{(l+1)(c-1)}$ e de b_{l_c}
8	se $(l+1) \leq n_{lg}$ e $(c+1) \leq n_{cg}$ faça
9	se $\begin{bmatrix} b_{(l+1)c} & b_{(l+1)(c+1)} \\ b_{l_c} & b_{l(c+1)} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$ faça
10	$i_n \leftarrow i_n + 1$
11	calcular o nó w'_{i_n} baseado nos centros de $b_{(l+1)(c+1)}$ e de b_{l_c}
12	se $(l-1) > 0$ e $(c+1) \leq n_{cg}$ faça
13	se $\begin{bmatrix} b_{l_c} & b_{l(c+1)} \\ b_{(l-1)c} & b_{(l-1)(c+1)} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ faça
14	$i_n \leftarrow i_n + 1$
15	calcular o nó w'_{i_n} baseado nos centros de $b_{(l-1)(c+1)}$ e de b_{l_c}
16	se $(l-1) > 0$ e $(c-1) > 0$ faça
17	se $\begin{bmatrix} b_{l(c-1)} & b_{l_c} \\ b_{(l-1)(c-1)} & b_{(l-1)c} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ faça
18	$i_n \leftarrow i_n + 1$
19	calcular o nó w'_{i_n} baseado nos centros de $b_{(l-1)(c-1)}$ e de b_{l_c}
20	$n_n \leftarrow i_n$

Pode-se observar que a matriz de elevações E poderia ser utilizada diretamente em substituição a B , mas por uma questão de simplificação na descrição do algoritmo, optou-se pela grade binária B . Este algoritmo é apresentado na Tabela 2.1. Observa-se também que a primeira célula (b_{11}) de uma grade é definida como a célula do canto inferior à esquerda da grade.

O segundo algoritmo conecta os nós através de arestas que não interceptam regiões não navegáveis de um ambiente de navegação. O algoritmo é apresentado na Tabela 2.2. Este algoritmo utiliza uma técnica para verificação bidimensional de situações de colisão de VANTs com elevações do ambiente de navegação, que é baseada no algoritmo proposto em (SHAPIRA, 1990) para cálculo da visibilidade entre pontos em um modelo digital de elevação, porém, utilizando a envergadura de um VANT (MEDEIROS et al., 2007).

Considerando o movimento de um VANT pelo segmento de reta que une dois nós w'_r e w'_s , a técnica é um processo iterativo que consiste em dividir tal movimento em n_q parcelas e analisar a situação de colisão do VANT em cada parcela de movimento. Como apresentado na Figura 2.6, a cada parcela, verifica-se se pelo menos um dos pontos q_i , q_{i1} e q_{i2} pertence a alguma célula não navegável. Se pelo menos um destes pontos pertencer a alguma célula obstáculo, então uma situação de colisão é detectada e a aresta não deve ser adicionada ao grafo de visibilidade, evitando uma possível colisão. Caso contrário, o processo iterativo continua até a detecção de uma situação de colisão ou até a verificação de todas as parcelas. Se nenhuma colisão for detectada, a aresta que une w'_r e w'_s é adicionada ao grafo de visibilidade. Observa-se que a distância d_s deve ser maior que metade da envergadura do VANT (e_v), como apresentado na Figura 2.6. O ponto q_i é definido por

$$q_i = w'_r + (i\Delta p_1)\vec{u}_2 \quad (2.4)$$

Em que: Δp_1 é distância entre duas parcelas; e \vec{u}_2 é um vetor unitário definido pelo nó w'_r e pelo nó w'_s .

Tabela 2.2 – Algoritmo para determinação das arestas de grafos de visibilidade através de modelos digitais de elevação.

Índice	Algoritmo para Determinação de Arestas de Grafos de Visibilidade
1	para i variando de 1 até n_n faça
2	para j variando de $(i+1)$ até n_n faça
3	$colisão \leftarrow 0$
4	$l \leftarrow 1$
5	enquanto $(l < n_q)$ e $colisão = 0$ faça
6	se alguma das células interceptadas por $\overline{q_{(l-1)1}q_{l1}}$, $\overline{q_{(l-1)2}q_{l2}}$ e $\overline{q_{l1}q_{l2}}$ for não navegável faça
7	$colisão \leftarrow 1$
8	senão faça
9	$l \leftarrow l + 1$
10	se $colisão = 0$ faça
11	adicionar a aresta que conecta o nó w'_i ao nó w'_j

Os pontos q_{i1} e q_{i2} são calculados através de q_i e da envergadura e_v .

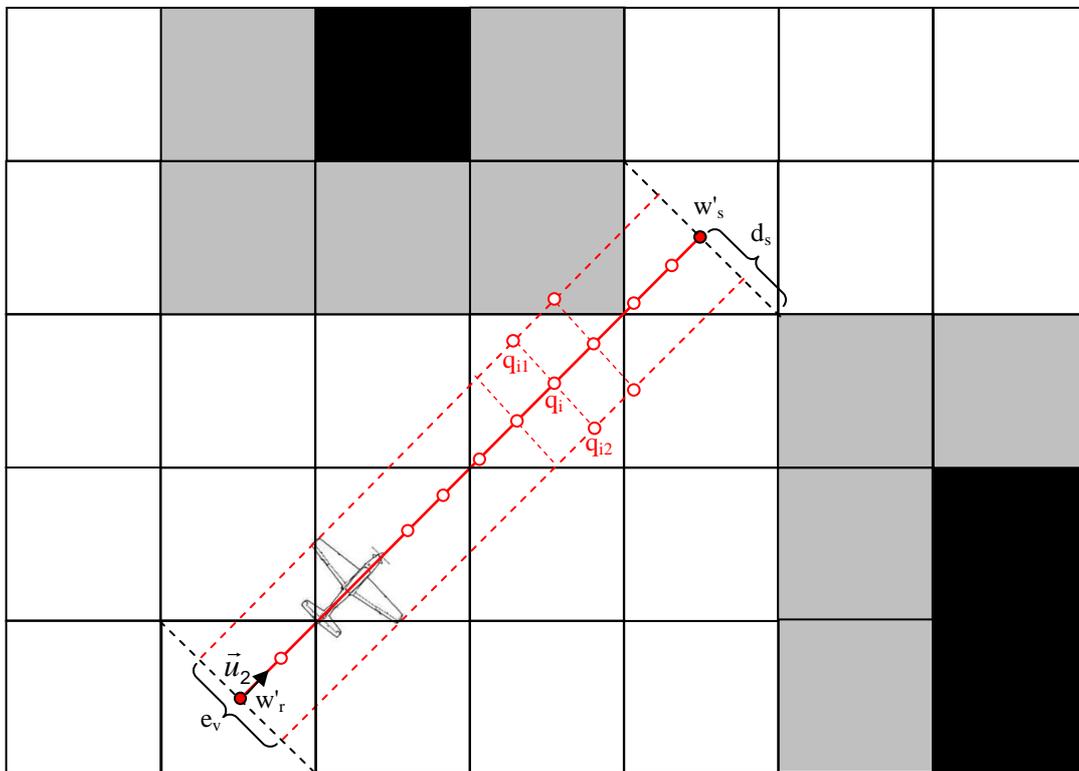


Figura 2.6 – Exemplo de verificação de situação de colisão na aresta que une dois nós.

O número de parcelas é definido por

$$n_q = \frac{d(w'_r, w'_s)}{\Delta p_1} \quad (2.5)$$

Em que: $d(w'_r, w'_s)$ é a distância entre os dois nós w'_r e w'_s do grafo.

Analisando a Tabela 2.2, é possível concluir que a complexidade computacional do algoritmo para determinação de arestas é $O(n_n^2 n_q)$, no pior caso.

O valor de Δp_1 é escolhido de modo que q_{i-1} e q_i pertençam a células cujas diferenças entre linhas ou entre colunas sejam menores ou iguais a 1, isto é, $|l_i - l_{i-1}| \leq 1$ e $|c_i - c_{i-1}| \leq 1$. Assim, se q_{i-1} e q_i pertencerem a células tais que $l_{i-1} \neq l_i$ e $c_{i-1} \neq c_i$, verifica-se se uma das células $b_{l_{i-1}c_i}$ e $b_{l_i c_{i-1}}$ interceptada pelo segmento de reta $\overline{q_{i-1}q_i}$ é não navegável. A mesma verificação é realizada para os demais segmentos de reta: $\overline{q_{i-1}q_{i2}}$; $\overline{q_{(i-1)1}q_{i1}}$; e $\overline{q_{(i-1)2}q_{i2}}$.

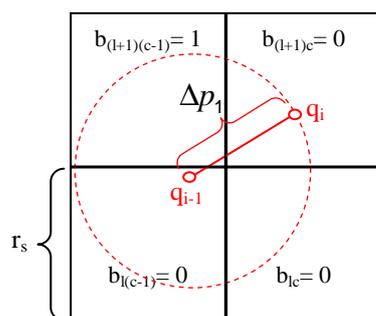


Figura 2.7 – Exemplo do problema de verificação de situação de colisão.

Foi proposta uma prova que garante que toda situação de colisão pode ser analisada no segmento de reta que une dois pontos q_{i-1} e q_i , se a distância entre os dois for inferior ao menor lado de uma célula da grade regular binária. Esta prova é descrita a seguir.

Lema 2 – Considerando o ambiente de navegação representado por uma grade regular B e $\Delta p_1 < r_s$, o segmento de reta $\overline{q_{i-1}q_i}$ é livre de colisão se as células $b_{l_{i-1}c_{i-1}}$ e $b_{l_i c_i}$ são navegáveis e se a célula $b_{l'c'}$, interceptada por $\overline{q_{i-1}q_i}$ quando $l_{i-1} \neq l_i$ e $c_{i-1} \neq c_i$, for navegável.

Prova 2 – Como $\Delta p_1 < r_s$ e r_s é o menor lado da célula, então o ponto q_i pode pertencer somente à vizinhança da célula $b_{l_{i-1}c_{i-1}}$, tal que $|l_i - l_{i-1}| \leq 1$ e $|c_i - c_{i-1}| \leq 1$. Isto pode ser provado, mostrando que a distância entre q_{i-1} e q_i é maior ou igual a r_s quando estes pontos pertencem a células com $|l_i - l_{i-1}| > 1$ ou $|c_i - c_{i-1}| > 1$. Portanto, o segmento de reta $\overline{q_{i-1}q_i}$ intercepta apenas uma célula $b_{l'c'}$, quando $l_{i-1} \neq l_i$ e $c_{i-1} \neq c_i$.

Quando algum ponto q_{i-1} ou q_i , ou algum ponto de intersecção pertence à borda de uma célula b_{lc} , todas as células que fazem fronteira a esta célula também são verificadas.

O lema apresentado pode ser aplicado aos demais segmentos de reta: $\overline{q_{i1}q_{i2}}$; $\overline{q_{(i-1)1}q_{i1}}$; e $\overline{q_{(i-1)2}q_{i2}}$. Porém, para que o lema não seja violado em $\overline{q_{i1}q_{i2}}$, a seguinte restrição deve ser considerada: $e_v < r_s$. É importante ser mencionado que a resolução r_s deve ser convertida de metros para graus, para a restrição com Δp_1 . Devido a diferenças de projeção entre a coordenada latitude e a coordenada longitude de um ponto (ROBINSON et al., 1995) ou à possível utilização de células não quadradas, a resolução r_s é definida em graus por

$$r_s = \min \left(\frac{\text{longitude}(c_{sup}) - \text{longitude}(c_{inf})}{n_{cg}}, \frac{\text{latitude}(c_{sup}) - \text{latitude}(c_{inf})}{n_{lg}} \right) \quad (2.6)$$

Em que: n_{cg} é o número de colunas de B ; n_{lg} é o número de linhas de B ; c_{inf} e c_{sup} são, respectivamente, o primeiro ponto inferior à esquerda e o último ponto superior à direita que delimitam o ambiente de navegação.

Os algoritmos mencionados para determinação de nós e de arestas foram aplicados à construção de grafos de visibilidade para ambientes de navegação obtidos com os modelos digitais de elevação apresentados na Figura 2.8a e na Figura 2.8b. Estes modelos digitais de elevação são do padrão da *Shuttle*

Radar Topography Mission (SRTM) (NASA, 2012) e possuem datum WGS-84 e projeção geográfica. As áreas mais claras das figuras são as áreas mais elevadas. A matriz E do modelo digital de elevação da Figura 2.8a é uma matriz quadrada de ordem 1205 com resolução $r_s = 30$ m, que corresponde à medida do lado de uma célula. É importante ser mencionado que modelos digitais de elevação do padrão SRTM possuem resolução de 90 m. Entretanto, foi feita uma aproximação para reduzir para 30 m a resolução do modelo digital de elevação da Figura 2.8a, com o objetivo de atender aos requisitos de algumas aplicações que são apresentados no Capítulo 3. Este primeiro modelo é delimitado pelos pontos $c_{inf} = (latitude, longitude)_{inf} = (-23.1632222, -46.0816667)$ e $c_{sup} = (latitude, longitude)_{sup} = (-22.8370639, -45.7293528)$. A matriz E do segundo modelo digital de elevação é uma matriz quadrada de ordem 1201 com $r_s = 90$ m. Este segundo modelo digital é delimitado pelos pontos $c_{inf} = (latitude, longitude)_{inf} = (-22.994583, -45.99875)$ e $c_{sup} = (latitude, longitude)_{sup} = (-22.000417, -45.002917)$.

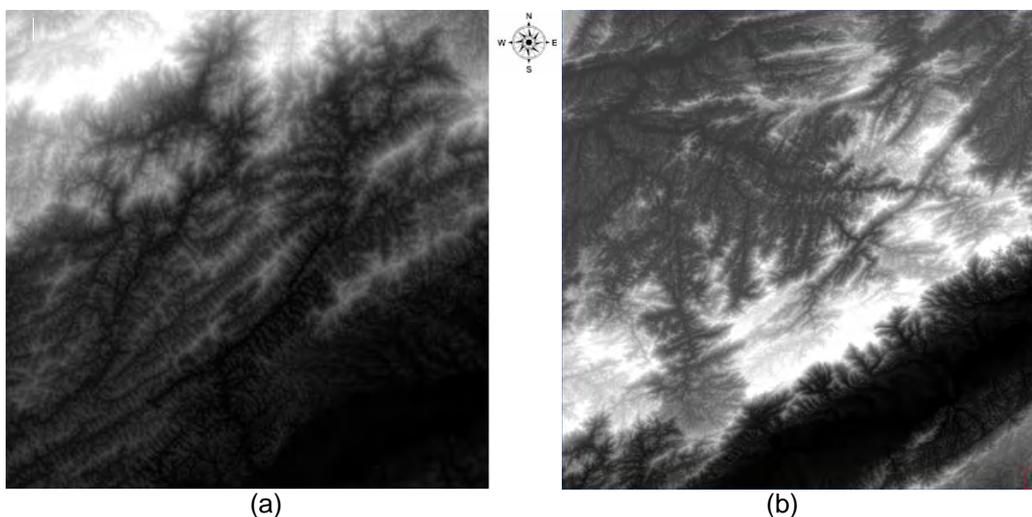


Figura 2.8 – Modelos digitais de elevação utilizados neste trabalho.

Os ambientes de navegação obtidos através destes dois modelos digitais de elevação são apresentados na Figura 2.9 e na Figura 2.10. No processo de criação destes ambientes, foi considerada uma altura de segurança $a_s = 300$ m. Informações a respeito dos grafos de visibilidade construídos para estes ambientes de navegação são descritas na Tabela 2.3 e na Tabela 2.4. Estes grafos são apresentados na Figura 2.11 e na Figura 2.12. Os nós e as arestas dos grafos são representados na cor cinza. Os grafos de visibilidade foram construídos considerando a envergadura do VANT Predator, $e_v = 14.8$ m.

Tabela 2.3 – Informações dos grafos de visibilidade construídos através do primeiro modelo digital de elevação.

Informações	$a_n = 1100$ m	$a_n = 1300$ m	$a_n = 1550$ m	$a_n = 1800$ m
d_s	21.22 m	21.22 m	21.22 m	21.22 m
Δp_1	0.00027 grau	0.00027 grau	0.00027 grau	0.00027 grau
Número de nós (n_n)	1489	756	360	215
Número de arestas (n_a)	65241	75318	23913	5463
Tempo para construção	121 s	72.4 s	21.3 s	2.7 s

Tabela 2.4 – Informações dos grafos de visibilidade construídos através do segundo modelo digital de elevação.

Informações	$a_n = 1300$ m	$a_n = 1500$ m	$a_n = 1900$ m	$a_n = 2200$ m
d_s	63.64 m	63.64 m	63.64 m	63.64 m
Δp_1	0.00082 grau	0.00082 grau	0.00082 grau	0.00082 grau
Número de nós (n_n)	5088	3937	2261	531
Número de arestas (n_a)	427926	490707	226356	44619
Tempo para construção	1053.5 s	983.4 s	312.4 s	48.6 s

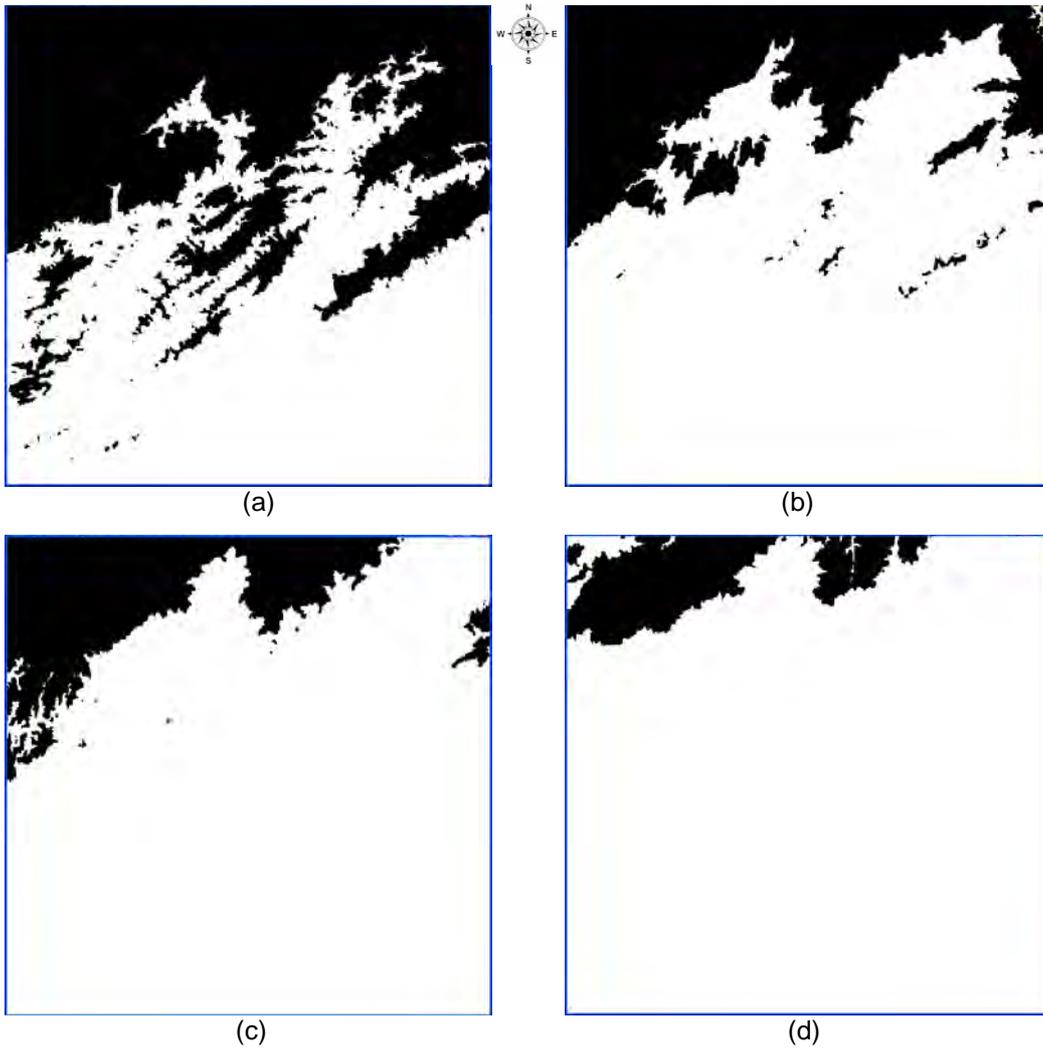


Figura 2.9 – Ambientes de navegação obtidos com o primeiro modelo digital de elevação, considerando: $a_s = 300$ m; (a) $a_n = 1100$ m; (b) $a_n = 1300$ m; (c) $a_n = 1550$ m; e (d) $a_n = 1800$ m.

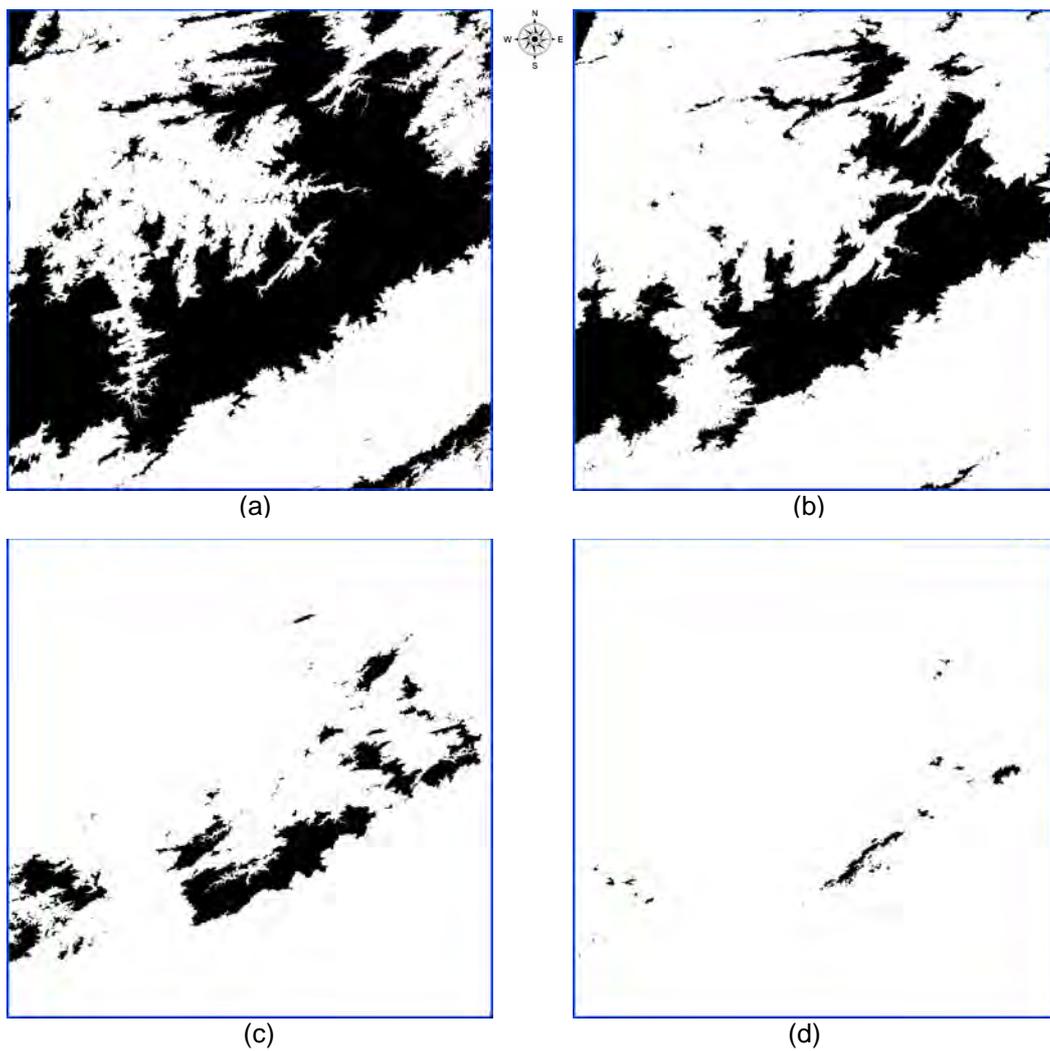


Figura 2.10 – Ambientes de navegação obtidos com o segundo modelo digital de elevação, considerando: $a_s = 300$ m; (a) $a_n = 1300$ m; (b) $a_n = 1500$ m; (c) $a_n = 1900$ m; e (d) $a_n = 2200$ m.

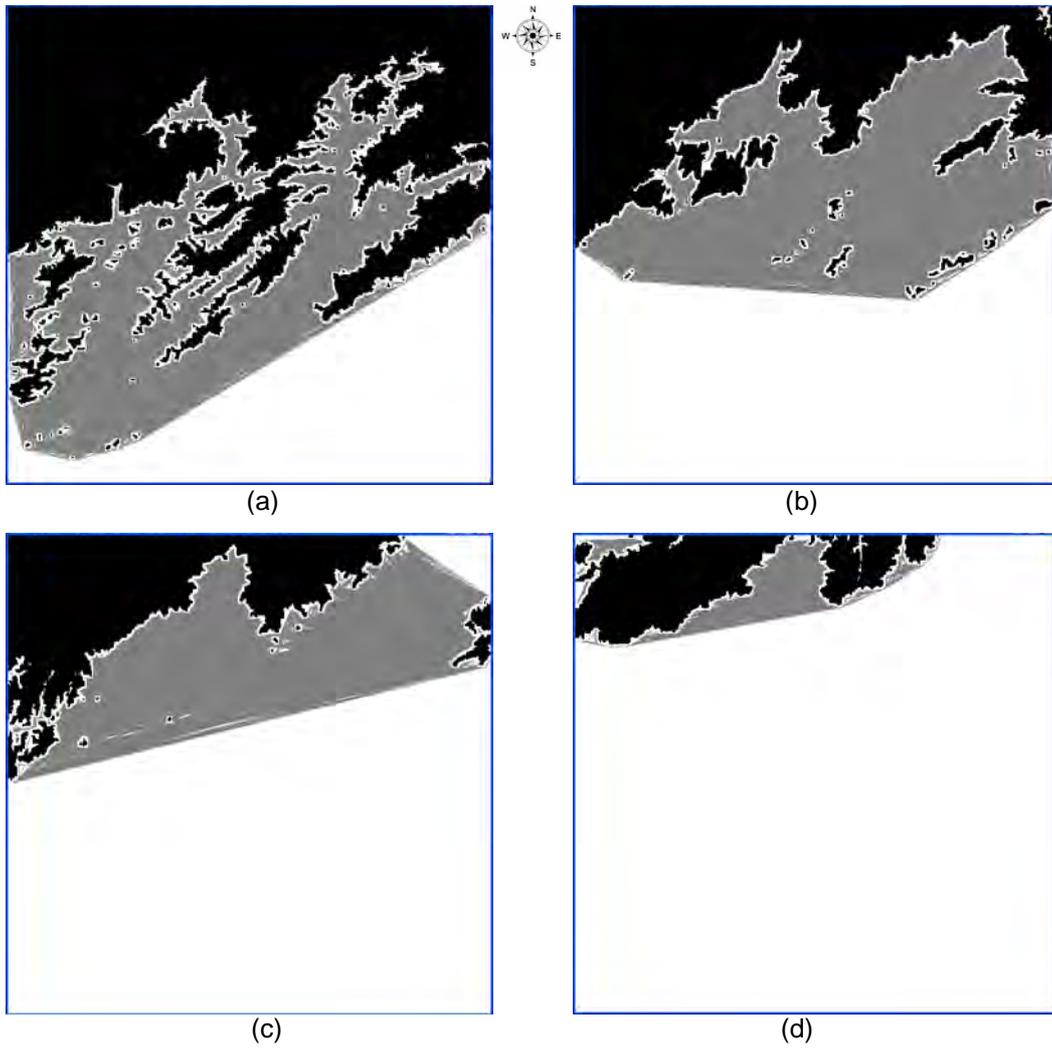


Figura 2.11 – Conjuntos de grafos de visibilidade construídos para os ambientes de navegação com: (a) $a_n = 1100$ m; (b) $a_n = 1300$ m; (c) $a_n = 1550$ m; e (d) $a_n = 1800$ m.

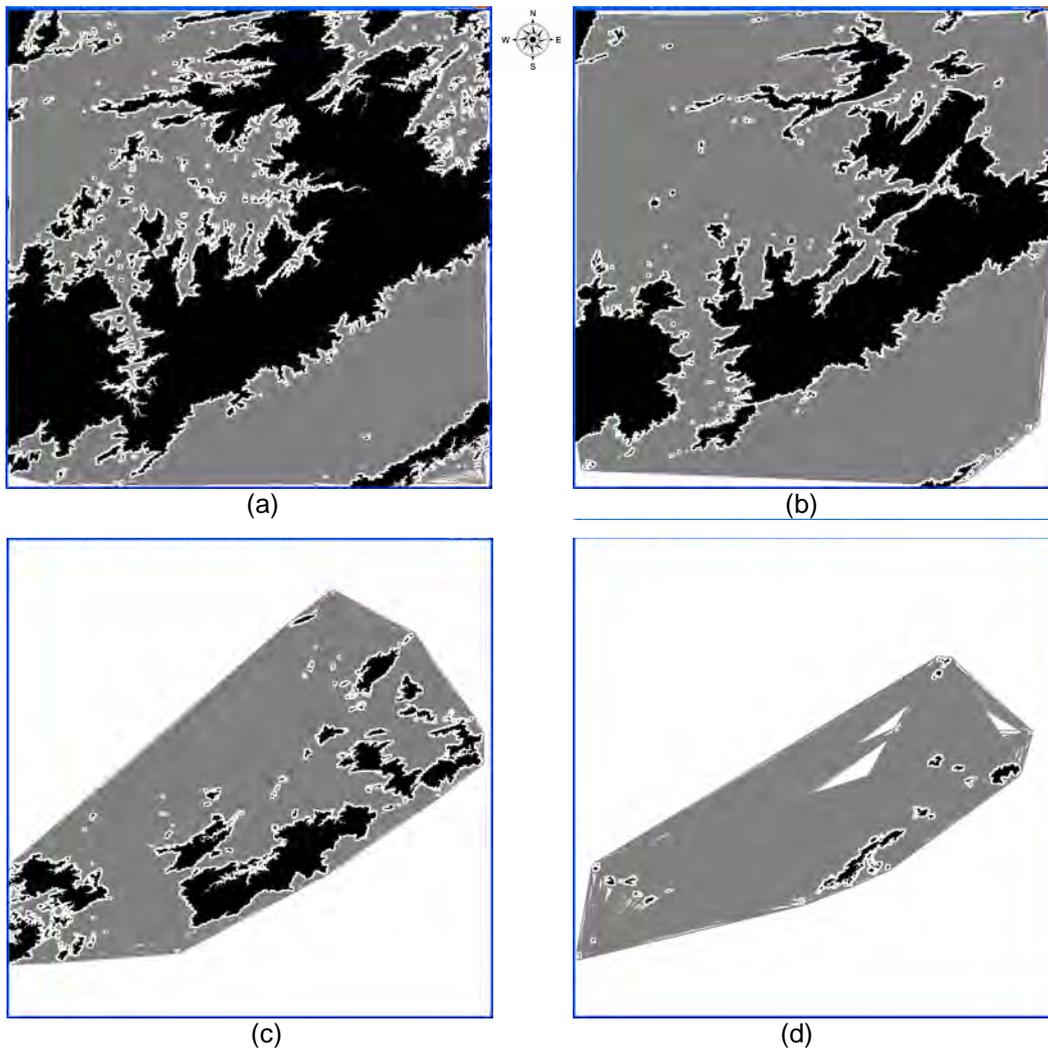


Figura 2.12 – Conjuntos de grafos de visibilidade construídos para os ambientes de navegação com: (a) $a_n = 1300$ m; (b) $a_n = 1500$ m; (c) $a_n = 1900$ m; e (d) $a_n = 2200$ m.

Analisando os resultados obtidos, observa-se que os algoritmos são aplicados diretamente aos modelos digitais de elevação para a construção de grafos de visibilidade, não sendo necessária a conversão dos conjuntos de células não navegáveis em representações poligonais ou a utilização de qualquer estrutura auxiliar. Os algoritmos constroem grafos de visibilidade para problemas de planejamento de rotas ou trajetórias, associando a informação de envergadura

do veículo. Esta informação garante segurança à navegação de um VANT pelas arestas dos grafos de visibilidade construídos.

Este trabalho considera a navegação bidimensional, ou seja, a navegação a uma altitude constante. Entretanto, os dois algoritmos podem ser utilizados na construção de camadas de grades binárias através da variação de a_n . Deste modo, estes algoritmos podem ser aplicados na modelagem tridimensional de ambientes de navegação aérea.

3 CONSTRUÇÃO DE GRAFOS DE VISIBILIDADE COM AVALIAÇÃO DOS EFEITOS DA QUEDA DE VANTS

Pesquisas têm sido desenvolvidas com o propósito de elaborar regulamentações e requisitos visando à certificação de navegação aérea para VANTS (DALAMAGKIDIS et al., 2008c; OLIVEIRA et al., 2009).

A principal finalidade da certificação de navegação aérea para VANTS é garantir um nível de segurança para a navegação destes veículos equivalente ou superior ao nível de segurança dos veículos aéreos tripulados, reduzindo a probabilidade do envolvimento de vidas humanas em possíveis acidentes com tais veículos (DALAMAGKIDIS et al., 2008c).

O trabalho desenvolvido e apresentado neste capítulo procura unir o planejamento de trajetórias para VANTS com métricas de segurança para a navegação destes veículos.

3.1. Revisão bibliográfica e motivação

A aplicação dos métodos para planejamento de trajetórias está focada em evitar situações de colisão de um VANT com obstáculos do ambiente de navegação e em evitar situações de colisão entre VANTS dispostos em agrupamentos (GOERZEN et al., 2010; LAVALLE, 2006). Porém, estes métodos não consideram a avaliação de riscos à população em acidentes com queda de VANTS. Uma trajetória de navegação para um VANT deve ser livre de situações de colisão e dinamicamente viável, mas é desejável que a mesma seja planejada de modo a reduzir o risco de fatalidades em acidentes com queda de VANTS. Portanto, foi proposto um método para modelar ambientes de navegação através da avaliação dos efeitos da queda de VANTS e através de modelos digitais de elevação. Este método permite restringir as regiões navegáveis àquelas que não representam risco de colisão aos VANTS e que não representam risco elevado à população na ocorrência de queda destes veículos. O método proposto foi publicado em (MEDEIROS e SILVA, 2010a).

A representação destas regiões navegáveis por meio de grafos de visibilidade mescla então o problema de planejamento de rotas e trajetórias com a avaliação dos riscos à população em possíveis acidentes com queda de VANTs.

Neste capítulo, é apresentado o método proposto em (MEDEIROS e SILVA, 2010a) para modelagem de ambientes de navegação. Também é descrita uma análise de rotas de navegação planejadas com grafos de visibilidade construídos através destas modelagens.

3.2. Método para modelagem de ambientes de navegação através de modelos digitais de elevação e de análises dos efeitos da queda de VANTs

O método proposto modela como regiões não navegáveis aquelas que representam situações de colisão ao VANT e aquelas regiões que representam um elevado risco de fatalidade à população na ocorrência da queda de um VANT.

As regiões que representam situações de colisão ao VANT são modeladas pelo processo descrito no capítulo anterior, através de um modelo digital de elevação, de uma altitude de navegação e de uma altura de segurança. As regiões que representam elevado risco de fatalidade à população são determinadas por uma métrica para a avaliação dos efeitos de impactos de VANTs na superfície sobrevoada. Esta métrica é o período mínimo aceitável entre dois acidentes consecutivos com impacto de VANT na superfície (DALAMAGKIDIS et al., 2008a; DALAMAGKIDIS et al., 2008c). Especificando um limite para este período, é possível estimar o número esperado de fatalidades e, assim, classificar a região como navegável ou não navegável.

3.2.1. Avaliação dos efeitos de acidentes com queda de VANTs

Como mencionado anteriormente, a métrica apresentada em (DALAMAGKIDIS et al., 2008b) para avaliar os efeitos de acidentes com impacto de VANTs na superfície sobrevoada é o período mínimo aceitável (t_m) entre dois acidentes consecutivos deste tipo e com fatalidade. A métrica t_m é definida por

$$t_m = \frac{n_{ef}}{n_{es}} \quad (3.1)$$

Em que: n_{ef} é o número esperado de fatalidades; e n_{es} é o nível de segurança de referência baseado na navegação aérea tripulada e definido como $n_{es} = 10^{-7}$ fatalidade por hora (h) de voo para acidentes com impacto na superfície sobrevoada. Este valor foi obtido através de análises de taxas de acidentes com fatalidades em linhas aéreas norte-americanas no período de 1983 a 2006, considerando fatalidades em solo e não considerando as fatalidades de membros da tripulação e passageiros das aeronaves. Um estudo detalhado do cálculo do valor de n_{es} é apresentado em (DALAMAGKIDIS et al., 2008c).

O número esperado de fatalidades é definido por

$$n_{ef} = a_e d_p p_f \quad (3.2)$$

Em que: a_e é a área exposta ao acidente; d_p é a densidade populacional da superfície exposta ao acidente; e p_f é a probabilidade da ocorrência de uma fatalidade.

A área exposta ao acidente é definida por

$$a_e = e_v \left(c_v + \frac{a_p}{\text{sen}(\alpha_v)} \right) \quad (3.3)$$

Em que: e_v é a envergadura de um VANT, isto é, o comprimento da ponta de uma asa à ponta da outra para um VANT de asa fixa, ou o comprimento de uma pá de hélice à outra para um VANT de asa rotativa; c_v é o comprimento do

veículo; a_p é a altura média de uma pessoa; α_v é o ângulo de descida ou queda do VANT, definido pelo eixo longitudinal do veículo e a superfície de impacto. Observa-se que no cálculo de a_e não são considerados o arrasto do veículo a partir do ponto de colisão ou a fragmentação do veículo.

Neste trabalho, foi utilizado o modelo matemático proposto em (DALAMAGKIDIS et al., 2008b) para calcular a probabilidade de fatalidade em acidentes com impacto na superfície. Neste modelo, a probabilidade de fatalidade é definida por

$$p_f = \frac{1}{1 + \sqrt{\frac{p_2}{p_3} \left(\frac{p_3}{m_v (1.4 v_v^2)} \right)^{\frac{1}{4p_1}}} \quad (3.4)$$

Em que: m_v é a massa do veículo; v_v é a velocidade do veículo; $p_1 \in (0,1]$ indica quão exposta ao impacto está uma pessoa, e o aumento deste parâmetro indica o aumento da proteção da pessoa exposta ao acidente; p_2 é a energia de impacto necessária para uma probabilidade de fatalidade de 50%, quando $p_1 = 0.5$; p_3 é o limiar da energia de impacto necessária para causar uma fatalidade, quando p_1 tende a zero.

Distribuições espaciais de t_m são apresentadas em (DALAMAGKIDIS et al., 2008b) para seis tipos de VANTs: Global Hawk; Predator; Neptune; Fire Scout; RMAX II; e Maxi Joker 2.

Como as áreas mais populosas são as mais restritivas para a navegação de VANTs, em termos de segurança à população, surgiu a idéia de diferenciar áreas urbanas de áreas rurais a fim de tornar mais preciso o cálculo de t_m . Portanto, um modelo matemático foi proposto neste trabalho para estimar densidades populacionais com diferenciação entre áreas urbanas e áreas rurais de cada município do ambiente de navegação. No modelo proposto, a densidade d_p é definida por

$$d_p = d_{lc} = u_{lc} \left(\frac{P_{um_{lc}}}{a_{um_{lc}}} \right) + (1 - u_{lc}) \left(\frac{P_{rm_{lc}}}{a_{rm_{lc}}} \right) \quad (3.5)$$

Em que: d_{lc} é uma célula da matriz de densidade populacional D ; $p_{um_{lc}}$ é um elemento de P_u que corresponde à população urbana do município de código m_{lc} da superfície representada pela célula b_{lc} ; $p_{rm_{lc}}$ é um elemento de P_r , que corresponde à população rural do município de código m_{lc} ; $a_{um_{lc}}$, elemento de A_u , é a área urbana do município de código m_{lc} ; $a_{rm_{lc}}$, elemento de A_r , é a área rural do município de código m_{lc} ; e u_{lc} é uma célula binária da matriz de áreas urbanas (U) e assume o valor 1 para indicar que a região representada pela célula binária b_{lc} contém alguma parcela de área urbana.

A principal vantagem da diferenciação entre área urbana e rural é evitar a distribuição de densidade homogênea pelo município, tornando mais preciso o processo de estimação da densidade populacional e, por consequência, a estimação de t_m .

3.2.2. Modelagem do ambiente de navegação

Baseado no modelo matemático da métrica t_m , descrito no capítulo anterior, as regiões navegáveis e não navegáveis de um ambiente de navegação podem ser modeladas através da seguinte redefinição de uma célula b_{lc}

$$b_{lc} = \begin{cases} 0, & \text{se} \begin{cases} e_{lc} < (a_n - a_s) e \\ t_{mlc} \leq l_{t_m} \end{cases} \\ 1, & \text{se} \begin{cases} e_{lc} \geq (a_n - a_s) e \\ t_{mlc} > l_{t_m} \end{cases} \end{cases} \quad (3.6)$$

Em que: t_{mlc} é uma célula da matriz T_m , cujo valor é definido pela Equação 3.1 e corresponde ao período mínimo aceitável entre dois acidentes consecutivos na região representada pela célula equivalente b_{lc} ; l_{t_m} é um limite para t_m que é

utilizado para restringir as regiões navegáveis para um VANT, garantindo segurança às populações do ambiente de navegação.

O método proposto para a criação da grade binária B através da aplicação da Equação 3.6 pode ser observado na Figura 3.1.

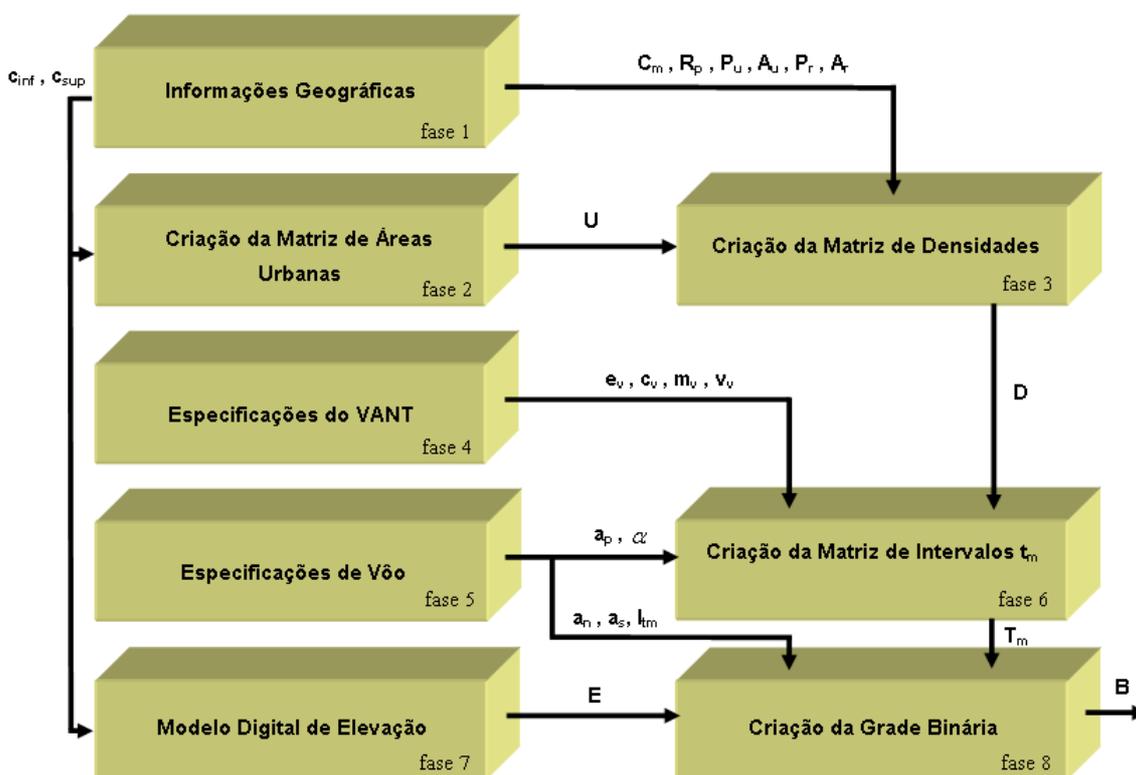


Figura 3.1 – Método para modelagem de ambientes de navegação através de modelos digitais de elevação e de avaliações dos efeitos da queda de VANTs.

Fonte: Medeiros e Silva (2010a)

O método integra em uma única representação do ambiente de navegação, a grade regular binária, seis diferentes camadas de informações: especificações do VANT; especificações de voo; uma imagem georeferenciada (I) da superfície do ambiente de navegação; informações geográficas do ambiente de navegação, incluindo informações dos municípios; representações poligonais dos municípios; e modelos digitais de elevação.

Neste capítulo, o estudo foi realizado com ambientes de navegação obtidos através do modelo digital de elevação da Figura 2.8a.

3.2.3. Aquisição de informações geográficas (fase 1)

A primeira fase do método consiste na obtenção do conjunto das seguintes informações geográficas: os pontos que delimitam o ambiente de navegação C_{inf} e C_{sup} ; a representação poligonal ($r_{pc_{mi}}$) do município de código c_{mi} , para i variando de 1 até o número de municípios (n_m); o código c_{mi} , elemento de C_m , é um código padronizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE) (IBGE, 2005); a população urbana ($p_{uc_{mi}}$) do município de código c_{mi} ; a população rural ($p_{rc_{mi}}$) do município de código c_{mi} ; e a área urbana ($a_{uc_{mi}}$) e a área rural ($a_{rc_{mi}}$) de cada município de código c_{mi} . É importante ser mencionado que a representação $r_{pc_{mi}}$ é um elemento de R_p . As informações a respeito da população, da área e dos códigos dos municípios foram obtidas de trabalhos desenvolvidos pela Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) (MIRANDA et al., 2005) e pelo IBGE (IBGE, 2005).

As informações geográficas descritas são utilizadas na criação da matriz de áreas urbanas U , na criação da matriz de densidades D e na aquisição da matriz de elevações E do modelo digital de elevação.

3.2.4. Criação da matriz de áreas urbanas (fase 2)

O objetivo da segunda fase é a determinação das áreas com as maiores densidades populacionais do ambiente de navegação, em outras palavras, as áreas urbanas. Neste trabalho, uma célula do ambiente de navegação é considerada urbana, se a mesma for uma área construída de um município, sendo composta por complexos residenciais, comerciais, industriais, religiosos, de transporte, de lazer, etc. O processo de classificação de imagens foi adotado por permitir uma determinação mais precisa das áreas urbanizadas (MIRANDA et al., 2005).

A determinação das áreas urbanas é realizada através de três processos: a classificação da imagem georeferenciada I ; a criação de uma representação binária (U') para a matriz de classificação (C), que é o resultado do processo de classificação; e a simplificação da representação binária U' . Esta fase é apresentada na Figura 3.2.

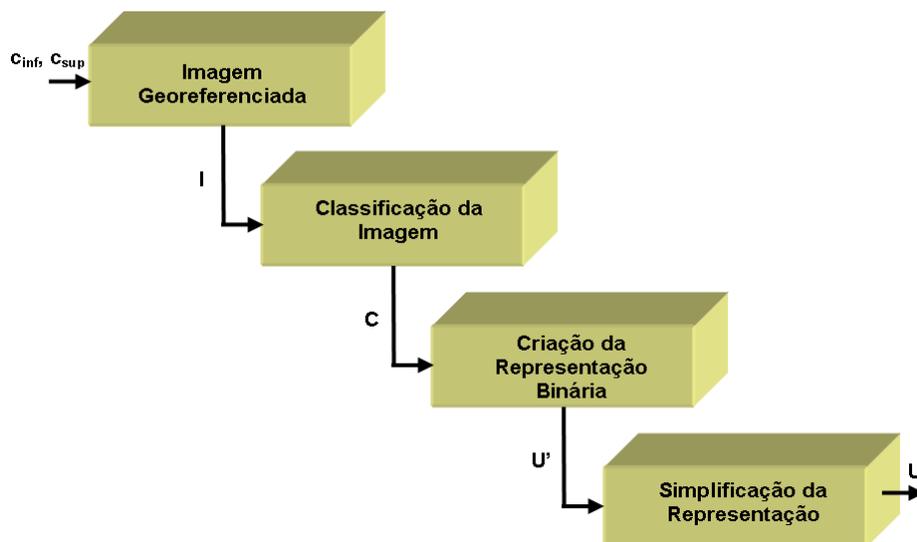


Figura 3.2 – Fase de criação da matriz de áreas urbanas.

Fonte: Medeiros e Silva (2010a)

Neste trabalho, foi utilizada uma imagem georeferenciada I com 1205 por 1205 pixels e composta por três bandas espectrais, capturada por sensores do satélite Landsat 5 e ortorretificada em (ZALOTI JUNIOR, 2007). Esta imagem corresponde à superfície do modelo digital de elevação apresentado na Figura 2.8a e possui uma resolução espacial de 30 m, ou seja, cada célula i_c representa uma região com comprimento e altura iguais a 30 m.

O método de máxima verossimilhança (RICHARDS e JIA, 2006) foi empregado no processo supervisionado de classificação da imagem I , considerando oito classes: cultura (ξ_1); água (ξ_2); floresta (ξ_3); solo exposto (ξ_4); sombra (ξ_5); nuvem (ξ_6); pastagem (ξ_7); e área urbana (ξ_8). Assim, as áreas rurais são definidas pelas sete primeiras classes. Estas classes foram definidas com base

no trabalho desenvolvido em (ZALOTI JUNIOR, 2007). As respectivas amostras de treinamento das oito classes são apresentadas na Figura 3.3. Nesta figura, os polígonos alaranjados representam as amostras de culturas, os polígonos azuis indicam as amostras de água, os polígonos na cor verde-mar indicam as amostras de florestas, os polígonos vermelhos representam as amostras de solo exposto, os polígonos amarelos representam as amostras de sombras, os polígonos brancos indicam as amostras de nuvens, os polígonos verdes representam as amostras de pastagens, os polígonos na cor ciano representam as amostras de áreas urbanas. O método de máxima verossimilhança avalia a função de probabilidade de pertencimento de um pixel i_{lc} para cada uma das classes ξ_p e atribui o pixel à classe com a maior probabilidade, como pode ser verificado na Equação 3.7.

$$i_{lc} \in \xi_p, \text{ se } p(i_{lc} / \xi_p)p(\xi_p) > p(i_{lc} / \xi_q)p(\xi_q) \text{ para todo } q \neq p \quad (3.7)$$

Em que: $p(i_{lc} / \xi_p)$ é a probabilidade de encontrar um pixel da classe ψ_p na célula i_{lc} , considerando as amostras de treinamento; e $p(\xi_p)$ é a probabilidade da classe ξ_p ocorrer na imagem.

Uma explicação mais detalhada do método de máxima verossimilhança é apresentada em (RICHARDS e JIA, 2006). O aplicativo de computador ENVI 4.3 (ENVI, 2010) foi usado no processo de classificação. Os resultados dos três processos para a criação da matriz U são apresentados na Figura 3.4.

A imagem foi classificada e um índice kappa de 95.59% foi obtido pelo aplicativo ENVI no processo de classificação. O índice kappa é uma medida de conformidade que é usada para avaliar a precisão de processos de classificação, comparando valores observados com valores de referência. O valor de kappa é diretamente proporcional à eficiência da classificação obtida. O segundo processo é a criação da representação U' através da matriz de classificação (C). Cada célula binária u'_{lc} recebe o valor 1, se a célula correspondente c_{lc} é classificada como área urbana.

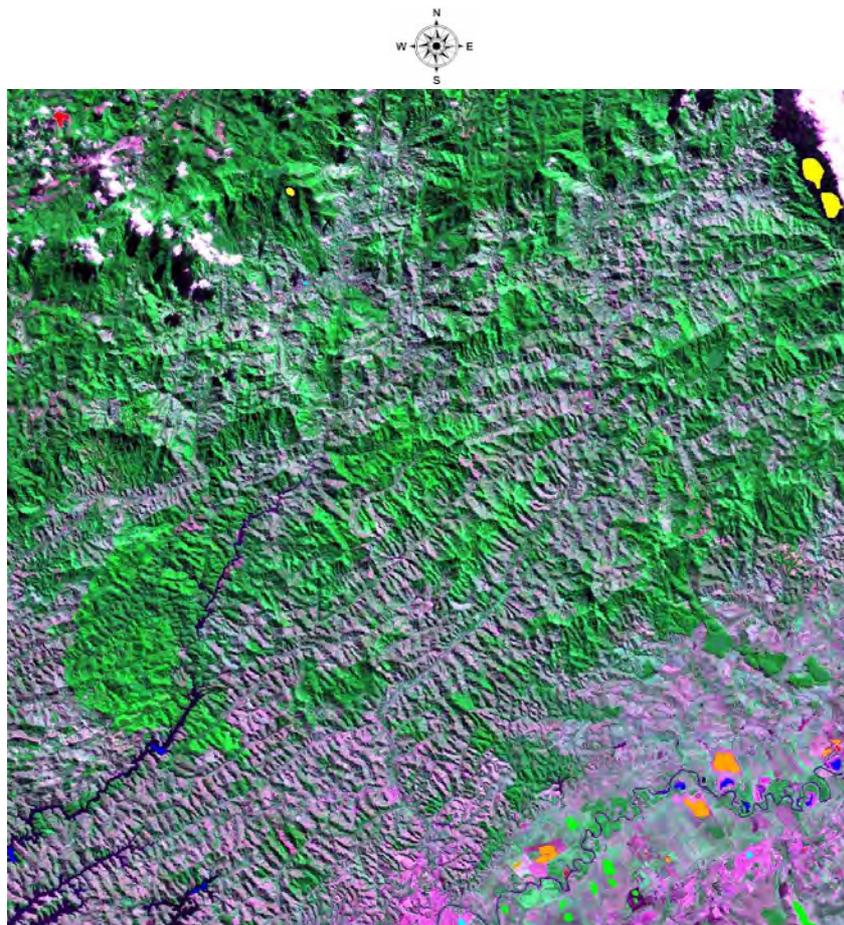


Figura 3.3 – Amostras de treinamento das oito classes do processo de classificação.

Fonte: Medeiros e Silva (2010a)

O terceiro processo simplifica a matriz U' pela aplicação de um filtro baseado na área dos polígonos definidos pelas áreas urbanas. No processo de filtragem, todos os polígonos compostos por células urbanas com área inferior a 9000 m^2 são eliminados da matriz. Esta área de simplificação corresponde à área de dez células da grade B . Em seguida, há a aplicação dos operadores morfológicos de dilatação e de fechamento para preenchimento de regiões (VAN DEN BOOMGARD e VAN BALEN, 1992), com o objetivo de unir áreas urbanas mais próximas. Na Figura 3.4b, as áreas urbanas são representadas na cor ciano. Na Figura 3.4c e na Figura 3.4d, as células urbanas são representadas por células pretas.

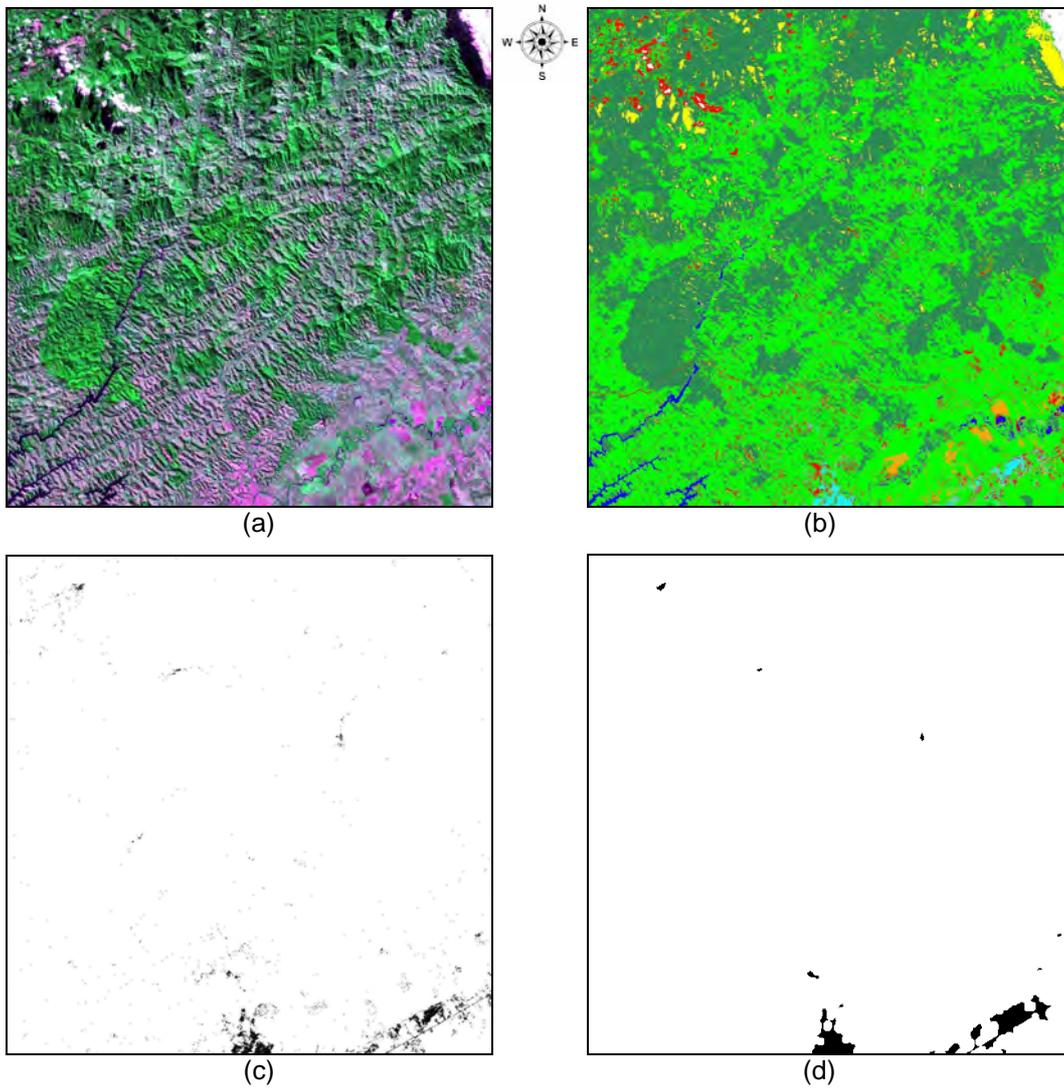


Figura 3.4 – Resultados dos processos de criação da matriz de áreas urbanas: (a) imagem I ; (b) classificação C ; (c) representação binária U^b ; e (d) a matriz U .

Fonte: Medeiros e Silva (2010a)

3.2.5. Criação da matriz de densidades populacionais (fase 3)

Os objetivos desta fase são: a criação da matriz de códigos de municípios (M), cuja célula m_{lc} é usada para indicar o código do município de maior densidade populacional e que possui parte de sua área contida na célula correspondente b_{lc} ; e, então, criar a matriz de densidades populacionais (D), diferenciando áreas urbanas de áreas rurais através da matriz U .

Um algoritmo foi proposto neste trabalho para determinar o código de cada célula m_{lc} através da representação poligonal $r_{pc_{mi}}$ de cada município de código c_{mi} . A representação poligonal dos municípios da superfície do modelo digital de elevação apresentado na Figura 2.8a é apresentada na Figura 3.5.

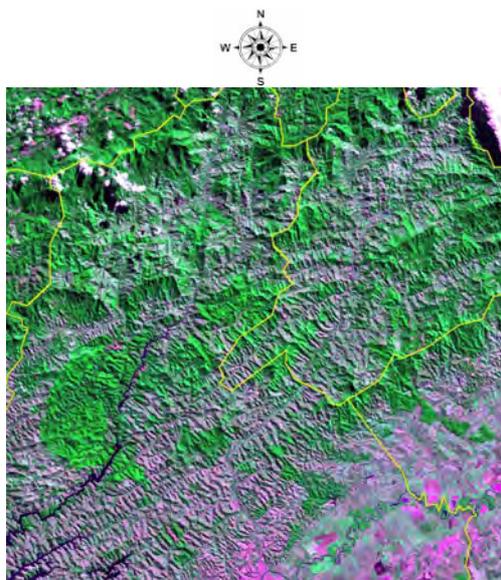


Figura 3.5 – Representação poligonal dos municípios da superfície do modelo digital de elevação analisado.

Inicialmente, este algoritmo identifica o código c_{mi} de cinco pontos que descrevem a região representada por b_{lc} . Cada célula m_{lc} recebe o código do município com maior densidade populacional entre os municípios identificados pelos cinco pontos. Um exemplo do esquema de determinação do código de uma célula m_{lc} através de cinco pontos de b_{lc} é apresentado na Figura 3.6.

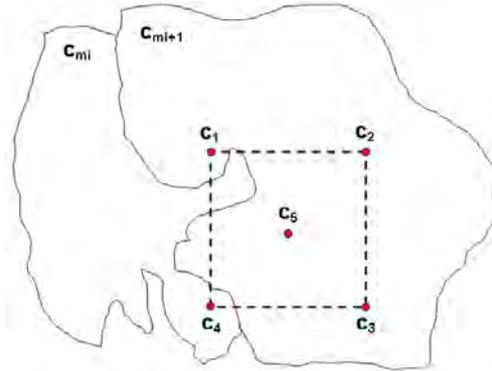


Figura 3.6 – Exemplo da determinação do código m_{lc} através de cinco pontos da região representada por b_{lc} .

Fonte: Medeiros e Silva (2010a)

Tabela 3.1 – Algoritmo para criação da matriz de densidades populacionais D e da matriz de códigos de municípios M .

Índice	Algoritmo para Criação da Matriz D e da Matriz M
1	para l variando de 1 até n_{lg} faça
2	para c variando de 1 até n_{cg} faça
3	$d_{temp} \leftarrow 0$
4	para j variando de 1 até 5 faça
5	calcular o ponto c_j da região representada por b_{lc}
6	$i \leftarrow 1$
7	$s \leftarrow 0$
8	enquanto $(i \leq n_m)$ e $(s = 0)$ faça
9	se c_j é um ponto interno do polígono $r_{pc_{mi}}$ então faça
10	$s \leftarrow 1$
11	senão faça
12	$i \leftarrow i + 1$
13	se $s = 1$ então faça
14	$d_p \leftarrow u_{lc} \left(\frac{p_{uc_{mi}}}{a_{uc_{mi}}} \right) + (1 - u_{lc}) \left(\frac{p_{rc_{mi}}}{a_{rc_{mi}}} \right)$
15	se $d_p > d_{temp}$ então faça
16	$m_{lc} \leftarrow c_{mi}$
17	$d_{temp} \leftarrow d_p$
18	contar o número de células urbanas ($n_{um_{lc}}$) com código m_{lc}
19	para l variando de 1 até n_{lg} faça
20	para c variando de 1 até n_{cg} faça
21	$a_{um_{lc}} \leftarrow n_{um_{lc}} r_s^2$
22	$a_{rm_{lc}} \leftarrow a_{m_{lc}} - a_{um_{lc}}$
23	$d_{lc} \leftarrow u_{lc} \left(\frac{p_{um_{lc}}}{a_{um_{lc}}} \right) + (1 - u_{lc}) \left(\frac{p_{rm_{lc}}}{a_{rm_{lc}}} \right)$

O algoritmo para criação da matriz D e da matriz M é apresentado na Tabela 3.1. Neste algoritmo, a_{m_c} é a área total de um município de código m_c . O bloco do algoritmo da linha 1 à linha 18 cria a matriz de códigos M e o bloco restante calcula a matriz de densidades D , com diferenciação entre áreas urbanas e rurais.

3.2.6. Aquisição de especificações do VANT (fase 4)

Esta fase consiste na aquisição de um conjunto de especificações do VANT, que são usadas na criação da matriz T_m . Estas especificações do VANT são a envergadura e_v , o comprimento c_v , a massa máxima na decolagem m_v , e a velocidade do veículo v_v . Neste trabalho foram realizados estudos com as especificações de seis tipos de VANTs: RQ-4 Global Hawk (WIKIPEDIA, 2009d); MQ-1 Predator (WIKIPEDIA, 2009c); KZO (WIKIPEDIA, 2009b); Apoená 3000 (WIKIPEDIA, 2009a); R-MAX (SATO, 2003); e ZALA 421-08 (WIKIPEDIA, 2009e). As especificações destes VANTs são apresentadas na Tabela 3.2.

Tabela 3.2 – Especificações dos VANTs utilizadas neste trabalho.

VANT	e_v	c_v	m_v	v_v
RQ-4 Global Hawk	35.4 m	13.5 m	12134 kg	650 km/h
MQ-1 Predator	14.8 m	8.22 m	1020 kg	140 km/h
KZO	3.42 m	2.28 m	161 kg	220 km/h
Apoena 3000	3.97 m	2.63 m	56 kg	115 km/h
R-MAX	3.12 m	3.63 m	94 kg	75 km/h
ZALA 421-08	0.8 m	0.41 m	1.9 kg	120 km/h

Fonte: Medeiros e Silva (2010a)

3.2.7. Aquisição de especificações de voo (fase 5)

As especificações de voo são a altitude de navegação a_n , a altura de segurança a_s , o limite l_{t_m} para t_m , a altura média de uma pessoa a_p e o ângulo de descida do veículo α_v . Como mencionado anteriormente, as duas primeiras informações são usadas para a determinação das áreas que representam risco de colisão à navegação de um VANT, na fase de criação da grade binária B .

Analisando a Equação 3.6, verifica-se que o limite l_{t_m} é usado na última fase para a determinação das regiões não navegáveis. As demais especificações são utilizadas na fase de criação da matriz de intervalos de tempo T_m . Neste trabalho, foi especificado $a_p = 1.8$ m e $\alpha_v = 60^\circ$.

3.2.8. Criação da matriz da métrica t_m (fase 6)

Através das informações provenientes das fases três, quatro e cinco, a fase seis do método consiste na aplicação da Equação 3.1 para o cálculo da métrica t_m para cada região representada por uma célula b_{lc} . Os intervalos calculados são armazenados nas células correspondentes t_{mlc} . A Figura 3.7 apresenta a distribuição espacial de t_m para os seis tipos de VANTs e para a superfície do ambiente de navegação considerados.

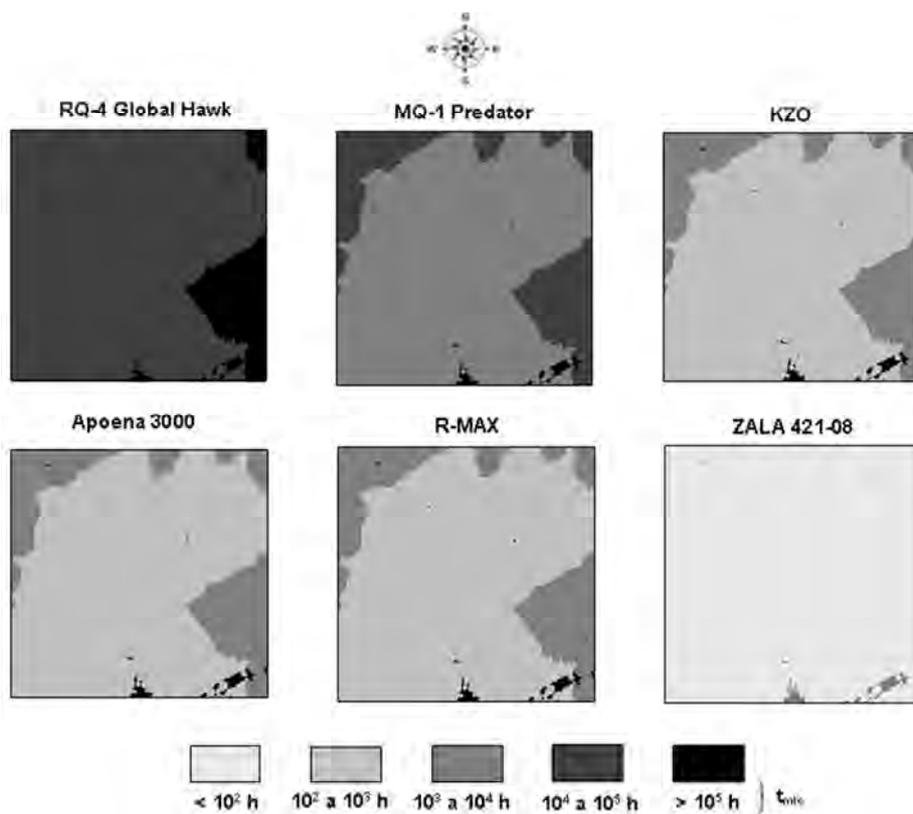


Figura 3.7 – Distribuição espacial de t_m para o ambiente de navegação considerado neste trabalho.

Fonte: Medeiros e Silva (2010a)

Os parâmetros p_2 e p_3 , apresentados na página 56, foram especificados como 10^6 Joules e 100 Joules (J), como sugerido em (DALAMAGKIDIS et al., 2008b). Embora um número maior de pessoas esteja exposto ao acidente em áreas urbanas, devido ao elevado número de construções nestas áreas, foi considerado que uma pessoa está menos exposta ao acidente em uma área urbana do que em uma área rural (Medeiros e Silva, 2010a). Portanto, o parâmetro p_1 foi especificado como 0.6 para áreas urbanas e 0.3 para áreas rurais. A Tabela 3.3 apresenta os valores mínimos e máximos das matrizes T_m criadas para cada VANT.

Tabela 3.3 – Valores mínimos e máximos de T_m para cada VANT.

VANT	mínimo t_{mlc}	máximo t_{mlc}
RQ-4 Global Hawk	34758.0 h	27525000.0 h
MQ-1 Predator	9434.0 h	3659700.0 h
KZO	903.1 h	280290.0 h
Apoena 3000	915.1 h	166360.0 h
R-MAX	813.1 h	140640.0 h
ZALA 421-08	22.9 h	5078.8 h

Fonte: Medeiros e Silva (2010a)

3.2.9. Criação da matriz de elevações (fase 7)

A sétima fase é a aquisição da matriz de elevações E do modelo digital de elevações com base nos pontos limitantes c_{inf} e c_{sup} , como mencionado anteriormente. Nesta fase, foi usado o modelo digital de elevação apresentado na Figura 2.8a.

É importante ser mencionado que este modelo digital de elevação possui resolução de 90 m. Entretanto, como a resolução deste modelo era diferente da resolução da imagem georeferenciada I , foi necessário elaborar um algoritmo para a criação de uma matriz de elevações E com a mesma resolução e com mesmo número de células da matriz I , isto é, uma matriz quadrada de 1205 por 1205 células e com 30 m de resolução r_s . O algoritmo foi fundamentado no algoritmo de simplificação descrito na fase três. Porém, neste algoritmo para alteração de resolução, uma célula e_{lc} recebe a maior elevação dos cinco pontos que descrevem a região representada por uma célula b_{lc} .

3.2.10. Criação da grade binária (fase 8)

Finalmente, na oitava fase, a grade binária B é criada pela aplicação da Equação 3.6, considerando as informações provenientes das fases cinco, seis e sete.

Grades binárias foram criadas através de matrizes T_m obtidas através de especificações do VANT MQ-1 Predator e através de variações de a_n e l_{t_m} . No processo de criação destas grades, a altura de segurança foi especificada como $a_s = 300$ m. Estas grades são apresentadas na Figura 3.8. Nesta figura, as células brancas representam as células navegáveis do ambiente de navegação. As células verdes representam as células rurais com $t_{mcl} > l_{t_m}$. As células na cor ciano representam as células urbanas com $t_{mcl} > l_{t_m}$. As células pretas indicam as células com $e_{lc} \geq (a_n - a_s)$.

Analisando o método para criação de grades binárias e os resultados obtidos, verifica-se que diferentes ambientes de navegação podem ser criados com a mesma informação fornecida pelas fases um, dois, três e sete. Neste caso, estas quatro fases são efetuadas uma única vez e diferentes ambientes de navegação podem ser criados com variações das especificações de VANTs e das especificações de voo. A complexidade computacional do método para modelagem de ambientes de navegação é $O(n_{lg}n_{cg})$ (MEDEIROS e SILVA, 2010a).

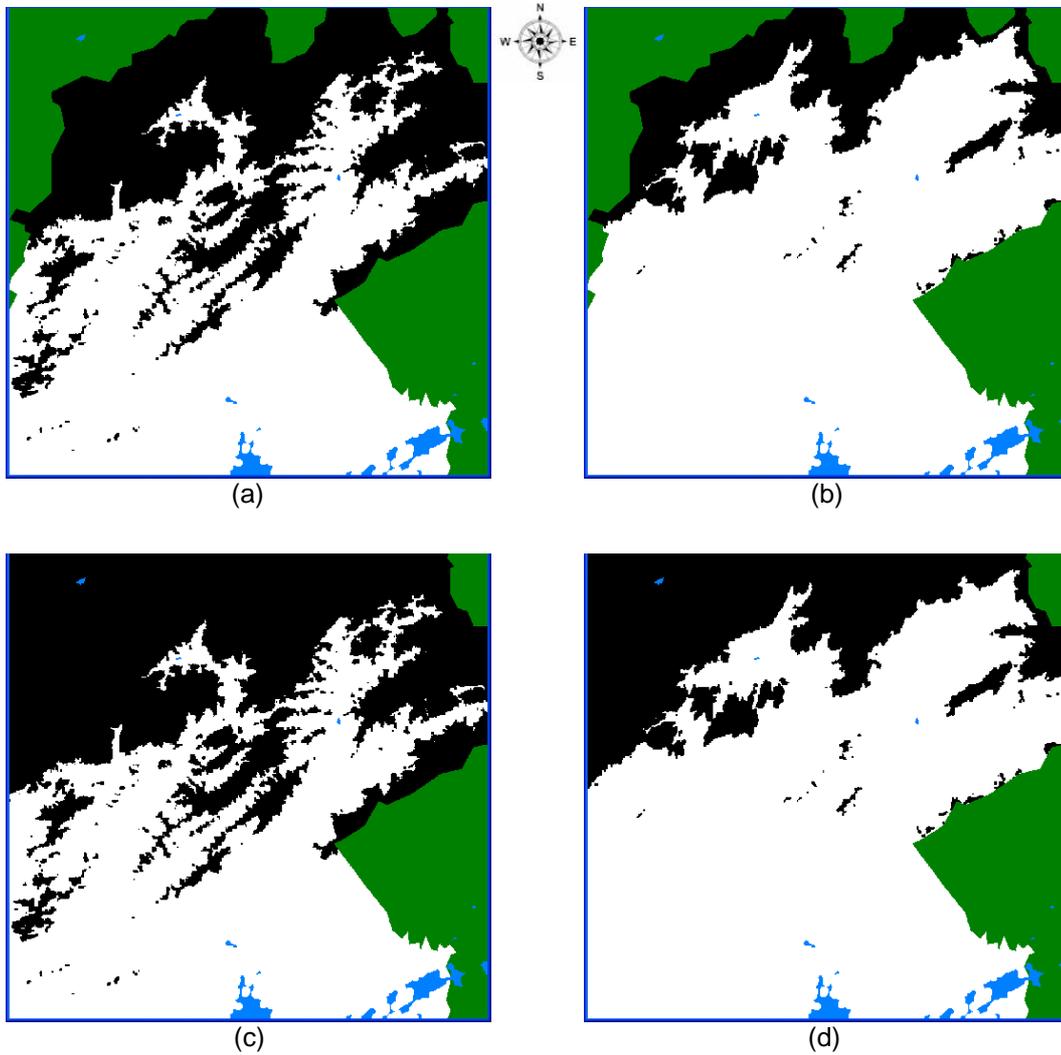


Figura 3.8 – Grades binárias criadas, considerando: (a) $a_n = 1100$ m e $l_{t_m} = 104$ h; (b) $a_n = 1300$ m e $l_{t_m} = 104$ h; (c) $a_n = 1100$ m e $l_{t_m} = 20000$ h; e (d) $a_n = 1300$ m e $l_{t_m} = 20000$ h.

Fonte: Medeiros e Silva (2010a)

3.3. Construção de grafos de visibilidade através da métrica t_m

Os algoritmos descritos no Capítulo 2 foram aplicados à construção de grafos de visibilidade através de grades binárias construídas pelo método apresentado neste capítulo. A Figura 3.9 e a Figura 3.10 apresentam alguns destes conjuntos de grafos de visibilidade construídos, considerando a altura $a_s = 300$ m, as especificações do VANT MQ-1 Predator e $l_{t_m} = 10^4$ h para os grafos da Figura 3.9, e as especificações do VANT RQ-4 Global Hawk e $l_{t_m} = 200000$ h para os grafos da Figura 3.10.

Nestas figuras, os nós e arestas dos grafos são representados na cor vermelha. Os polígonos cinza indicam as envoltórias de segurança em torno das regiões não navegáveis definidas pela Equação 3.6. Neste trabalho, a envoltória de segurança é definida com base na distância de segurança d_s .

Analisando os resultados obtidos, pode-se constatar que a estimacão do número esperado de fatalidades nas regiões navegáveis corresponde ao controle do parâmetro l_{t_m} na Equação 3.6. Através da Equação 3.1 e da Equação 3.6, verifica-se que o parâmetro l_{t_m} é um limite para o número esperado de fatalidades n_{ef} . A redução de l_{t_m} para um VANT causa o aumento do número de células b_{lc} não navegáveis, aumentando a restrição para a navegação do veículo. Observa-se que o decréscimo do parâmetro l_{t_m} permite a diminuição do n_{ef} nas regiões navegáveis das grades binárias criadas pelo método apresentado. Também é importante ser observado que a especificação do parâmetro l_{t_m} pode estar associada com o objetivo da missão de um VANT. Assim, supondo VANTs idênticos, mas com diferentes objetivos de missão, podem ser utilizados diferentes valores de l_{t_m} para um mesmo ambiente de navegação.

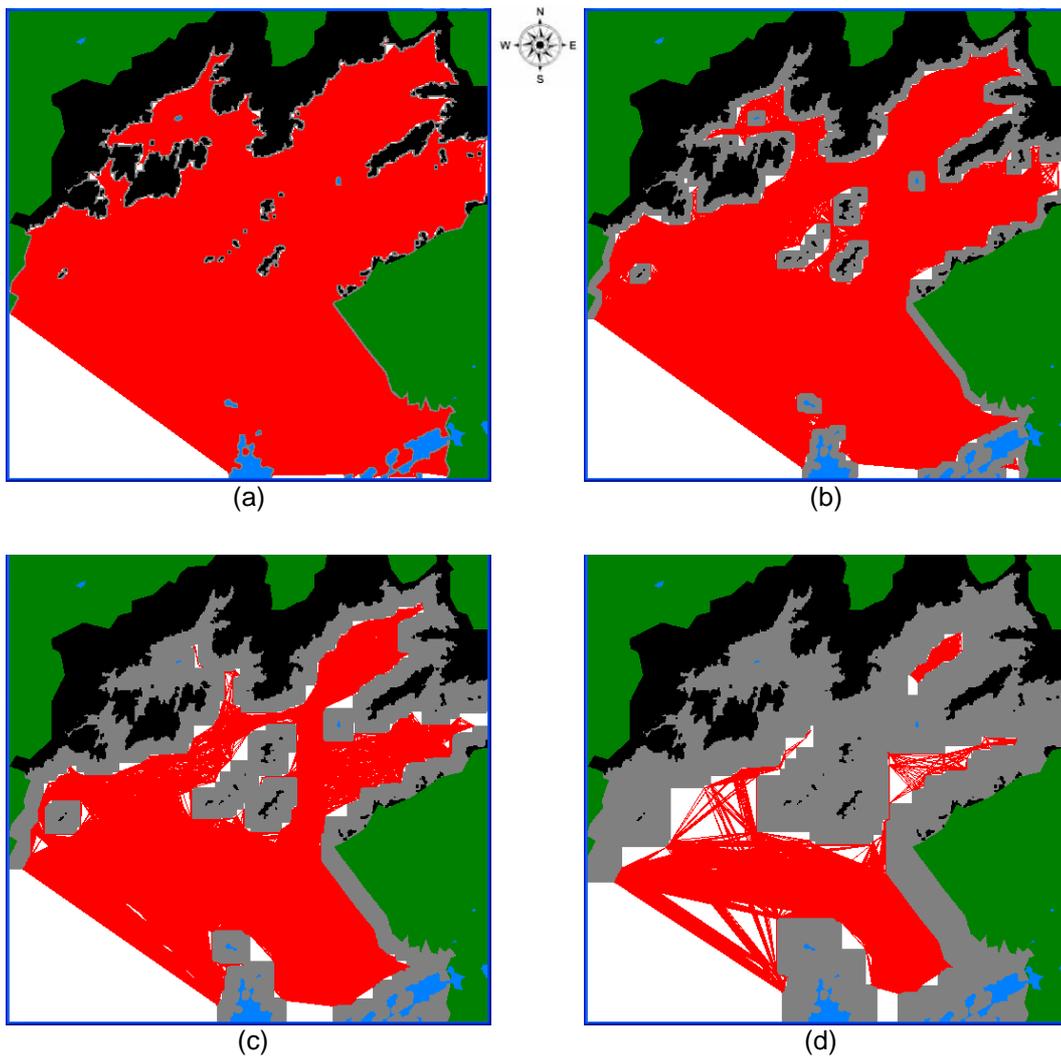


Figura 3.9 – Grafos de visibilidade construídos, considerando: (a) $d_s = 100$ m; (b) $d_s = 500$ m; (c) $d_s = 1000$ m; e $d_s = 2000$ m.
 Fonte: Medeiros e Silva (2010a)

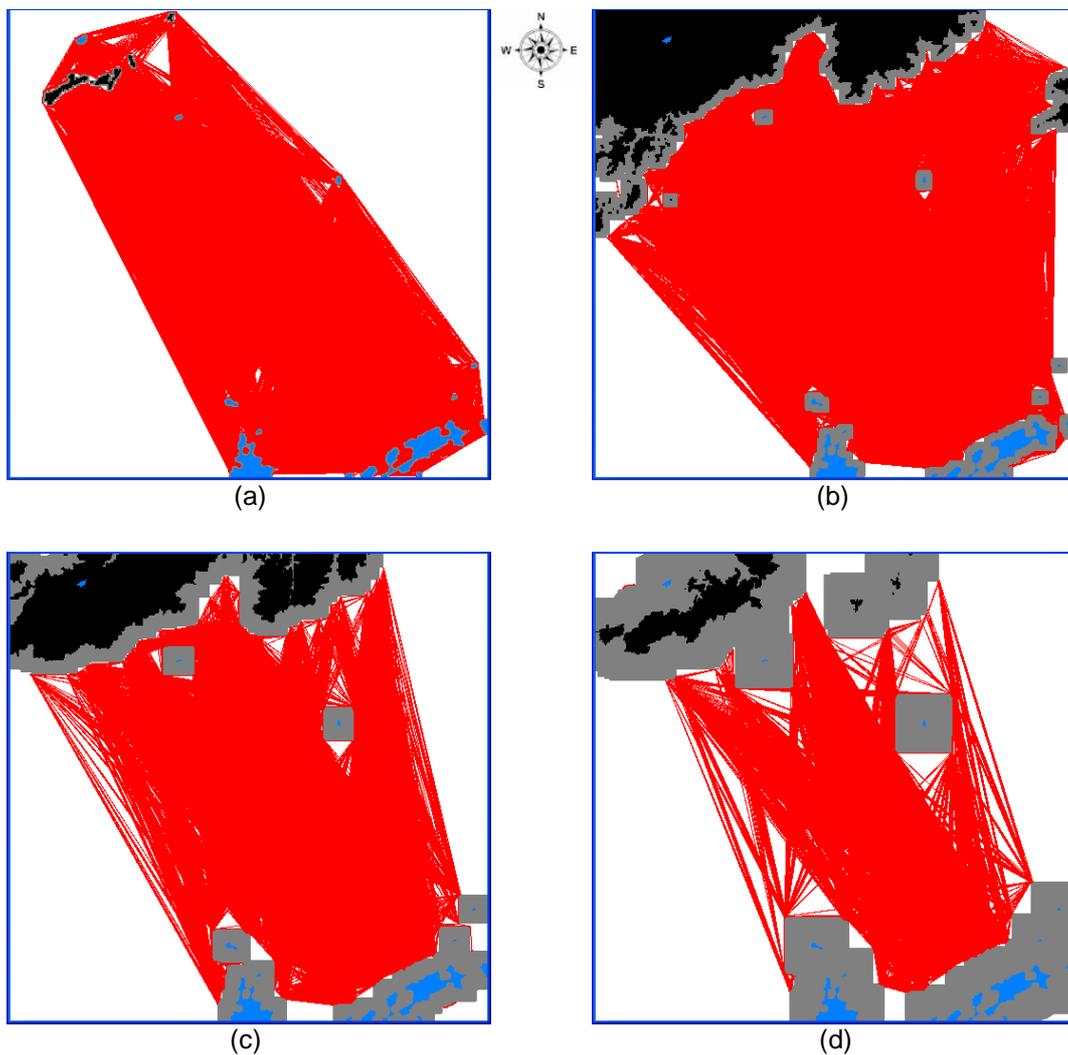


Figura 3.10 – Grafos de visibilidade construídos, considerando: (a) $a_n = 2200$ m e $d_s = 100$ m; (b) $a_n = 1600$ m e $d_s = 500$ m; (c) $a_n = 1800$ m e $d_s = 1000$ m; e (d) $a_n = 2000$ m e $d_s = 2000$ m.

Fonte: Medeiros e Silva (2010a)

As grades binárias criadas pelo método apresentado podem ser usadas no estudo de outros tipos de representação computacional de regiões navegáveis como, por exemplo, os diagramas de Voronoi e as RRTs.

3.4. Planejamento de rotas com grafos de visibilidade construídos por meio da métrica t_m

O algoritmo Dijkstra clássico (DIJKSTRA, 1959) foi aplicado ao planejamento de rotas de menor extensão através dos grafos de visibilidade construídos pelo método apresentado neste capítulo. Considerando um grafo não direcionado (N,A) , onde N e A são o conjunto de nós e de arestas do grafo, respectivamente, o problema de menor rota (BAZARAA et al., 1990; DASKIN, 1995; PAPADIMITRIOU e STEIGLITZ, 1998; PINTO, 2004) é definido matematicamente por

Minimizar

$$\sum_{i_n=1}^{n_n} \sum_{\substack{j_n=1 \\ j_n \neq i_n}}^{n_n} y_{i_n j_n} x_{i_n j_n} \quad (3.8)$$

Sujeito a

$$\sum_{\substack{j_n=1 \\ j_n \neq i_n}}^{n_n} x_{i_n j_n} - \sum_{\substack{j_n=1 \\ j_n \neq i_n}}^{n_n} x_{j_n i_n} = \begin{cases} 1, & \text{se } w'_{i_n} = w'_o \\ -1, & \text{se } w'_{i_n} = w'_d \\ 0, & \text{caso contrário} \end{cases} \quad \forall i_n \quad (3.9)$$

$$x_{i_n j_n} = \begin{cases} 1, & \text{se a aresta que conecta o nó de índice } i_n \\ & \text{com o nó de índice } j_n \text{ pertence ao caminho} \\ 0, & \text{se não pertencer} \end{cases} \quad (3.10)$$

Em que: $y_{i_n j_n}$ é o custo da aresta que conecta o nó de índice i_n ao nó de índice j_n , ou seja, no caso de planejamento de rotas $y_{i_n j_n} = d(w'_{i_n}, w'_{j_n})$, que corresponde à distância entre o nó w'_{i_n} e o nó w'_{j_n} ; w'_o é o nó ou coordenada de origem da rota ou trajetória a ser planejada no grafo; w'_d é o nó ou coordenada de destino da rota ou trajetória a ser planejada no grafo. Observa-se que as coordenadas w'_o e w'_d são inseridas no grafo de visibilidade e podem ser quaisquer coordenadas navegáveis do ambiente de navegação.

A restrição expressa pela Equação 3.9 supõe que exista apenas uma conexão com fluxo de saída, isto é, sentido de navegação de saída do nó w'_o ; exista apenas uma conexão com fluxo de entrada no nó w'_d ; o resultado da otimização seja uma sequência de nós conectados com fluxo que se inicia em w'_o e finaliza em w'_d .

O algoritmo Dijkstra para solução do problema de menor caminho entre dois nós é apresentado na Tabela 3.4. Neste algoritmo, são consideradas as seguintes definições: a é o nó avaliado; v é um nó vizinho do nó a , sendo que os nós v e a são conectados por uma única aresta; $f(a,v)$ é a função de custo, que calcula o custo da rota entre o nó ou coordenada de origem w'_o e o nó v ; $d'(a)$ é o custo entre o nó a e o nó de origem w'_o ; $d(a,v)$ é o custo da aresta que conecta o nó a e o nó v ; $p_r(w'_{i_n})$ é o nó anterior ao nó w'_{i_n} e vizinho do mesmo; e R é a rota planejada entre a coordenada w'_o e a coordenada w'_d .

Tabela 3.4 – Algoritmo Dijkstra para planejamento de rotas de menor extensão entre dois nós de um grafo.

Índice	Algoritmo Dijkstra para Planejamento de Rotas entre Dois Nós
1	para i_n variando de 1 até n_n faça
2	armazenar w'_{i_n} em Q
3	$d'(w'_{i_n}) \leftarrow \begin{cases} 0, & \text{para } w'_{i_n} = w'_o \\ \infty, & \text{para } w'_{i_n} \neq w'_o \end{cases}$
4	$a \leftarrow w'_o$
5	enquanto $a \neq w'_d$ faça
6	remover a de Q
7	para cada vizinho v de a faça
8	$f(a,v) \leftarrow d'(a) + d(a,v)$
9	se $f(a,v) < d'(v)$ então faça
10	$d'(v) \leftarrow f(a,v)$
11	$p_r(v) \leftarrow a$
12	$a \leftarrow w'_{i_n}$, tal que $d'(w'_{i_n})$ é o mínimo ($d'(w'_{i_n})$), para todo $w'_{i_n} \in Q$
13	enquanto $a \neq w'_o$ faça
14	armazenar a na pilha R
15	$a \leftarrow p_r(a)$

Neste trabalho, o custo da aresta que conecta dois nós vizinhos é a distância entre as coordenadas que definem estes nós. A coordenada de origem w_1 e a coordenada de destino w_{n_r} da rota são os respectivos nós w'_o e w'_d .

3.4.1. Análise das rotas planejadas

Esta seção apresenta uma comparação entre os níveis de segurança à população de rotas planejadas com grafos de visibilidade construídos através de dois tipos de grades binárias. O primeiro tipo são as grades binárias criadas pelo método apresentado neste capítulo, em que a métrica t_m é utilizada para a determinação das regiões não navegáveis. O outro tipo são as grades binárias que não consideram a avaliação dos efeitos da queda de VANTs, como os grafos apresentados no Capítulo 2. A Figura 3.11 apresenta um exemplo de rota planejada pelo algoritmo Dijkstra, considerando um grafo de visibilidade construído através de uma grade binária obtida com a métrica t_m e um grafo de visibilidade construído por uma grade binária obtida sem a métrica t_m .

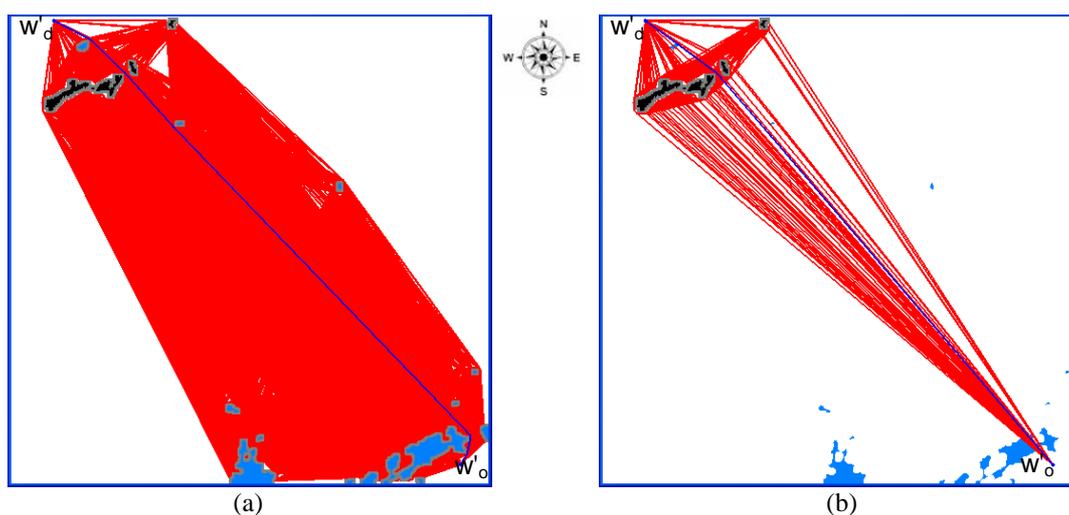


Figura 3.11 – Rotas planejadas com grafos de visibilidade construídos através de: (a) grade binária obtida com a métrica t_m ; e (b) grade binária obtida sem a métrica t_m .

Fonte: Medeiros e Silva (2010a)

Na Figura 3.11, as rotas são representadas na cor azul escuro. A métrica t_m da grade binária da Figura 3.11a foi calculada com base nas especificações do VANT RQ-4 Global Hawk. A latitude e a longitude da coordenada inicial w'_o são iguais a -23.15 e -45.75, respectivamente. A latitude e a longitude da coordenada w'_d são iguais a -22.84 e -46.05, respectivamente. As rotas foram

planejadas considerando $a_n = 2200$ m, $a_s = 300$ m e $d_s = 200$ m. A Tabela 3.5 apresenta comparações entre a segurança proporcionada por estas rotas planejadas. O valor máximo de t_m e o valor máximo de n_{ef} ao longo destas rotas são apresentados nesta tabela.

Tabela 3.5 – Análise da segurança das rotas planejadas.

Rota Planejada	l_{t_m}	máximo t_m	máximo n_{ef}
Grade Binária Obtida com t_m	200000 h	144581.564 h	0.0145 fatalidade
Grade Binária Obtida sem t_m	200000 h	27524573.867 h	2.7525 fatalidades

Qualquer aresta do grafo de visibilidade construído através de grade binária com a métrica t_m não intercepta regiões não navegáveis definidas por células b_{lc} com $e_{lc} \geq (a_n - a_s)$ e por células b_{lc} com $t_{mlc} > l_{t_m}$. Portanto, qualquer rota planejada através de grafos de visibilidade deste tipo é livre de colisão e garante segurança à população do ambiente de navegação, considerando os valores especificados para a_n , a_s , l_{t_m} e d_s . Analisando a Figura 3.11a e a Tabela 3.5, verifica-se que a rota apresentada é livre de colisão e a métrica t_m é inferior ao limite l_{t_m} ao longo desta rota. Na ocorrência da queda de um VANT Global Hawk durante a navegação por esta rota, o número esperado de fatalidades é inferior ou igual a $n_{es} l_{t_m}$. Assim, a segurança à população pode ser controlada no processo de construção deste tipo de grafo de visibilidade por meio do ajuste de l_{t_m} .

Como descrito no Capítulo 2, um grafo de visibilidade clássico, construído por meio de uma grade binária sem o uso da métrica t_m , permite o planejamento de rotas livres de situações de colisão, considerando as regiões não navegáveis definidas pelas células b_{lc} com $e_{lc} \geq (a_n - a_s)$. Devido à restrição imposta pela métrica t_m , a área não navegável da grade binária com tal métrica pode ser menor do que a área não navegável da grade sem a métrica t_m . Deste modo, as rotas planejadas com um grafo de visibilidade clássico podem ser menores que as rotas planejadas com o grafo associado à métrica t_m , como apresentado

na Figura 3.11. Entretanto, não há qualquer controle da segurança à população em possíveis acidentes com queda de VANTs, pois a métrica t_m não é considerada na construção do grafo de visibilidade clássico. Por exemplo, considerando que um valor inferior ou igual a 200000 h é aceitável para t_m , a rota apresentada na Figura 3.11b é livre de colisão, porém, viola este nível aceitável de segurança porque intercepta regiões com $t_m > l_{t_m}$. Na Figura 3.11b, as regiões com $t_m > l_{t_m}$ foram introduzidas com o propósito de demonstrar a falta de segurança da rota planejada.

3.4.2. Limite para a distância de segurança

Esta subseção aborda a relação entre a distância de segurança d_s e a distribuição espacial da métrica t_m .

A Figura 3.12 apresenta um exemplo de trajetória da queda de um VANT. Nesta figura, o comprimento do segmento de reta $\overline{z_1 z_2}$ indica a distância entre o segmento de reta $\overline{w_{i-1} w_i}$ e o ponto de impacto na superfície. As coordenadas w_{i-1} e w_i são elementos de uma rota ou trajetória de navegação.

Como apresentado na Figura 3.13, a envoltória de segurança das regiões não navegáveis e a distância d_s devem ser especificadas de modo a evitar um impacto de um VANT em qualquer superfície com $t_m > l_{t_m}$. Considerando $\Delta_1 = 0$, qualquer valor de d_s menor que $a_n / \text{tg}(\beta_v)$ pode causar o impacto de um VANT em uma região com $t_m > l_{t_m}$. Portanto, um modo de garantir mais segurança para evitar quedas de VANTs em regiões com $t_m > l_{t_m}$ é calcular um limite inferior para a envoltória de segurança e para a distância d_s fundamentado na altitude de navegação a_n e no ângulo β_v .

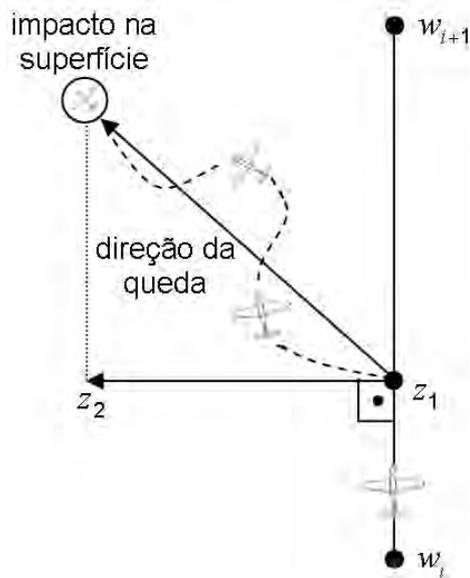


Figura 3.12 – Exemplo de uma trajetória de queda de um VANT.
 Fonte: Medeiros e Silva (2010a)

Assim, o limite inferior l_{d_s} para a envoltória de segurança e para d_s foi proposto neste trabalho. Este limite é definido por

$$l_{d_s} = \Delta_1 + \frac{a_n}{\text{tg}(\beta_v)} \quad (3.8)$$

Em que: Δ_1 é a distância entre a região com $t_m > l_{t_m}$ e o ponto de colisão na superfície; e, como definido anteriormente, β_v é ângulo formado pela superfície de colisão e a projeção da direção de queda do VANT.

Exemplos de rotas planejadas com dois conjuntos de grafos de visibilidade são apresentados na Figura 3.14. A diferença entre as grades binárias que modelam os ambientes de navegação é definida pela variação da envoltória de segurança e da distância d_s . Pode-se verificar que a segunda rota é menor que a primeira. Entretanto, a menor distância entre a segunda rota e alguma região não navegável é $d_s = 500$ m, o que viola o limite l_{d_s} e torna esta rota menos segura para a população do que a primeira. Isto pode ser provado através da

aplicação da Equação 3.8 que retorna $l_{d_s} = 1254.701$ m, para $\beta_v = 60^\circ$, $a_n = 2000$ m e $\Delta_1 = 100$ m.

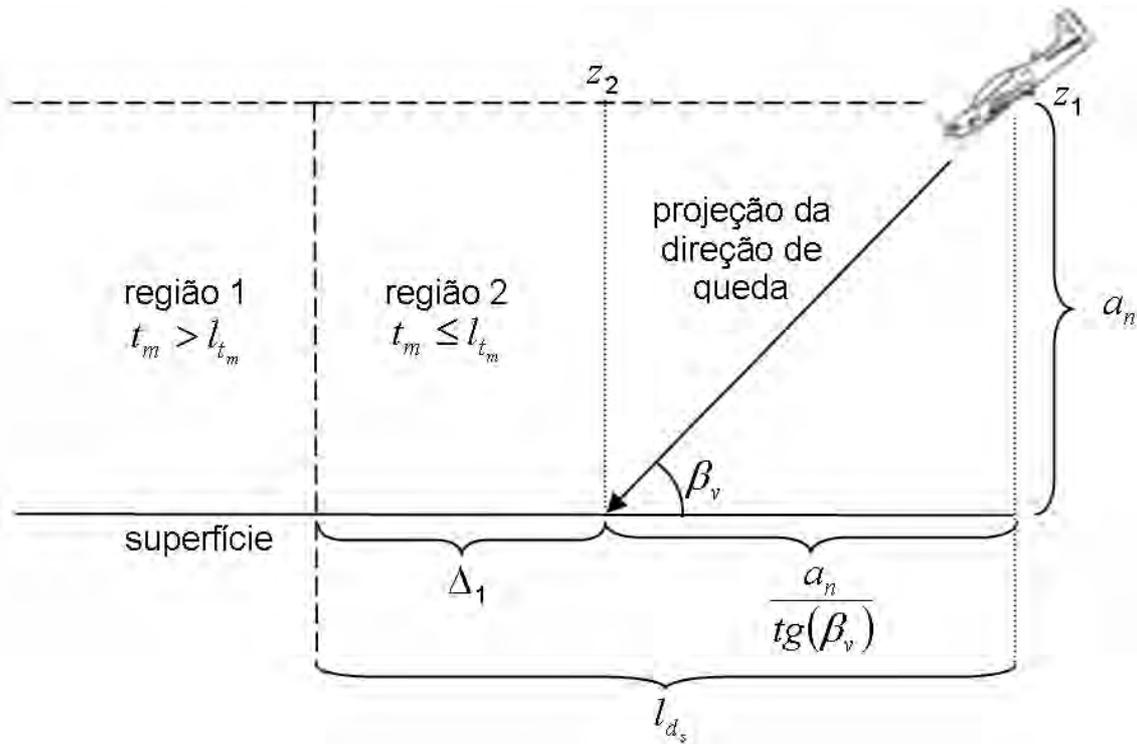


Figura 3.13 – Esquema para definição de l_{d_s} .

Fonte: Medeiros e Silva (2010a)

Analisando as Equações 3.1, 3.2 e 3.3, observa-se que o valor de t_m aumenta quando o valor de α_v diminui. Além deste fato, $t_m \leq l_{t_m}$ para qualquer ângulo de descida superior ou igual ao ângulo de queda α_v usado na criação da grade binária. Assim, o decréscimo de α_v usado na criação da grade binária permite mais segurança à população do ambiente de navegação.

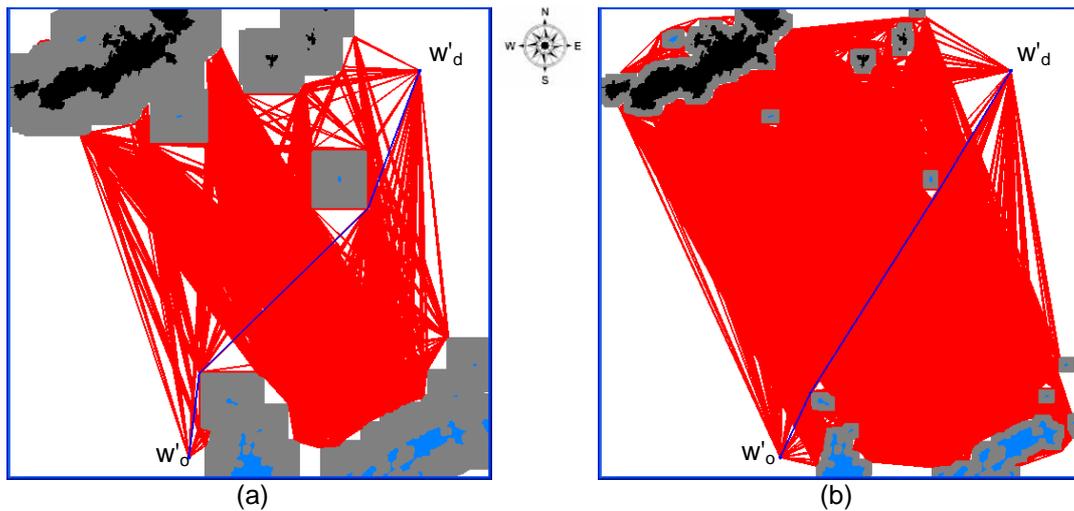


Figura 3.14 – Rotas planejadas através de grafos de visibilidade construídos, considerando as especificações do VANT RQ-4 Global Hawk, $a_n = 2000$ m, $a_s = 300$ m, $l_{t_m} = 200000$ h, (a) $d_s = 2000$ m e (b) $d_s = 500$ m.
 Fonte: Medeiros e Silva (2010a)

Através dos resultados obtidos, pode-se observar que os grafos de visibilidade construídos pelo método apresentado permitem o planejamento de rotas com controle do número esperado de fatalidades em acidentes com queda de VANTs. A especificação da envoltória de segurança e de d_s por meio do limite l_{d_s} controla a distância entre o VANT e as regiões não navegáveis, o que torna possível o aumento de segurança para as populações das regiões com $t_m > l_{t_m}$. Este limite pode ser relacionado com regulamentações futuras de segurança para VANTs.

Embora o estudo deste capítulo tenha sido realizado com o planejamento de rotas, o mesmo estudo se aplica ao planejamento de trajetórias para VANTs.

4 SIMPLIFICAÇÃO DE GRAFOS DE VISIBILIDADE

Grafos de visibilidade podem apresentar um elevado número de nós e de arestas, quando construídos através de regiões não navegáveis definidas por modelos digitais de elevação (MEDEIROS e SILVA, 2008). Isto pode ser confirmado pelos resultados obtidos no Capítulo 2. Outro exemplo deste problema é o grafo apresentado na Figura 4.1 que possui 339 nós e 11319 arestas. Neste exemplo, os polígonos pretos indicam as regiões não navegáveis, e os nós e arestas dos grafos de visibilidade são representados na cor cinza.

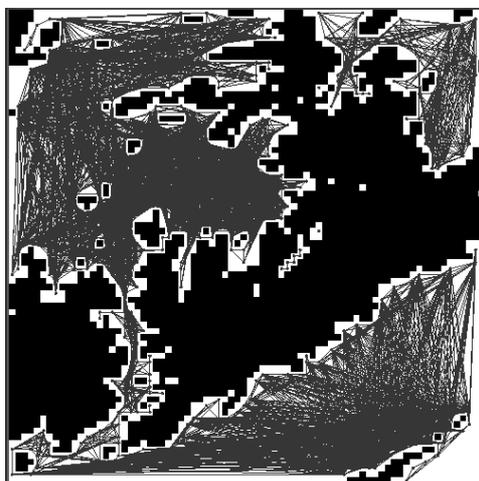


Figura 4.1 – Exemplo de grafo de visibilidade construído através de um modelo digital de elevação.

Fonte: Medeiros e Silva (2008)

O aumento do número de nós (n_n) e do número de arestas (n_a) em um grafo causa o aumento dos custos computacionais de tempo e de armazenamento para métodos de planejamento de rotas e trajetórias baseado em grafos. Um exemplo é aplicação do algoritmo Dijkstra ao problema de planejamento de rotas com as regiões navegáveis do ambiente de navegação representadas por grafos de visibilidade. Este algoritmo apresenta complexidade $O(n_a \log(n_n))$ no pior caso, considerando a implementação do grafo como uma lista de adjacência e utilizando busca binária para o cálculo do nó com menor custo.

Este problema motivou o estudo da simplificação de grafos de visibilidade através da redução do número de nós e de arestas.

Neste trabalho, foram identificadas e estudadas duas abordagens para a simplificação de um grafo de visibilidade. A primeira abordagem é a simplificação antes da construção dos grafos de visibilidade. Neste caso, ocorre a simplificação dos polígonos que representam as regiões não navegáveis do ambiente de navegação através da redução dos vértices que determinam os nós dos grafos de visibilidade. A outra abordagem é a simplificação dos grafos de visibilidade já construídos. Neste caso, métodos são aplicados diretamente aos grafos de visibilidade, visando à eliminação de nós ou arestas.

A seção 4.1 apresenta a abordagem baseada na simplificação dos polígonos que representam as regiões não navegáveis do ambiente de navegação. A seção 4.2 apresenta a segunda abordagem baseada na simplificação dos grafos de visibilidade já construídos.

4.1. Simplificação por redução de vértices de polígonos

Nesta abordagem, o número de vértices de um polígono é reduzido através da aplicação de métodos para o cálculo de envoltórias convexas ou métodos baseados em métricas ou parâmetros para eliminação de vértices (DAVIS JUNIOR e QUEIROZ, 2005).

4.1.1. Métodos para o cálculo de envoltórias convexas

São exemplos de métodos para o cálculo da envoltória convexa de um conjunto de coordenadas ou pontos: o algoritmo Quickhull (EDDY, 1977; BYKAT, 1978; GREEN e SILVERMAN, 1979); o algoritmo Mergehull (PREPARATA e SHAMOS, 1985); o algoritmo Varredura de Graham (GRAHAM, 1972); e o algoritmo Marcha de Jarvis (JARVIS, 1973). Porém, o cálculo da envoltória convexa de um conjunto de vértices de polígonos que

representam regiões não navegáveis pode gerar regiões de invisibilidade, causando perda de área navegável, como apresentado no exemplo da Figura 4.2, ou o aumento da rota ou trajetória planejada através do grafo construído através dos polígonos simplificados. Neste exemplo, os polígonos pretos representam as regiões não navegáveis do ambiente. Os polígonos cinza representam as regiões de invisibilidade geradas pelos vértices intermediários convexos após o cálculo da envoltória convexa.

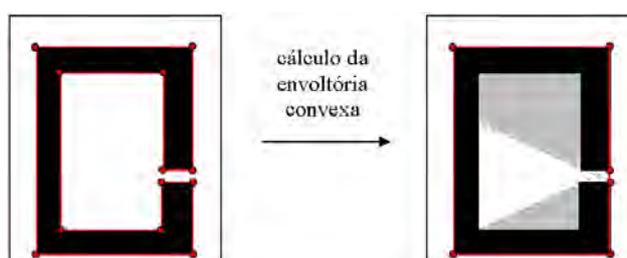


Figura 4.2 – Exemplo de cálculo de envoltória convexa.

Neste trabalho, a região de invisibilidade de um polígono é definida como aquela em que seus pontos internos não podem ser conectados diretamente a qualquer vértice intermediário convexo do polígono, considerando tais conexões como segmentos de reta que não cortam regiões não navegáveis. Portanto, não é possível planejar uma rota de navegação entre qualquer ponto de uma área de invisibilidade e qualquer nó do grafo de visibilidade construído através de vértices intermediários convexos do mesmo polígono.

O algoritmo varredura de Graham foi aplicado a cada polígono que representa as regiões não navegáveis dos ambientes de navegação apresentados na Figura 4.3b e Figura 4.4b. Estes ambientes de navegação são os mesmos ambientes apresentados na Figura 2.9a e na Figura 2.10b. Porém, estes ambientes são novamente apresentados com a finalidade de destacar as representações poligonais dos mesmos. Na Figura 4.3b e na Figura 4.4b, os polígonos vermelhos indicam os polígonos a serem simplificados pelo processo de criação de envoltórias convexas. Observa-se que as lacunas das regiões não navegáveis não são consideradas neste trabalho.

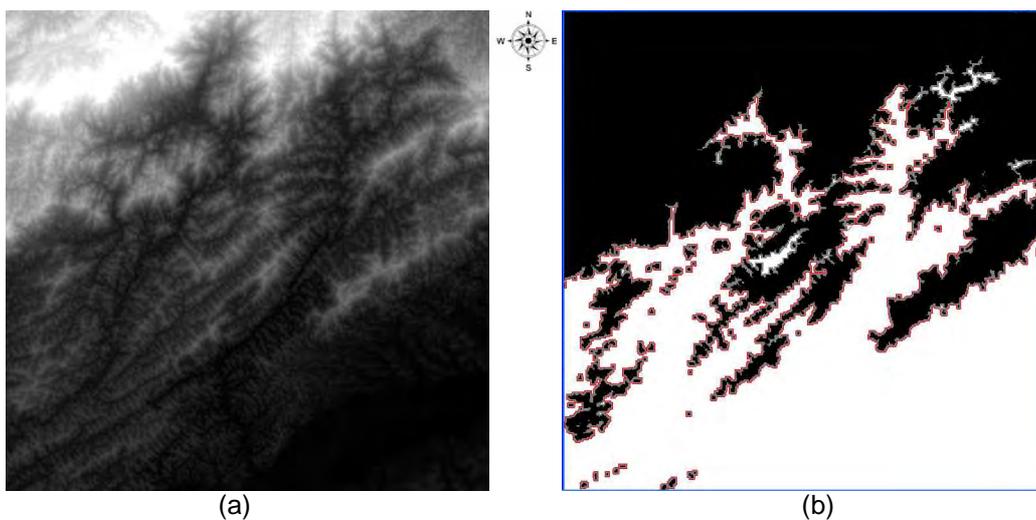


Figura 4.3 – (a) Modelo digital de elevação e respectivo (b) ambiente de navegação considerando uma altitude de navegação de 1100 metros e uma altura de segurança de 300 metros.

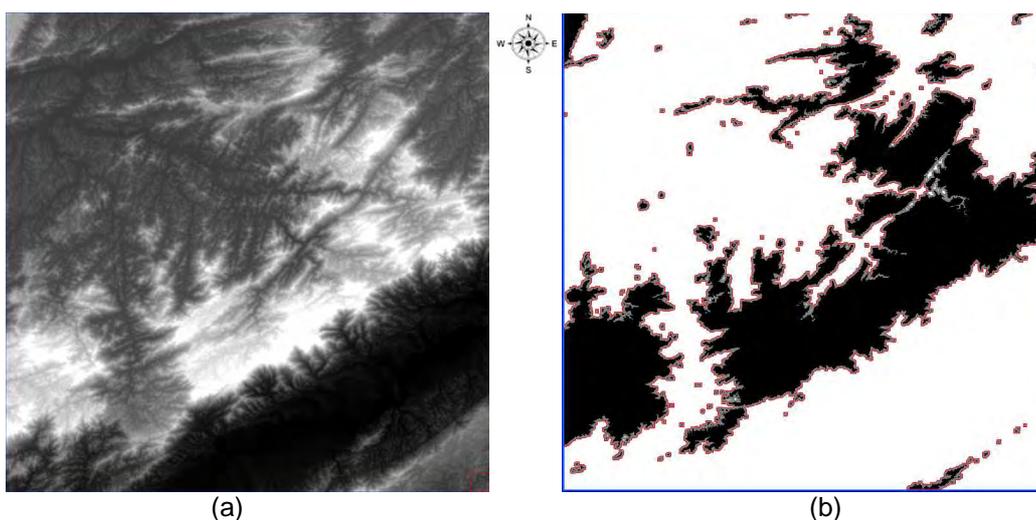


Figura 4.4 – (a) Modelo digital de elevação e respectivo (b) ambiente de navegação considerando uma altitude de navegação de 1500 metros e uma altura de segurança de 300 metros.

Os resultados obtidos com a aplicação do algoritmo de Graham são apresentados na Figura 4.5 e na Figura 4.6. Os polígonos e pontos vermelhos representam as envoltórias convexas obtidas para cada polígono. As regiões na cor cinza representam envoltórias de segurança das regiões não navegáveis.

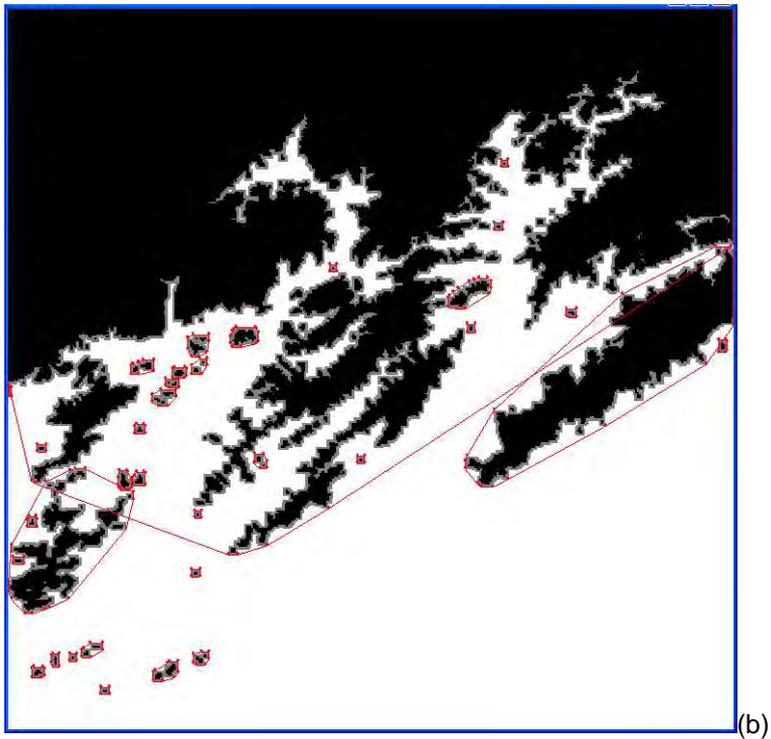
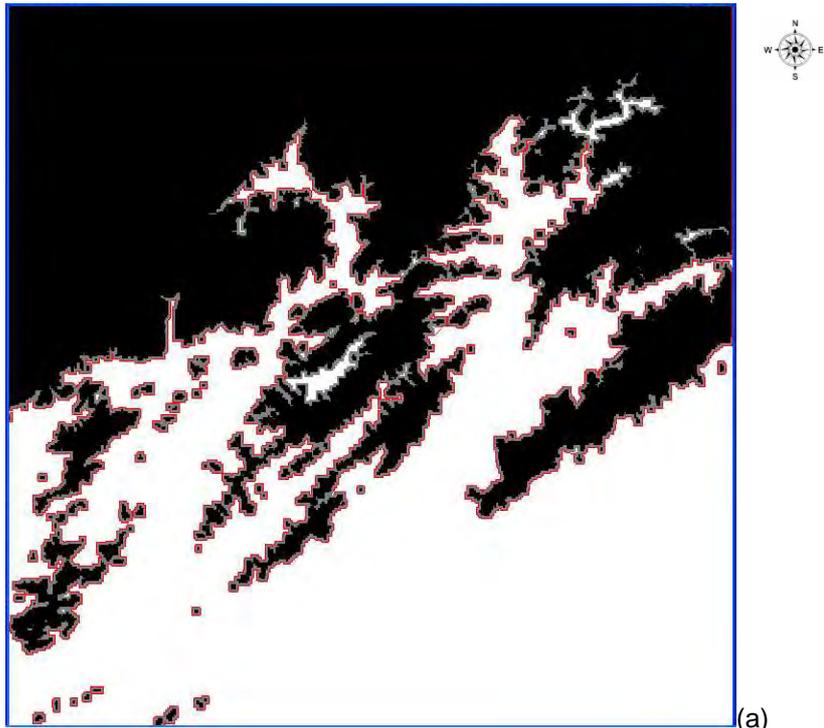


Figura 4.5 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) envoltórias convexas obtidas com a aplicação do algoritmo de Graham a tais representações.

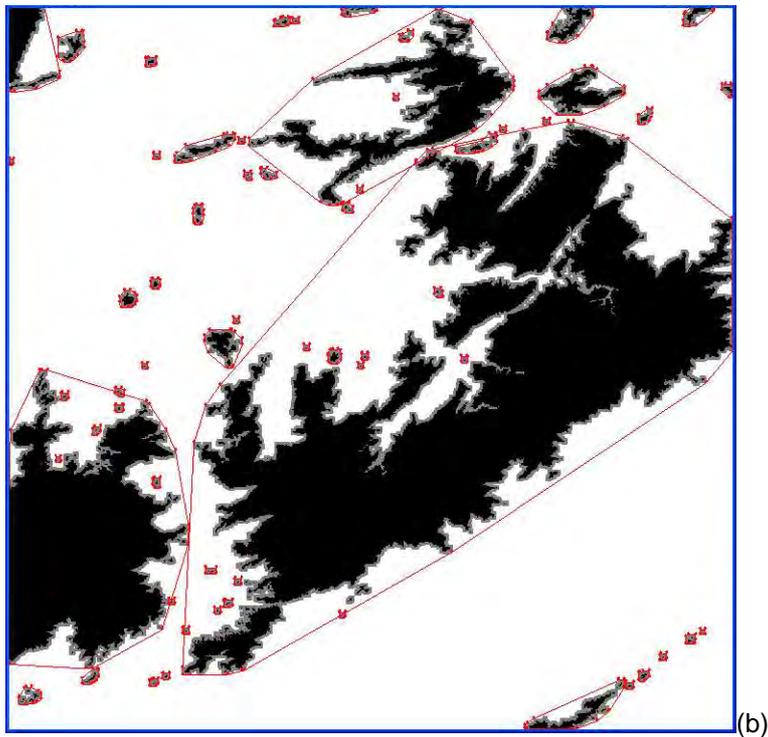
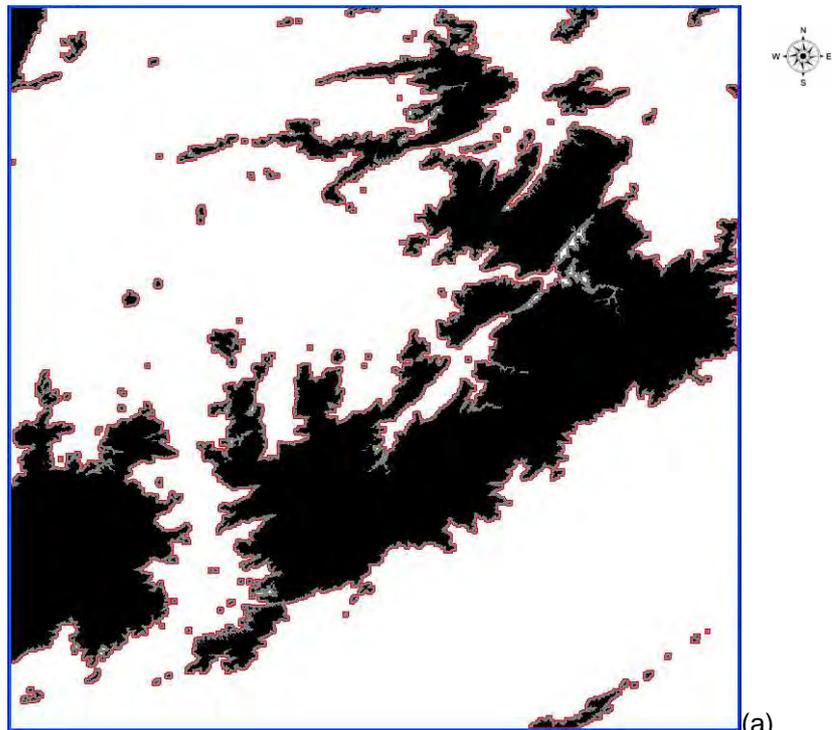


Figura 4.6 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) envoltórias convexas obtidas com a aplicação do algoritmo de Graham a tais representações.

Comparações entre os ambientes de navegação originais e os ambientes de navegação obtidos após a simplificação pela criação de envoltórias convexas são apresentadas na Tabela 4.1 e na Tabela 4.2. Analisando os resultados, comprova-se que a simplificação por envoltória convexa ocasionou a criação de regiões de invisibilidade com significativa área total.

É importante ser observado que uma região de invisibilidade determinada pelos vértices intermediários convexos de um polígono pode ser parcialmente ou totalmente visível aos vértices intermediários convexos de outro polígono. Entretanto, neste trabalho, a área total de invisibilidade é o somatório das áreas das regiões de invisibilidade calculadas separadamente para cada polígono que representa o ambiente de navegação.

Tabela 4.1 – Comparações entre a representação poligonal do primeiro ambiente de navegação e a representação obtida após a simplificação por envoltória convexa.

Características	Original	Simplificado
Número de vértices intermediários convexos	1489	207
Área total de invisibilidade	-	190.1 km ²
Área total do ambiente de navegação	1306.9 km ²	1306.9 km ²
Relação da área de invisibilidade com a área total	-	14.55%
Área navegável total	654.8 km ²	464.7 km ²
Relação da área de invisibilidade com a área navegável total	-	29.04%

Tabela 4.2 – Comparações entre a representação poligonal do segundo ambiente de navegação e a representação obtida após a simplificação por envoltória convexa.

Características	Original	Simplificado
Número de vértices intermediários convexos	3937	479
Área total de invisibilidade	-	4422.7 km ²
Área total do ambiente de navegação	11683.5 km ²	11683.5 km ²
Relação da área de invisibilidade com a área total	-	37.86%
Área navegável total	7739.1 km ²	3316.4 km ²
Relação da área de invisibilidade com a área navegável total	-	57.15%

Considerando o contexto de planejamento de rotas ou trajetórias baseado em grafos de visibilidade e analisando os resultados obtidos, pode-se concluir que a eliminação de um vértice intermediário convexo deve estar relacionada com a área de invisibilidade gerada pelo mesmo.

4.1.2. Métodos baseados em métricas para eliminação de vértices

Estes métodos caracterizam-se pelo ajuste de parâmetros baseados em métricas para a eliminação de vértices de um polígono. Alguns dos métodos baseados em métricas para a eliminação de vértices são: o algoritmo do k -ésimo vértice (TOBLER, 1964); o algoritmo Lang (LANG, 1969); o algoritmo Ramer-Douglas-Peucker (RAMER, 1972; DOUGLAS e PEUCKER, 1973); o algoritmo Reumann-Witkam (REUMANN e WITKAM, 1973); o algoritmo de Jenks (JENKS, 1981); o algoritmo Opheim (OPHEIM, 1981); o algoritmo Visvalingam-Whyatt (VISVALINGAM e WHYATT, 1993); o algoritmo Zhao-Saalfeld (ZHAO e SAALFELD, 1997); o algoritmo baseado em agrupamento de vértices, proposto em (KANAYA et al., 2005); algoritmo que utiliza, no processo de simplificação, ângulos formados por sequências orientadas de vértices (RANGAYYAN et al., 2008); o algoritmo proposto em (TANG et al., 2011), que se baseia em triangulação de Delaunay; e o algoritmo de simplificação proposto em (PARK e YU, 2011) que utiliza uma combinação de métricas quantitativas definidas por vértices como, por exemplo, distâncias e ângulos.

Analisando o funcionamento destes algoritmos, verifica-se que os mesmos não relacionam a eliminação de vértices intermediários convexos com as áreas de invisibilidade geradas. Além disso, verifica-se que nestes algoritmos a eliminação de vértices intermediários convexos pode causar o problema de ocultação de obstáculos, quando aplicados à simplificação de polígonos que representam as regiões não navegáveis de um ambiente de navegação.

No problema de ocultação, alguma região não navegável é modelada como navegável no polígono resultante do processo de simplificação. Esta ocultação pode ocasionar situações de colisão, caso o grafo de visibilidade gerado

através do polígono simplificado seja utilizado para planejar rotas ou trajetórias de navegação para um veículo. Considerando os vértices de um polígono numerados no sentido horário, a ocultação de regiões não navegáveis ocorre quando um vértice intermediário convexo v_{i_v} é eliminado entre dois vértices v_{i_v-p} e v_{i_v+q} e a sequência de vértices $v_{i_v-p}v_{i_v}v_{i_v+q}$ está no sentido horário. A Figura 4.7 apresenta um exemplo de uma situação de colisão gerada pelo problema de ocultação de obstáculos. O polígono preto indica a região não navegável original. Os segmentos de reta pontilhados na cor cinza representam o contorno do polígono simplificado. O grafo de visibilidade construído através dos vértices do polígono simplificado é representado por nós e arestas na cor cinza. A rota de navegação é representada em vermelho. Através do exemplo, pode-se perceber que a rota planejada intercepta uma região não navegável.

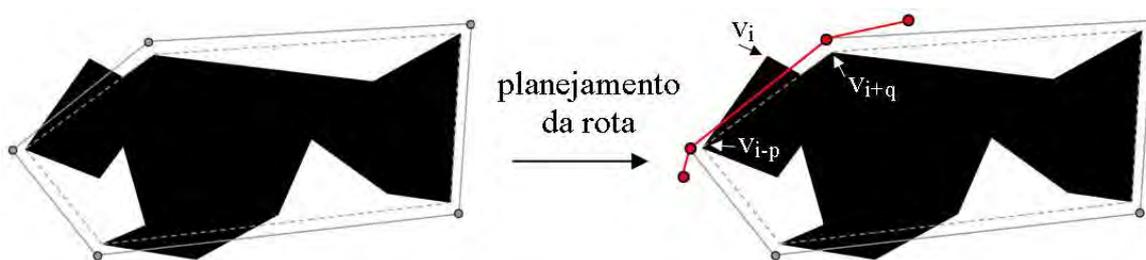


Figura 4.7 – Exemplo do problema de ocultação de obstáculos no planejamento de rotas.

Neste trabalho, o algoritmo Ramer-Douglas-Peucker e o algoritmo Visvalingam-Whyatt foram selecionados para exemplificar o problema de ocultação de obstáculos, sendo aplicados ao problema de simplificação dos polígonos que representam os ambientes de navegação da Figura 4.3b e da Figura 4.4b. O algoritmo Ramer-Douglas-Peucker foi escolhido por ser o algoritmo mais utilizado em problemas de simplificação de poligonais (DAVIS JUNIOR e QUEIROZ, 2005). Já o algoritmo Visvalingam-Whyatt foi escolhido por utilizar a área entre três vértices consecutivos como métrica para eliminação de vértices. As áreas das sequências de três vértices consecutivos são ordenadas de modo

crescente, e o algoritmo elimina o vértice central de cada sequência enquanto a área da sequência for inferior a uma área limite a_{limite} . O algoritmo Visvalingam-Whyatt é apresentado na Tabela 4.3.

Tabela 4.3 – Algoritmo Visvalingam-Whyatt.

Índice	Algoritmo Visvalingam-Whyatt
1	calcular a área a_{i_v} de cada sequência de três vértices do polígono
2	armazenar as áreas em A
3	gerar a ordenação crescente das áreas armazenadas em A
4	$i_v \leftarrow 1$
5	faça
6	se $a_{i_v} < a_{limite}$ então faça
7	eliminar o segundo vértice da sequência de índice i_v
8	calcular as novas áreas geradas devido à eliminação do vértice
9	gerar nova ordenação crescente das áreas armazenadas em A
10	atualizar n_v
11	enquanto $(a_{i_v} < a_{limite})$ e $(n_v > 3)$

O algoritmo Ramer-Douglas-Peucker utiliza, para a eliminação de vértices, a métrica distância ortogonal dos vértices de uma sequência $[v_i, v_{i+1}, \dots, v_{f-1}, v_f]$ ao segmento de reta que une os vértices que delimitam tal sequência. Se a distância $d_{máx}$ do vértice v_k de maior distância ortogonal for superior à distância limite d_{limite} , então o algoritmo é acionado recursivamente para as sequências de vértices $[v_i, \dots, v_k]$ e $[v_k, \dots, v_f]$. Caso contrário, todos os vértices entre v_i e v_f são eliminados. O algoritmo Ramer-Douglas-Peucker é apresentado na Tabela 4.4.

Tabela 4.4 – Algoritmo Ramer-Douglas-Peucker.

Índice	Algoritmo Ramer-Douglas-Peucker (i, f)
1	$d_{máx} \leftarrow 0$
2	$i_{máx} \leftarrow 0$
3	para i_v variando de $(i+1)$ até $(f-1)$ faça
4	$d \leftarrow$ distância ortogonal entre o vértice v_{i_v} e o segmento de reta $\overline{v_i v_f}$
5	se $d > d_{máx}$ então faça
6	$d_{máx} \leftarrow d$
7	$i_{máx} \leftarrow i_v$
8	se $d_{máx} \geq d_{limite}$ então faça
9	aplicar o algoritmo Ramer-Douglas-Peucker ($i, i_{máx}$)
10	aplicar o algoritmo Ramer-Douglas-Peucker ($i_{máx}, f$)
11	senão faça
11	eliminar vértices v_{i_v} tal que $i < i_v < f$

A Tabela 4.5 apresenta uma comparação entre os resultados obtidos com a aplicação do algoritmo Ramer-Douglas-Peucker, considerando o ambiente de navegação apresentado na Figura 4.3b para diferentes valores de d_{limite} . Os respectivos polígonos simplificados pelo algoritmo são apresentados nas Figuras 4.8, 4.9 e 4.10.

Tabela 4.5 – Comparação entre a representação poligonal do primeiro ambiente de navegação e as representações simplificadas pelo algoritmo Ramer-Douglas-Peucker.

Características	Original	Simplificado		
		$d_{limite} = 200 \text{ m}$	$d_{limite} = 500 \text{ m}$	$d_{limite} = 1000 \text{ m}$
Número total de vértices	2668	703	324	197
Redução	-	73.65%	87.86%	92.62%

A Tabela 4.6 apresenta uma comparação entre os resultados obtidos com a aplicação do algoritmo Ramer-Douglas-Peucker, considerando o ambiente de navegação apresentado na Figura 4.4b para diferentes valores de d_{limite} . Os respectivos polígonos simplificados pelo algoritmo são apresentados nas Figuras 4.11, 4.12 e 4.13.

Tabela 4.6 – Comparação entre a representação poligonal do segundo ambiente de navegação e as representações simplificadas pelo algoritmo Ramer-Douglas-Peucker.

Características	Original	Simplificado		
		$d_{limite} = 200 \text{ m}$	$d_{limite} = 500 \text{ m}$	$d_{limite} = 1000 \text{ m}$
Número total de vértices	7649	1824	947	515
Redução	-	76.16%	87.62%	93.27%

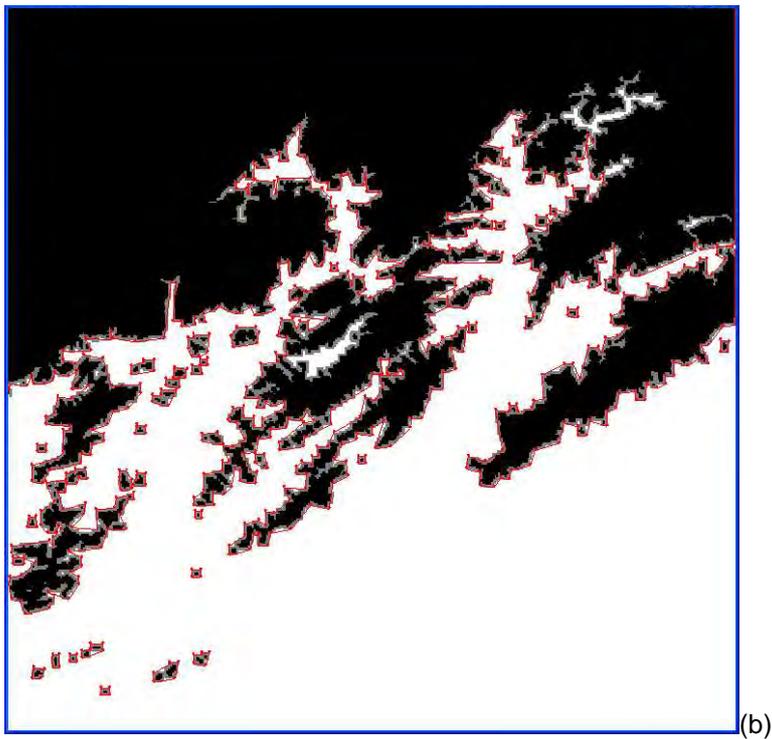
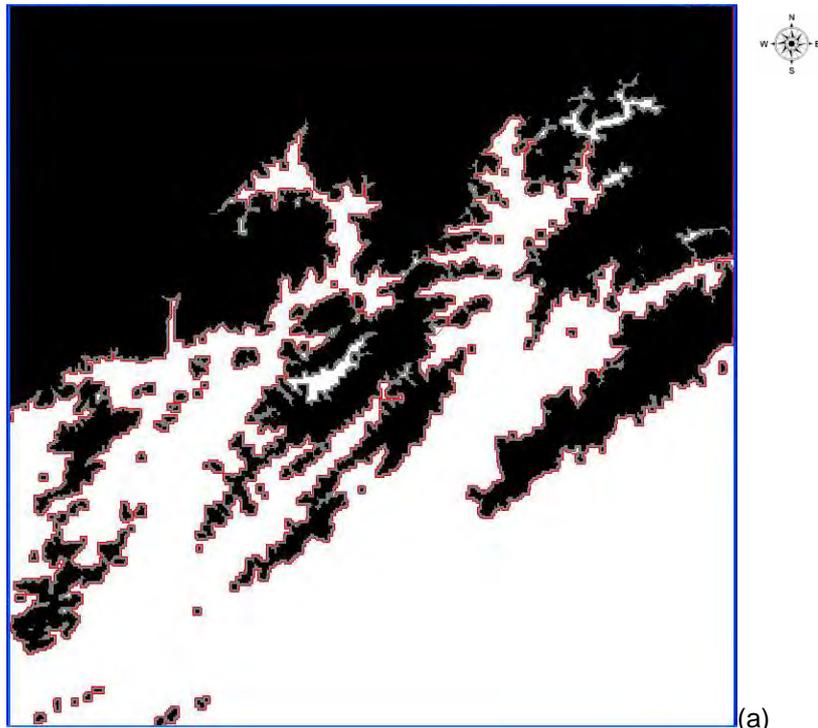


Figura 4.8 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 200$ m.

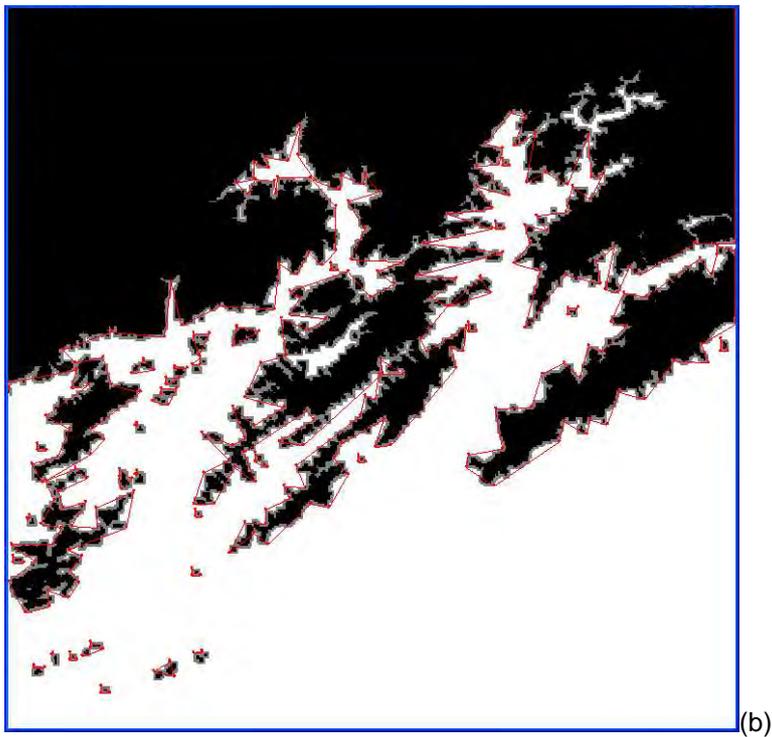
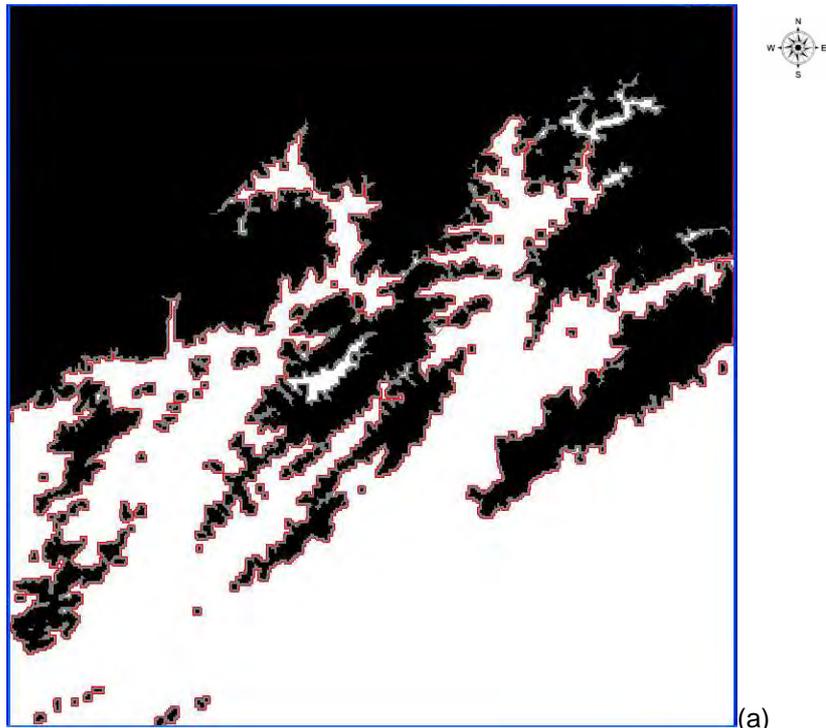


Figura 4.9 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 500$ m.

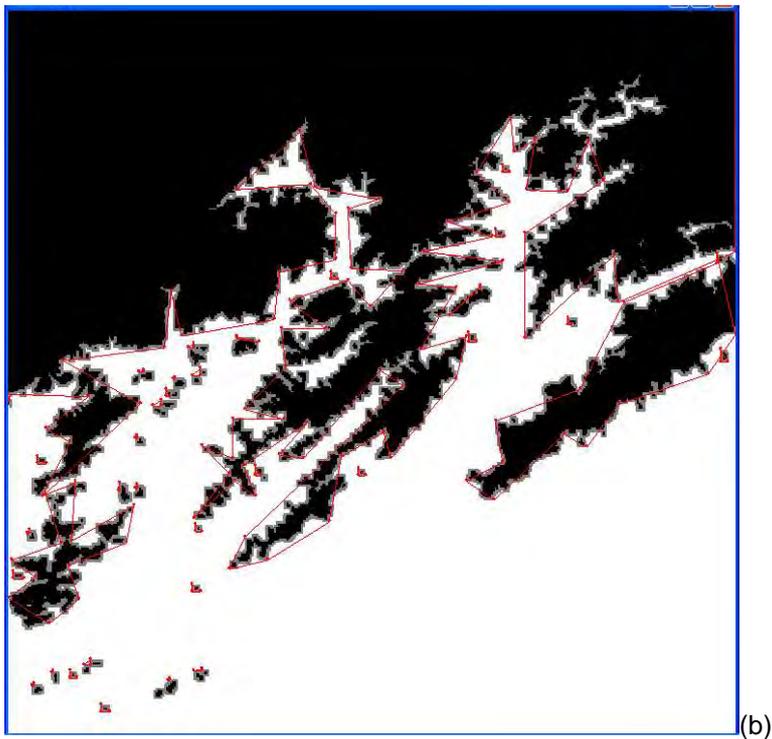
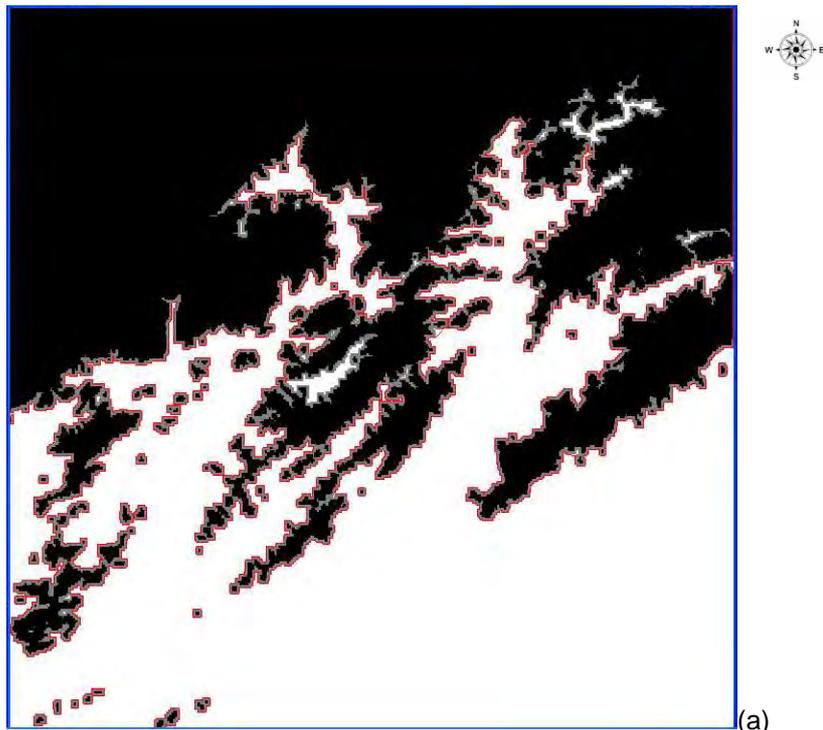


Figura 4.10 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 1000$ m.

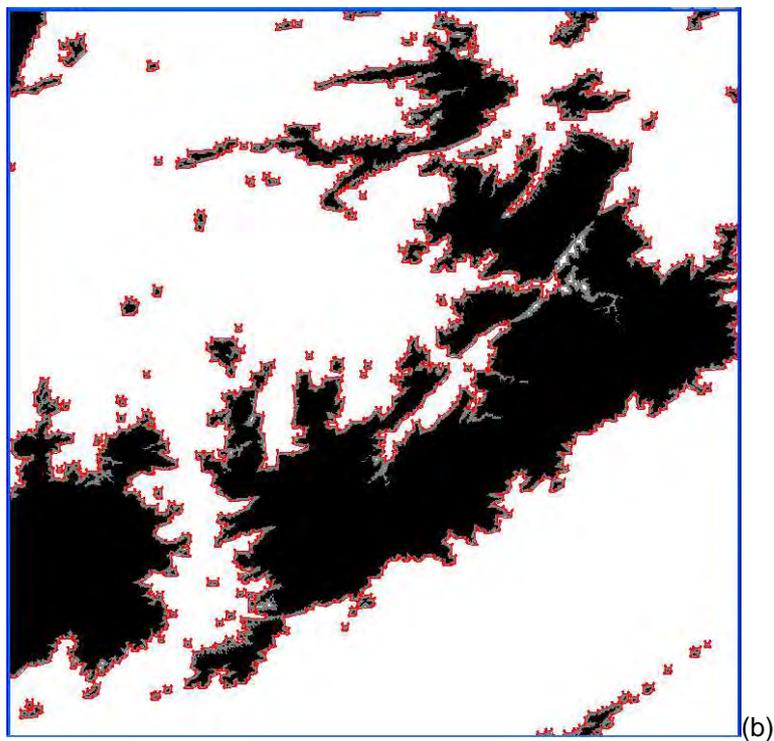
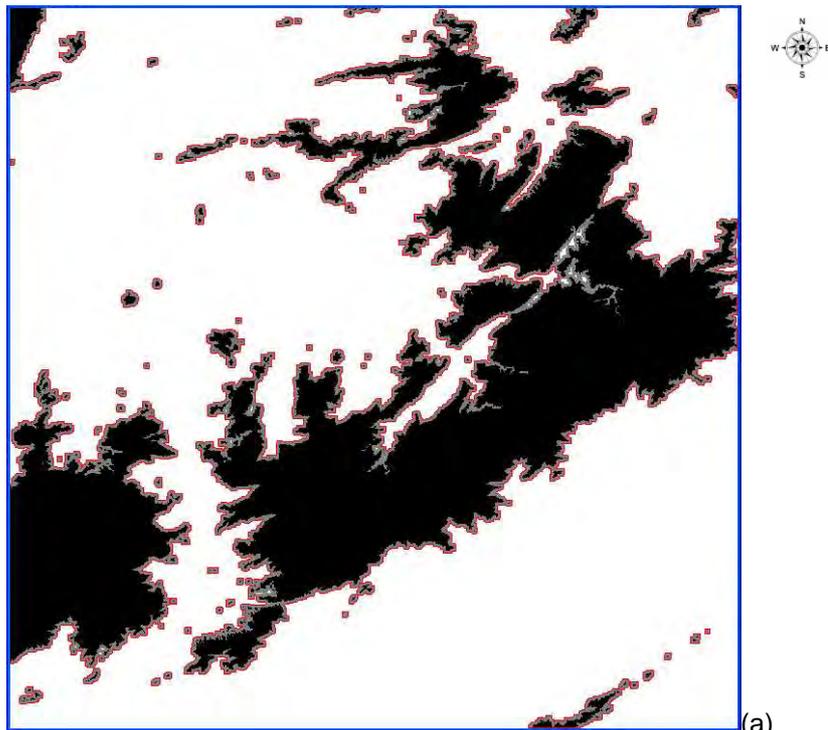


Figura 4.11 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 200$ m.

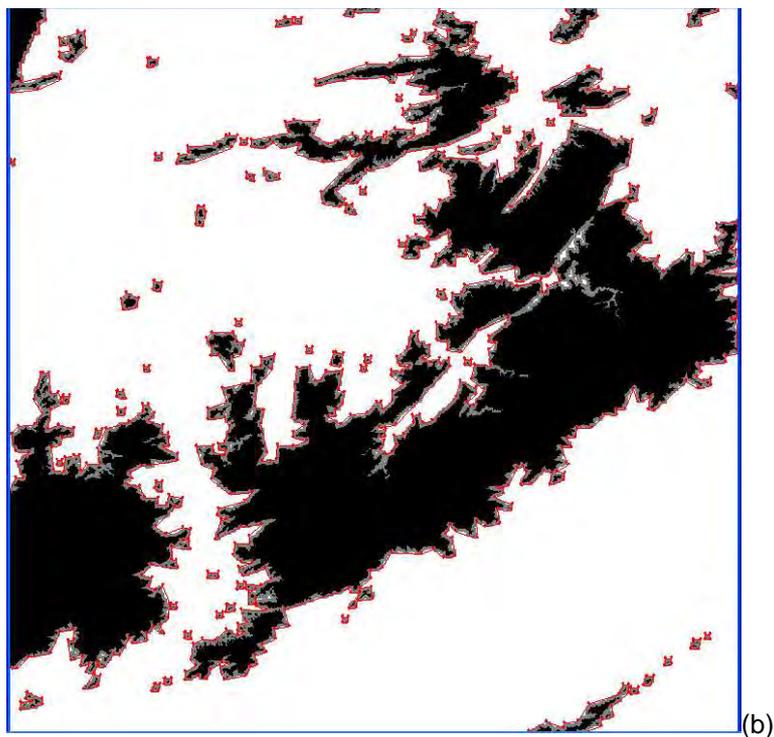
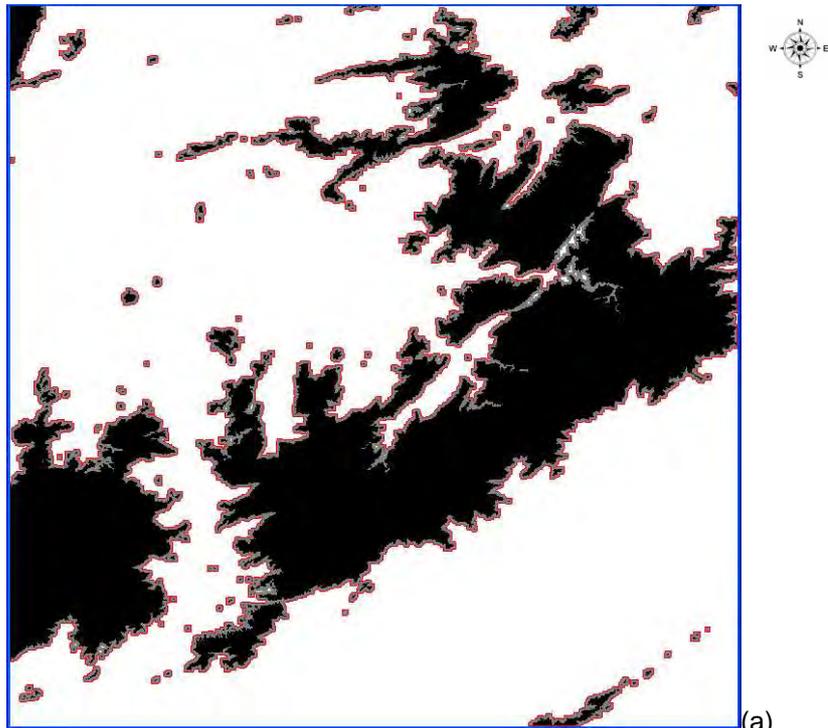


Figura 4.12 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 500$ m.

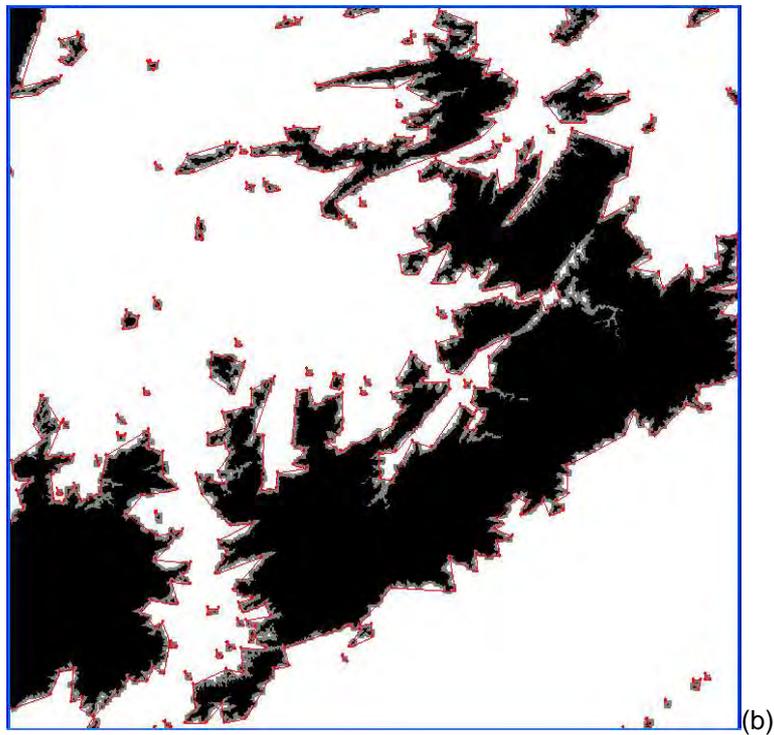
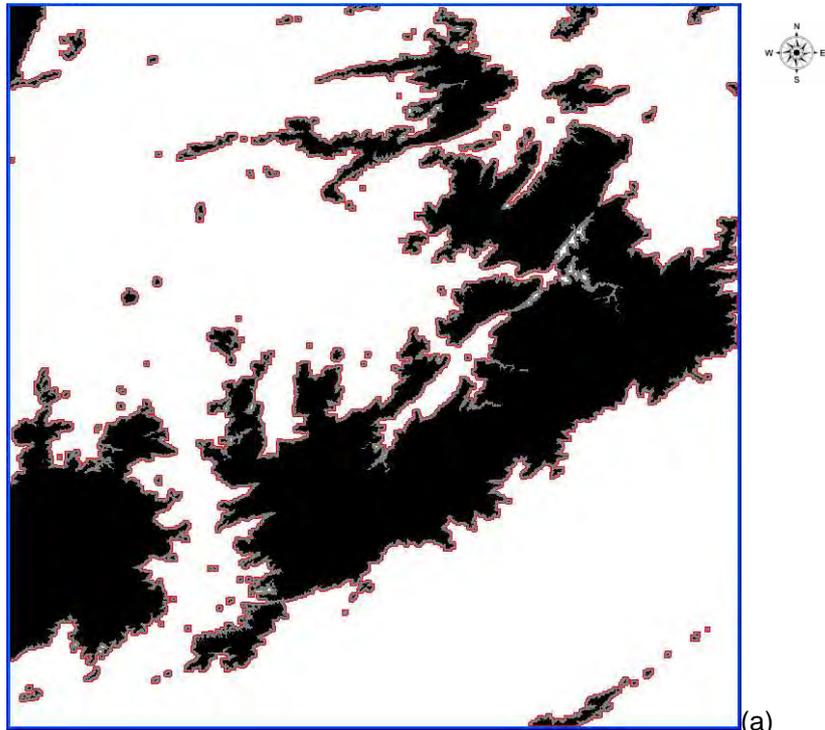


Figura 4.13 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) Simplificação destas representações pela aplicação do algoritmo Ramer-Douglas-Peucker, considerando $d_{limite} = 1000$ m.

A Tabela 4.7 apresenta uma comparação entre os resultados obtidos com a aplicação do algoritmo Visvalingam-Whyatt, considerando o ambiente de navegação apresentado na Figura 4.3b para diferentes valores de a_{limite} . Os respectivos polígonos simplificados pelo algoritmo são apresentados nas Figuras 4.14, 4.15 e 4.16.

Tabela 4.7 – Comparação entre a representação poligonal do primeiro ambiente de navegação e as representações simplificadas pelo algoritmo Visvalingam-Whyatt.

Características	Original	Simplificado		
		$a_{limite} = 0.05 \text{ km}^2$	$a_{limite} = 0.5 \text{ km}^2$	$a_{limite} = 5 \text{ km}^2$
Número total de vértices	2668	898	251	125
Redução	-	66.35%	90.60%	95.32%

A Tabela 4.8 apresenta uma comparação entre os resultados obtidos com a aplicação do algoritmo Visvalingam-Whyatt, considerando o ambiente de navegação apresentado na Figura 4.4b para diferentes valores de a_{limite} . Os respectivos polígonos simplificados pelo algoritmo são apresentados nas Figuras 4.17, 4.18 e 4.19.

Tabela 4.8 – Comparação entre a representação poligonal do segundo ambiente de navegação e as representações simplificadas pelo algoritmo Visvalingam-Whyatt.

Características	Original	Simplificado		
		$a_{limite} = 0.05 \text{ km}^2$	$a_{limite} = 0.5 \text{ km}^2$	$a_{limite} = 5 \text{ km}^2$
Número total de vértices	7649	2702	861	303
Redução	-	64.68%	88.75%	96.04%

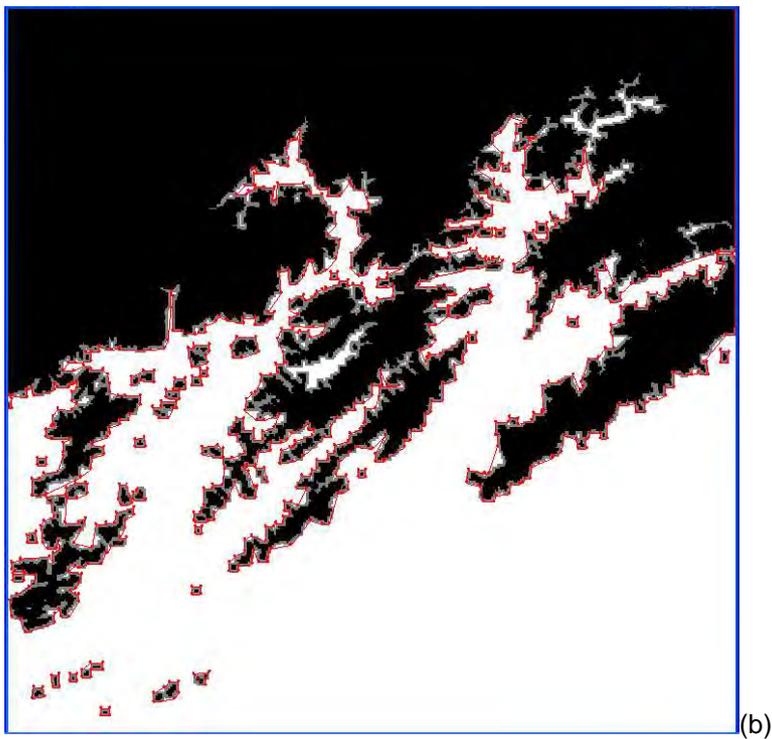
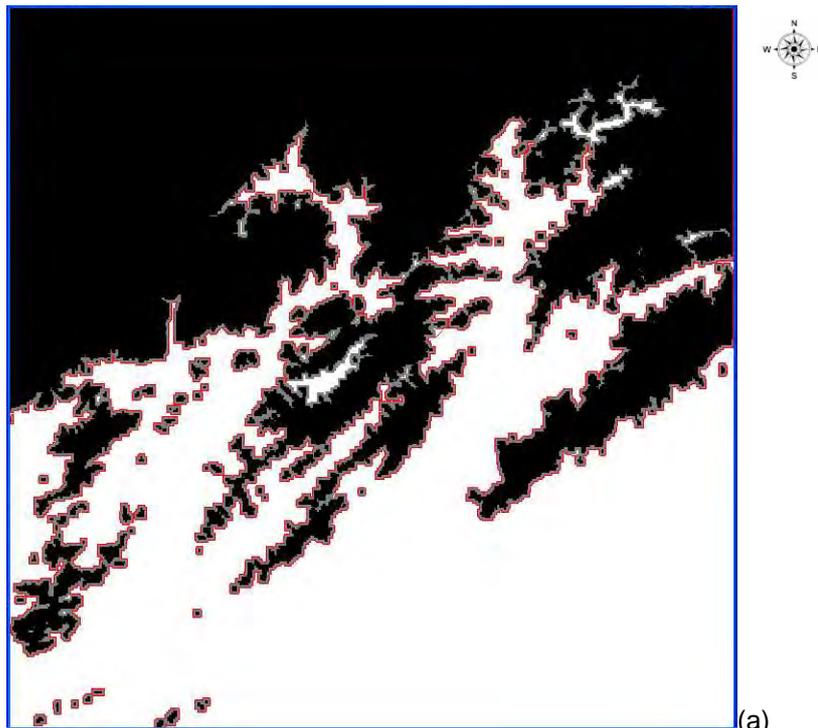


Figura 4.14 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 0.05 \text{ km}^2$.

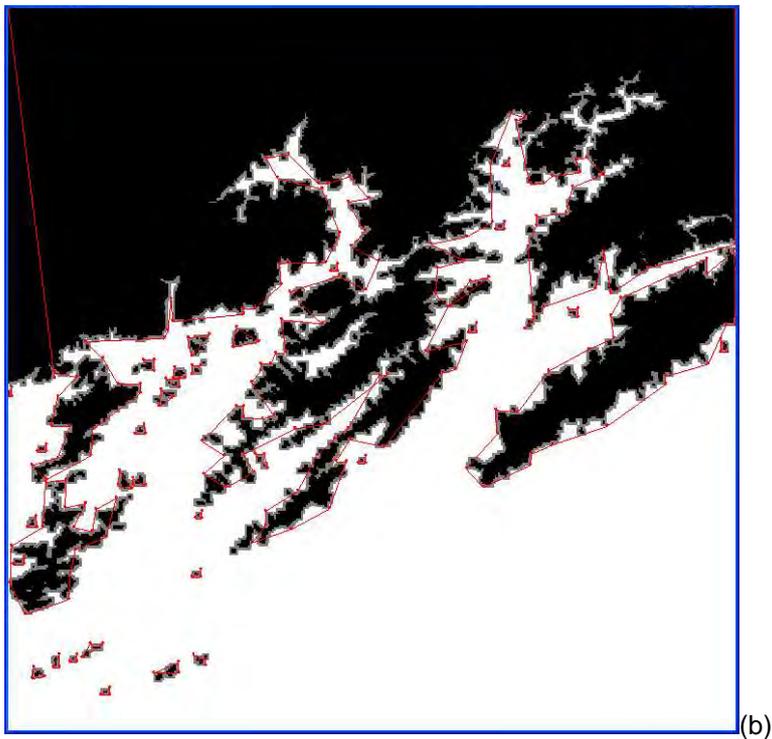
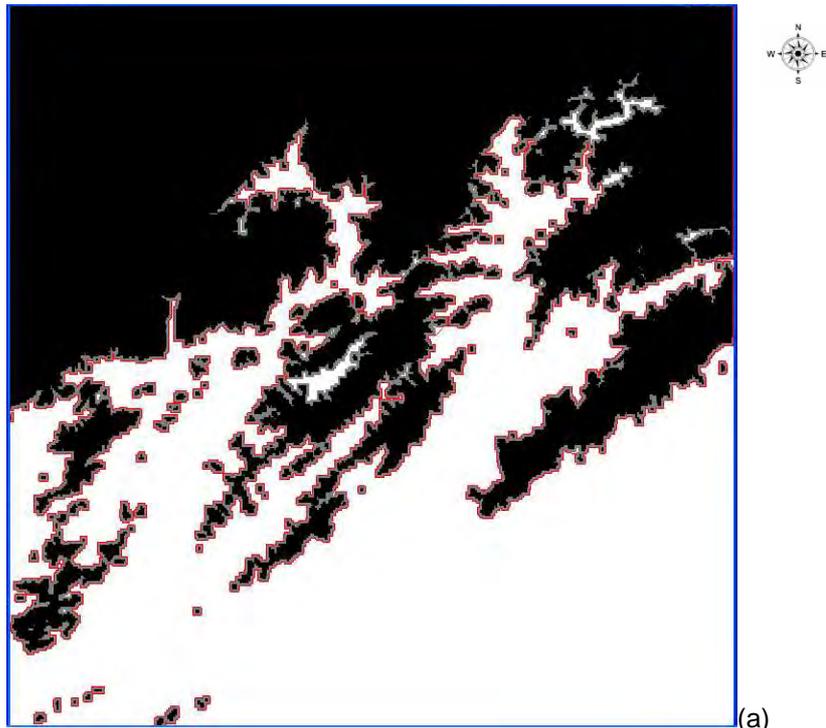


Figura 4.15 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 0.5 \text{ km}^2$.

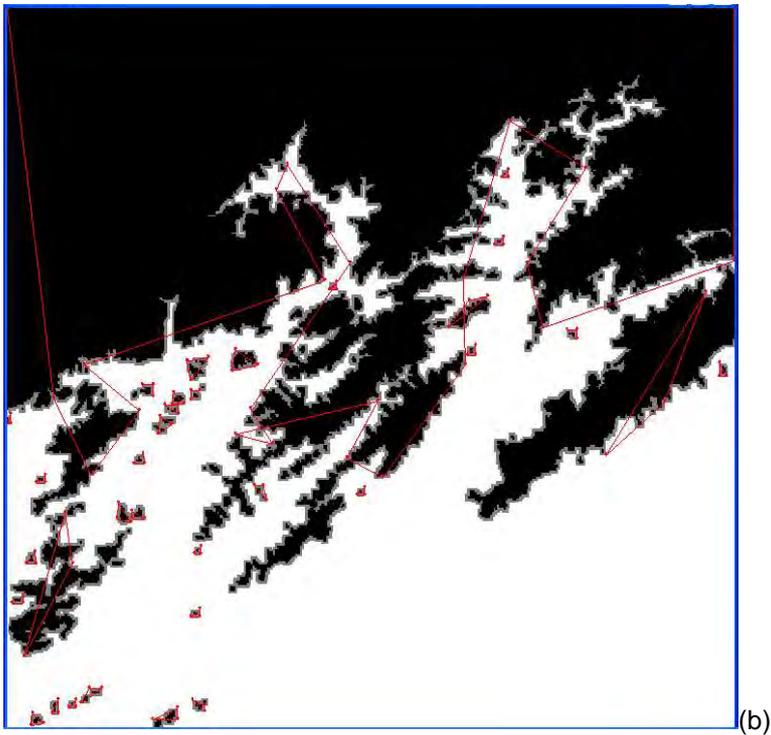
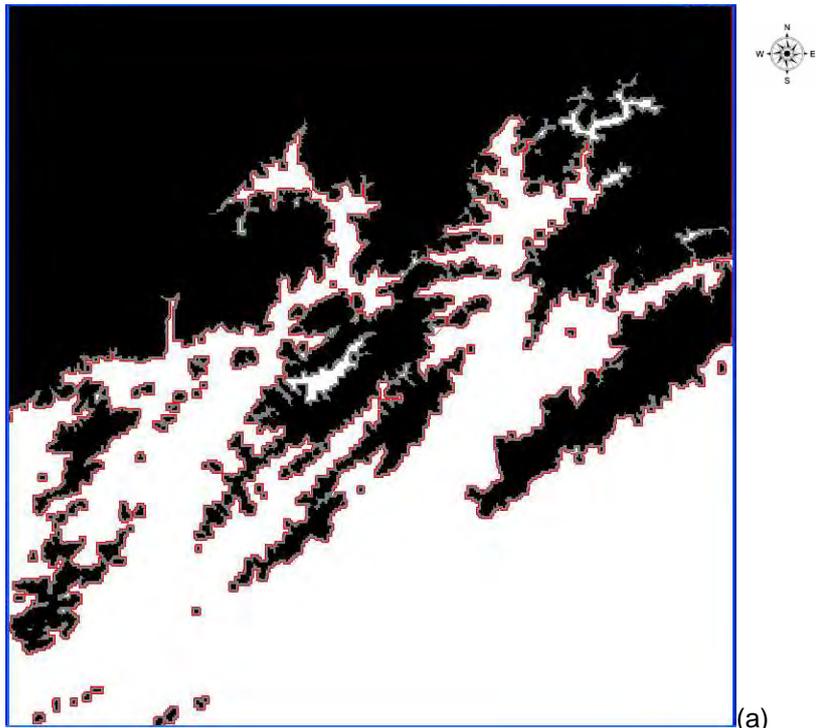


Figura 4.16 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 5 \text{ km}^2$.

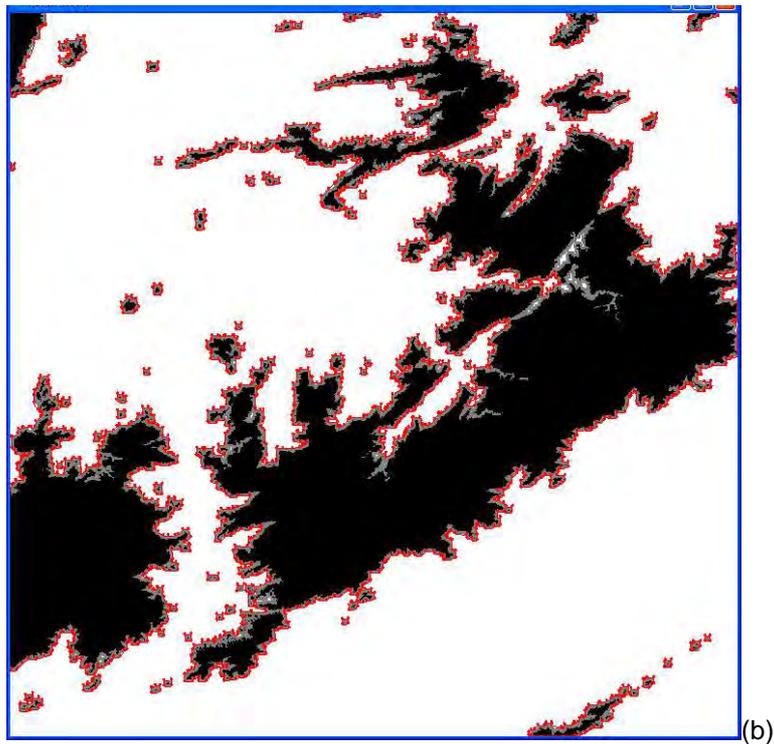
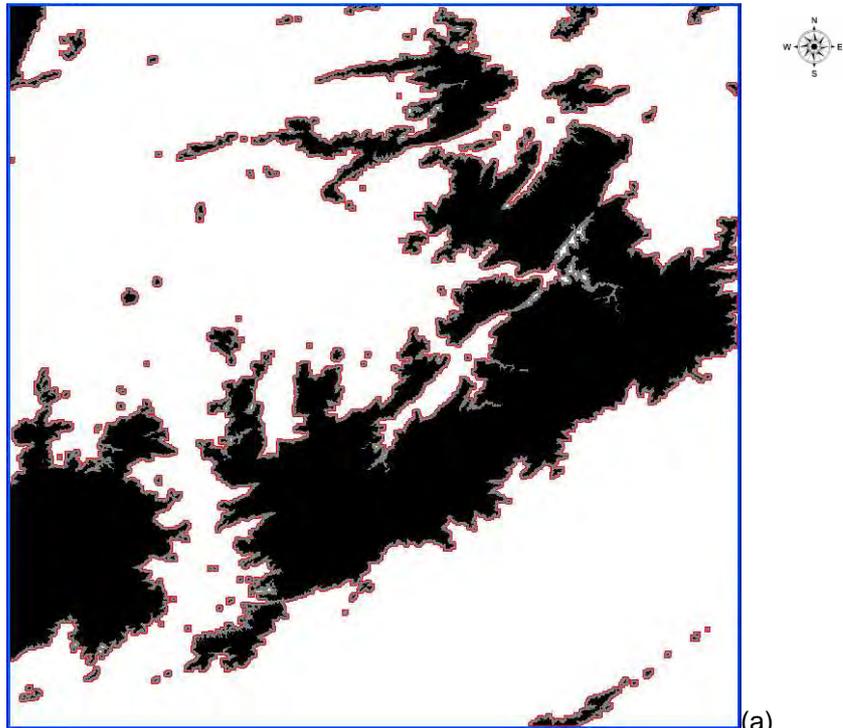


Figura 4.17 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 0.05 \text{ km}^2$.

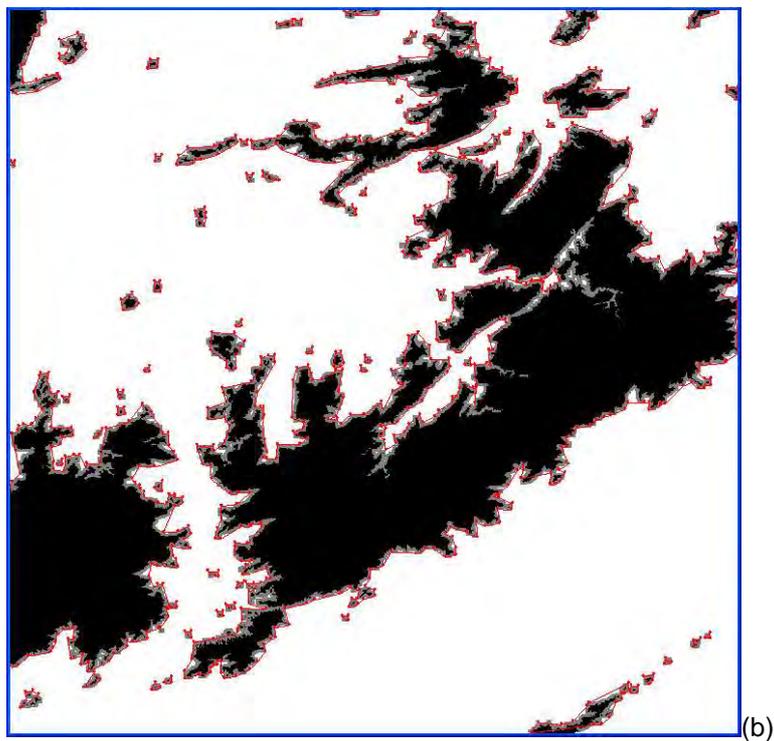
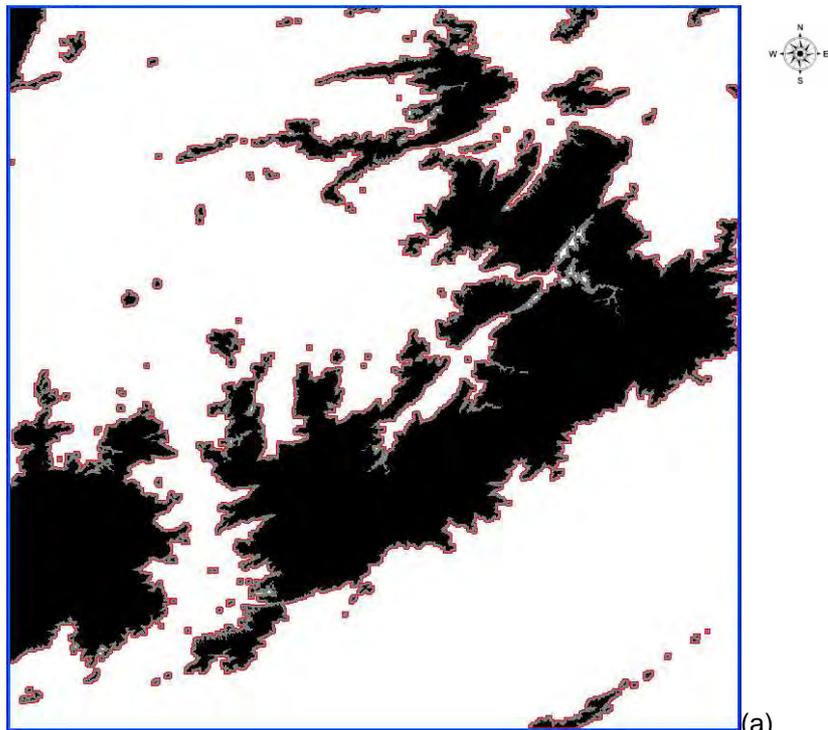


Figura 4.18 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 0.5 \text{ km}^2$.

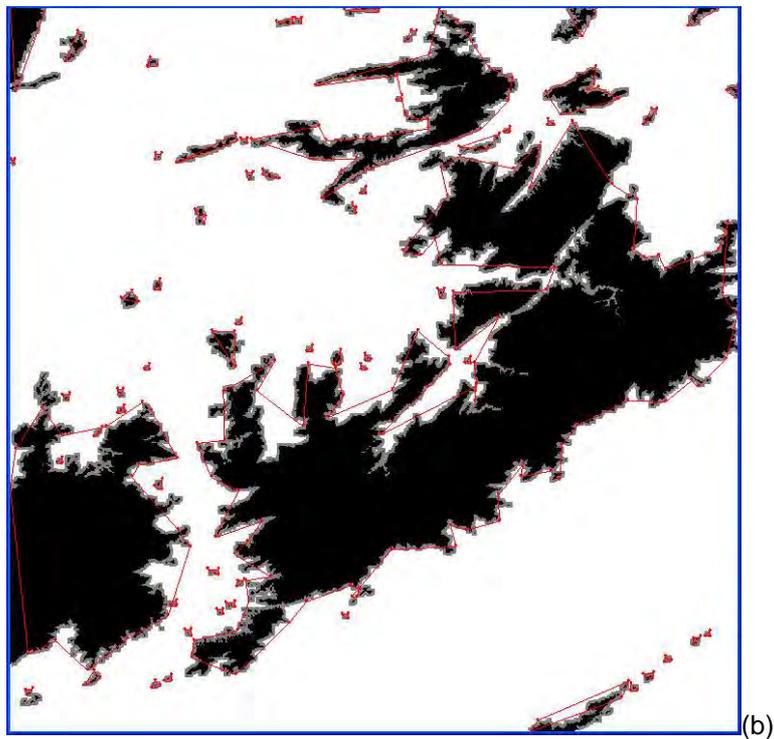
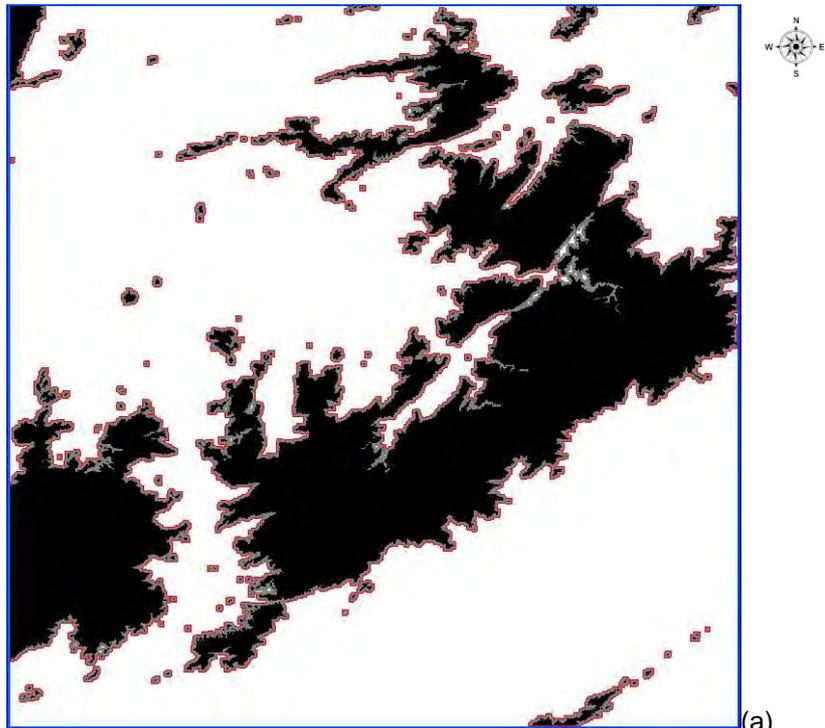


Figura 4.19 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do Visvalingam-Whyatt, considerando $a_{limite} = 5 \text{ km}^2$.

Analisando os resultados obtidos com a aplicação do algoritmo Ramer-Douglas-Peucker e com a aplicação do algoritmo Visvalingam-Whyatt à simplificação de representações poligonais de obstáculos de ambientes de navegação, verifica-se a ocorrência do problema de ocultação de obstáculos em todos os resultados.

Portanto, os problemas identificados com a aplicação dos métodos para geração de envoltórias convexas e dos métodos para simplificação de polígonos motivaram a elaboração de um novo algoritmo para simplificação de polígonos, visando à solução de tais problemas. Este algoritmo é apresentado na próxima subseção.

4.1.3. Algoritmo para a simplificação de polígonos baseado na área de invisibilidade entre vértices intermediários convexas

O algoritmo proposto neste trabalho utiliza as áreas de invisibilidade entre vértices intermediários convexas como métrica para a eliminação dos mesmos. Deste modo, é possível controlar o total de área de invisibilidade gerada pela eliminação de vértices intermediários convexas, o que soluciona o primeiro problema identificado na Subseção 1.1.1. Considerando uma lista original de vértices V_o e uma lista de vértices a serem eliminados V_e , ambos numerados no sentido horário, o algoritmo é um processo iterativo que consiste em duas principais etapas: calcular a área de invisibilidade gerada por dois vértices intermediários convexas v_{ei} e v_{ej} , e eliminar os vértices entre v_{ei} e v_{ej} , se a área de invisibilidade calculada a_i somada à área total de invisibilidade a_{ti} for inferior ou igual a uma área de invisibilidade limitante a_{il} . A área de invisibilidade é calculada considerando a lista de vértices V_o , isto é, os vértices entre os vértices v_{oi} e v_{oj} equivalentes a v_{ei} e v_{ej} . A utilização de V_o garante a associação da área com a representação original dos obstáculos, mesmo após as eliminações de vértices. Portanto, as regiões de invisibilidade geradas por v_{ei} e v_{ej} são as respectivas intersecções entre as regiões de invisibilidade geradas por v_{ei} e as regiões de invisibilidade geradas por v_{ej} , considerando os vértices

v_{ok} , tal que $i < k < j$. Um exemplo de regiões de invisibilidade geradas entre dois vértices intermediários convexos v_{ei} e v_{ej} é apresentado na Figura 4.20. Neste exemplo, os segmentos de reta pretos e tracejados indicam o polígono simplificado resultante da eliminação de vértices das iterações anteriores. Os demais segmentos de reta indicam o polígono original. Os polígonos com preenchimento quadriculado e pontilhado indicam as áreas de invisibilidade geradas por v_{ei} . Os polígonos com preenchimento cinza indicam as áreas de invisibilidade geradas por v_{ej} . Portanto, as regiões quadriculadas na cor cinza representam as regiões de invisibilidade geradas pelos vértices v_{ei} e v_{ej} .

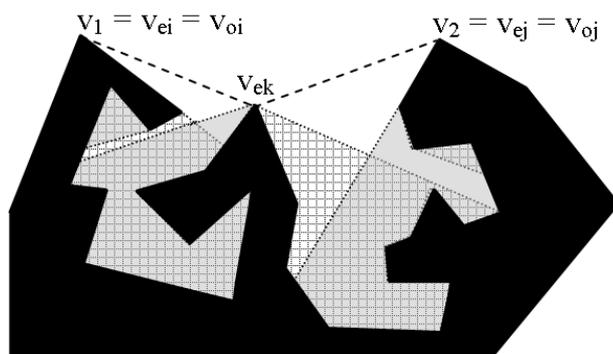


Figura 4.20 – Exemplo de regiões de invisibilidade geradas por dois vértices intermediários convexos v_{ei} e v_{ej} .

O algoritmo proposto é apresentado na Tabela 4.9. A linha 8 deste algoritmo soluciona o problema de ocultação de obstáculos através da determinação do vértice intermediário convexo v_2 baseado na lista atualizada de vértices V_e . Deste modo, o algoritmo garante que a sequência $v_1v_kv_2$ está no sentido anti-horário, para todo vértice v_{ek} entre v_1 e v_2 , considerando os vértices do polígono numerados no sentido horário. Analisando o exemplo da Figura 4.20, pode-se observar que todo vértice v_{ek} forma uma sequência $v_1v_kv_2$ no sentido anti-horário, considerando $v_1 = v_{ei}$ e $v_2 = v_{ej}$. Um vértice v_{ek} intermediário convexo em determinada iteração será eliminado em uma iteração posterior, caso este vértice torne-se côncavo e a restrição de área de invisibilidade limitante não seja violada.

A área de invisibilidade formada por dois vértices $v_1 = v_{ei}$ e $v_2 = v_{ej}$ não pode ser calculada de maneira eficiente pelo algoritmo, se v_2 não é visível a v_1 , isto é, se algum segmento de reta ou lado $\overline{v_{ep}v_{ep+1}}$ intercepta o segmento $\overline{v_1v_2}$, para qualquer $j < p < i$. A Figura 4.21 apresenta um exemplo deste problema. Na linha 9 do algoritmo, é verificada a visibilidade entre v_1 e v_2 com o objetivo de solucionar o problema mencionado, evitando o cálculo errôneo da área de invisibilidade entre estes dois vértices.

Tabela 4.9 – Algoritmo para simplificação de polígonos baseado em áreas de invisibilidade entre vértices intermediários convexos.

Índice	Algoritmo para Simplificação de Polígonos Baseado em Áreas de Invisibilidade
1	$V_e \leftarrow V_o$
2	$eliminação \leftarrow 1$
3	$a_{ii} \leftarrow 0$
4	$v_1 \leftarrow primeiro(V_e)$
5	enquanto $(a_{ii} < a_{ii})$ e $((eliminação = 1)$ ou $((v_1=v_2)$ e $(v_2 \neq primeiro(V_e))))$ faça
6	se $v_1 = primeiro(V_e)$ faça
7	$eliminação \leftarrow 0$
8	$v_2 \leftarrow$ primeiro vértice intermediário convexo em V_e depois de v_1
9	$visibilidade \leftarrow$ visível(v_1, v_2)
10	se $visibilidade = 0$ faça
11	se $v_1 \neq ultimo(V_e)$ faça
12	$v_1 \leftarrow próximo(v_1)$
13	senão faça
14	$v_1 \leftarrow v_{e1}$
15	senão faça
16	se $v_2 \neq próximo(v_1)$
17	$a_i \leftarrow$ área de invisibilidade entre os equivalentes a v_1 e v_2 em V_o
18	se $(a_i + a_{ii}) \leq a_{ii}$ faça
19	$a_{ii} \leftarrow a_{ii} + a_i$
20	$eliminação \leftarrow 1$
21	eliminar os vértices entre v_1 e v_2 em V_e
22	se $v_2 \neq ultimo(V_e)$ faça
23	$v_1 \leftarrow v_2$
24	senão faça
25	$v_1 \leftarrow primeiro(V_e)$
26	senão faça
27	$eliminação \leftarrow 0$
28	senão faça
29	se $v_2 \neq ultimo(V_e)$ faça
30	$v_1 \leftarrow v_2$
31	senão faça
32	$v_1 \leftarrow primeiro(V_e)$
33	$v_2 \leftarrow v_1$

Analisando o algoritmo, foi verificado que, no pior caso, a complexidade computacional do algoritmo é $O(n_v^5)$, em que n_v é o número de vértices do

polígono. Esta complexidade foi obtida, através de uma análise da execução do bloco compreendido entre as linhas 5 e 33, que é executado $O(n_v^2)$ vezes.

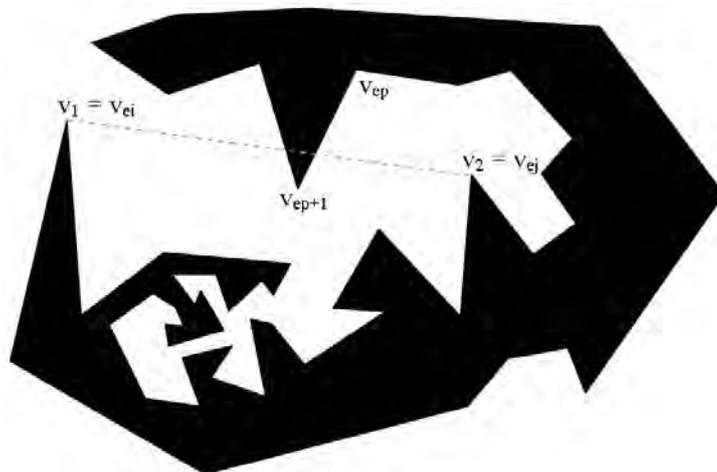


Figura 4.21 – Exemplo do problema de invisibilidade entre os vértices v_1 e v_2 .

O bloco é composto por três principais etapas: a determinação do vértice intermediário convexo v_2 , cujo algoritmo é apresentado na Tabela 4.10; a verificação da visibilidade entre v_1 e v_2 , cujo algoritmo é apresentado na Tabela 4.11; e o cálculo da área de invisibilidade gerada entre os vértices v_1 e v_2 , que é apresentado na Tabela 4.12. As complexidades de tempo da primeira e da segunda etapa são iguais a $O(n_v)$.

Tabela 4.10 – Algoritmo para determinação do vértice intermediário convexo v_2 .

Índice	Algoritmo para Determinação do Vértice Intermediário convexo v_2
1	$convexo \leftarrow 0$
2	enquanto ($convexo = 0$) e ($v_2 \neq \text{último}(V_e)$) faça
3	se a sequência anterior(v_2) v_2 próximo(v_2) está no sentido horário faça
4	$convexo \leftarrow 1$
5	se $convexo = 0$ faça
6	$v_2 \leftarrow \text{próximo}(v_2)$
7	se ($convexo = 0$) e ($v_2 = \text{último}(V_e)$) faça
8	se a sequência anterior(v_2) v_2 primeiro(V_e) estiver no sentido horário faça
9	$convexo \leftarrow 1$
10	se $convexo = 0$ faça
11	$\text{último}(V_e) \leftarrow v_1$
12	$v_1 \leftarrow \text{primeiro}(V_e)$
13	$v_2 \leftarrow \text{próximo}(v_1)$

Tabela 4.11 – Algoritmo para verificação da visibilidade entre os vértices v_1 e v_2 .

Índice	Algoritmo para Verificação da Visibilidade entre os Vértices v_1 e v_2
1	se $v_2 = \text{próximo}(v_1)$ faça
2	visibilidade \leftarrow 1
3	senão faça
4	$v \leftarrow \text{primeiro}(V_e)$
5	visibilidade \leftarrow 1
6	enquanto (visibilidade = 1) e ($v \neq \text{último}(V_e)$) faça
7	se $\text{próximo}(v) \neq v_1$ faça
8	se houver intersecção entre os segmentos de reta $\overline{v_1v_2}$ e $\overline{v\text{próximo}(v)}$ faça
9	visibilidade \leftarrow 0
10	senão faça
11	$v \leftarrow \text{próximo}(v)$
12	senão faça
13	se $v_2 \neq \text{último}(V_e)$ faça
14	$v \leftarrow \text{próximo}(v_2)$
15	senão faça
16	$v \leftarrow \text{primeiro}(V_e)$
17	se (visibilidade = 1) e ($v = \text{último}(V_e)$) faça
18	se houver intersecção entre os segmentos de reta $\overline{v_1v_2}$ e $\overline{v\text{primeiro}(V_e)}$ faça
19	visibilidade \leftarrow 0
20	retornar visibilidade

Tabela 4.12 – Algoritmo para cálculo da área de invisibilidade entre os vértices v_1 e v_2 .

Índice	Algoritmo para Cálculo da Área de Invisibilidade entre os Vértices v_1 e v_2
1	$v \leftarrow \text{próximo}(\text{equivalente}(v_1, V_o))$
2	enquanto ($v \neq \text{equivalente}(v_2, V_o)$) e ($\text{próximo}(v) \neq \text{equivalente}(v_2, V_o)$) faça
3	se v é convexo faça
4	determinar o ponto de intersecção v' entre o prolongamento do segmento de reta $\overline{v_1v}$ e o lado $\overline{v\text{próximo}(v)}$ do polígono
5	montar o polígono p_1 baseado no vértice v e no ponto de intersecção v'
6	armazenar p_1 na lista L_1 de regiões de invisibilidade geradas por v_1
6	$v \leftarrow \text{próximo}(v)$
8	$v \leftarrow \text{anterior}(\text{equivalente}(v_2, V_o))$
9	enquanto ($v \neq \text{equivalente}(v_1, V_o)$) e ($\text{anterior}(v) \neq \text{equivalente}(v_1, V_o)$) faça
10	se v é convexo faça
11	determinar o ponto de intersecção v'' entre o prolongamento do segmento de reta $\overline{v_2v}$ e o lado $\overline{vanterior(v)}$ do polígono
12	montar o polígono p_2 baseado no vértice v e no ponto de intersecção v''
13	armazenar p_2 na lista L_2 de regiões de invisibilidade geradas por v_2
14	$v \leftarrow \text{anterior}(v)$
15	área \leftarrow 0
16	para cada polígono p_1 de L_1 faça
17	para cada polígono p_2 de L_2 faça
18	$p \leftarrow$ intersecção entre o polígono p_1 e o polígono p_2
19	área \leftarrow área + área do polígono p
20	retornar área

Inicialmente, a complexidade computacional da terceira etapa era $O(n_v^4)$, pois a intersecção entre polígonos, apresentada na linha 18 da Tabela 4.12, era obtida através da verificação da intersecção entre cada vértice das regiões de

invisibilidade geradas por v_1 e os vértices das regiões de invisibilidade geradas por v_2 . Porém, analisando as regiões de invisibilidade geradas, verificou-se que a intersecção entre uma região p_1 de invisibilidade gerada por v_1 e uma região p_2 gerada por v_2 ocorre apenas em duas situações: quando o fecho de p_1 , que corresponde ao segmento de reta $\overline{v_1v'}$, intercepta o fecho de p_2 , que corresponde ao segmento de reta $\overline{v''v_2}$; ou quando algum vértice de p_1 for ponto interno de p_2 ou vice-versa. A complexidade da verificação de intersecção é $O(1)$ e a complexidade da segunda situação é $O(n_v)$, o que reduziu a complexidade computacional da terceira etapa para $O(n_v^3)$. No melhor caso, a complexidade do algoritmo é $O(n_v)$ e ocorre quando os vértices do polígono são todos intermediários convexos.

O algoritmo da Tabela 4.9 foi aplicado ao problema de simplificação das representações poligonais dos ambientes de navegação apresentados na Figura 4.3b e na Figura 4.4b. Na aplicação, foi considerada uma área total para todos os polígonos do ambiente de navegação, isto é, a área total de invisibilidade é a soma das áreas de invisibilidade geradas no processo de simplificação de cada polígono do ambiente. Isto permitiu o controle da simplificação baseada na área total de invisibilidade. A Tabela 4.13 e a Tabela 4.14 apresentam comparações entre os resultados obtidos com a aplicação do algoritmo, considerando os ambientes de navegação mencionados. As respectivas representações poligonais simplificadas pelo algoritmo são apresentadas nas Figuras 4.22 a 4.31. Analisando a Tabela 4.13 e a Tabela 4.14, pode-se verificar que a maior simplificação possível da representação do primeiro ambiente de navegação gera uma área total de invisibilidade igual a 55.4 km^2 e a maior simplificação possível do segundo ambiente de navegação gera uma área total de invisibilidade igual a 306.5 km^2 .

Pode-se observar também que, neste caso, a maior simplificação possível não corresponde a uma envoltória convexa, pois, a restrição de visibilidade limita a simplificação.

Tabela 4.13 – Comparação entre a representação poligonal do primeiro ambiente de navegação e as representações simplificadas pelo algoritmo proposto.

Características	Original	Simplificado				
		$a_{ij} = 5 \text{ km}^2$	$a_{ij} = 10 \text{ km}^2$	$a_{ij} = 20 \text{ km}^2$	$a_{ij} = 50 \text{ km}^2$	$a_{ij} = 100 \text{ km}^2$
Número total de vértices	2668	570	459	389	284	264
Redução	-	78.64%	82.80%	85.42%	89.36%	90.11%
Área total (km ²)	1306.83	-				
Área total de invisibilidade (km ²)	-	4.8	10	16.3	46	55.4
Porcentagem em relação à área total	-	0.37%	0.77%	1.25%	3.52%	4.24%
Área total navegável (km ²)	654.8	-				
Porcentagem em relação à área navegável	-	0.74%	1.53%	2.49%	7.03%	8.47%

Tabela 4.14 – Comparação entre a representação poligonal do segundo ambiente de navegação e as representações simplificadas pelo algoritmo proposto.

Características	Original	Simplificado				
		$a_{ij} = 50 \text{ km}^2$	$a_{ij} = 100 \text{ km}^2$	$a_{ij} = 200 \text{ km}^2$	$a_{ij} = 300 \text{ km}^2$	$a_{ij} = 400 \text{ km}^2$
Número total de vértices	7649	1297	841	710	571	567
Redução	-	83.05%	89.01%	90.72%	92.54%	92.59%
Área total (km ²)	11683.45	-				
Área total de invisibilidade (km ²)	-	49.5	93.7	175.3	297.9	306.5
Porcentagem em relação à área total	-	0.43%	0.82%	1.50%	2.55%	2.63%
Área total navegável (km ²)	7739.1	-				
Porcentagem em relação à área navegável	-	0.64%	1.22%	2.27%	3.85%	3.97%

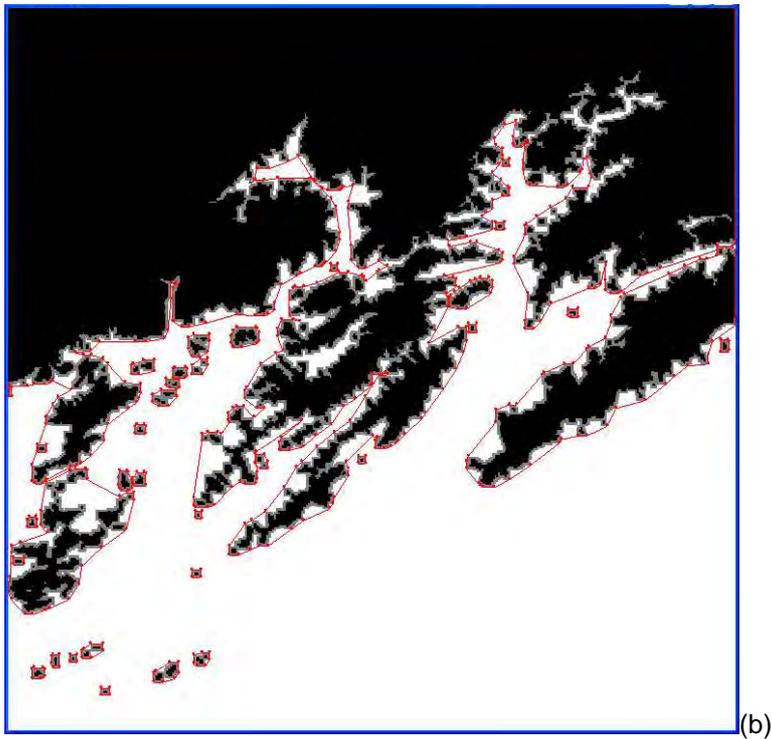
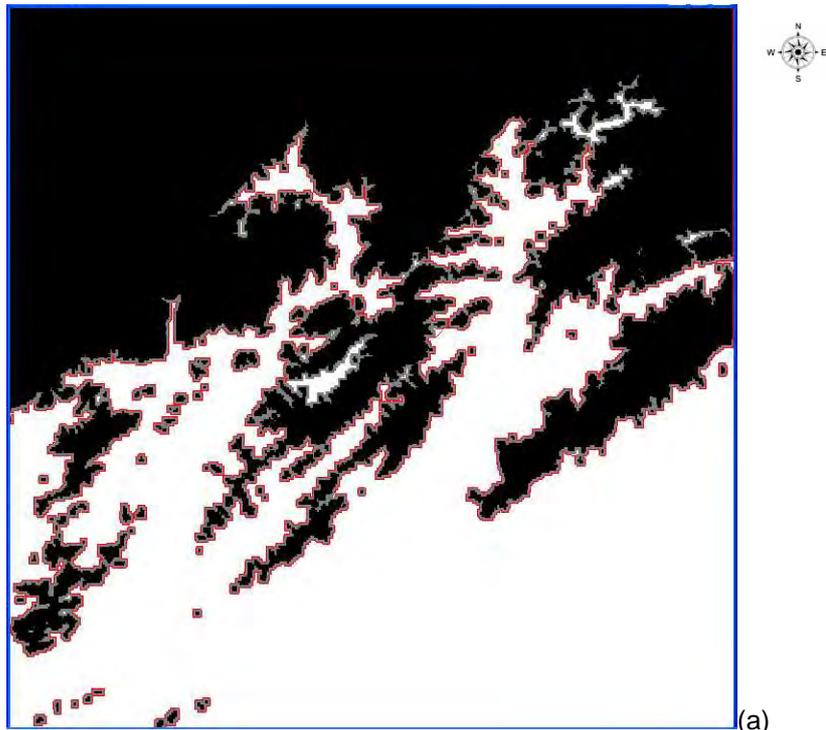


Figura 4.22 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 5 \text{ km}^2$.

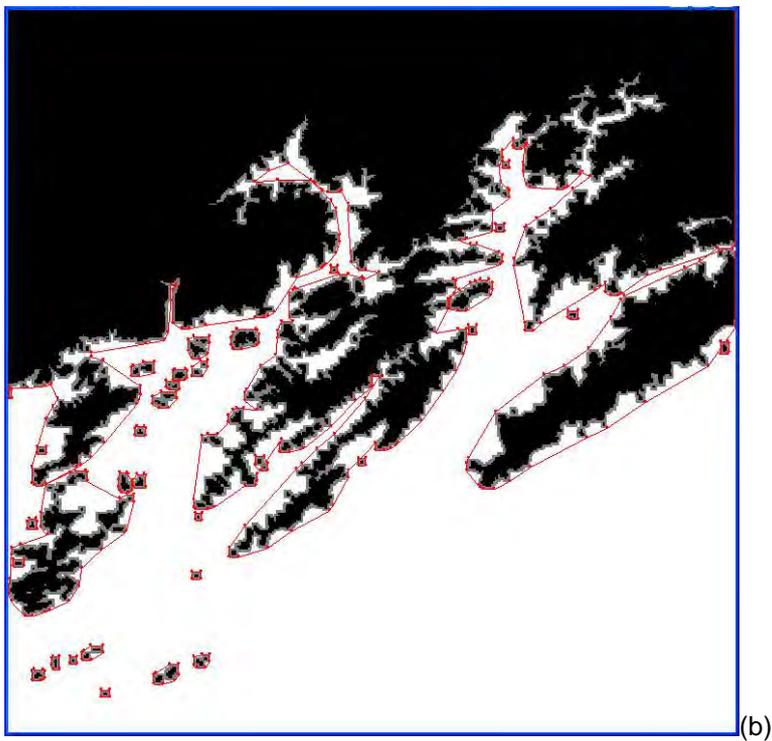
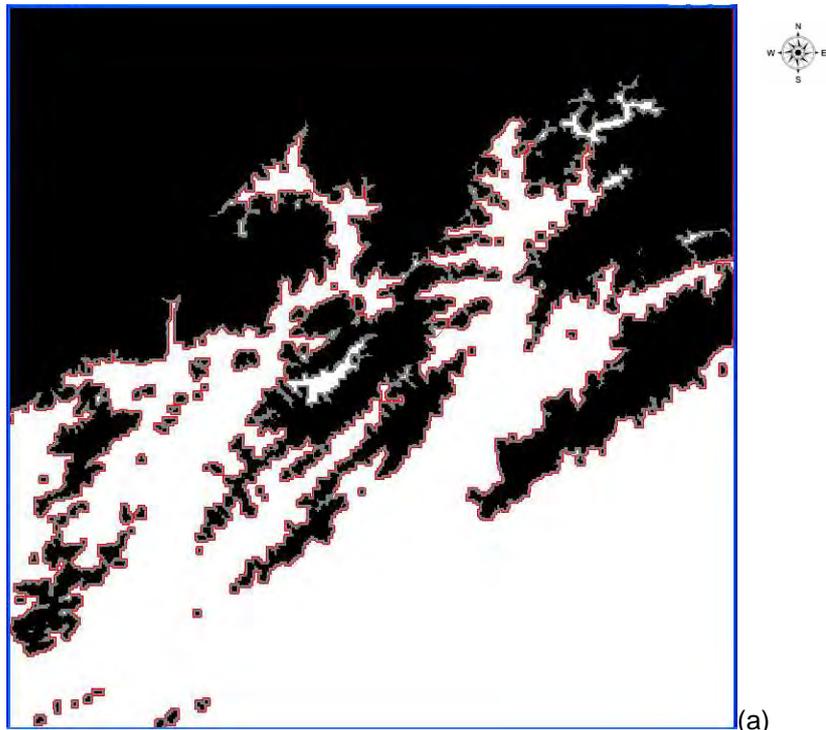


Figura 4.23 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 10 \text{ km}^2$.

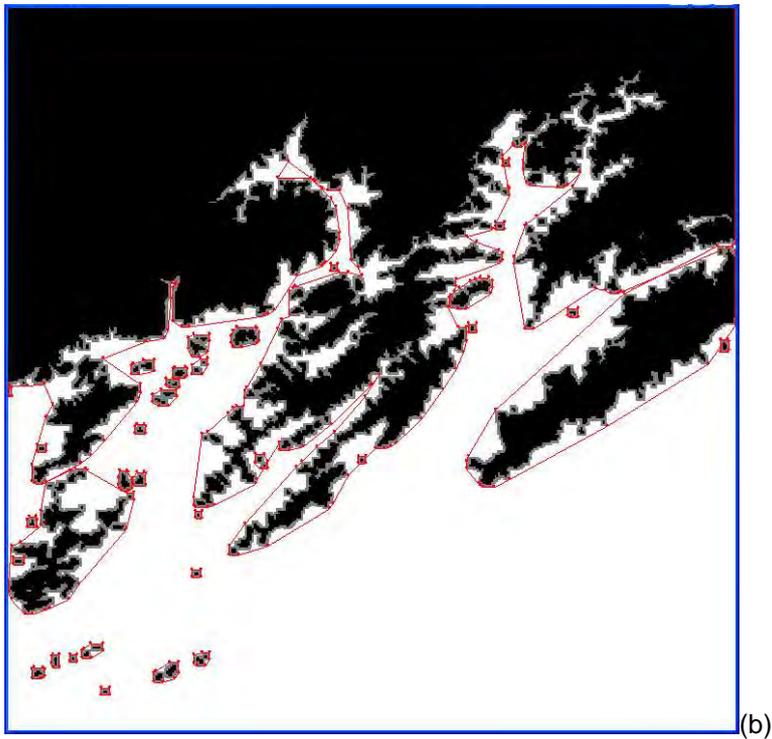
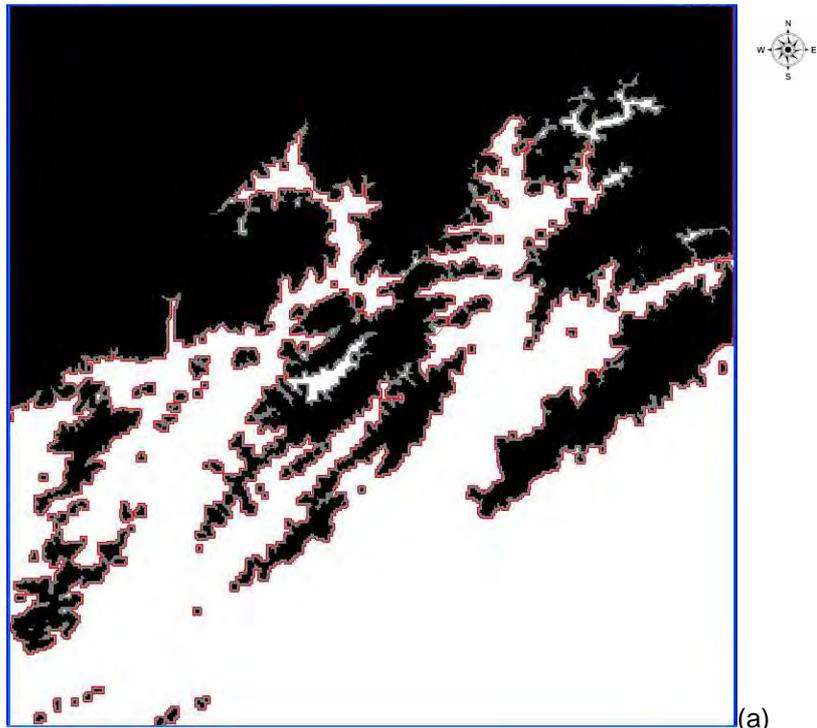


Figura 4.24 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 20 \text{ km}^2$.

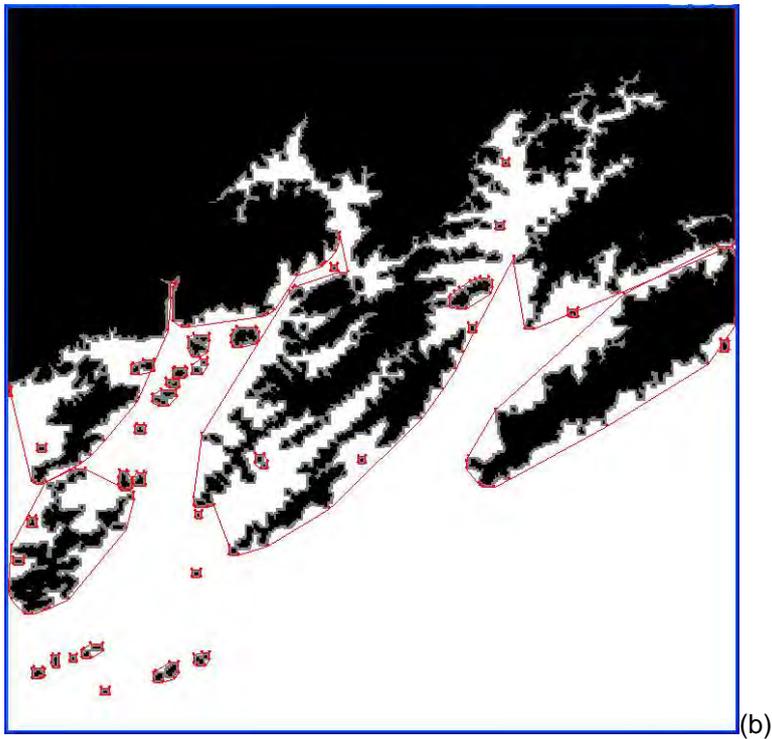
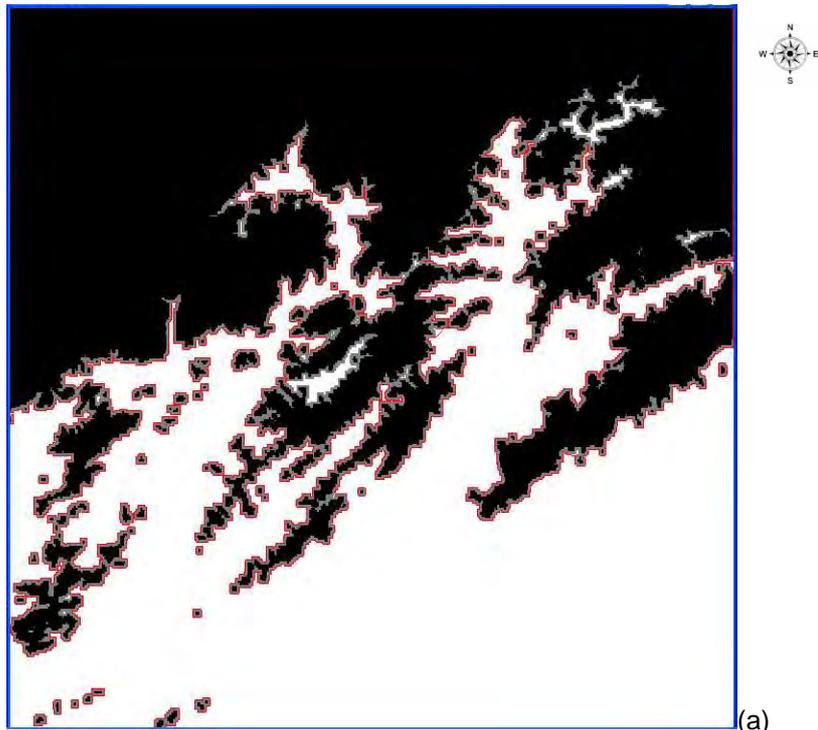


Figura 4.25 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 50 \text{ km}^2$.

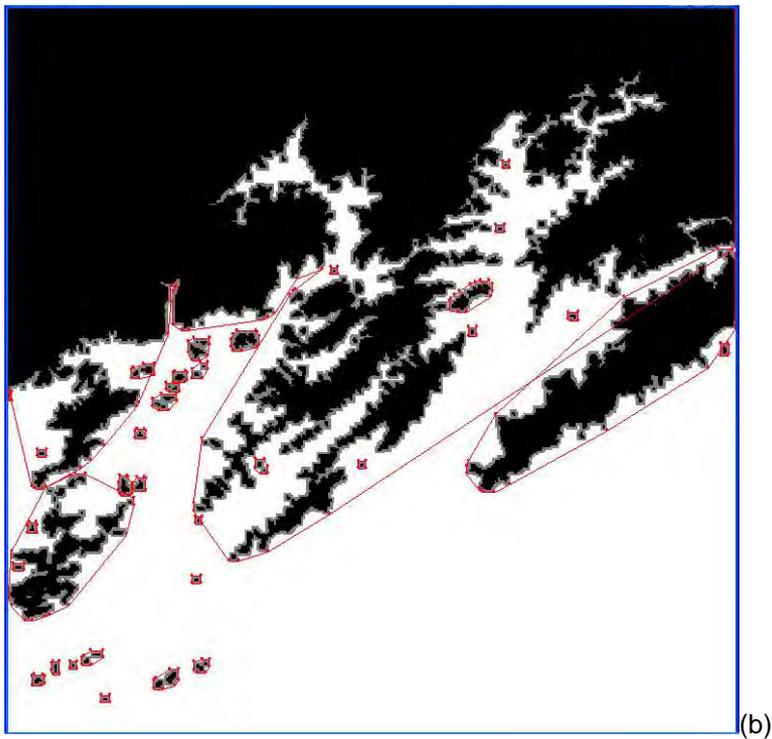
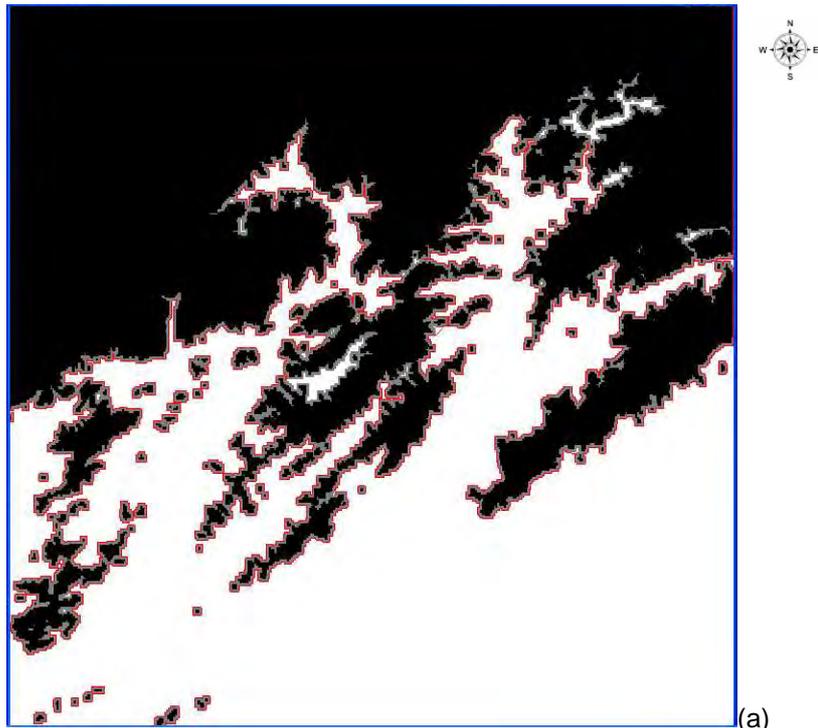


Figura 4.26 – (a) Representações poligonais originais do primeiro ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 100 \text{ km}^2$.

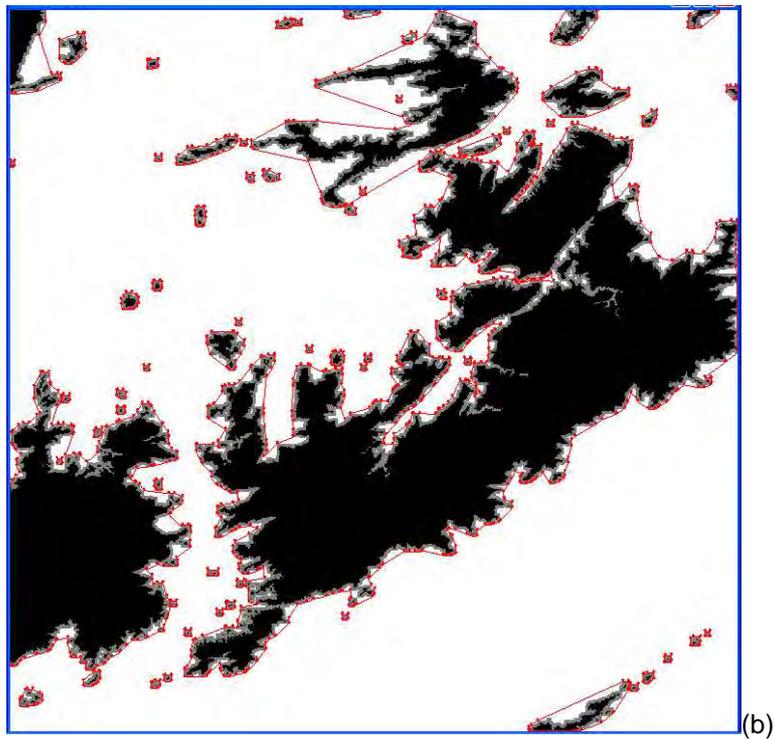
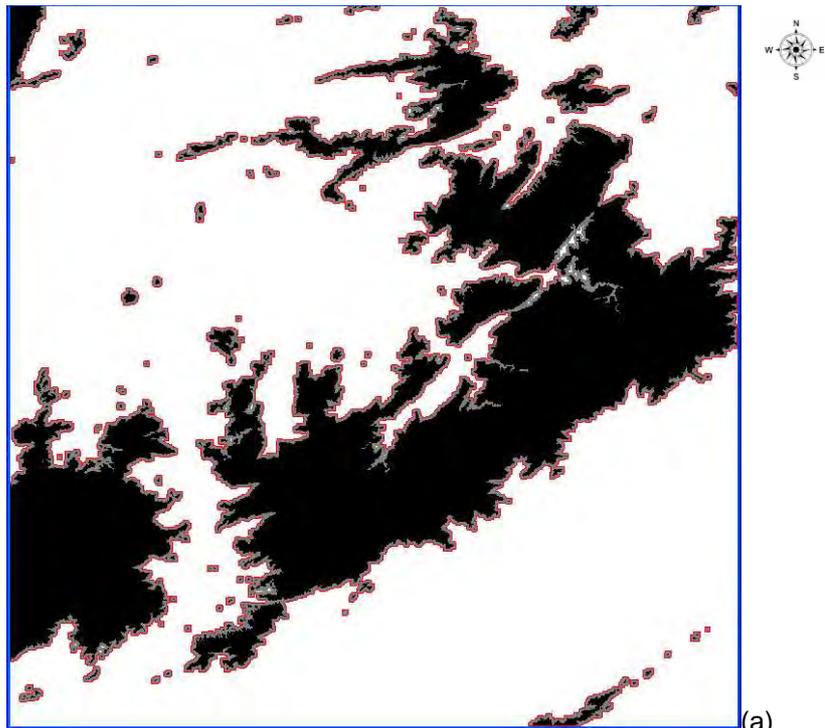


Figura 4.27 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 50 \text{ km}^2$.

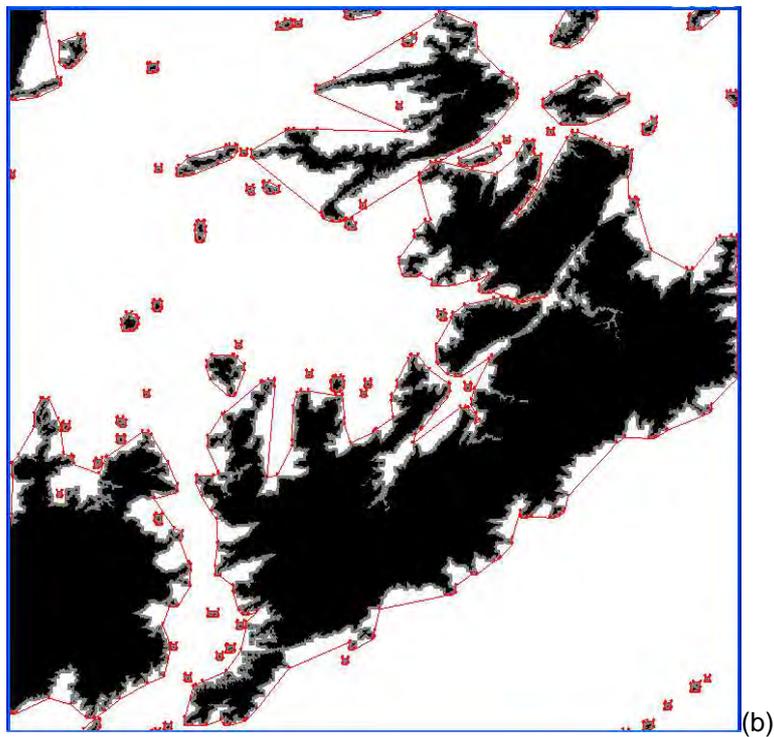
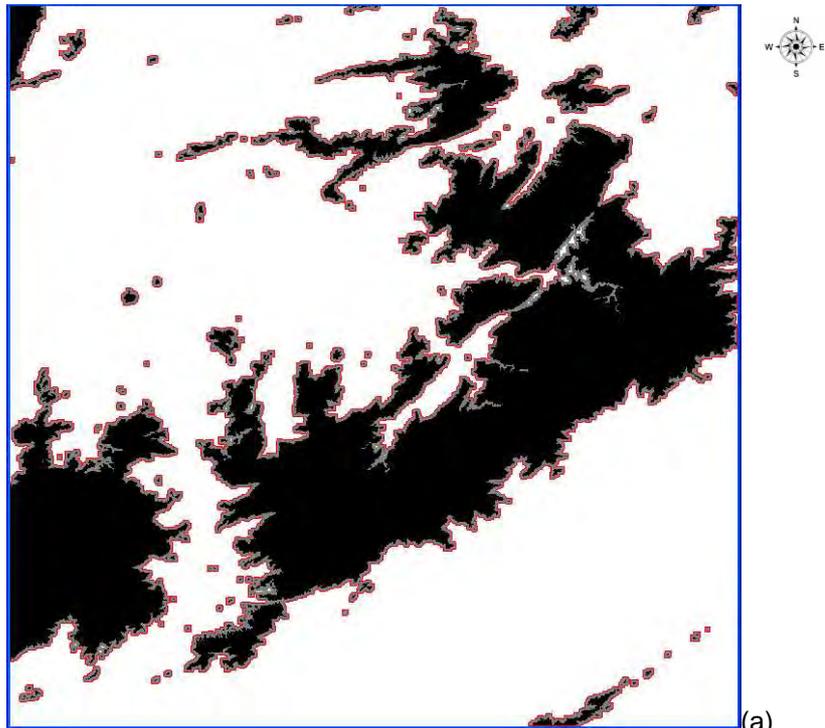


Figura 4.28 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 100 \text{ km}^2$.

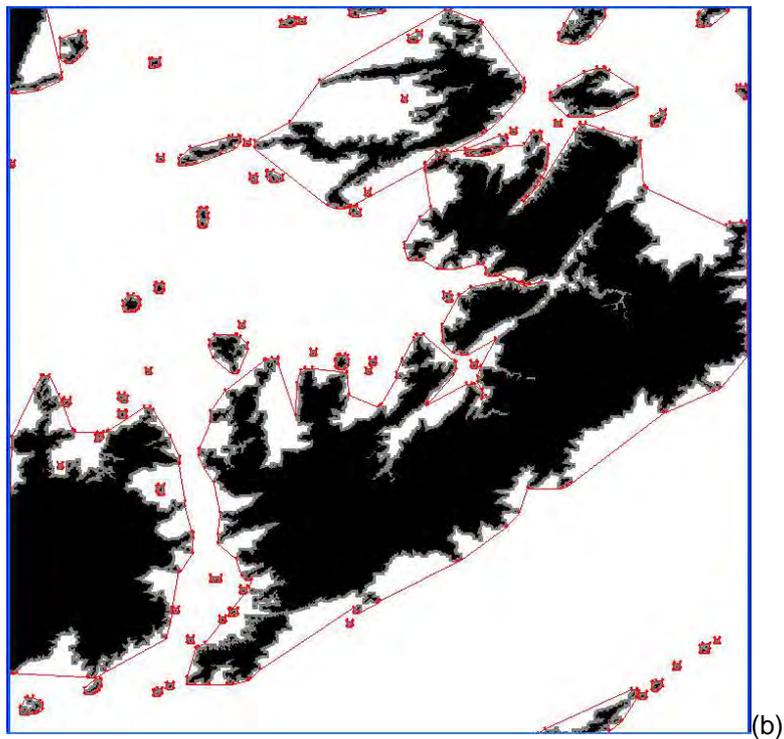
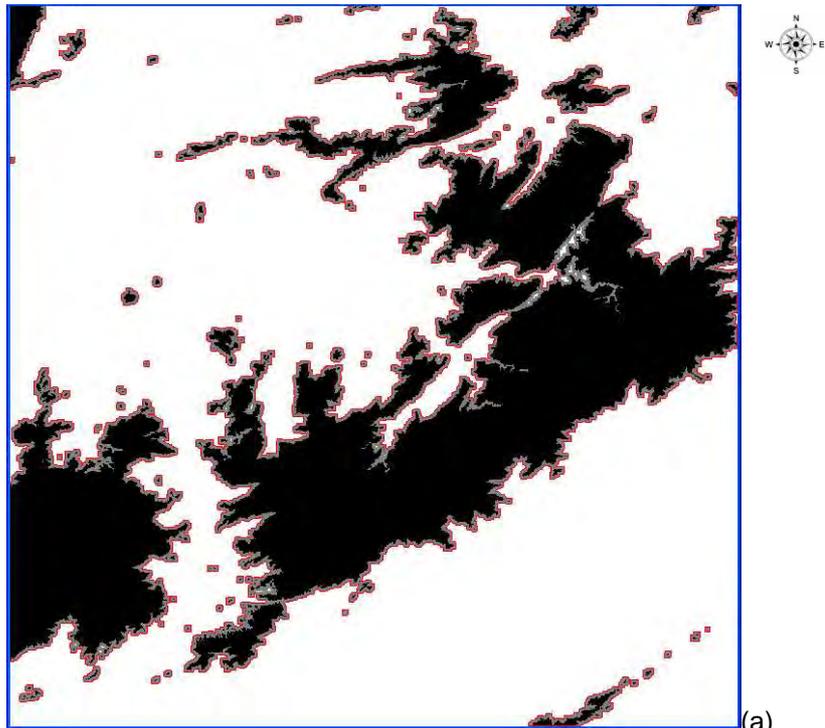


Figura 4.29 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 200 \text{ km}^2$.

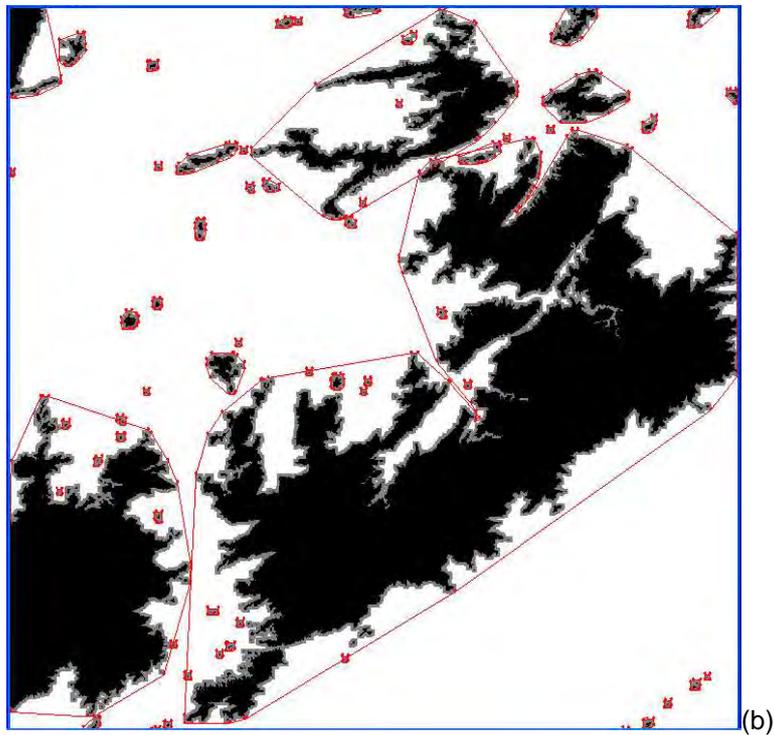
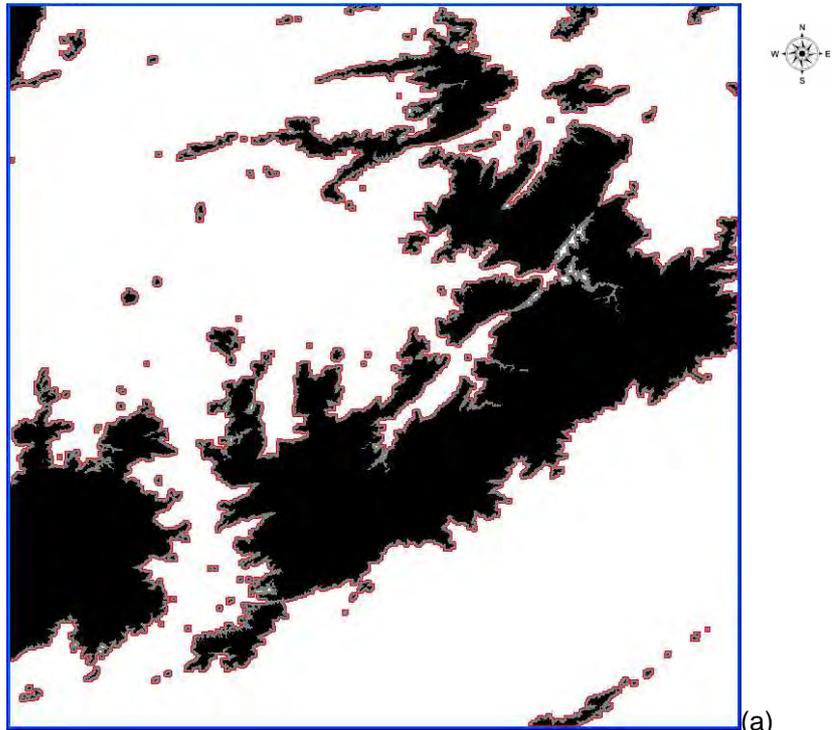


Figura 4.30 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 300 \text{ km}^2$.

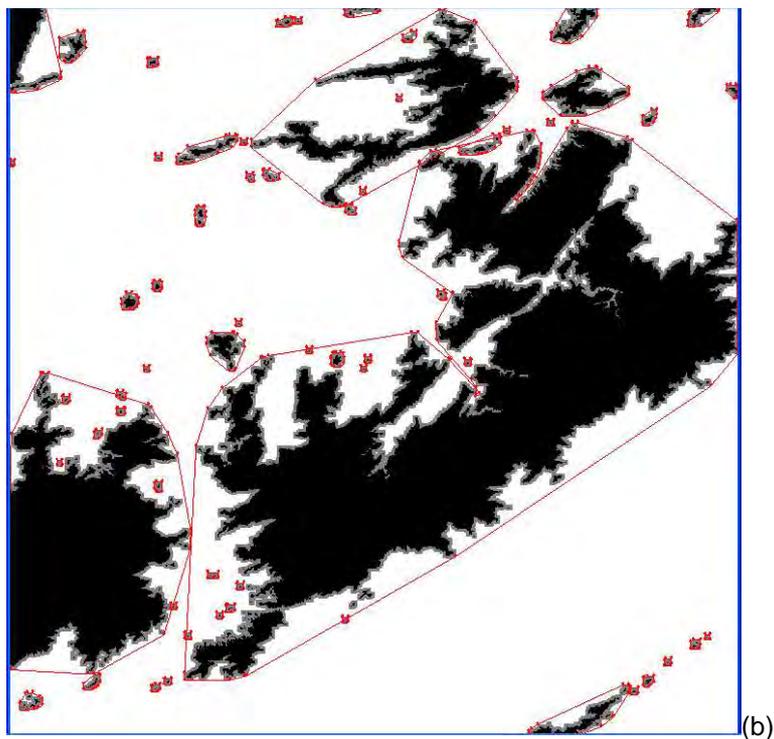
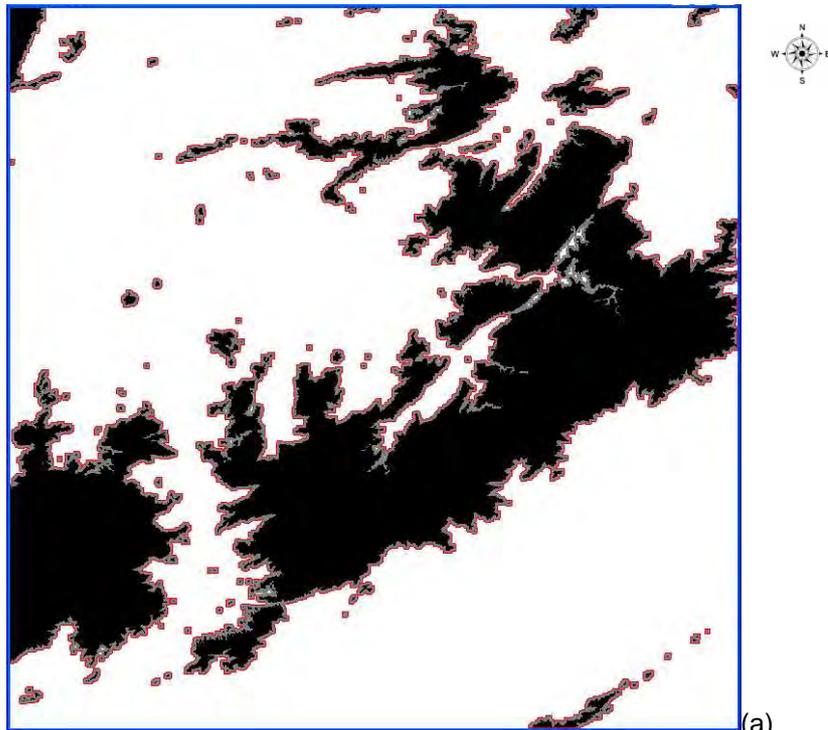


Figura 4.31 – (a) Representações poligonais originais do segundo ambiente de navegação e (b) simplificação destas representações pela aplicação do algoritmo proposto, considerando $a_{ij} = 400 \text{ km}^2$.

Os resultados comprovam a eficiência do algoritmo porque não há ocorrência de ocultação de obstáculos e o controle do processo de simplificação é realizado através do ajuste da área total de invisibilidade, solucionando assim, os dois problemas identificados nos algoritmos para simplificação de polígonos mencionados na revisão apresentada anteriormente.

Analisando os resultados apresentados na Tabela 4.13 e a Tabela 4.14, pode-se observar que o algoritmo proposto permite uma redução significativa do número de vértices dos polígonos com a criação de uma pequena área de invisibilidade. Este fato ocorre porque os vértices intermediários convexos com maior visibilidade em um polígono são preservados no processo de simplificação. Por exemplo, o algoritmo reduz 78.64% dos vértices do primeiro ambiente de navegação, gerando apenas 4.8 km² de área de invisibilidade. Esta área corresponde a 0.37% da área total do primeiro ambiente de navegação e 0.74% da área navegável total do mesmo ambiente. Para o segundo ambiente de navegação, o algoritmo reduz 83.28% dos vértices deste ambiente, gerando apenas 49.5 km² de área de invisibilidade. Esta área de invisibilidade corresponde a 0.43% da área total do segundo ambiente de navegação e 0.64% da área navegável total do mesmo ambiente.

O primeiro efeito observado da redução do número de vértices é a redução significativa do tempo necessário para a construção de grafos de visibilidade baseada nos ambientes simplificados pelo algoritmo proposto. Outro efeito observado é a redução significativa do número de nós e de arestas dos grafos construídos através dos ambientes de navegação simplificados pelo algoritmo proposto. O grafo de visibilidade, construído através da simplificação com criação de 4.8 km² de área de invisibilidade, apresenta uma redução de 61.72% de nós e 69.24% de arestas em relação ao primeiro grafo de visibilidade original, isto é, o grafo construído através da representação poligonal original do primeiro ambiente de navegação. O grafo de visibilidade, construído através da simplificação com criação de 49.5 km² de área de invisibilidade, apresenta uma redução de 67.06% de nós e 80.75% de arestas

em relação ao grafo construído através da representação poligonal original do segundo ambiente de navegação.

A Tabela 4.15 e a Tabela 4.16 apresentam comparações entre características dos grafos construídos através das representações originais dos ambientes de navegação e características dos grafos construídos através das representações simplificadas. A Figura 4.32 e a Figura 4.33 apresentam os grafos de visibilidade construídos para os dois ambientes de navegação avaliados neste trabalho. Os grafos de visibilidade são representados pelos nós e arestas na cor cinza.

Tabela 4.15 – Comparação entre os grafos construídos considerando a representação original do primeiro ambiente de navegação e as representações simplificadas pelo algoritmo proposto.

Características	Original	Simplificado				
		$a_{ij} = 5 \text{ km}^2$	$a_{ij} = 10 \text{ km}^2$	$a_{ij} = 20 \text{ km}^2$	$a_{ij} = 50 \text{ km}^2$	$a_{ij} = 100 \text{ km}^2$
Número total de nós	1489	570	459	389	284	264
Redução	-	61.72%	69.18%	73.88%	80.93%	82.27%
Número total de arestas	65241	20073	15237	12474	7959	7101
Redução	-	69.24%	76.65%	80.88%	87.80%	89.12%
Tempo para construção	121.4 s	25.2 s	17.4 s	13.7 s	8.2 s	7.2 s
Redução	-	79.24%	85.69%	88.78%	93.32%	94.09%

Tabela 4.16 – Comparação entre os grafos construídos considerando a representação original do segundo ambiente de navegação e as representações simplificadas pelo algoritmo proposto.

Características	Original	Simplificado				
		$a_{ij} = 50 \text{ km}^2$	$a_{ij} = 100 \text{ km}^2$	$a_{ij} = 200 \text{ km}^2$	$a_{ij} = 300 \text{ km}^2$	$a_{ij} = 400 \text{ km}^2$
Número total de nós	3937	1297	841	710	571	567
Redução	-	67.06%	78.64%	81.97%	85.50%	85.60%
Número total de arestas	490707	94476	54159	43302	30522	29958
Redução	-	80.75%	88.97%	91.18%	93.78%	93.90%
Tempo para construção	982.8 s	138.6 s	74.5 s	58.3 s	43.5 s	41.5 s
Redução	-	85.91%	92.43%	94.07%	95.58%	95.78%

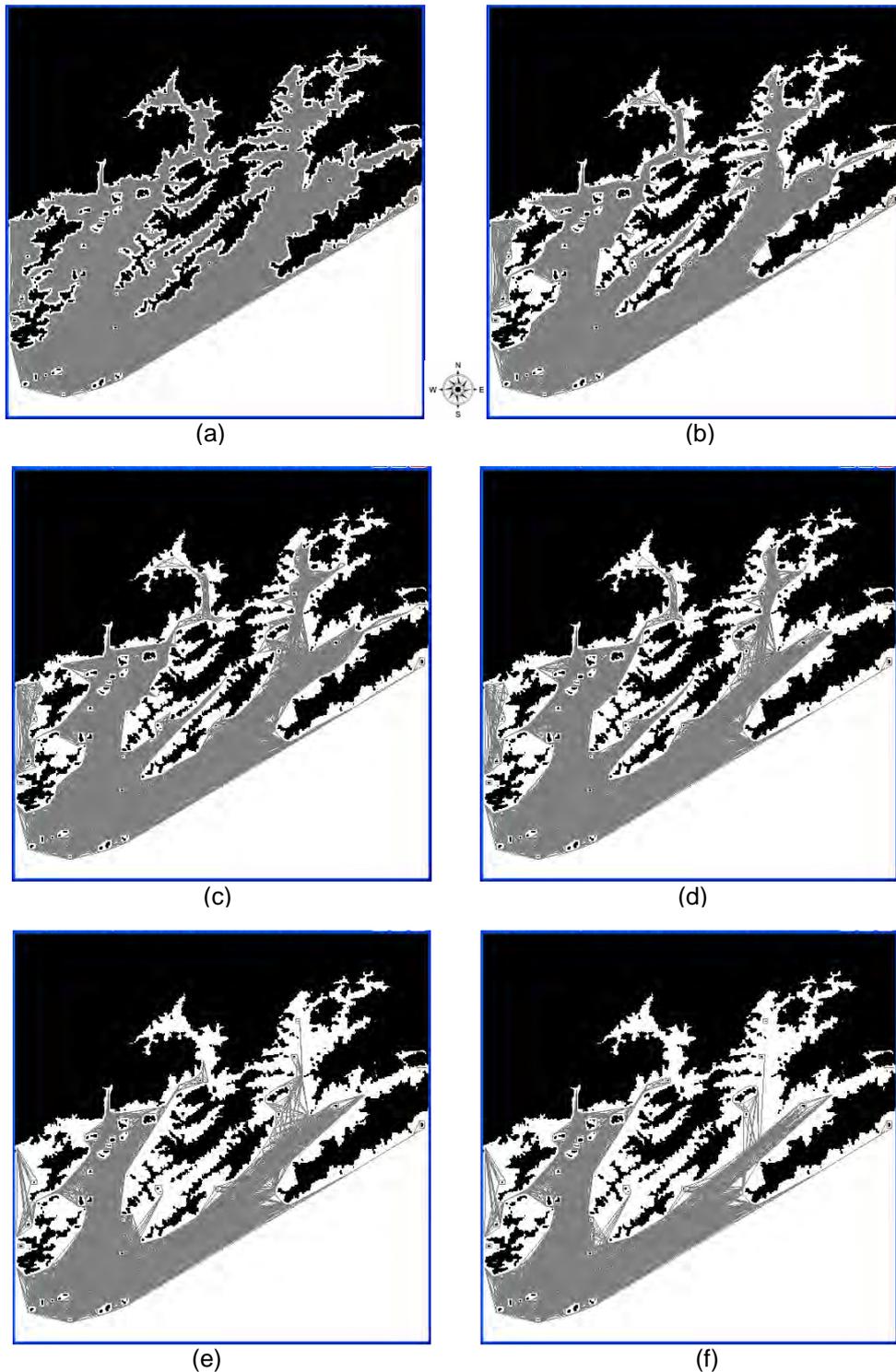


Figura 4.32 – Grafos de visibilidade construídos com a (a) representação original do primeiro ambiente de navegação e com as representações simplificadas, considerando: (b) $a_{ii} = 5 \text{ km}^2$; (c) $a_{ii} = 10 \text{ km}^2$; (d) $a_{ii} = 20 \text{ km}^2$; (e) $a_{ii} = 50 \text{ km}^2$; e (f) $a_{ii} = 100 \text{ km}^2$.

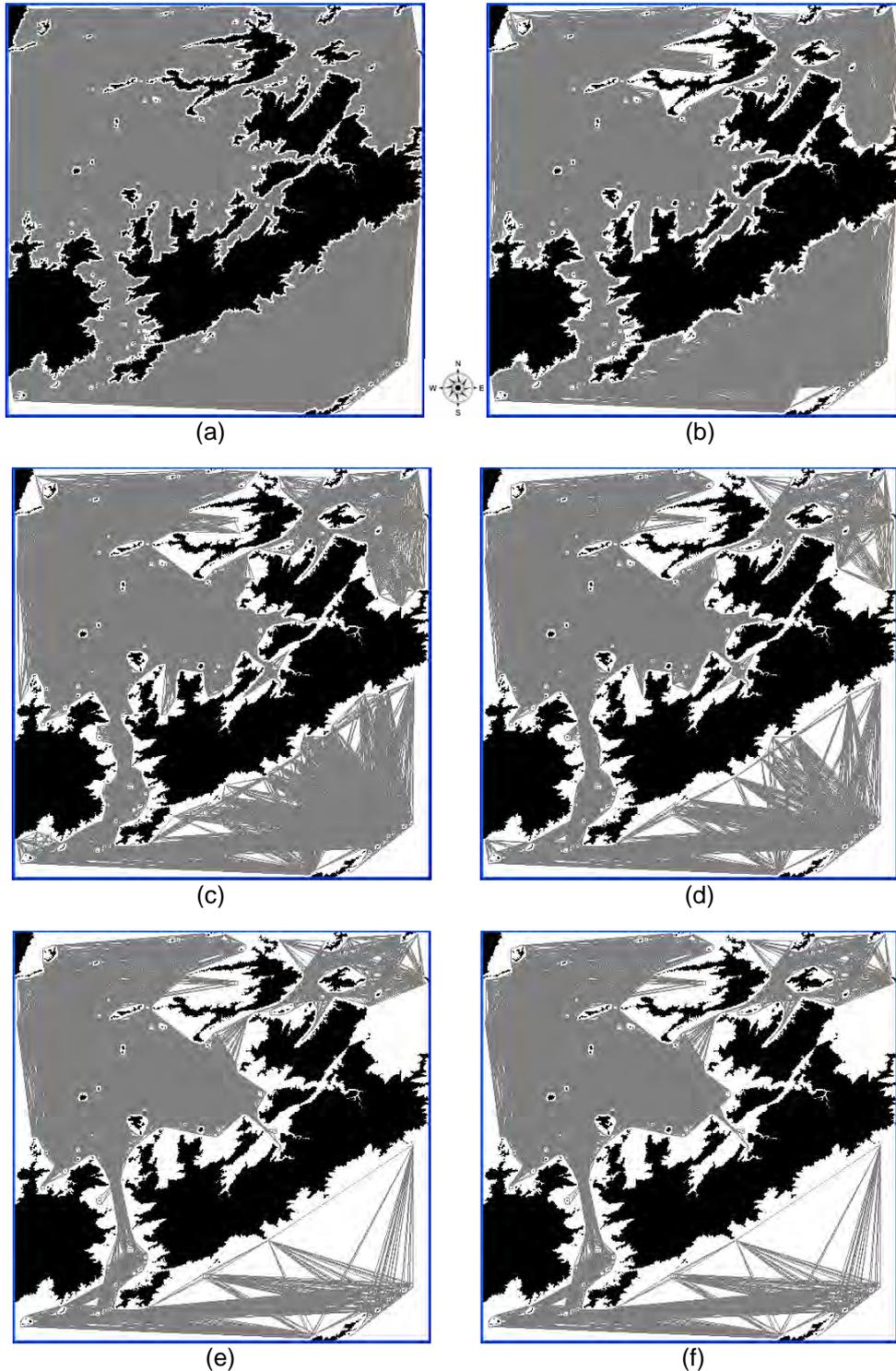


Figura 4.33 – Grafos de visibilidade construídos com a (a) representação original do segundo ambiente de navegação e com as representações simplificadas, considerando: (b) $a_{ii} = 50 \text{ km}^2$; (c) $a_{ii} = 100 \text{ km}^2$; (d) $a_{ii} = 200 \text{ km}^2$; (e) $a_{ii} = 300 \text{ km}^2$; e (f) $a_{ii} = 400 \text{ km}^2$.

Como era esperado, o segundo efeito observado da redução do número de vértices foi o decréscimo do tempo necessário para o planejamento automático de trajetórias de navegação, utilizando o algoritmo Dijkstra baseado em elevações (MEDEIROS e SILVA, 2010b). A Tabela 4.17 apresenta os resultados obtidos com o planejamento de movimento entre a coordenada de origem $w'_o = (latitude_o, longitude_o) = (-22.91, -45.95)$ e coordenada de destino $w'_d = (latitude_d, longitude_d) = (-23.0, -45.735)$, considerando: o raio de curva de 400 m; a envergadura do veículo igual a 15 metros; o grafo de visibilidade através da representação original do primeiro ambiente de navegação; e os grafos construídos através das representações simplificadas pelo algoritmo proposto, isto é, os grafos simplificados. A Tabela 4.18 apresenta os resultados obtidos com o planejamento de movimento entre a coordenada de origem $w'_o = (latitude_o, longitude_o) = (-22.91, -45.95)$ e coordenada de destino $w'_d = (latitude_d, longitude_d) = (-22.58, -45.482)$, considerando: o raio de curva igual a 1500 m; a envergadura do veículo igual a 15 m; o grafo de visibilidade através da representação original do segundo ambiente de navegação; e os grafos simplificados pelo algoritmo proposto.

O tempo necessário para o planejamento de uma trajetória de navegação considerando o grafo de visibilidade da Figura 4.32b é 26.81% inferior ao tempo necessário para o planejamento da mesma trajetória, baseado no grafo da representação original da Figura 4.32a. Uma redução também significativa foi observada com o segundo ambiente de navegação. O tempo necessário para o planejamento de uma trajetória considerando o grafo da Figura 4.33b é 64.93% inferior ao tempo necessário para o planejamento da mesma trajetória, considerando o grafo da Figura 4.33a.

Entretanto, outro efeito esperado foi observado: a redução do número de vértices pode causar o aumento do comprimento da trajetória planejada ou até mesmo impossibilitar o planejamento da mesma. A simplificação do primeiro ambiente de navegação com $a_{ij} = 20 \text{ km}^2$ ocasionou um aumento de 0.003% do comprimento da trajetória planejada. A simplificação do segundo ambiente de

navegação ocasionou um aumento de 0.082% do comprimento da trajetória planejada, considerando $a_{ij} = 400 \text{ km}^2$.

Tabela 4.17 – Comparação entre as trajetórias planejadas considerando o grafo de visibilidade original do primeiro ambiente de navegação e os grafos simplificados pelo algoritmo proposto.

Características	Original	Simplificado				
		$a_{ij} = 5 \text{ km}^2$	$a_{ij} = 10 \text{ km}^2$	$a_{ij} = 20 \text{ km}^2$	$a_{ij} = 50 \text{ km}^2$	$a_{ij} = 100 \text{ km}^2$
Comprimento da trajetória	52928.24 m	52929.8 m	52929.8 m	52929.8 m	-	-
Aumento	-	0.003%	0.003%	0.003%	-	-
Tempo para planejamento	2.2 s	1.6 s	1.6 s	1.6 s	-	-
Redução	-	26.81%	26.81%	26.81%	-	-

Tabela 4.18 – Comparação entre as trajetórias planejadas considerando o grafo de visibilidade original do segundo ambiente de navegação e os grafos simplificados pelo algoritmo proposto.

Características	Original	Simplificado				
		$a_{ij} = 50 \text{ km}^2$	$a_{ij} = 100 \text{ km}^2$	$a_{ij} = 200 \text{ km}^2$	$a_{ij} = 300 \text{ km}^2$	$a_{ij} = 400 \text{ km}^2$
Comprimento da trajetória	222237.62 m	222244.13 m	222244.13 m	222305.3 m	222419.6 m	222419.6 m
Aumento	-	0.003%	0.003%	0.031%	0.082%	0.082%
Tempo para planejamento	7.5 s	2.7s	2.1 s	2 s	1.8 s	1.8 s
Redução	-	64.93%	72.03%	74.11%	76.41%	76.41%

Analisando a Tabela 4.17, observa-se que não foi possível planejar trajetórias através dos dois últimos grafos de visibilidade construídos a partir das representações simplificadas pelo algoritmo proposto. As Figuras 4.34 a 4.37 apresentam as trajetórias planejadas referentes aos dados apresentados na Tabela 4.17 e as Figuras 4.38 a 4.43 apresentam as trajetórias planejadas referentes aos dados da Tabela 4.18. Nestas figuras, cada trajetória planejada é representada em vermelho. As circunferências tracejadas vermelhas representam os locais em que o veículo deve realizar uma curva baseado no raio de curva utilizado no planejamento.

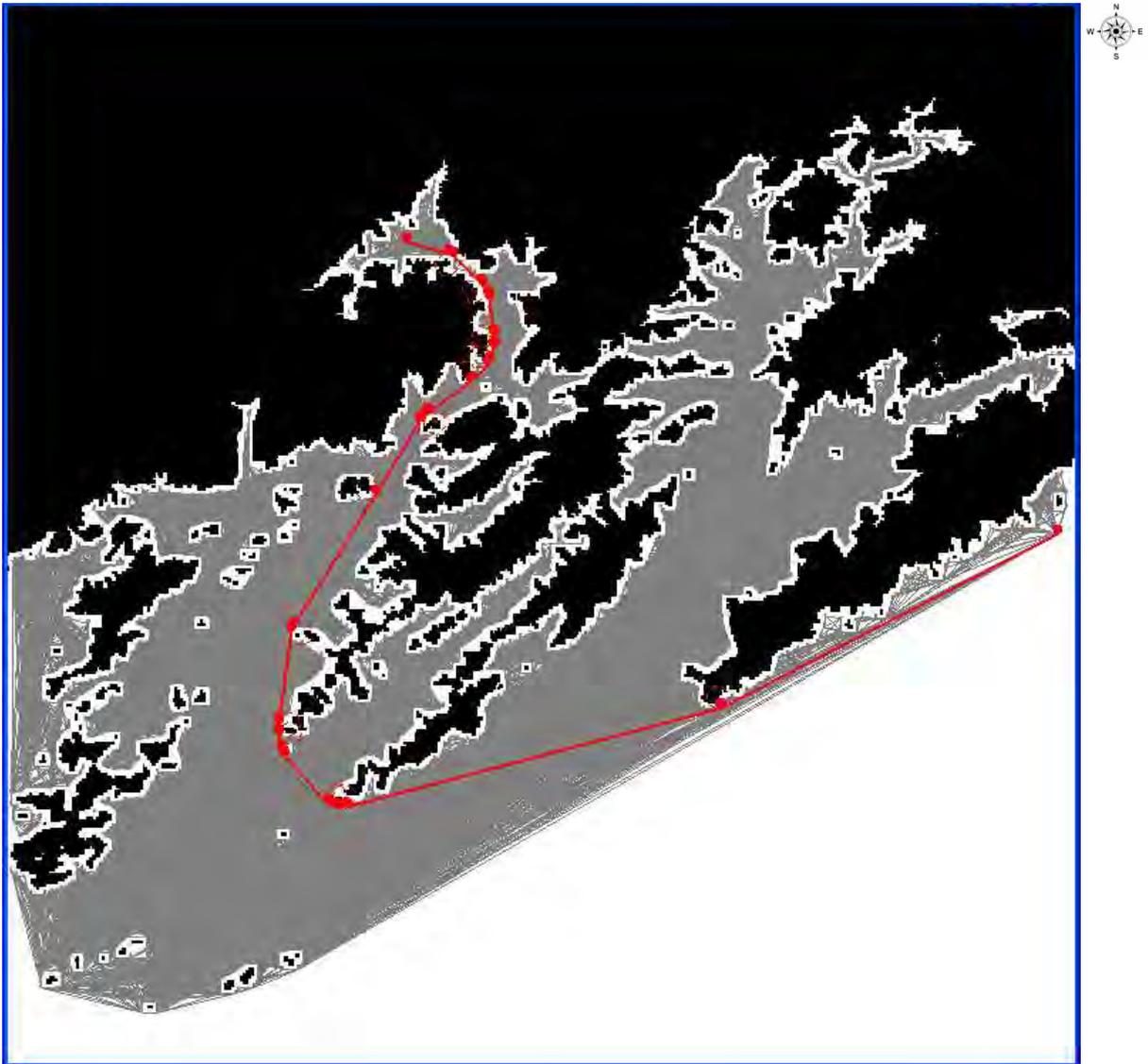


Figura 4.34 – Trajetória planejada com o grafo de visibilidade original do primeiro ambiente de navegação.

É importante ser observado que o veículo não percorrerá toda a circunferência tracejada, apenas a curva definida pelo arco entre duas coordenadas w_i e w_{i+1} da trajetória, tal que a coordenada central da extensão deste arco é a mais próxima do encontro w' dos prolongamentos dos segmentos de reta $\overline{w_{i-1}w_i}$ e $\overline{w_{i+1}w_{i+2}}$.

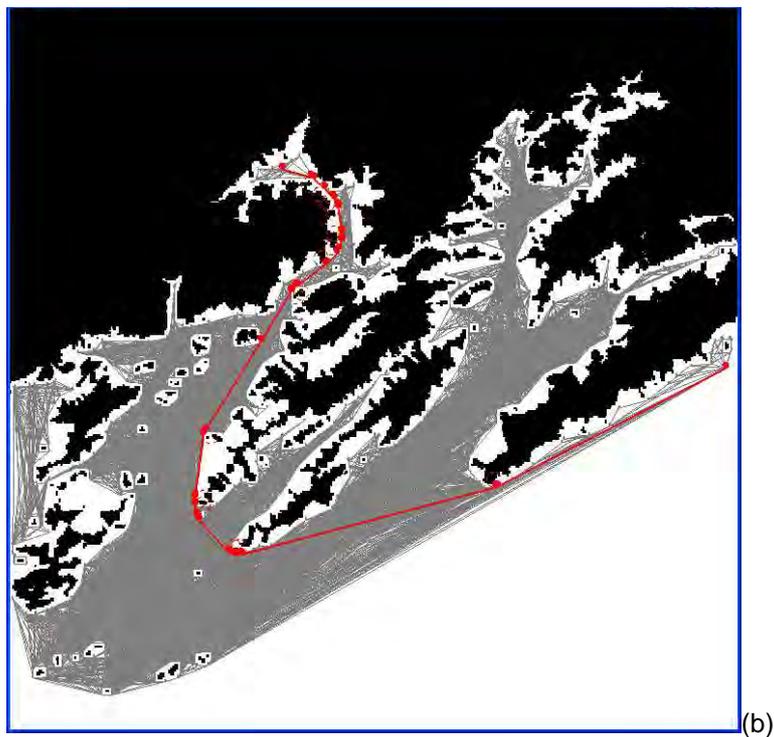
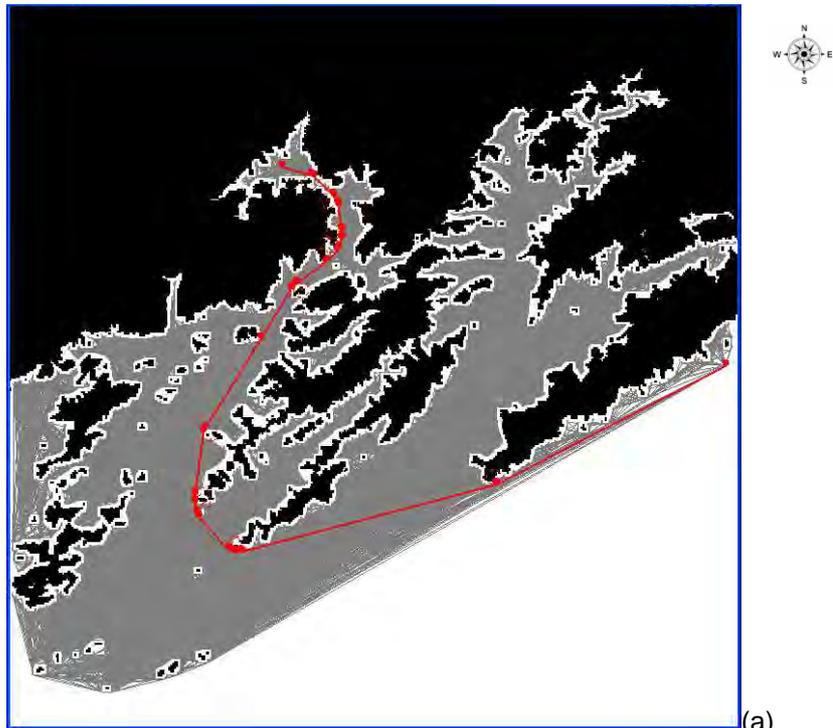


Figura 4.35 – Trajetórias planejadas com (a) o grafo de visibilidade original do primeiro ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 5 \text{ km}^2$.

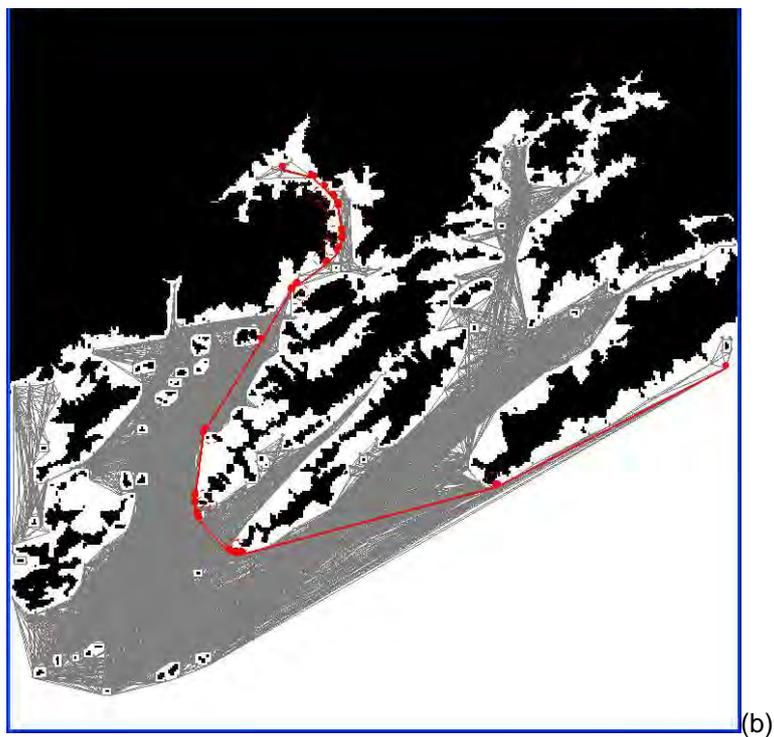
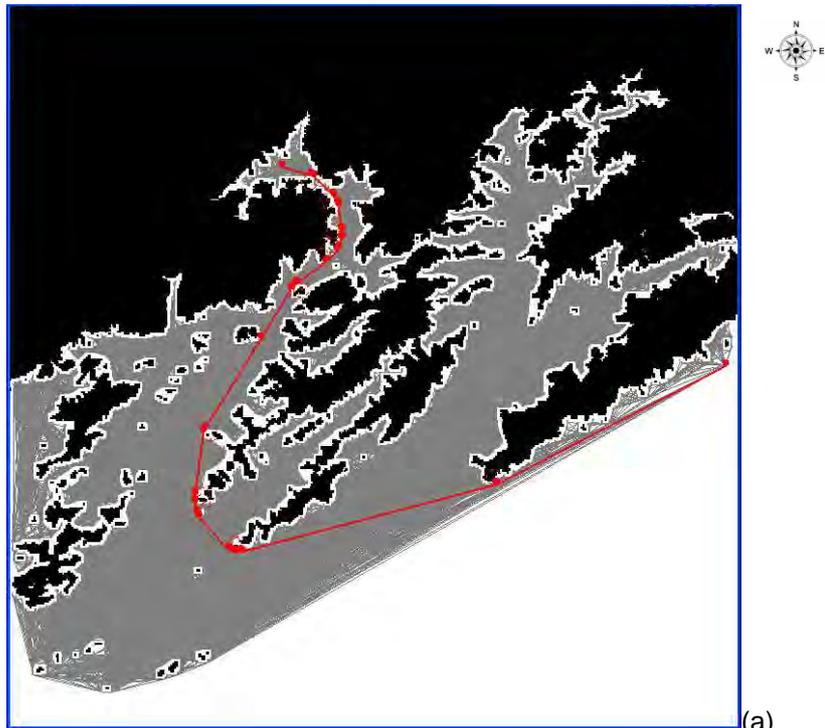


Figura 4.36 – Trajetórias planejadas com (a) o grafo de visibilidade original do primeiro ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 10 \text{ km}^2$.

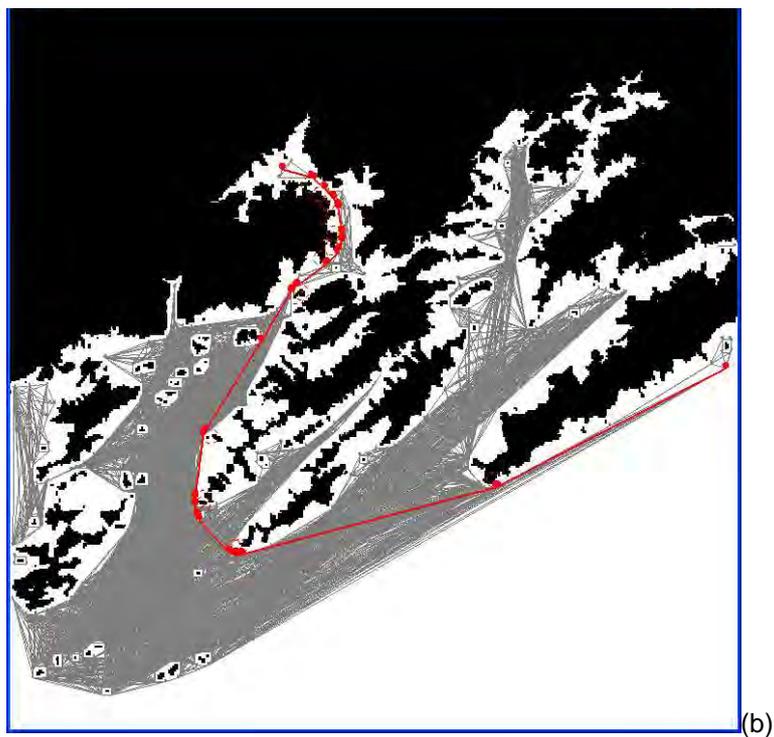
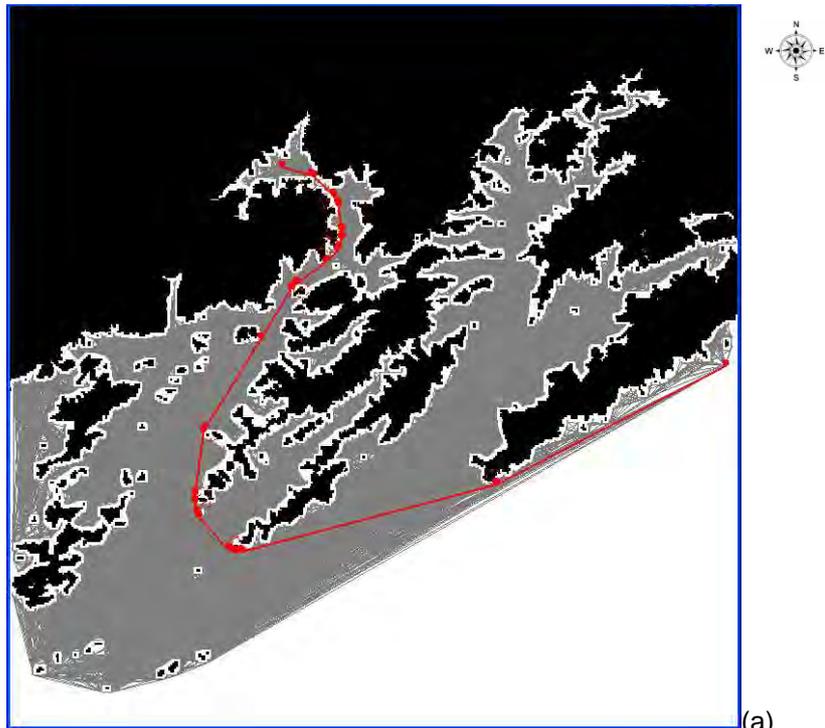


Figura 4.37 – Trajetórias planejadas com (a) o grafo de visibilidade original do primeiro ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 20 \text{ km}^2$.

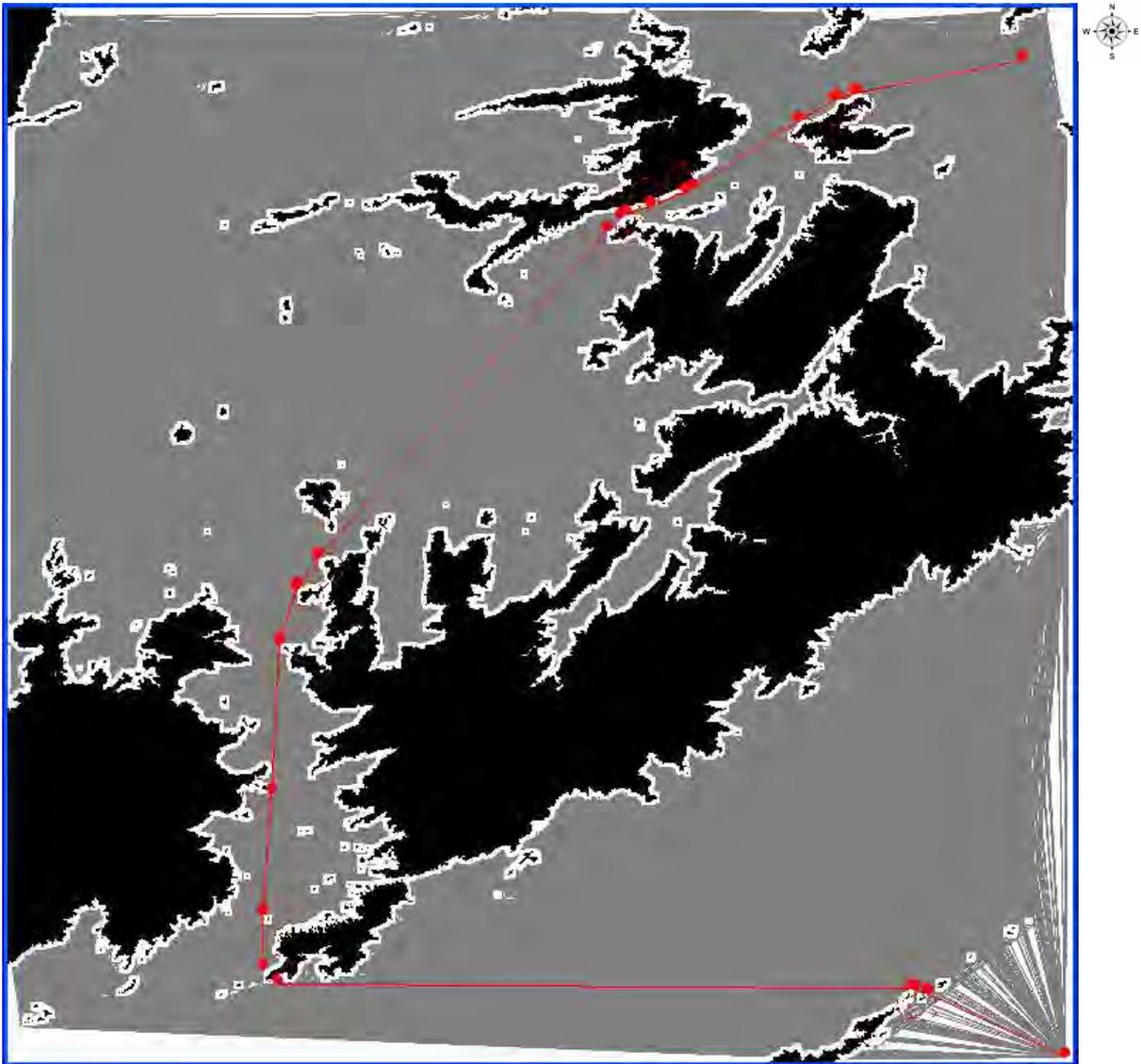


Figura 4.38 – Trajetória planejada com o grafo de visibilidade original do segundo ambiente de navegação.

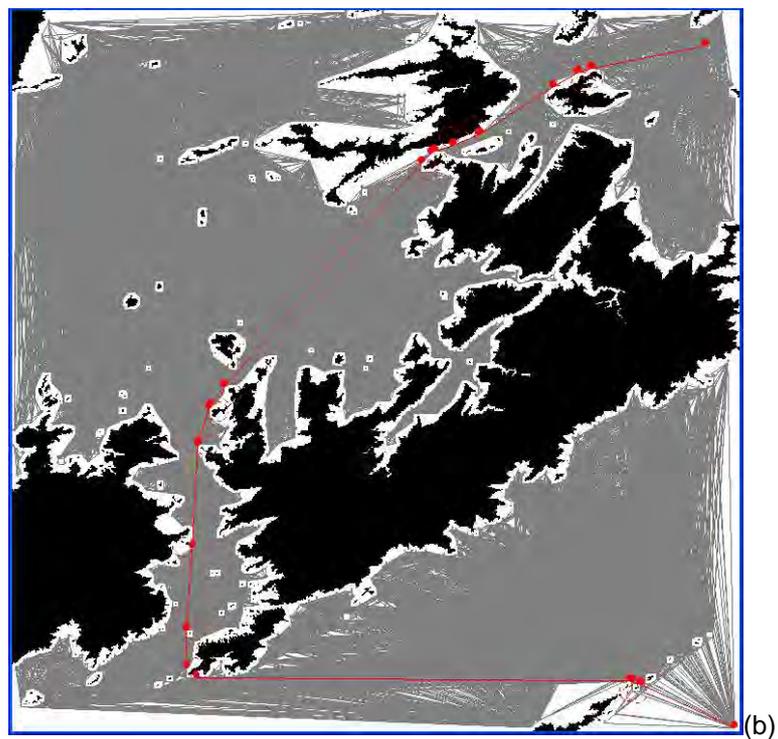
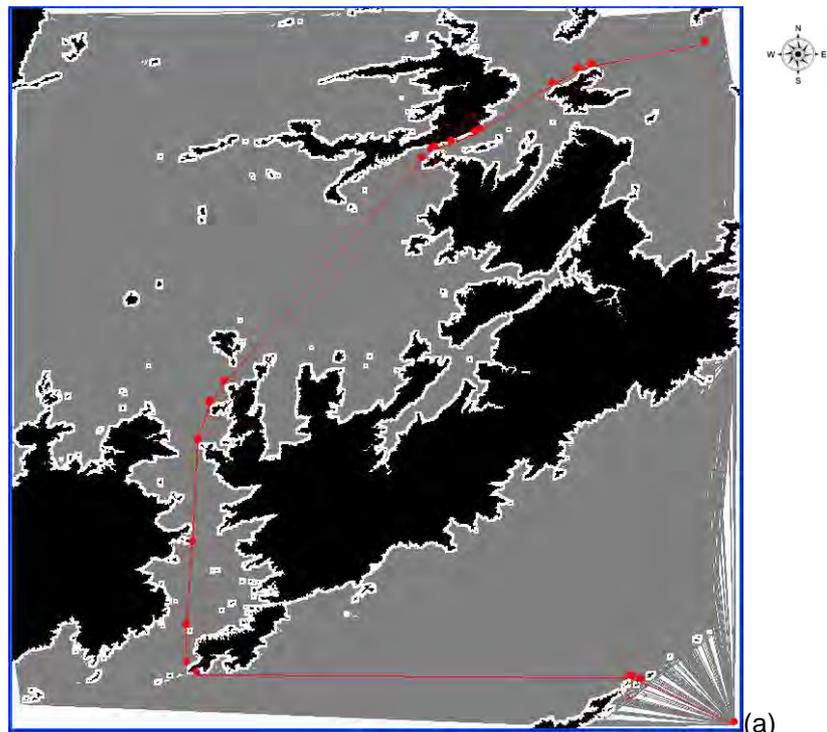


Figura 4.39 – Trajetórias planejadas com (a) o grafo de visibilidade original do segundo ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 50 \text{ km}^2$.

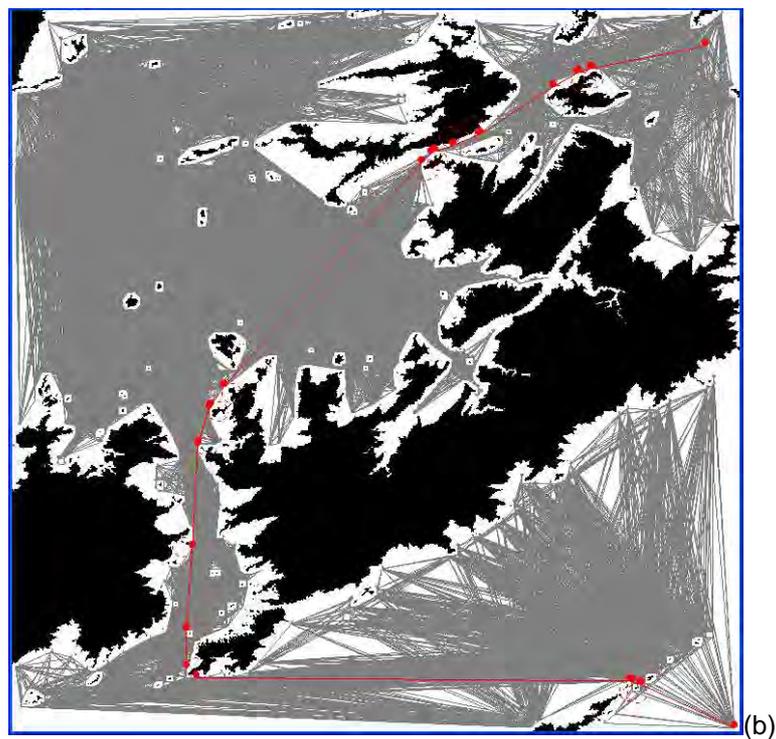
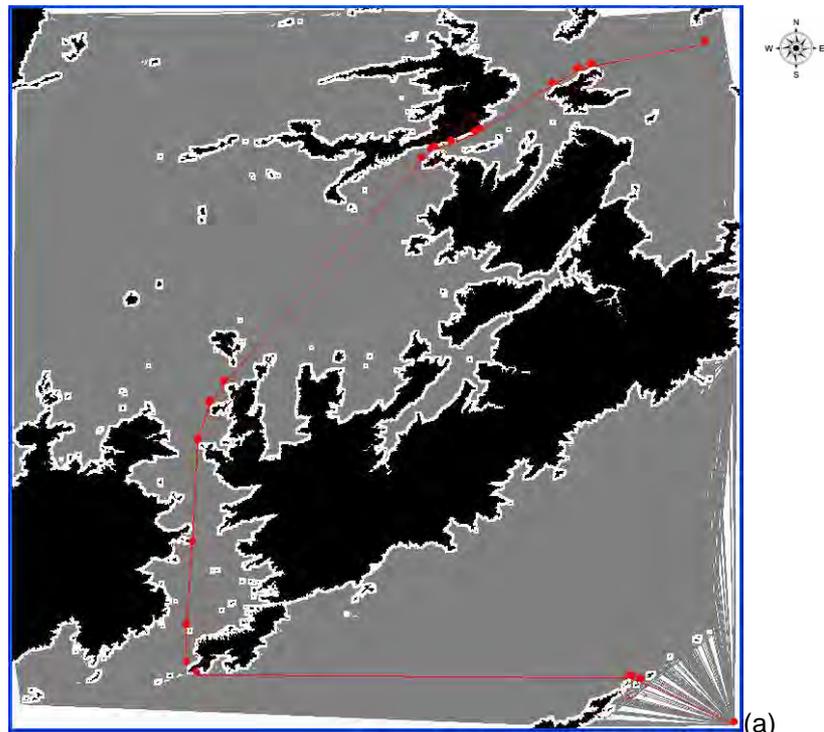


Figura 4.40 – Trajetórias planejadas com (a) o grafo de visibilidade original do segundo ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 100 \text{ km}^2$.

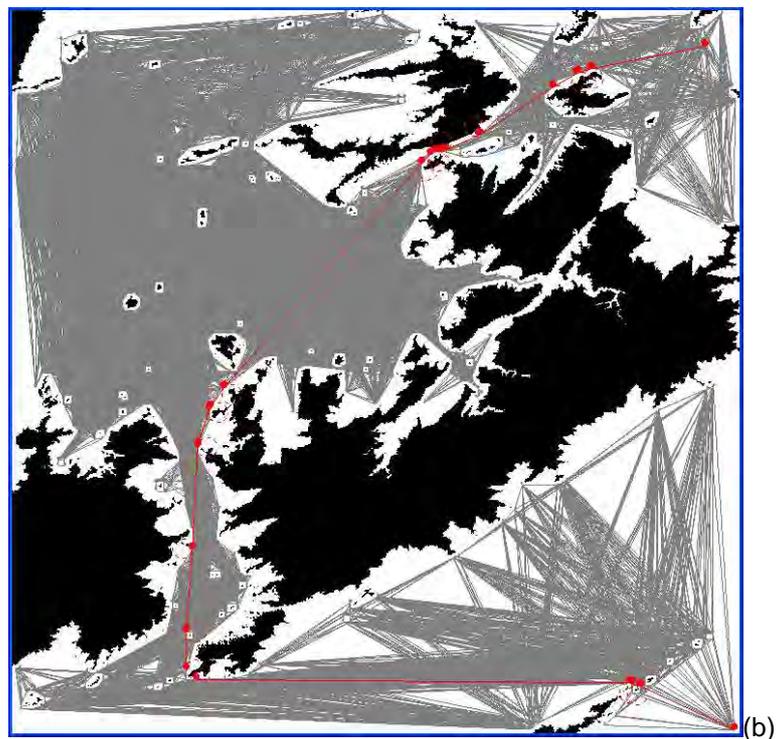
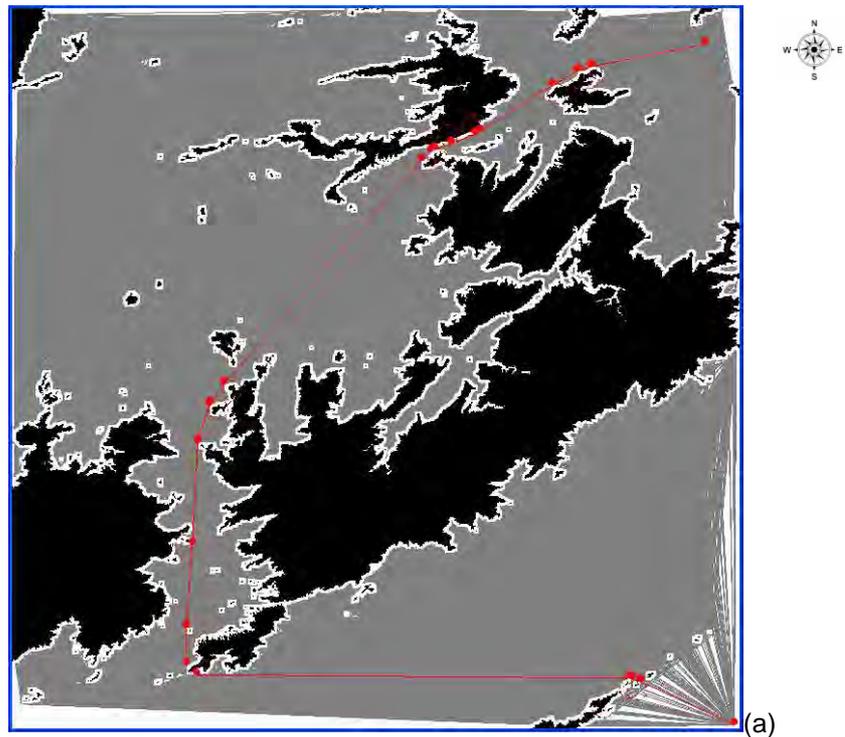


Figura 4.41 – Trajetórias planejadas com (a) o grafo de visibilidade original do segundo ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 200 \text{ km}^2$.

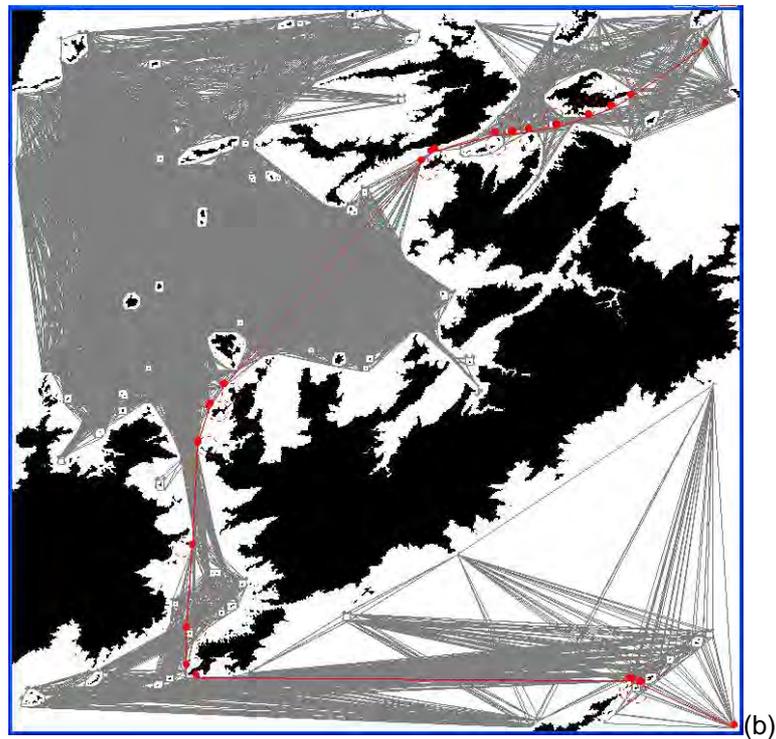
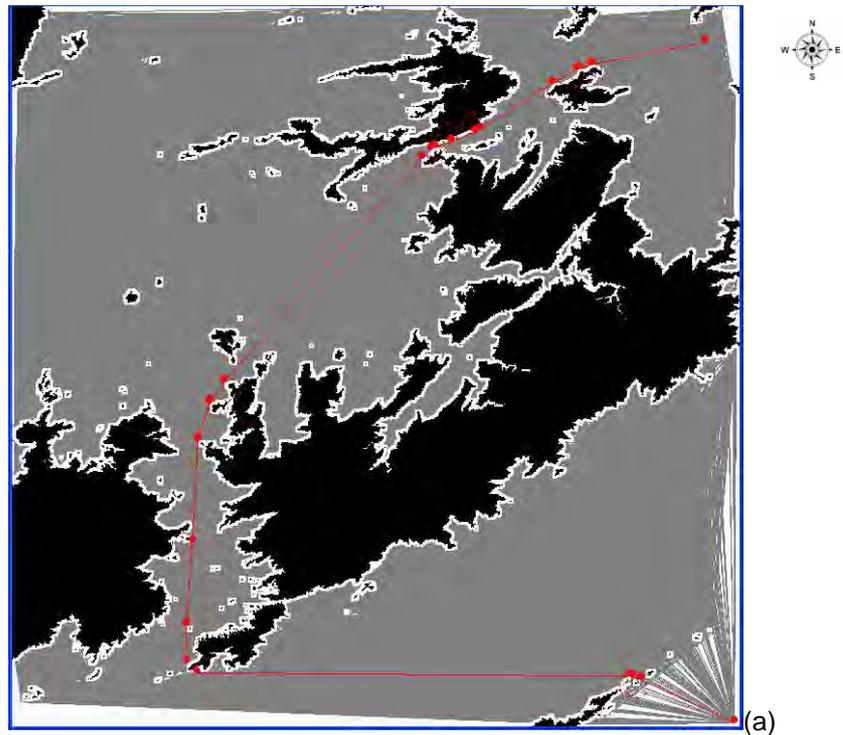


Figura 4.42 – Trajetórias planejadas com (a) o grafo de visibilidade original do segundo ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 300 \text{ km}^2$.

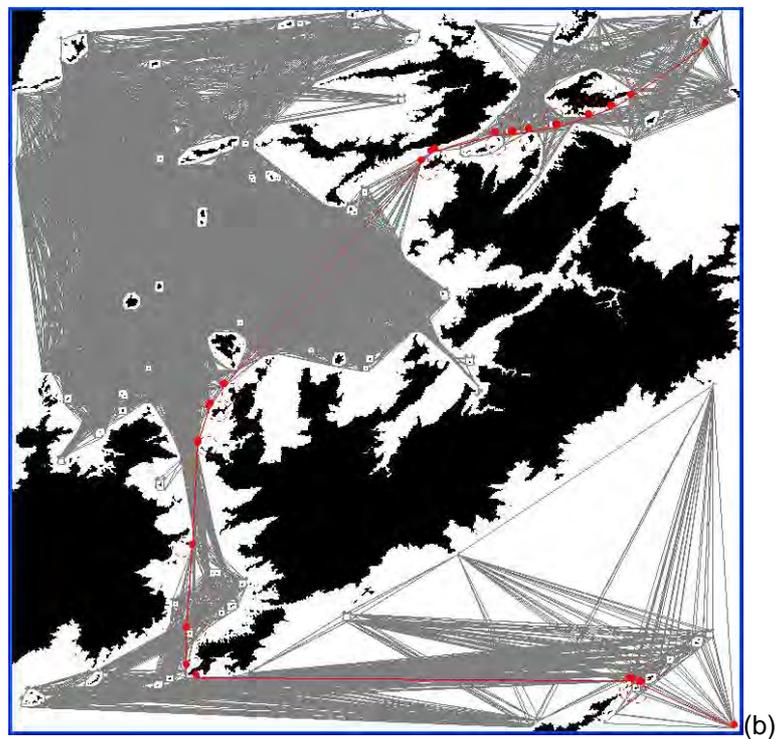
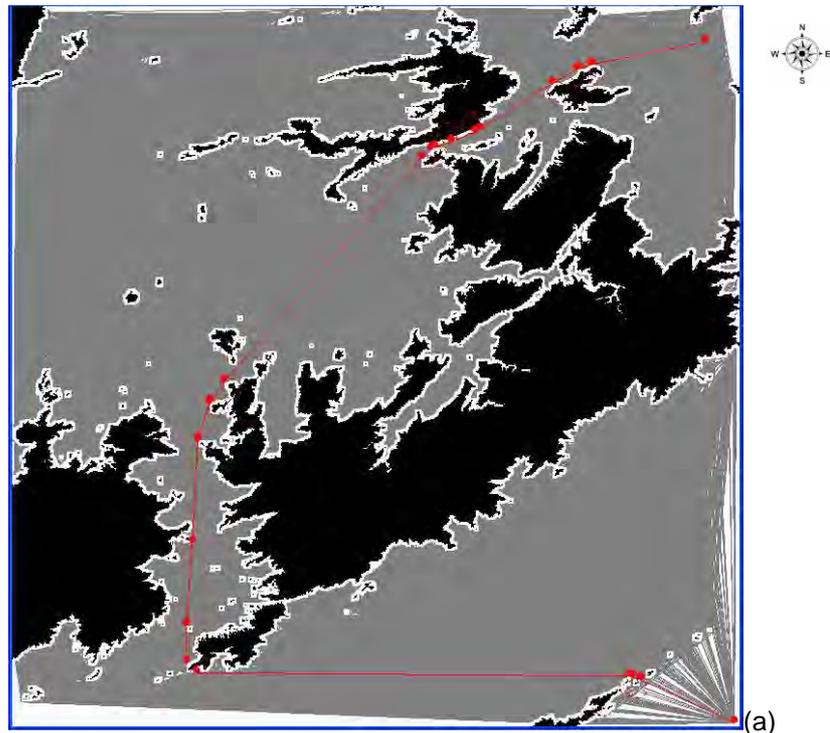


Figura 4.43 – Trajetórias planejadas com (a) o grafo de visibilidade original do segundo ambiente de navegação e com (b) o grafo simplificado do primeiro ambiente de navegação, considerando $a_{ij} = 400 \text{ km}^2$.

Analisando o algoritmo proposto, pode-se perceber que, a cada ciclo, o polígono é simplificado enquanto a restrição de área de invisibilidade limitante não for violada, enquanto a restrição de visibilidade for atendida ou até que todos os vértices de V_e sejam intermediários convexos. Cada ciclo consiste na análise do primeiro ao último vértice de V_e . Porém, o algoritmo foi alterado de modo que um ciclo corresponde à análise sequencial de todos os vértices de todos os polígonos. Assim, a cada ciclo, o conjunto de polígonos é simplificado. O problema verificado é que a restrição de área de invisibilidade limitante pode ser violada para as k primeiras duplas de vértices intermediários convexos v_{ei} e v_{ej} , gerando a eliminação de um número reduzido de vértices em comparação com o número de vértices que seriam eliminados se as demais duplas de vértices intermediários convexos fossem avaliadas. Por exemplo, se a restrição de área de invisibilidade limitante for atendida na simplificação dos primeiros polígonos, os demais polígonos do ambiente de navegação não são simplificados. Portanto, o modo mais eficiente de simplificação seria o cálculo e a ordenação crescente de todas as áreas de invisibilidade dos vértices intermediários convexos de todos os polígonos, e a eliminação dos vértices começando pelas menores áreas de invisibilidade enquanto a restrição de área de invisibilidade resultante não for violada.

Assim, um trabalho futuro é a alteração do algoritmo para utilizar a simplificação por ordenação de áreas de invisibilidade entre vértices intermediários convexos.

4.2. Simplificação por eliminação de nós ou de arestas de grafos

Há diversos trabalhos que abordam a simplificação de grafos baseada na eliminação de nós ou de arestas. Nestes trabalhos, os métodos utilizados baseiam-se em métricas e limiares para a eliminação. São exemplos destas métricas e limiares: medidas de centralidade do nó avaliado (SABIDUSSI, 1966), como a centralidade de grau, que expressa o número de nós conectados ao nó avaliado; a distância entre o nó avaliado e um nó de

referência (HENNESSEY et al., 2008); o custo da aresta avaliada, como a simplificação por árvore geradora mínima (*minimum spanning tree*)(Ban e Sen, 2003); o número de nós em comum conectados a dois nós avaliados (GILBERT e LEVCHENKO, 2004); o número de tangentes entre obstáculos poligonais, como no algoritmo linha de varredura rotacional (*rotational sweep line algorithm*) (CHOSSET et al., 2005); etc.

Porém, nos métodos mencionados e destinados à eliminação de nós, a eliminação não está vinculada à área de invisibilidade gerada pelo vértice intermediário convexo que define tal nó ou ao problema de ocultação de obstáculos.

Como apresentado no Capítulo 2, grafos de visibilidade possuem a propriedade de permitir o planejamento da menor rota entre duas coordenadas navegáveis de um ambiente de navegação devido à conexão entre nós definidos pelos vértices intermediários convexos das regiões não navegáveis (DE BERG et al., 2000). O problema verificado com a simplificação através da eliminação de arestas é que o grafo simplificado viola a definição de grafo de visibilidade e, portanto, não possui a propriedade mencionada.

Para exemplificar este problema, foi elaborado um algoritmo para simplificação de grafos que utiliza o agrupamento de nós baseado na distância média entre os mesmos (SHAEFFER, 2007). Posteriormente, trajetórias foram planejadas com os grafos simplificados.

No algoritmo elaborado, os nós de um grafo de visibilidade são agrupados através do algoritmo *k*-médias e todos os grupos são conectados por arestas que unem os dois nós mais próximos de cada grupo e que não interceptam regiões não navegáveis. O algoritmo é apresentado na Tabela 4.19.

Tabela 4.19 – Algoritmo para simplificação de grafos de visibilidade baseado em agrupamento de nós.

Índice	Algoritmo para Simplificação Baseado em Agrupamento de Nós
1	agrupar os nós do grafo através do algoritmo k -médias
2	para i variando de 1 até $k-1$ faça
3	para j variando de $i+1$ até k faça
4	determinar os dois vértices mais próximos entre o grupo i e o grupo j
5	eliminar todas as arestas que conectam os nós dos dois grupos i e j com exceção da aresta que conecta os dois vértices mais próximos

O algoritmo da Tabela 4.19 foi aplicado à simplificação dos grafos de visibilidade construídos através das representações originais dos dois ambientes de navegação analisados neste trabalho. A Tabela 4.20 e a Tabela 4.21 apresentam comparações entre os grafos simplificados pelo algoritmo proposto. A Figura 4.44 e a Figura 4.45 apresentam tais grafos simplificados.

Tabela 4.20 – Comparação entre o grafo de visibilidade original do primeiro ambiente de navegação e os grafos simplificados pelo algoritmo baseado em agrupamento de nós.

Características	Original	Simplificado				
		$k = 5$	$k = 10$	$k = 20$	$k = 30$	$k = 40$
Número de Arestas	65241	43773	36504	30156	24459	23655
Redução	-	32.91%	44.05%	53.78%	62.51%	63.75%
Tempo para simplificação	-	248.9 s	250.6 s	253.6 s	258.6 s	255.3 s

Tabela 4.21 – Comparação entre o grafo de visibilidade original do segundo ambiente de navegação e os grafos simplificados pelo algoritmo baseado em agrupamento de nós.

Características	Original	Simplificado		
		$k = 10$	$k = 40$	$k = 100$
Número de arestas	490707	234699	125337	86226
Redução	-	52.18%	74.46%	82.43%
Tempo para simplificação	-	1962.1 s	2023.1 s	2029.3 s

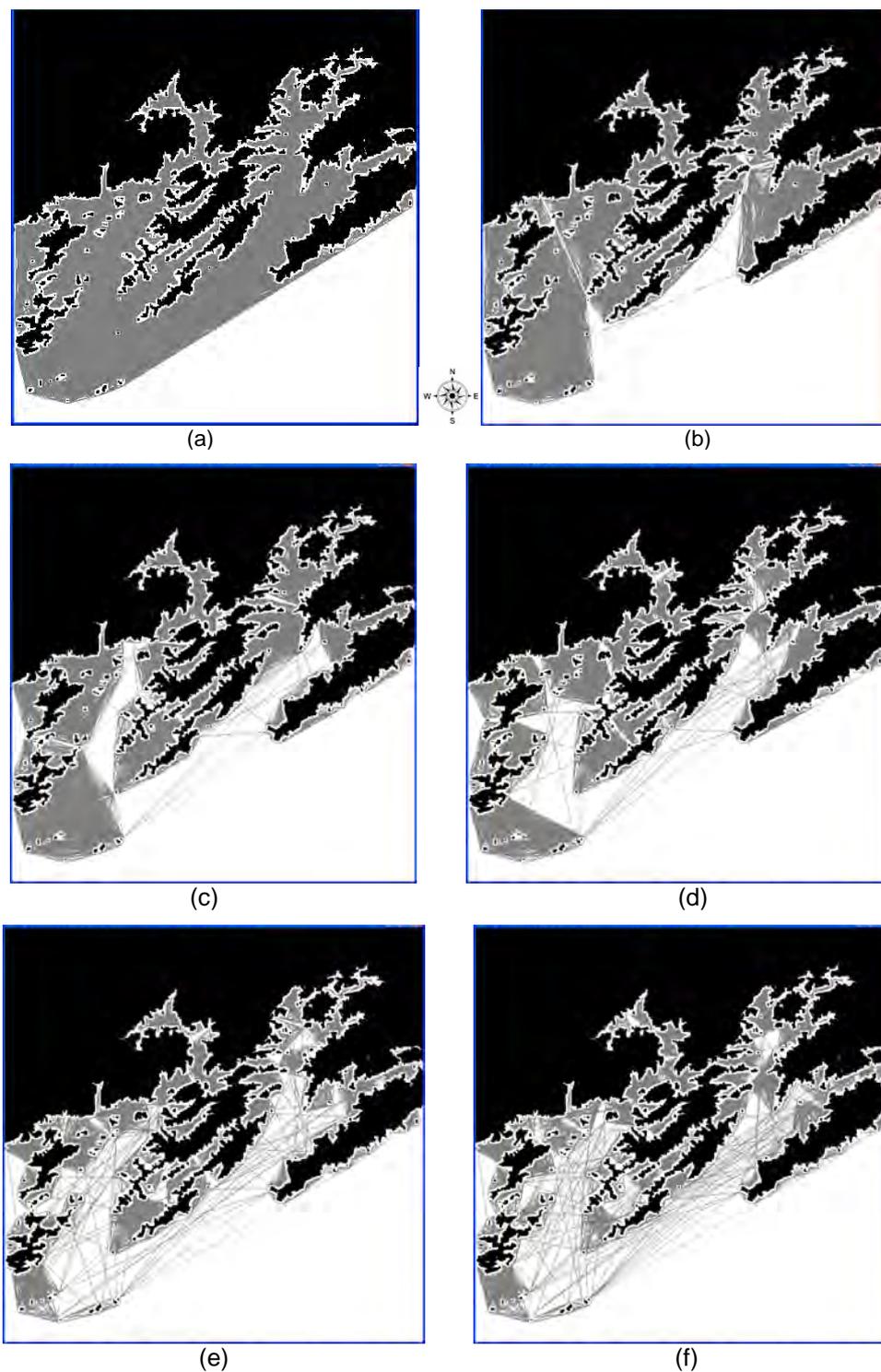


Figura 4.44 – (a) Grafo de visibilidade construído com a representação original do primeiro ambiente de navegação e grafos simplificados, considerando: (b) $k = 5$ grupos; (c) $k = 10$ grupos; (d) $k = 20$ grupos; (e) $k = 30$ grupos; e (f) $k = 40$ grupos.

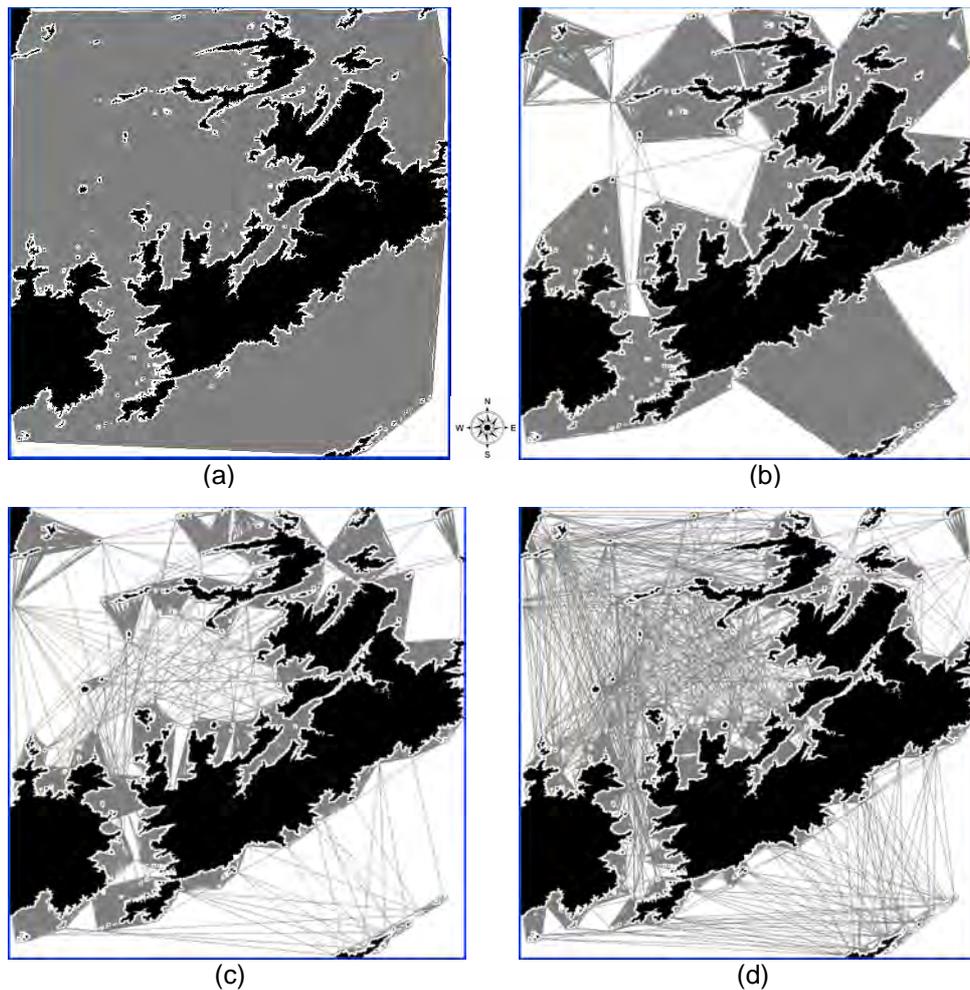


Figura 4.45 – (a) Grafo de visibilidade construído com a representação original do segundo ambiente de navegação e grafos simplificados, considerando: (b) $k = 10$ grupos; (c) $k = 40$ grupos; e (d) $k = 100$ grupos.

A Figura 4.46 e a Figura 4.47 apresentam exemplos de trajetórias planejadas pelo algoritmo Dijkstra proposto em (MEDEIROS e SILVA, 2010b), utilizando alguns dos grafos simplificados. Apesar da mesma distribuição de nós, observa-se que as trajetórias planejadas não possuem a menor extensão possível através de uma comparação com as trajetórias planejadas com os grafos de visibilidade originais. A trajetória da Figura 4.46 possui uma extensão de 53381.95 m e a trajetória da Figura 4.47 possui uma extensão de 242019.72 m.

Nestes exemplos, foram considerados os mesmos raios de curva e coordenadas de origem e de destino considerados para as trajetórias planejadas com os grafos de visibilidade obtidos pelas simplificações geradas pelo algoritmo de simplificação proposto neste trabalho.

Embora o elevado número de nós e de arestas tenha sido constatado também na construção de grafos de visibilidade associados com a avaliação dos efeitos da queda de VANTs, neste capítulo, foi estudado somente a simplificação de grafos construídos por meio de modelos digitais de elevações. Contudo, como foi apresentado neste capítulo, os problemas identificados e as soluções encontradas estão relacionados com as regiões não navegáveis e independem se estas regiões são determinadas por modelos digitais de elevação ou por distribuições espaciais da métrica para avaliação dos efeitos da queda de VANTs.

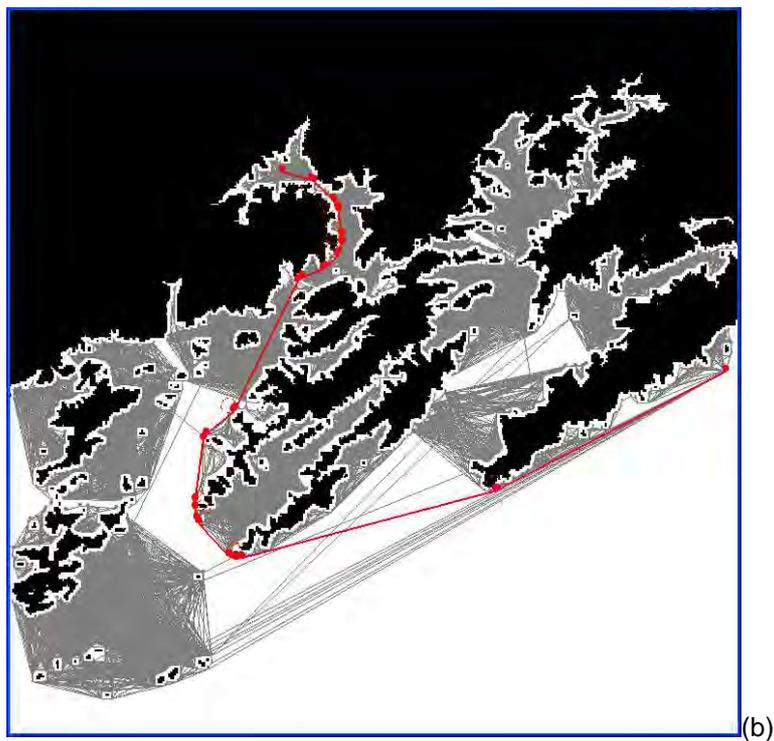
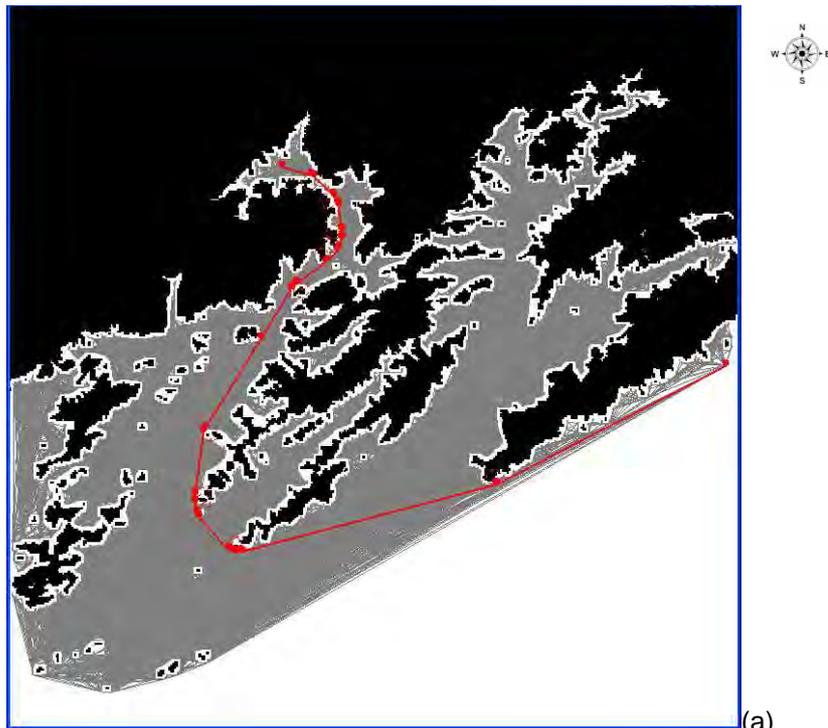


Figura 4.46 – Trajetórias planejadas com (a) o grafo simplificado original do primeiro ambiente de navegação e com (b) o grafo simplificado, considerando $k = 10$ grupos.

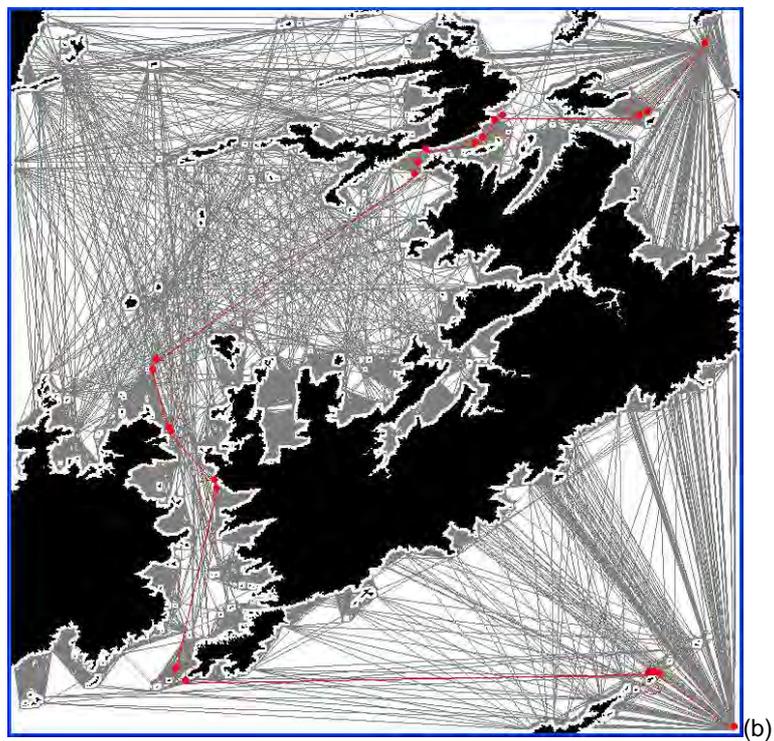
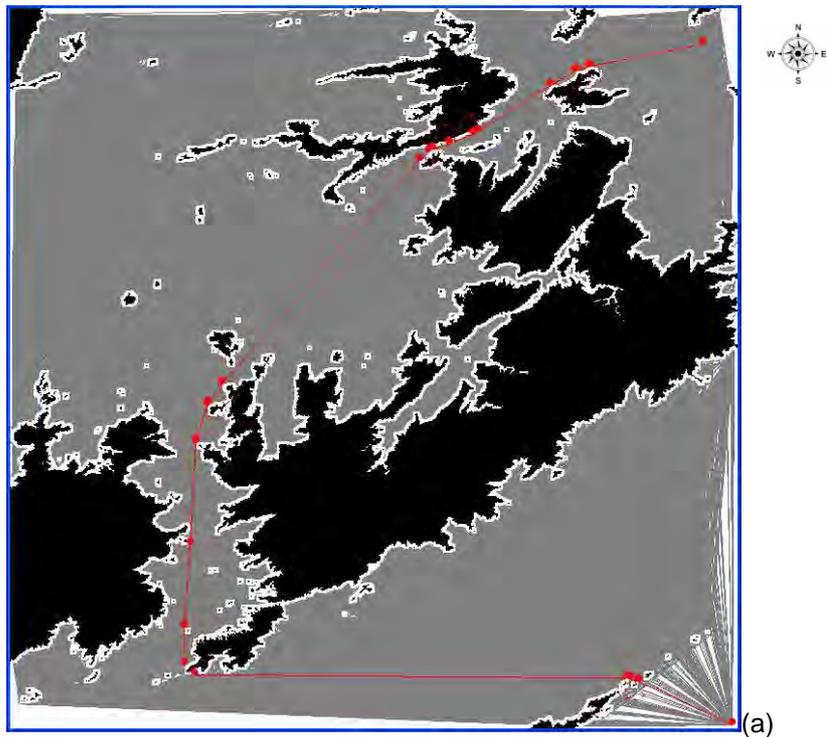


Figura 4.47 – Trajetórias planejadas com (a) o grafo simplificado original do segundo ambiente de navegação e com (b) o grafo simplificado, considerando $k = 100$ grupos.

5 PLANEJAMENTO AUTOMÁTICO DE TRAJETÓRIAS DE NAVEGAÇÃO PARA VANTS

Este trabalho aborda o problema de planejamento automático de trajetória ou movimento para VANTS considerando constantes a altitude e o módulo da velocidade de navegação. Fundamentando-se nestas considerações, o movimento bidimensional de um VANT pode ser modelado matematicamente como o movimento de um carro de Dubins (BRY e ROY, 2011; TSOURDOS et al., 2011; DING et al., 2010; ENG et al., 2010; CHITSAZ e LAVALLE, 2007; ANDERSON et al., 2005; LAVALLE, 2006), sendo definido por

$$\begin{cases} \dot{x}_v = v_v \cos \theta_v \\ \dot{y}_v = v_v \sin \theta_v \\ \dot{\theta}_v = \omega_v \in \left\{ -\frac{v_v}{r_c}, 0, \frac{v_v}{r_c} \right\} \end{cases} \quad (5.1)$$

Em que: (x_v, y_v) é a posição do VANT; v_v é a velocidade do VANT; θ_v é o ângulo formado pelo eixo x e o eixo longitudinal do VANT; ω_v é a velocidade angular do VANT; e r_c é o raio de curva da trajetória do VANT.

5.1. Revisão bibliográfica e motivação

Como mencionado anteriormente, neste trabalho, as regiões navegáveis de um ambiente de navegação para VANTS são representadas por grafos de visibilidade.

Há três principais abordagens para o planejamento de trajetórias para VANTS usando grafos de visibilidade. As três abordagens são apresentadas através dos esquemas simplificados da Figura 5.1.

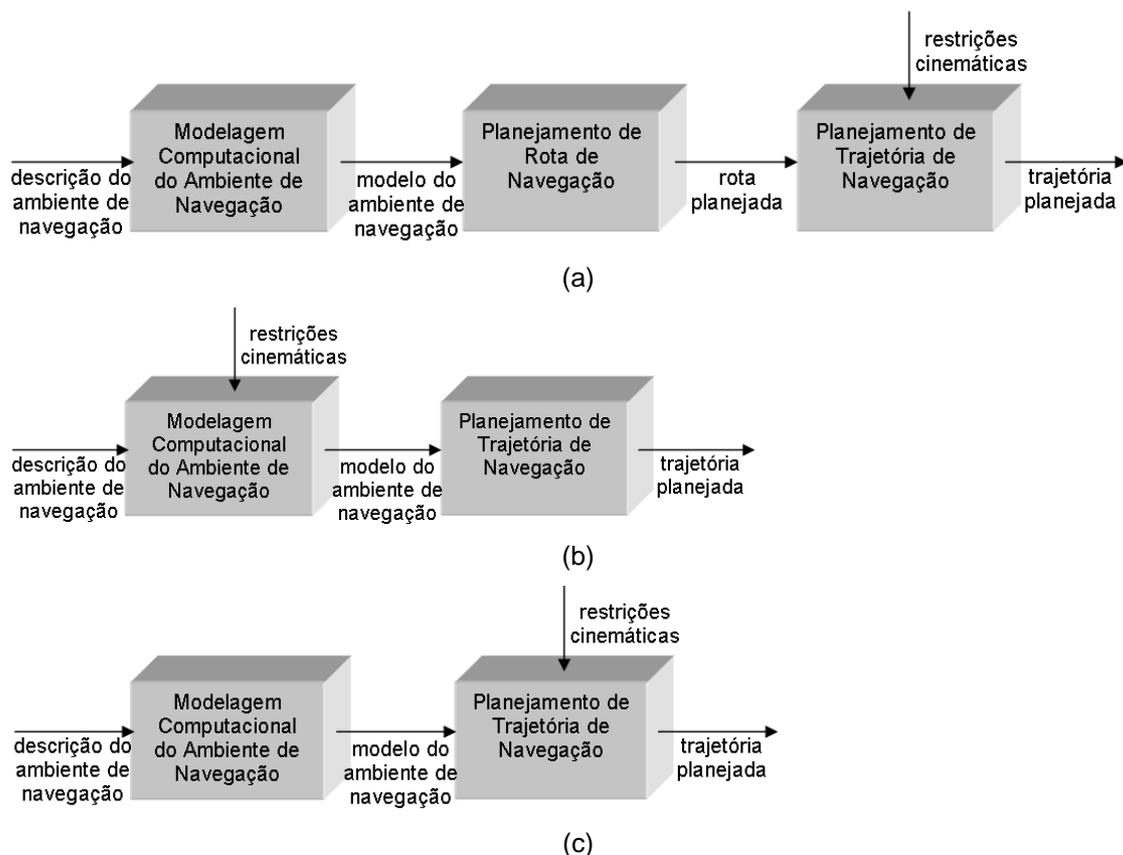


Figura 5.1 – Três principais abordagens para o problema de planejamento automático de trajetórias através de grafos de visibilidade.

Na primeira abordagem (TSOURDOS et al., 2011; TIN, 2004; BELLINGHAM et al., 2003), apresentada na Figura 5.1a, uma rota de navegação é planejada por meio de um grafo de visibilidade e depois é convertida em uma trajetória através da aplicação de métodos de suavização, como por exemplo, as trajetórias de Dubins (DUBINS, 1957; CHITSAZ e LAVALLE, 2007; ANDERSON et al., 2005), as curvas de Bézier (NETO et al., 2010), ou *B-splines* (THOMAS et al., 2003). O problema desta abordagem é que, devido à distribuição espacial dos obstáculos, pode não ser possível a suavização e conversão da rota planejada em uma trajetória dinamicamente viável e livre de colisão. Exemplos deste problema são apresentados na Figura 5.2. Nesta figura, a rota de menor extensão é representada na cor azul. A trajetória é representada na cor vermelha. O grafo de visibilidade é representado pelos nós

e arestas na cor cinza. As circunferências cinza tracejadas são apenas estruturas auxiliares para a construção dos arcos das trajetórias. Através desta figura, observa-se que a trajetória obtida pelo processo de suavização da rota por curvas de Dubins apresenta situações de colisão para o VANT.

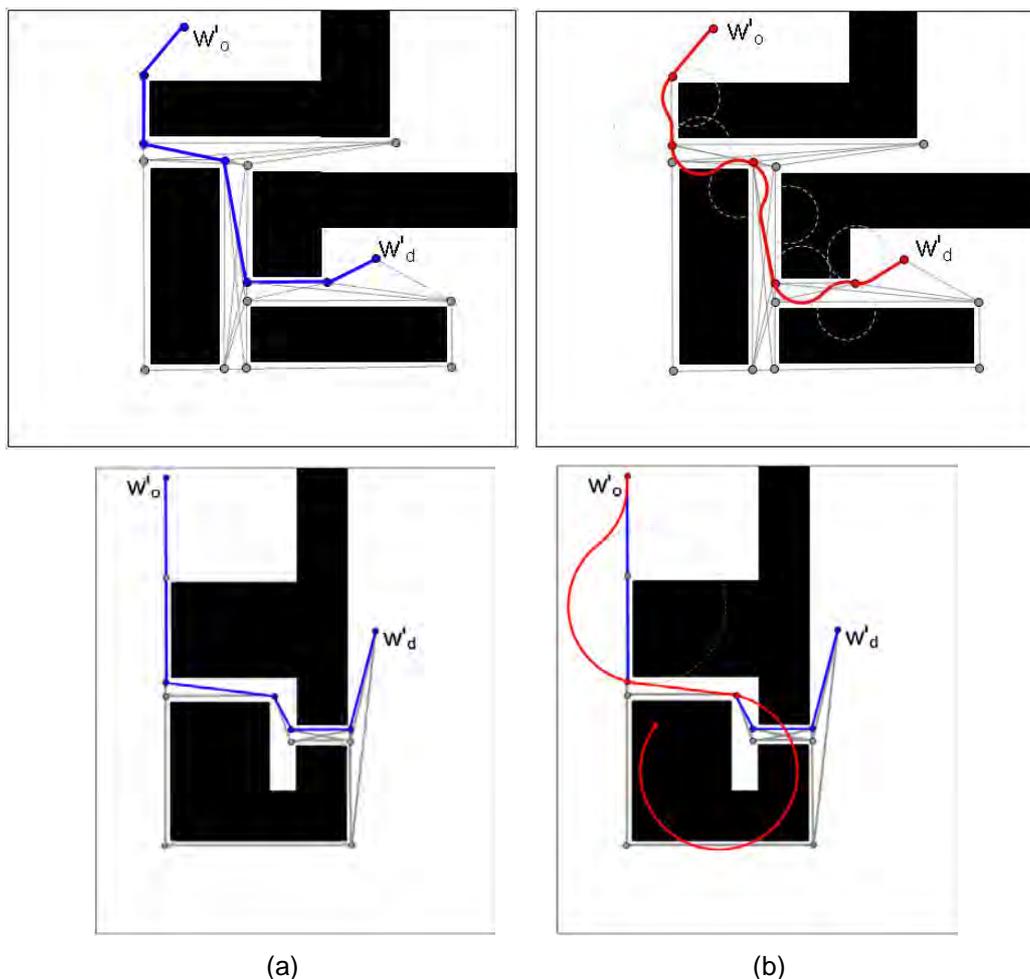


Figura 5.2 – Exemplos de tentativas de planejamento de (b) trajetórias através de suavizações de (a) rotas planejadas.

Em (LEE e KIM, 2007) foi proposta a segunda abordagem para a solução do problema e apresentada na Figura 5.1b: a associação do raio de curva ao grafo de visibilidade, que permite o planejamento direto de trajetória, sem a necessidade da aplicação de métodos de suavização após o planejamento. No Grafo de Visibilidade Modificado (GVM) proposto em (LEE e KIM, 2007) cada nó do grafo define duas circunferências fixas que são utilizadas para o cálculo

dos arcos de uma trajetória de navegação. O raio destas circunferências é o raio de curva da trajetória do VANT. O primeiro problema desta abordagem é o custo computacional exigido para verificação de colisões nas circunferências durante o processo de construção de um grafo de visibilidade ou de adaptação de um grafo já construído. Outro problema observado nesta abordagem é que um GVM só pode ser utilizado para o planejamento de trajetórias com o mesmo raio de curva utilizado em sua construção. Um exemplo de GVM é apresentado na Figura 5.3. Nesta figura, a trajetória planejada através do GVM é representada na cor vermelha. As regiões não navegáveis são representadas pelos polígonos cinza B_1 e B_2 .

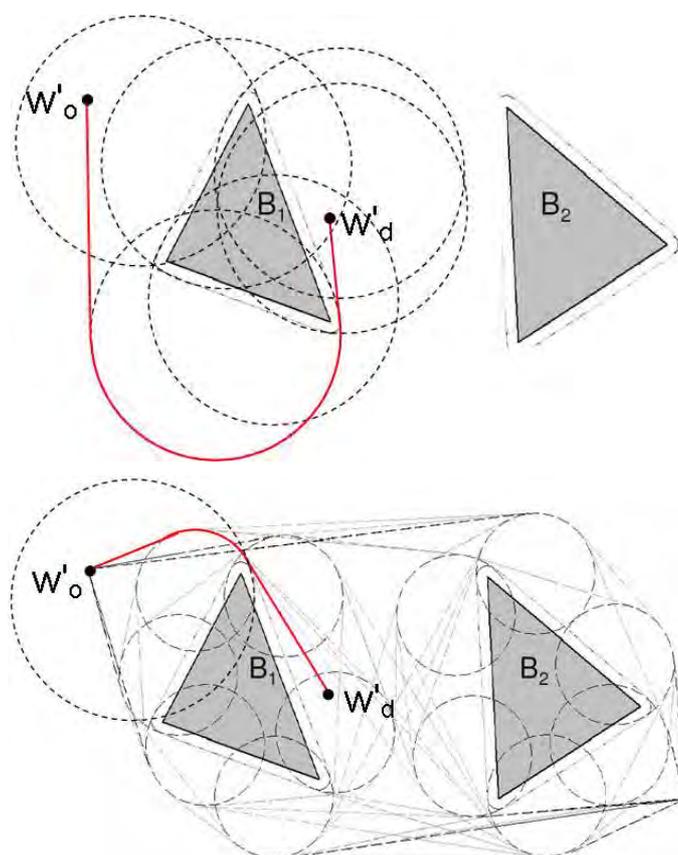


Figura 5.3 – Exemplos de trajetórias planejadas com um GVM

Fonte: adaptado de Lee e Kim 2007.

Um exemplo da terceira abordagem do problema é o Algoritmo Dijkstra Modificado (ADM) proposto em (KUWATA, 2003), que soluciona o problema do

planejamento da menor trajetória para VANTs baseado em grafos de visibilidade, sem a necessidade da aplicação de métodos de suavização após o planejamento. Nesta abordagem, o raio de curva é utilizado no processo de planejamento a partir de um grafo de visibilidade clássico e, assim, o resultado é uma trajetória dinamicamente viável e livre de colisão. Devido às verificações de colisão nos arcos de trajetória, o planejamento de trajetórias associado à restrição cinemática apresenta um custo computacional maior que os processos de planejamento das outras abordagens. Entretanto, além de não apresentar o problema que ocorre na primeira abordagem, a terceira abordagem permite o planejamento de trajetórias através de um mesmo grafo de visibilidade clássico, sendo que, para cada trajetória, é utilizado um diferente raio de curva.

O ADM é fundamentado no problema de menor caminho com destino único. Assim, o resultado da aplicação do ADM é uma árvore com as menores trajetórias entre o nó de destino e os demais nós do grafo (KUWATA e HOW, 2004). No ADM, os nós e arestas são criados através de representações retangulares das regiões não navegáveis ou obstáculos de um ambiente de navegação. No pior caso, o ADM possui complexidade computacional $O(n_o n_a \log(n_n))$, onde n_o é o número de obstáculos, considerando a implementação do grafo como uma lista de adjacência e utilizando busca binária para o cálculo do nó com menor custo.

A complexidade computacional do método de verificação de colisões do ADM é $O(n_o)$, no pior caso. Este método consiste em verificar todo o conjunto de obstáculos em busca de algum obstáculo que possa interceptar um arco da trajetória sendo planejada. O problema é que ambientes de navegação definidos por modelos digitais de elevação apresentam elevado número de obstáculos retangulares. Visando à solução deste problema, foi proposto o Algoritmo Dijkstra baseado em Elevações do terreno (ADE), que é uma adaptação do ADM, tornando-o independente do número de obstáculos do ambiente. Inicialmente, o ADE foi projetado para o planejamento de trajetórias

livres de colisão para VANTs de asa fixa através da representação de um ambiente de navegação por grafos de visibilidade e modelos digitais de elevação. Um estudo relacionado ao algoritmo ADE foi publicado em (MEDEIROS e SILVA, 2010b).

O ADE apresenta três principais distinções em relação ao ADM. A primeira distinção é que o ADE soluciona o problema do planejamento da trajetória entre duas coordenadas e o ADM soluciona o problema da trajetória entre a coordenada de destino e os demais nós do grafo de visibilidade. A segunda distinção é que o ADE planeja trajetórias da classe k -trajetórias proposta em (ANDERSON et al., 2005). A última distinção é que uma heurística de busca local baseada em grade regular foi utilizada para verificar situações de colisão nos arcos das trajetórias planejadas pelo ADE (MEDEIROS e SILVA, 2010b). Esta heurística é invariante ao número de obstáculos do ambiente de navegação e apresenta complexidade computacional $O(n_p)$ no pior caso, onde n_p é o número de verificações. Considerando ambientes de navegação com elevado número de obstáculos, esta heurística permite um decréscimo significativo do tempo computacional em comparação com o ADM.

5.2. Algoritmo Dijkstra baseado em elevações de terreno

Como mencionado anteriormente, o ADE soluciona o problema de planejamento de k -trajetórias para VANTs entre duas coordenadas do ambiente de navegação. Este algoritmo fundamenta-se em grafos de visibilidade e modelos digitais de elevação.

Em (ANDERSON et al., 2005) é proposta a classe k -trajetórias baseada em restrições cinemáticas de um VANT. Uma k -trajetória é uma curva de Dubins cujo comprimento e cuja forma podem ser ajustados pelo parâmetro k . Uma explicação detalhada das curvas de Dubins é apresentada no Apêndice C. Um exemplo de uma k -trajetória é apresentado na Figura 5.4. Nesta figura, a k -trajetória é representada em vermelho. Dada uma rota de navegação composta pelas coordenadas de navegação w_{i-1} , w_i e w_{i+1} , uma k -trajetória é a única

trajetória de menor extensão que permite a navegação da coordenada w_{i-1} até a coordenada w_{i+1} passando diretamente pelo ponto $p(k)$, calculada em função de w_i . Uma prova desta propriedade é mencionada em (ANDERSON et al., 2005). O ponto $p(k)$ é definido por

$$p(k) = w_i + kr_c \left(\frac{1}{\text{sen}\left(\frac{\beta}{2}\right)} - 1 \right) \vec{q} \quad (5.1)$$

Em que: c_i é o centro da circunferência que define $p(k)$; w_{ie1} e w_{ie2} são coordenadas de entrada dos arcos definidos pelas circunferências de centro c_{i-1} e c_i , respectivamente; w_{is1} e w_{is2} são coordenadas de saída dos arcos definidos pelas circunferências de centro c_i e c_{i+1} , respectivamente; \vec{q} é um vetor unitário.

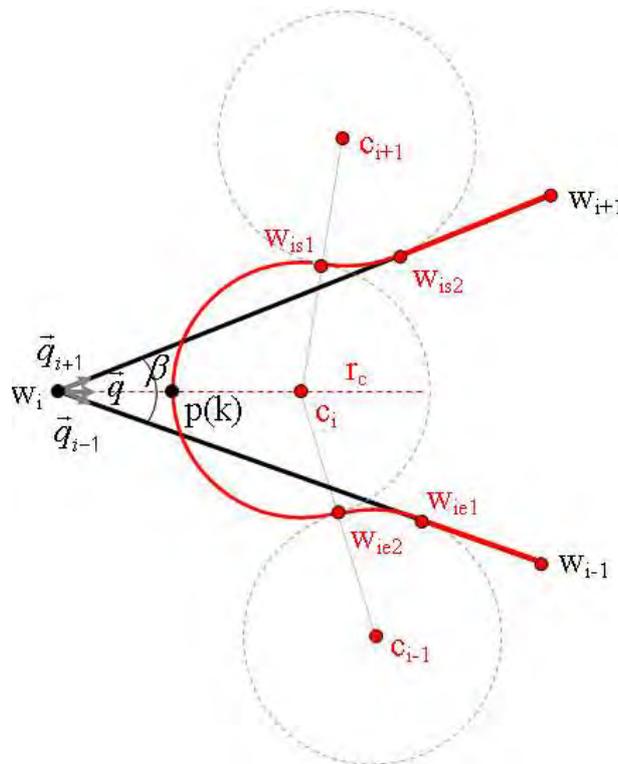


Figura 5.4 – Exemplo de uma k -trajetória.

O ADE planeja 1-trajetórias, embora a idéia inicial fosse utilizar diferentes valores de k durante o planejamento. A 1-trajetória é a menor dentre as k -

trajetórias (ANDERSON et al., 2005) e um exemplo desta trajetória é apresentado na Figura 5.5.

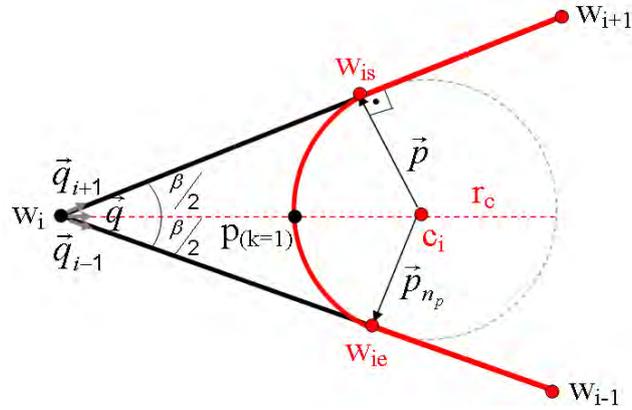


Figura 5.5 – Exemplo de uma 1-trajetória.

O ADE é apresentado na Tabela 5.1 e o ADM é apresentado na Tabela 5.2. Nestes algoritmos, são consideradas as seguintes definições: a é o nó avaliado; v é um nó vizinho do nó a , sendo que os nós v e a são conectados por uma aresta; $f(a,v)$ é a função de custo, que calcula o custo da rota entre o nó ou coordenada de origem (w'_o) e o nó v ; $h_{cs}(a_e, a_s, E)$ é a heurística utilizada no ADE para verificar se um arco entre a_e e a_s é livre de situação de colisão, através de análises da matriz de elevações E ; $d'(a)$ é o custo entre o nó a e o nó w'_o ; $d(a,v)$ é o custo da aresta que conecta o nó a e o nó v ; a_e e a_s são respectivamente as coordenadas de entrada e de saída do arco definido pelo raio de curva r_c e pelos segmentos de reta $\overline{p_r(a)a}$ e \overline{av} , como apresentado na Figura 5.6; $p_r(a)$ é o nó anterior ao nó a e vizinho do mesmo; $p_r(a)_s$ é a coordenada de saída do arco definido pelos segmentos de reta $\overline{p_r(p_r(a))p_r(a)}$ e $\overline{p_r(a)a}$; c_a é o centro da circunferência que intercepta o nó a , no ADM; c_{ae} é o centro da circunferência que define a coordenada de entrada a_e no ADM; c_{as} é o centro da circunferência que define a coordenada de saída a_s no ADM; T é a trajetória planejada entre a coordenada w'_o e a coordenada de destino (w'_d). O custo de uma aresta é o comprimento da trajetória definida pelos dois nós conectados pela aresta. No ADE, quando o nó analisado a é igual a w'_o , $p_r(p_r(a))$, $p_r(a)$, $p_r(a)_s$, a_e , a_s são considerados iguais a a .

Tabela 5.1 – Algoritmo Dijkstra baseado em elevações para planejamento de 1-trajetórias entre dois nós de um grafo.

Índice	Algoritmo Dijkstra Baseado em Elevações (ADE)
1	para i_n variando de 1 até n_n faça
2	armazenar w'_{i_n} em Q
3	$d'(w'_{i_n}) \leftarrow \begin{cases} 0, & \text{para } w'_{i_n} = w'_o \\ \infty, & \text{para } w'_{i_n} \neq w'_o \end{cases}$
4	$a \leftarrow w'_o$
5	enquanto $a \neq w'_d$ faça
6	remover a de Q
7	para cada vizinho v de a faça
8	$f(a,v) \leftarrow d'(a) + d(a,v)$
9	determinar a_e, a_s e $p_r(a)_s$ através de $p_r(p_r(a)), p_r(a), a$ e v
10	se $d(a,a_s) > d(a,v)$ ou $d(a,a_e) > d(a,p_r(a)_s)$ ou $d(a,a_e) > d(a,p_r(a))$ então faça
11	$f(a,v) \leftarrow \infty$
12	senão faça
13	se $h_{cs}(a_e, a_s, E) = 1$ então faça
14	$f(a,v) \leftarrow \infty$
15	se $f(a,v) < d'(v)$ faça
16	$d'(v) \leftarrow f(a,v)$
17	$p_r(v) \leftarrow a$
18	$a \leftarrow w'_{i_n}$, tal que $d'(w'_{i_n})$ é o mínimo ($d'(w'_{i_n})$), para todo $w'_{i_n} \in Q$
19	enquanto $a \neq w'_d$ faça
20	armazenar a na pilha T
21	$a \leftarrow p_r(a)$

Tabela 5.2 – Algoritmo Dijkstra Modificado.

Índice	Algoritmo Dijkstra Modificado (ADM)
1	para i_n variando de 1 até n_n faça
2	armazenar w'_{i_n} em Q
3	$d'(w'_{i_n}) \leftarrow \begin{cases} 0, & \text{para } w'_{i_n} = w'_d \\ \infty, & \text{para } w'_{i_n} \neq w'_d \end{cases}$
4	$a \leftarrow w'_d$
5	colocar a_e em a
6	enquanto $Q \neq \emptyset$ faça
7	$a \leftarrow w'_{i_n}$, tal que $d'(w'_{i_n})$ é o mínimo ($d'(w'_{i_n})$), para todo $w'_{i_n} \in Q$
8	posicionar duas circunferências de centro c_a e c_{a_s} em torno do nó a
9	remover a de Q
10	para cada vizinho v de a faça
11	posicionar uma circunferência de centro c_{ae} em torno de a
12	se não é possível colocar a_e no segmento de reta ou aresta av faça
13	$f(a,v) \leftarrow \infty$
14	senão faça
15	através de pesquisa no conjunto de todos os obstáculos, se os arcos entre a_e e a_s não são livres de colisão faça
16	$f(a,v) \leftarrow \infty$
17	se $f(a,v) < d'(v)$ faça
18	$d'(v) \leftarrow f(a,v)$
19	$p_r(v) \leftarrow a$

Na Figura 5.6, w'_{i_n} , w'_{j_n} e w'_{l_n} são nós de um grafo, tal que $i_n \neq j_n \neq l_n$. Uma 1-trajetória planejada pelo ADE é composta por: um segmento de reta $\overline{w_1 w_{2e}}$; um arco de w_{2e} a w_{2s} ; uma sequência de segmentos de reta $\overline{w_{(i-1)s} w_{ie}}$ conectados por arcos de w_{ie} a w_{is} , para i variando de 3 até $n_t - 1$; e um segmento de reta $\overline{w_{(n_t-1)s} w_{n_t}}$, onde n_t é o número de elementos da trajetória. É importante ser observado que $w_1 = w'_o$, $w_{n_t} = w'_d$ e as demais coordenadas da trajetória são coordenadas de entrada e de saída obtidas através de coordenadas w'_{i_n} do grafo de visibilidade. Como pode ser verificado na Figura 5.7, observa-se que as 1-trajetórias planejadas pelo ADE possuem menor extensão que as trajetórias planejadas pelo ADM para uma mesma sequência de nós de um grafo.

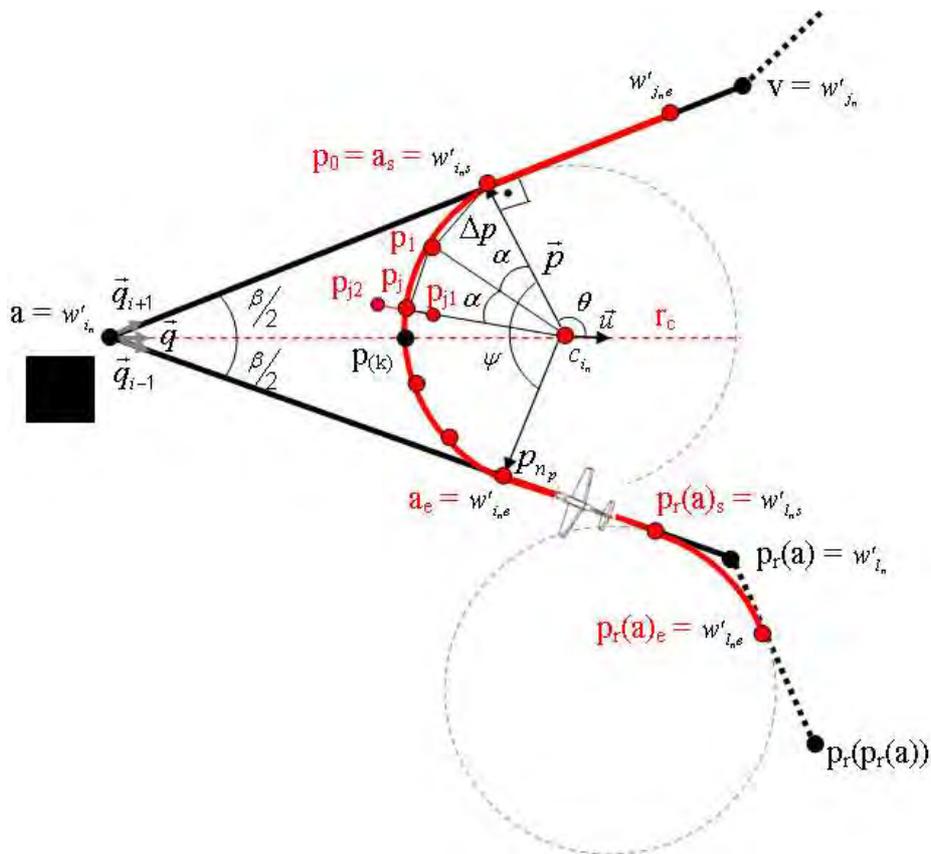


Figura 5.6 – Exemplo de parte de uma 1-trajetória planejada pelo ADE.

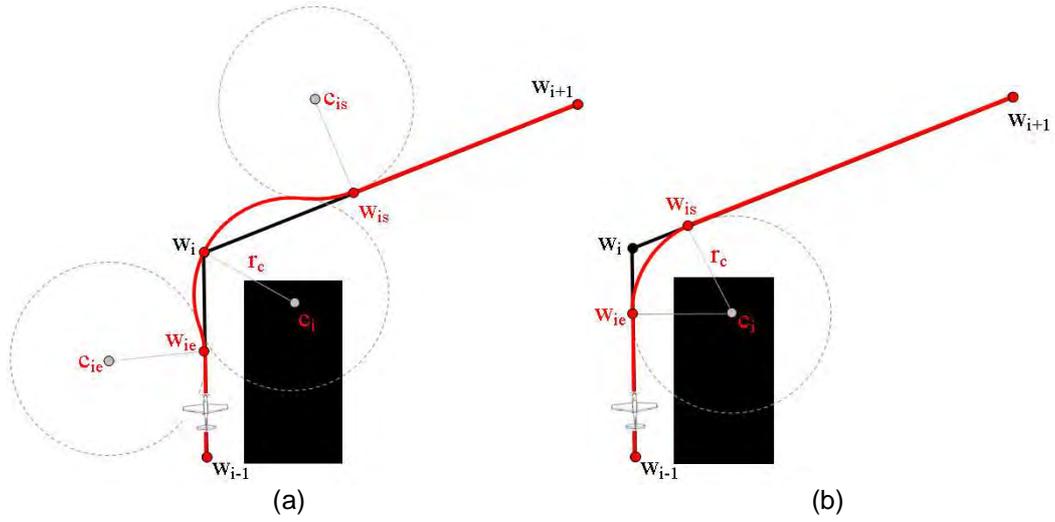


Figura 5.7 – Comparação entre (a) trajetória planejada pelo ADM e (b) 1-trajetória planejada pelo ADE.

Devido à restrição do raio de curva, as coordenadas w_{ie} e w_{is} não devem violar as restrições impostas pelas Equações 5.2, 5.3, 5.4 e 5.5. Estas restrições garantem a conexão entre os segmentos de uma 1-trajetória para i variando de 1 até n_t . Estas restrições estão asseguradas na linha 10 do ADE, como apresentado na Tabela 5.1.

$$d(w_i, w_{ie}) \leq d(w_i, w_{(i-1)_b}), \forall i \quad (5.2)$$

$$d(w_i, w_{ie}) \leq d(w_i, w_{(i-1)_s}), \forall i \quad (5.3)$$

$$d(w_i, w_{is}) \leq d(w_i, w_{(i+1)_e}), \forall i \quad (5.4)$$

$$d(w_i, w_{is}) \leq d(w_i, w_{(i+1)_b}), \forall i \quad (5.5)$$

Em que: d é a distância entre duas coordenadas.

Cada $\overline{w_{(i-1)_s}w_{ie}}$ é um segmento de reta da aresta que conecta os nós $w'_{j_{(i-1)}}$ e w'_{j_i} do grafo de visibilidade, correspondentes a w_{i-1} e w_i , para $j_{(i-1)} \neq j_i$, $1 \leq j_{(i-1)} \leq n_n$ e $1 \leq j_i \leq n_n$. Este fato garante que os segmentos de reta de uma

trajetória planejada pelo ADE são sempre livres de situações de colisão, como explicado no Capítulo 2.

Entretanto, o grafo de visibilidade não assegura que os arcos de uma trajetória sejam livres de colisão. Portanto, foi necessário utilizar uma heurística (h_{cs}) para verificar se os arcos de uma trajetória são livres de colisão. Analisando a Tabela 5.1, observa-se que a linha 13 do ADE corresponde à aplicação desta heurística.

Assim, o problema de planejamento de menor 1-trajetória para VANTs é formulado neste trabalho por

Minimizar

$$\sum_{i_n=1}^{n_n} \sum_{\substack{j_n=1 \\ j_n \neq i_n}}^{n_n} y'_{i_n j_n} x'_{i_n j_n} \quad (5.6)$$

Sujeito a

$$\sum_{\substack{j_n=1 \\ j_n \neq i_n}}^{n_n} x'_{i_n j_n} - \sum_{\substack{j_n=1 \\ j_n \neq i_n}}^{n_n} x'_{j_n i_n} = \begin{cases} 1, & \text{se } w'_{i_n} = w'_o \\ -1, & \text{se } w'_{i_n} = w'_d \\ 0, & \text{caso contrário} \end{cases} \quad \forall i_n \quad (5.7)$$

$$x'_{i_n j_n} = \begin{cases} 1, & \text{se a conexão entre } w'_{i_n} \text{ e } w'_{j_n}, \text{ incluindo arcos e} \\ & \text{segmentos de reta, pertence ao caminho} \\ 0, & \text{se não pertencer} \end{cases} \quad (5.8)$$

$$w'_{i_n e} - \left(w'_{i_n} + \frac{r_c \bar{q}_{j_n i_n}}{\text{tg}(\beta/2)} \right) = \bar{0}, \quad \forall i_n \neq j_n / x'_{j_n i_n} - x'_{i_n j_n} = 1 \quad (5.9)$$

$$w'_{i_n s} - \left(w'_{i_n} + \frac{r_c \bar{q}_{i_n j_n}}{\text{tg}(\beta/2)} \right) = \bar{0}, \quad \forall i_n \neq j_n / x'_{i_n j_n} - x'_{j_n i_n} = 1 \quad (5.10)$$

$$d(w'_{i_n}, w'_{i_n e}) - d(w'_{i_n}, w'_{j_n}) \leq 0, \quad \forall i_n \neq j_n / x'_{j_n i_n} - x'_{i_n j_n} = 1 \wedge w'_{i_n} \neq w'_o \wedge w'_{i_n} \neq w'_d, \quad \forall j_n \quad (5.11)$$

$$d(w'_{i_n}, w'_{i_n e}) - d(w'_{i_n}, w'_{j_n s}) \leq 0, \forall i_n \neq j_n / x'_{j_n i_n} - x'_{i_n j_n} = 1 \wedge w'_{i_n} \neq w'_o \wedge w'_{i_n} \neq w'_d, \forall j_n / w'_{j_n} \neq w'_o \wedge w'_{j_n} \neq w'_d \quad (5.12)$$

$$d(w'_{i_n}, w'_{i_n s}) - d(w'_{i_n}, w'_{j_n e}) \leq 0, \forall i_n \neq j_n / x'_{i_n j_n} - x'_{j_n i_n} = 1 \wedge w'_{i_n} \neq w'_o \wedge w'_{i_n} \neq w'_d, \forall j_n / w'_{j_n} \neq w'_o \wedge w'_{j_n} \neq w'_d \quad (5.13)$$

$$d(w'_{i_n}, w'_{i_n s}) - d(w'_{i_n}, w'_{j_n}) \leq 0, \forall i_n \neq j_n / x'_{i_n j_n} - x'_{j_n i_n} = 1 \wedge w'_{i_n} \neq w'_o \wedge w'_{i_n} \neq w'_d, \forall j_n \quad (5.14)$$

$$h(w'_{i_n e}, w'_{i_n s}, E) = 0, \forall x_{i_n j_n} = 1 \quad (5.15)$$

Em que: $\vec{q}_{j_n i_n}$ é o vetor unitário da aresta $\overline{w'_{i_n} w'_{j_n}}$, para $x'_{j_n i_n} - x'_{i_n j_n} = 1$; $\vec{q}_{i_n j_n}$ é o vetor unitário da aresta $\overline{w'_{i_n} w'_{j_n}}$, para $x'_{i_n j_n} - x'_{j_n i_n} = 1$; e $y'_{i_n j_n}$ é o comprimento da trajetória entre w'_{i_n} e w'_{j_n} . Se $w'_{i_n} = w'_o$, então $y'_{i_n j_n}$ é o somatório da extensão do segmento de reta $\overline{w'_o w'_{j_n e}}$ com a extensão do arco de $w'_{j_n e}$ a $w'_{j_n s}$. Se $w'_{i_n} \neq w'_o$, então $y'_{i_n j_n}$ é o somatório da extensão do segmento de reta $\overline{w'_{i_n s} w'_{j_n e}}$ com a extensão do arco de $w'_{j_n e}$ a $w'_{j_n s}$. Se $w'_{j_n} = w'_d$, então $y'_{i_n j_n}$ é a extensão do segmento de reta $\overline{w'_{i_n s} w'_d}$.

5.3. Heurística de busca local baseada em grades regulares

A primeira concepção do funcionamento da heurística era verificar a intersecção entre o arco de uma k -trajetória e o quadrilátero de células definido pelo arco, como apresentado na Figura 5.8. Entretanto, foi constatado que seria possível verificar somente as células sobrepostas pela área do arco de trajetória, como no algoritmo de verificação de colisão em arestas, apresentado no Capítulo 2. O objetivo inicial era utilizar a heurística para a verificação de colisões em diferentes tipos de k -trajetórias. Embora a heurística tenha sido projetada para a verificação de colisões em arcos de uma 1-trajetória, a mesma pode ser adaptada para outros valores de k .

A heurística utilizada neste trabalho baseia-se na técnica para verificação de colisões apresentada em (LAVALLE, 2006), que consiste na divisão da

trajetória de um robô em seções, a encapsulação de tais seções em polígonos e a verificação de quais obstáculos são interceptados pelos polígonos.

É importante ser mencionado que o exemplo da Figura 5.8 é apenas ilustrativo e não foi considerada a atitude correta do VANT baseada em seu ângulo de inclinação. Inclusive, neste trabalho, a heurística de verificação da colisão não considera a inclinação do VANT (λ_v) com o objetivo de aumentar a margem de segurança do mesmo em relação às regiões não navegáveis.

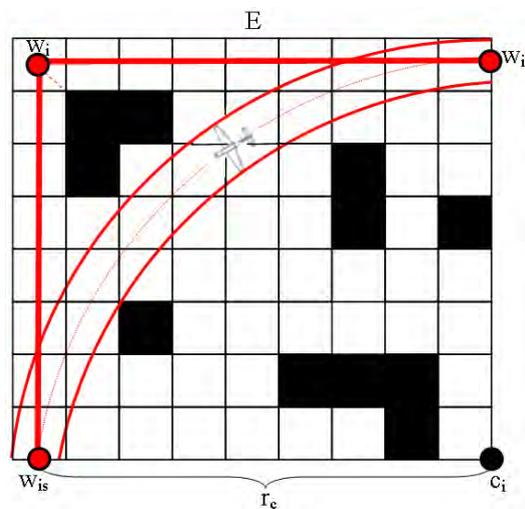


Figura 5.8 – Exemplo de verificação de colisão entre o arco de trajetória do VANT e a matriz de elevações E .

A heurística verifica se um VANT navegando por um arco intercepta alguma célula não navegável $e_{ic} \geq (a_n - a_s)$, através de uma busca local em E . Embora possa ser aplicado diretamente à grade B , representando qualquer tipo de região não navegável. Como apresentado na Figura 5.6, a heurística divide o arco de a_e a a_s em uma sequência de seções definidas por pontos p_j , para j variando de 1 até n_p , onde o número de iterações n_p é definido por

$$n_p = \frac{\psi}{\alpha} \quad (5.16)$$

Em que: ψ é o ângulo formado pelos segmentos de reta $\overline{c_i p_0}$ e $\overline{c_i p_{n_p}}$; e α é o ângulo utilizado para a divisão da trajetória em seções.

Analisando a Figura 5.9, a cada iteração j , a heurística verifica se o polígono, neste caso um pentágono, que contém a seção de trajetória do VANT entre os pontos p_{j-1} e p_j intercepta alguma célula não navegável b_{lc} , com $e_{lc} \geq (a_n - a_s)$. Na Figura 5.9b, l_{j-1} é igual a $(l_j - 1)$ e c_{j-1} é igual a $(c_j + 1)$. A heurística é apresentada na Tabela 5.3.

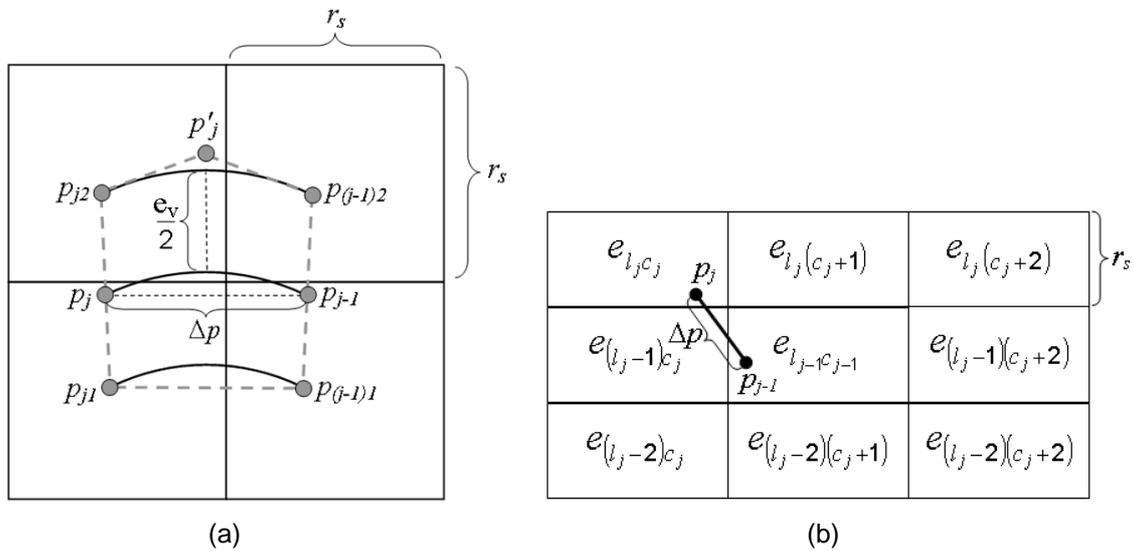


Figura 5.9 – (a) Verificação de colisão através de um polígono que contém a trajetória do VANT entre os pontos p_{j-1} e p_j com (b) busca local na matriz de elevações E .

Fonte: Medeiros e Silva (2010b)

Cada ponto p_j é definido por

$$p_j = \begin{cases} c_{i_n} + \begin{bmatrix} r_c \cos(\theta + j\alpha) \\ r_c \sin(\theta + j\alpha) \end{bmatrix}, & \text{se } c_{i_n}, p_0 \text{ e } w_i \text{ está no sentido anti-horário} \\ c_{i_n} + \begin{bmatrix} r_c \cos(\theta - j\alpha) \\ r_c \sin(\theta - j\alpha) \end{bmatrix}, & \text{se } c_{i_n}, p_0 \text{ e } w_i \text{ está no sentido horário} \end{cases} \quad (5.17)$$

Em que: θ é o ângulo de rotação composto pelo vetor \vec{p} e pelo vetor unitário \vec{u} ; e \vec{p} é o vetor definido pelo ponto central c e pelo ponto p_0 .

O ângulo α está associado com a distância Δp entre p_j e p_{j+1} . A distância Δp é baseada na distância entre dois estados de um VANT. Como pode ser observado através da Figura 5.9, a heurística elaborada neste trabalho avalia somente as células sobrepostas pelo arco de trajetória no processo de verificação de colisão.

Tabela 5.3 – Heurística para verificar se um arco de a_e até as é livre de colisão.

Índice	Heurística de Busca Local Baseada em Grade Regular (h_{cs})
1	calcular n_p
2	$colisão \leftarrow 0$
3	$j \leftarrow 1$
4	enquanto $j < n_p$ e $colisão = 0$ faça
5	calcular p_j, p_{j1}, p_{j2} e p'_j
6	através do lema proposto, se o polígono definido por $\overline{p_{(j-1)1}p_{(j-1)2}}, \overline{p_{(j-1)1}p_{j1}}, \overline{p_{(j-1)2}p'_j}, \overline{p_{j1}p_{j2}}$ e $\overline{p_{j2}p'_j}$ intercepta alguma célula não navegável b_{lc} então faça
7	$colisão \leftarrow 1$
8	senão faça
9	$j \leftarrow j + 1$
10	retornar $colisão$

O problema desta abordagem com discretização é garantir que toda situação de colisão possa ser detectada pelo processo de discretização de um arco de trajetória. Deste modo, neste trabalho, foi proposto um lema para demonstrar que a discretização proposta sempre permite a detecção de qualquer situação de colisão ao longo do arco da trajetória. A prova do lema proposto pode ser compreendida através da Figura 5.9 e da Figura 5.10. Na Figura 5.10, o quadrado preto representa uma célula da grade B ou da matriz de elevações E . Os arcos vermelhos indicam os arcos definidos pelas extremidades da asa do VANT.

Lema 3 – Considerando o ambiente de navegação representado por uma grade regular B , $e_v \leq \Delta p < r_s$, $c_v \leq \Delta p$, $\overline{p_{j1}p_{j2}} = e_v$, $\overline{p_{j1}p_j} = (e_v/2)$ e p'_j sendo a intersecção entre as tangentes do arco definidas por $p_{(j-1)2}$ e $p_{j2} \forall j$, a discretização de um arco de navegação em n_p seções encapsuladas por pentágonos definidos por $p_{j-1}, p_{(j-1)1}, p_{(j-1)2}, p_j, p_{j1}, p_{j2}$ e p'_j , permite a verificação de toda possível situação de colisão ao longo deste arco, se e somente se o

segmento de reta $\overline{p_{(j-1)}p_j}$ e todos segmentos de reta que definem os lados de tais pentágonos forem livres de colisão.

Prova 3 – Como $\Delta p < r_s$ e r_s é o menor lado da célula, aplicando então o lema 2 descrito no Capítulo 2 ao segmento de reta $\overline{p_{(j-1)}p_j}$ e aos segmentos de reta que compõem o pentágono: $\overline{p_{(j-1)1}p_{(j-1)2}}$; $\overline{p_{j1}p_{j2}}$; $\overline{p_{(j-1)1}p_{j1}}$; $\overline{p_{(j-1)2}p'_{j2}}$; e $\overline{p_{j2}p'_{j2}}$, pode-se verificar qualquer situação de colisão com tais segmentos de reta. Portanto, a única possível situação de colisão não detectada seria a existência de uma célula não navegável contida no pentágono e não interceptada por qualquer lado deste pentágono. Entretanto, não é possível existir uma célula contida no pentágono e que não seja verificada. A prova deste argumento fundamenta-se no fato que o segmento de reta $\overline{p_{(j-1)}p_j}$ divide o pentágono em duas regiões, um trapézio R_t e um pentágono R_p , e não é possível uma célula estar contida nestas regiões porque a área da célula (A_c) é superior à área de cada uma delas, A_t e A_p .

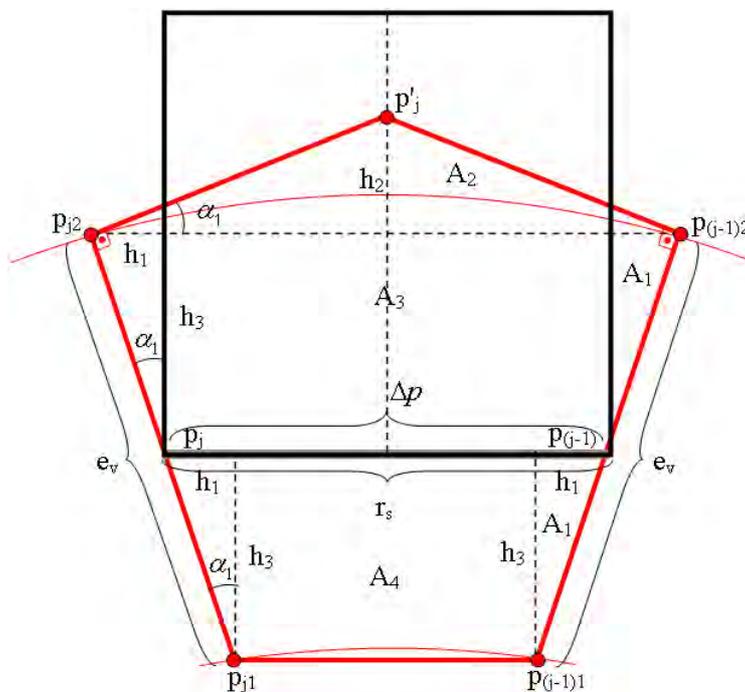


Figura 5.10 – Esquema para cálculo da área do pentágono que contém a trajetória do VANT entre os pontos p_{j-1} e p_j .

Analisando a Figura 5.10, verifica-se que

$$A_c = r_s^2 \quad (5.18)$$

$$A_t = 2A_1 + A_4 \quad (5.19)$$

$$A_p = 2A_1 + 2A_2 + A_3 \quad (5.20)$$

$$A_1 = \frac{h_1 h_3}{2} \quad (5.21)$$

$$A_2 = \frac{(\Delta p + 2h_1)h_2}{4} \quad (5.22)$$

$$A_3 = h_3 \Delta p \quad (5.23)$$

$$A_4 = (\Delta p - 2h_1)h_3 \quad (5.24)$$

$$h_1 = \left(\frac{e_v}{2}\right) \text{sen}(\alpha_1) \quad (5.25)$$

$$h_2 = \left(\frac{\Delta p + 2h_1}{2}\right) \text{tg}(\alpha_1) \quad (5.26)$$

$$h_3 = \left(\frac{e_v}{2}\right) \text{cos}(\alpha_1) \quad (5.27)$$

Realizando as devidas substituições, chega-se às equações para o cálculo de A_t e A_p .

$$A_t = \frac{e_v \text{cos}(\alpha_1)(2\Delta p - e_v \text{sen}(\alpha_1))}{4} \quad (5.28)$$

$$A_p = \frac{(\text{sen}(\alpha_1)(\Delta p^2 + e_v^2) + 2e_v \Delta p)}{4 \text{cos}(\alpha_1)} \quad (5.29)$$

Sabendo-se que $\alpha_1 = 18^\circ$ e que $e_v \leq \Delta p < r_s$, pode-se considerar que $e_v = \Delta p$, o que conduz a $A_t = 0.40206\Delta p^2$ e $A_p = 0.68819\Delta p^2$. Portanto, está provado que A_t e A_p são sempre menores que A_c , $\forall e_v \leq \Delta p < r_s$. Por conseqüência, prova-se

também que toda possível situação de colisão pode ser verificada com a discretização do arco de trajetória em pentágonos com as características mencionadas e baseando-se nas restrições consideradas.

A fim de associar a discretização com Δp e com o raio de curva, o ângulo α é definido por

$$\alpha = 2 \arcsin \left(\frac{\Delta p}{2r_c} \right) \quad (5.30)$$

Como apresentado na Figura 5.11, quando um VANT navega pelo arco de trajetória, a projeção da envergadura do VANT é igual a $e_v \cos(\lambda_v)$, onde λ_v é o ângulo de inclinação do VANT.

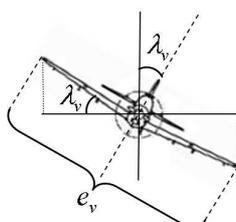


Figura 5.11 – Projeção da envergadura de um VANT.

Entretanto, como mencionado anteriormente, considerou-se neste trabalho que a projeção é a própria envergadura e_v para aumentar a margem de segurança do VANT.

5.4. Experimentos com o ADE

O ADE foi aplicado ao planejamento de trajetórias considerando os conjuntos de grafos de visibilidade construídos no Capítulo 2 e no Capítulo 4, como apresentado anteriormente. Entretanto, neste capítulo, serão analisados apenas os resultados obtidos com os grafos de visibilidade do Capítulo 2.

Inicialmente, um conjunto de especificações foi considerado visando à aplicação do ADE. Analisando as Equações 5.16 e 5.30, pode-se observar que

o valor de n_p decai quando o valor de Δp aumenta. Devido a este fato e sabendo-se que $\Delta p < r_s$, o valor de Δp foi especificado como $0.99r_s$ na aplicação do algoritmo. Analisando a Figura 5.6, observa-se que $0^\circ < \psi < 180^\circ$. Então, considerou-se que no pior caso da aplicação da heurística, ψ é igual a 179.99° e a cada iteração j desta heurística, três células são avaliadas para cada um dos lados do pentágono. Portanto, no pior caso, como o segmento de reta $\overline{p_{(j-1)1}p_{(j-1)2}}$ é avaliado na iteração anterior, o número de células avaliadas pela heurística é definido por

$$n_c = 7n_p \quad (5.31)$$

Através de uma análise do ADE e da heurística para detecção de colisões, é possível constatar que a complexidade computacional do ADE é $O(n_p n_a \log(n_n))$ no pior caso, considerando a implementação do grafo como uma lista de adjacência e utilizando busca binária para o cálculo do nó com menor custo. A Tabela 5.4 apresenta uma comparação, no pior caso, entre as aplicações diretas da heurística utilizada no ADE e do método utilizado no ADM para detectar situações de colisão em um arco, considerando $r_c = 300\text{m}$, $r_s = 30\text{ m}$, $\Delta p = 0.99r_s$ e $\psi = 179.99^\circ$ e os ambientes de navegação da Figura 5.12.

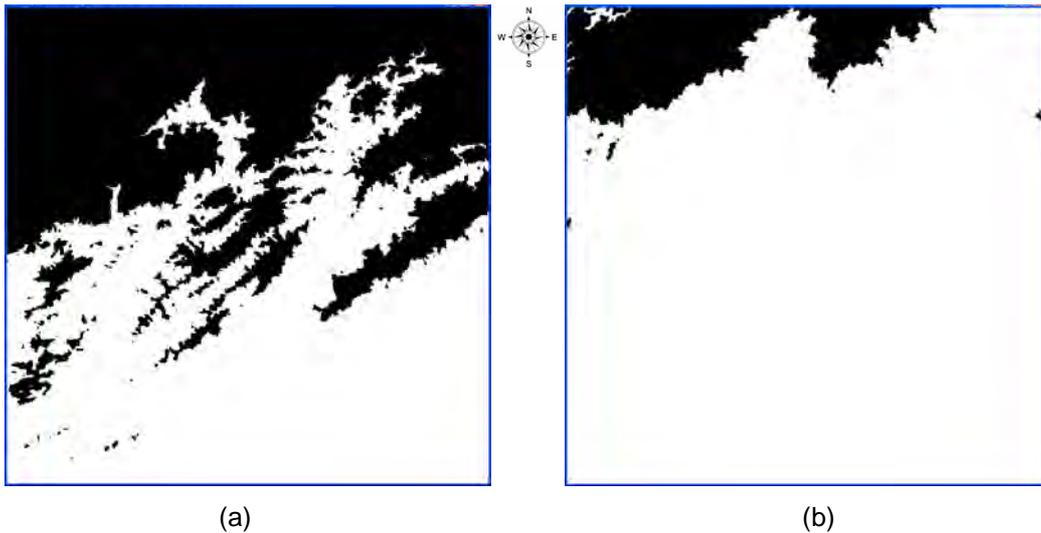


Figura 5.12 – Ambientes de navegação obtidos com $a_s = 300\text{ m}$, $d_s = 100\text{ m}$, (a) $a_n = 1100\text{ m}$ e (b) $a_n = 1700\text{ m}$.

Tabela 5.4 – Comparação entre a heurística baseada em discretização do arco de trajetória e o método usado no ADM.

Ambientes de Navegação	Heurística Utilizada $O(n_p)$	Método Usado no ADM $O(n_o)$
Figura 5.12a	$n_p = 32$ iterações, $n_c = 224$ células	$n_o = 724548$ células
Figura 5.12b	$n_p = 32$ iterações, $n_c = 224$ células	$n_o = 159478$ células

A heurística também foi comparada com a heurística baseada na concepção inicial da verificação de colisões: a avaliação da intersecção entre os arcos que delimitam o arco de trajetória e as células do quadrilátero definido por este arco de trajetória, como apresentado na Figura 5.8. A complexidade computacional da heurística baseada em avaliação do quadrilátero é $O(n_{cq})$, onde n_{cq} é o número de células do quadrilátero. A Tabela 5.5 apresenta uma comparação entre as duas heurísticas considerando um intervalo de valores para r_c e $r_s = 90$ m, $\psi = 179.99^\circ$ e $\Delta p = 0.99r_s$. As informações apresentadas na Tabela 5.5 são exibidas graficamente na Figura 5.13. As comparações são feitas em termos do número de verificações de colisão efetuadas por cada heurística no pior caso. O número de verificações da primeira heurística é calculado pela Equação 5.31 e o número de verificações da outra heurística é definido por

$$n_{vc} = 8 \left(\frac{r_c}{r_s} \right)^2 \quad (5.32)$$

Na Equação 5.32, o número oito corresponde ao resultado da multiplicação do número de arcos que delimitam o arco de trajetória pelo número de lados de uma célula. De modo similar, a Tabela 5.6 apresenta outra comparação entre as duas heurísticas considerando um intervalo de valores para r_s e $r_c = 200$ m, $\psi = 179.99^\circ$ e $\Delta p = 0.99r_s$. As informações apresentadas na Tabela 5.6 são exibidas graficamente na Figura 5.14.

Através dos resultados apresentados na Tabela 5.5, pode-se constatar que a abordagem com discretização do arco é mais eficiente que a abordagem com avaliação do quadrilátero para valores de r_c superiores a 120 m.

Tabela 5.5 – Comparação entre a heurística baseada em discretização do arco de trajetória e a heurística baseada em avaliação do quadrilátero de células, considerando a variação do raio de curva r_c .

r_c (m)	n_p	n_{cq}	Número de Verificações de Colisão	
			Discretização de Arco	Avaliação de Quadrilátero
50	0.714216	0.308642	4.999509785	2.469135802
60	0.938475	0.444444	6.569323515	3.555555556
70	1.138364	0.604938	7.968548384	4.839506173
80	1.329641	0.790123	9.307488056	6.320987654
90	1.516528	1.0	10.61569432	8.0
100	1.700791	1.234568	11.90553902	9.87654321
110	1.883335	1.493827	13.18334619	11.95061728
120	2.064679	1.777778	14.45275644	14.22222222
130	2.245149	2.08642	15.71604064	16.69135802
140	2.424957	2.419753	16.97470029	19.35802469
150	2.604253	2.777778	18.22977339	22.22222222
160	2.783143	3.160494	19.48200326	25.28395062
170	2.961705	3.567901	20.7319377	28.54320988
180	3.139999	4.0	21.97999018	32.0
190	3.318068	4.45679	23.22647914	35.65432099
200	3.495951	4.938272	24.47165411	39.50617284
210	3.673673	5.444444	25.71571359	43.55555556
220	3.85126	5.975309	26.95881761	47.80246914
230	4.028728	6.530864	28.20109668	52.24691358
240	4.206094	7.111111	29.44265837	56.88888889
250	4.38337	7.716049	30.68359217	61.72839506
260	4.560568	8.345679	31.92397315	66.7654321
270	4.737695	9.0	33.16386481	72.0
280	4.91476	9.679012	34.4033212	77.43209877
290	5.09177	10.38272	35.64238865	83.0617284
300	5.26873	11.11111	36.8811071	88.88888889
310	5.445644	11.8642	38.11951117	94.91358025
320	5.622519	12.64198	39.35763103	101.1358025
330	5.799356	13.44444	40.59549308	107.5555556
340	5.97616	14.2716	41.83312054	114.1728395
350	6.152933	15.12346	43.07053392	120.9876543

Tabela 5.6 – Comparação entre a heurística baseada em discretização do arco de trajetória e a heurística baseada em avaliação do quadrilátero de células, considerando a variação da resolução r_s .

r_s (m)	Δp (m)	n_p	n_{cq}	Número de Verificações de Colisão	
				Discretização de Arco	Avaliação de Quadrilátero
300	297.0	0.938475	0.444444	6.569323515	3.555555556
290	287.1	0.980945	0.475624	6.866616063	3.804994055
280	277.2	1.025689	0.510204	7.179825705	4.081632653
270	267.3	1.073043	0.548697	7.511302387	4.38957476
260	257.4	1.123378	0.591716	7.86364599	4.733727811
250	247.5	1.177112	0.64	8.239783909	5.12
240	237.6	1.234723	0.694444	8.643059747	5.555555556
230	227.7	1.296763	0.756144	9.077339379	6.049149338
220	217.8	1.363877	0.826446	9.54714178	6.611570248
210	207.9	1.436829	0.907029	10.05780416	7.256235828
200	198.0	1.516528	1.0	10.61569432	8.0
190	188.1	1.60407	1.108033	11.22848827	8.864265928
180	178.2	1.700791	1.234568	11.90553902	9.87654321
170	168.3	1.808339	1.384083	12.65837427	11.07266436
160	158.4	1.928769	1.5625	13.50137998	12.5
150	148.5	2.064679	1.777778	14.45275644	14.22222222
140	138.6	2.219412	2.040816	15.53588359	16.32653061
130	128.7	2.397331	2.366864	16.78131574	18.93491124
120	118.8	2.604253	2.777778	18.22977339	22.22222222
110	108.9	2.848109	3.305785	19.9367663	26.44628099
100	99.0	3.139999	4.0	21.97999018	32.0
90	89.1	3.495951	4.938272	24.47165411	39.50617284
80	79.2	3.940008	6.25	27.58005307	50.0
70	69.3	4.509947	8.163265	31.5696311	65.30612245
60	59.4	5.26873	11.11111	36.8811071	88.88888889
50	49.5	6.329679	16.0	44.30775137	128.0
40	39.6	7.91944	25.0	55.43607655	200.0
30	29.7	10.56684	44.44444	73.96791004	355.5555556
20	19.8	15.85838	100.0	111.008683	800.0
10	9.9	31.72649	400.0	222.0854527	3200.0

Analisando os resultados apresentados na Tabela 5.6, verifica-se que a abordagem com discretização do arco é mais eficiente que a outra abordagem para valores de r_s inferiores a 150 m. É importante ser observado, nesta segunda análise, que quanto menor o valor de r_s , em outras palavras, quanto maior o detalhamento do modelo digital de elevação, mais eficiente é a abordagem por discretização de arco de trajetória.

Exemplos de 1-trajetórias planejadas pelo ADE são apresentados nas Figuras 5.15 a 5.22, considerando $d_s = 100$ m. As Tabelas 5.7 a 5.14 apresentam informações a respeito destas trajetórias planejadas. Nas figuras, as trajetórias são representadas na cor vermelha. Como explicado anteriormente, as

circunferências vermelhas tracejadas indicam os locais de entrada e de saída de arcos de trajetória definidos pelo raio de curva.

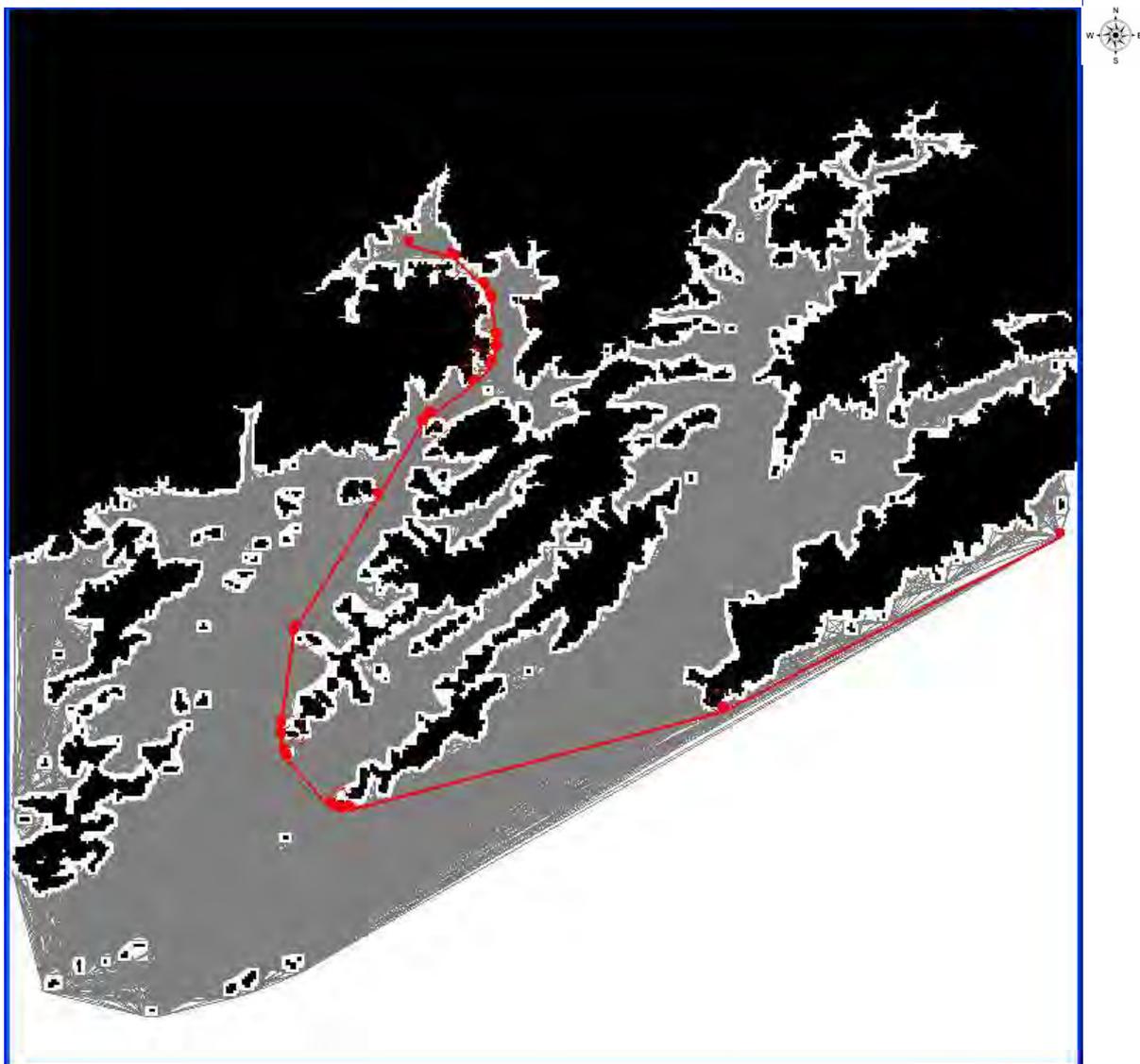


Figura 5.15 – Trajetória planejada pelo ADE entre as coordenadas $w'_o = (latitude_o, longitude_o) = (-22.91, -45.95)$ e $w'_d = (latitude_d, longitude_d) = (-23.0, -45.735)$ com um grafo de visibilidade de 1489 nós e 65241 arestas e com $r_c = 400$ m.

Tabela 5.7 – Informações referentes à 1-trajetória apresentada na Figura 5.15.

Extensão	52928.238 m
Tempo para planejamento	2.2s
r_c	400 m
n_t	36
latitude	longitude
-22.910000	-45.950000
-22.913574	-45.935926
-22.914420	-45.934369
-22.922631	-45.925499
-22.923474	-45.924827
-22.926474	-45.923098
-22.927732	-45.922659
-22.938435	-45.921046
-22.938970	-45.921006
-22.941624	-45.921006
-22.942784	-45.921198
-22.946547	-45.922482
-22.947963	-45.923378
-22.952828	-45.928383
-22.953223	-45.928867
-22.962639	-45.942711
-22.962973	-45.943132
-22.964917	-45.945231
-22.965610	-45.945812
-22.987934	-45.960165
-22.987992	-45.960204
-23.028692	-45.987501
-23.030191	-45.988075
-23.058087	-45.992018
-23.058590	-45.992053
-23.061593	-45.992053
-23.062116	-45.992015
-23.066643	-45.991348
-23.068649	-45.990345
-23.083259	-45.975857
-23.084322	-45.973306
-23.084322	-45.970562
-23.084217	-45.969701
-23.053826	-45.846522
-23.053572	-45.845819
-23.000000	-45.735000

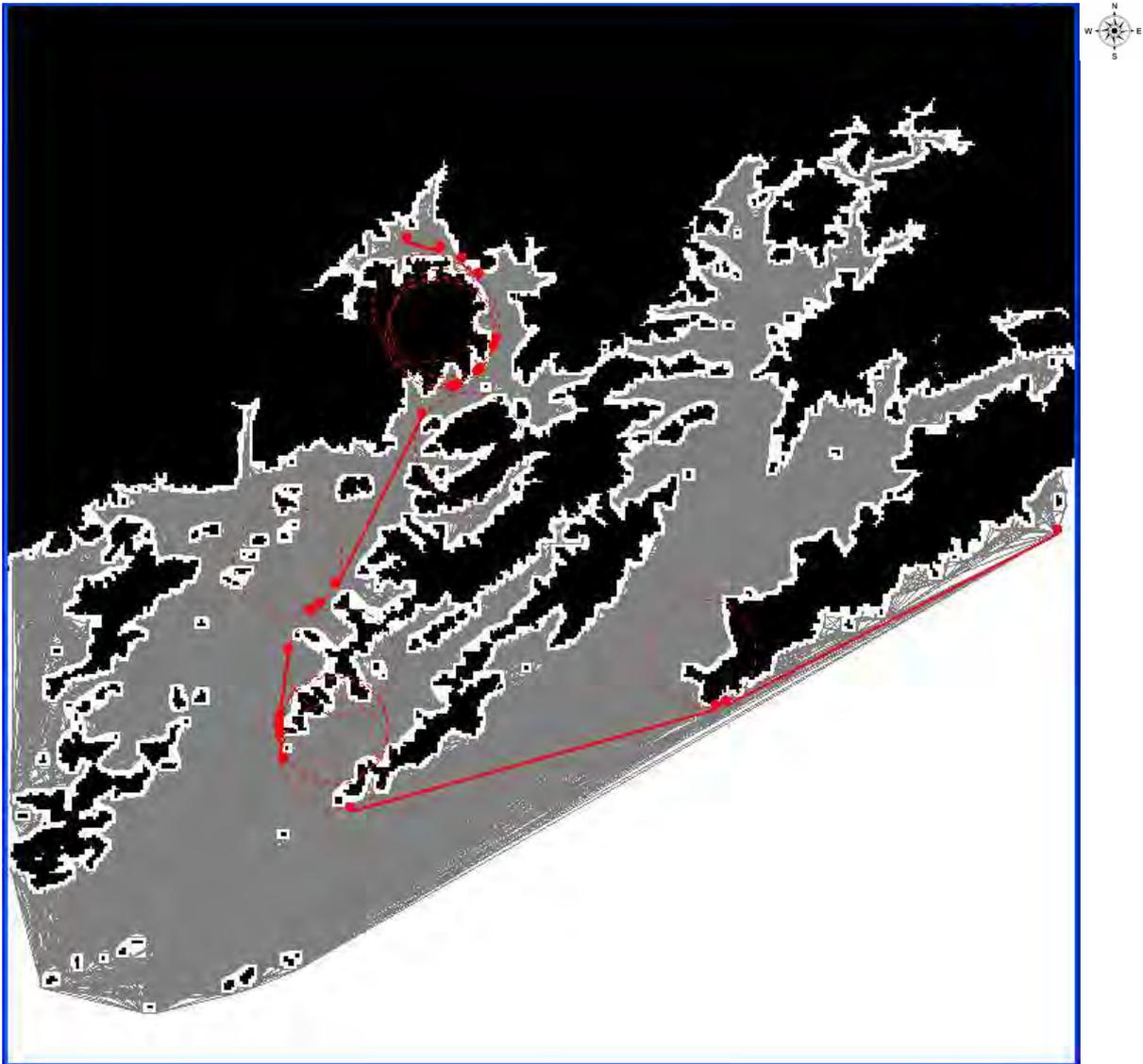


Figura 5.16 – Trajetória planejada pelo ADE entre as coordenadas $w'_o = (latitude_o, longitude_o) = (-22.91, -45.95)$ e $w'_d = (latitude_d, longitude_d) = (-23.0, -45.735)$ com um grafo de visibilidade de 1489 nós e 65241 arestas e com $r_c = 2000$ m.

Tabela 5.8 – Informações referentes à 1-trajetória apresentada na Figura 5.16.

Extensão	53700.483 m																																																		
Tempo para planejamento	2.3 s																																																		
r_c	2000 m																																																		
n_t	24																																																		
<table border="0"> <thead> <tr> <th>latitude</th> <th>longitude</th> </tr> </thead> <tbody> <tr><td>-22.910000</td><td>-45.950000</td></tr> <tr><td>-22.912828</td><td>-45.938862</td></tr> <tr><td>-22.916281</td><td>-45.931976</td></tr> <tr><td>-22.921001</td><td>-45.926149</td></tr> <tr><td>-22.940888</td><td>-45.920498</td></tr> <tr><td>-22.943333</td><td>-45.921352</td></tr> <tr><td>-22.950328</td><td>-45.925830</td></tr> <tr><td>-22.951139</td><td>-45.926669</td></tr> <tr><td>-22.955160</td><td>-45.933176</td></tr> <tr><td>-22.955890</td><td>-45.935244</td></tr> <tr><td>-22.964266</td><td>-45.945061</td></tr> <tr><td>-23.016626</td><td>-45.973464</td></tr> <tr><td>-23.022556</td><td>-45.978644</td></tr> <tr><td>-23.024898</td><td>-45.981843</td></tr> <tr><td>-23.036881</td><td>-45.989021</td></tr> <tr><td>-23.057086</td><td>-45.991876</td></tr> <tr><td>-23.059601</td><td>-45.992053</td></tr> <tr><td>-23.060682</td><td>-45.992053</td></tr> <tr><td>-23.063020</td><td>-45.991900</td></tr> <tr><td>-23.070631</td><td>-45.990902</td></tr> <tr><td>-23.085677</td><td>-45.968552</td></tr> <tr><td>-23.054180</td><td>-45.847858</td></tr> <tr><td>-23.052971</td><td>-45.844575</td></tr> <tr><td>-23.000000</td><td>-45.735000</td></tr> </tbody> </table>		latitude	longitude	-22.910000	-45.950000	-22.912828	-45.938862	-22.916281	-45.931976	-22.921001	-45.926149	-22.940888	-45.920498	-22.943333	-45.921352	-22.950328	-45.925830	-22.951139	-45.926669	-22.955160	-45.933176	-22.955890	-45.935244	-22.964266	-45.945061	-23.016626	-45.973464	-23.022556	-45.978644	-23.024898	-45.981843	-23.036881	-45.989021	-23.057086	-45.991876	-23.059601	-45.992053	-23.060682	-45.992053	-23.063020	-45.991900	-23.070631	-45.990902	-23.085677	-45.968552	-23.054180	-45.847858	-23.052971	-45.844575	-23.000000	-45.735000
latitude	longitude																																																		
-22.910000	-45.950000																																																		
-22.912828	-45.938862																																																		
-22.916281	-45.931976																																																		
-22.921001	-45.926149																																																		
-22.940888	-45.920498																																																		
-22.943333	-45.921352																																																		
-22.950328	-45.925830																																																		
-22.951139	-45.926669																																																		
-22.955160	-45.933176																																																		
-22.955890	-45.935244																																																		
-22.964266	-45.945061																																																		
-23.016626	-45.973464																																																		
-23.022556	-45.978644																																																		
-23.024898	-45.981843																																																		
-23.036881	-45.989021																																																		
-23.057086	-45.991876																																																		
-23.059601	-45.992053																																																		
-23.060682	-45.992053																																																		
-23.063020	-45.991900																																																		
-23.070631	-45.990902																																																		
-23.085677	-45.968552																																																		
-23.054180	-45.847858																																																		
-23.052971	-45.844575																																																		
-23.000000	-45.735000																																																		

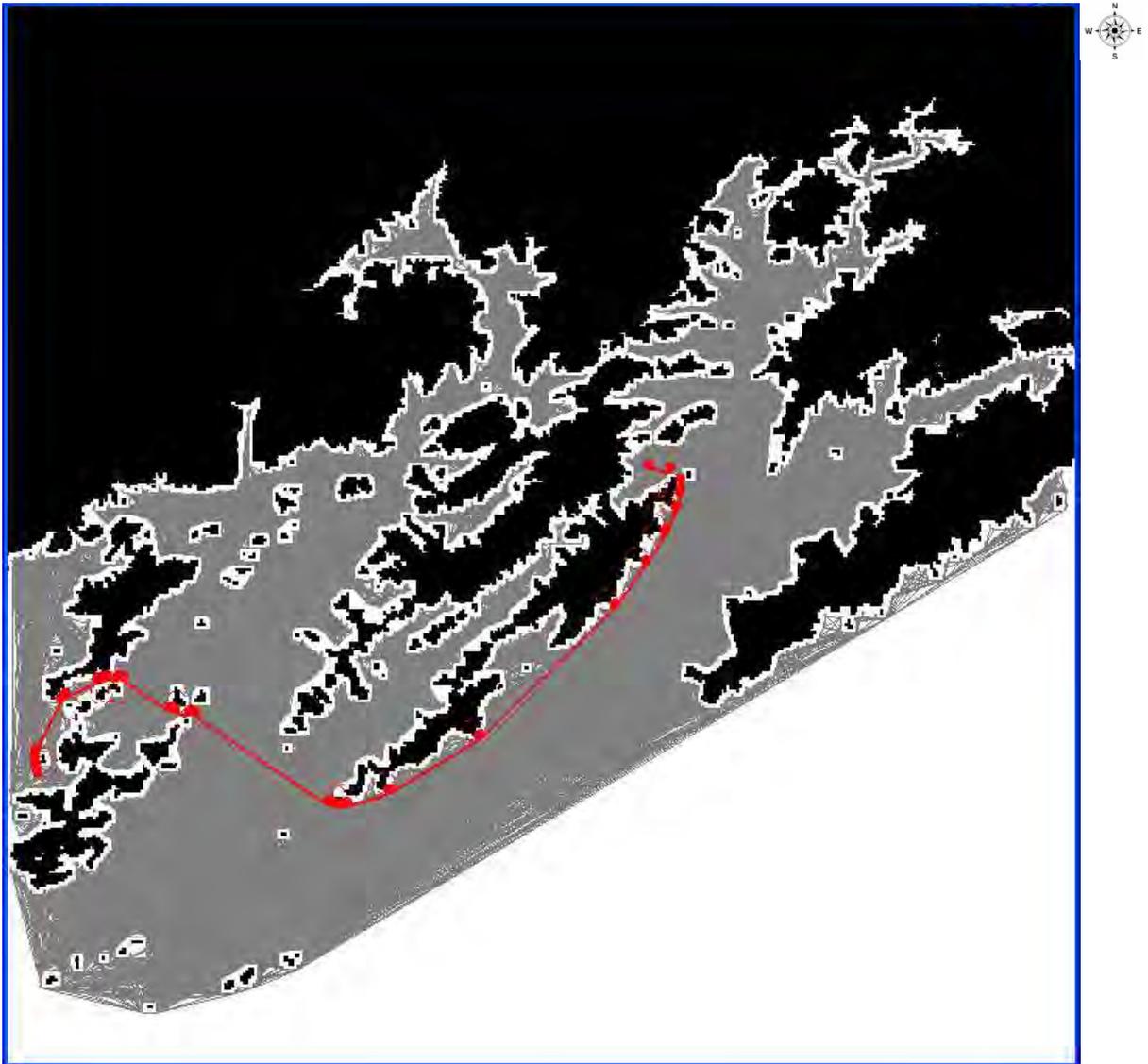


Figura 5.17 – Trajetória planejada pelo ADE entre as coordenadas $w'_o = (latitude_o, longitude_o) = (-23.075, -46.072)$ e $w'_d = (latitude_d, longitude_d) = (-22.98, -45.87)$ com um grafo de visibilidade de 1489 nós e 65241 arestas e com $r_c = 400$ m.

Tabela 5.9 – Informações referentes à 1-trajetória apresentada na Figura 5.17.

Extensão	33093.055 m
Tempo para planejamento	2.0 s
r_c	400 m
n_t	38
latitude	longitude
-23.075000	-46.072000
-23.072283	-46.072650
-23.071447	-46.072749
-23.069235	-46.072749
-23.067570	-46.072340
-23.052145	-46.064270
-23.050522	-46.062532
-23.045970	-46.052180
-23.045825	-46.051793
-23.045234	-46.049877
-23.045074	-46.048817
-23.045074	-46.045101
-23.045595	-46.043237
-23.054922	-46.027860
-23.055431	-46.026298
-23.055829	-46.021564
-23.056355	-46.019974
-23.083784	-45.975672
-23.084322	-45.973781
-23.084322	-45.970624
-23.084186	-45.969645
-23.080378	-45.956209
-23.080052	-45.955427
-23.063679	-45.926329
-23.063208	-45.925676
-23.023093	-45.881478
-23.022627	-45.881048
-23.009684	-45.871061
-23.009547	-45.870961
-23.000674	-45.864759
-23.000125	-45.864443
-22.993067	-45.861176
-22.992590	-45.860995
-22.988468	-45.859758
-22.987435	-45.859607
-22.984399	-45.859607
-22.980830	-45.862789
-22.980000	-45.870000

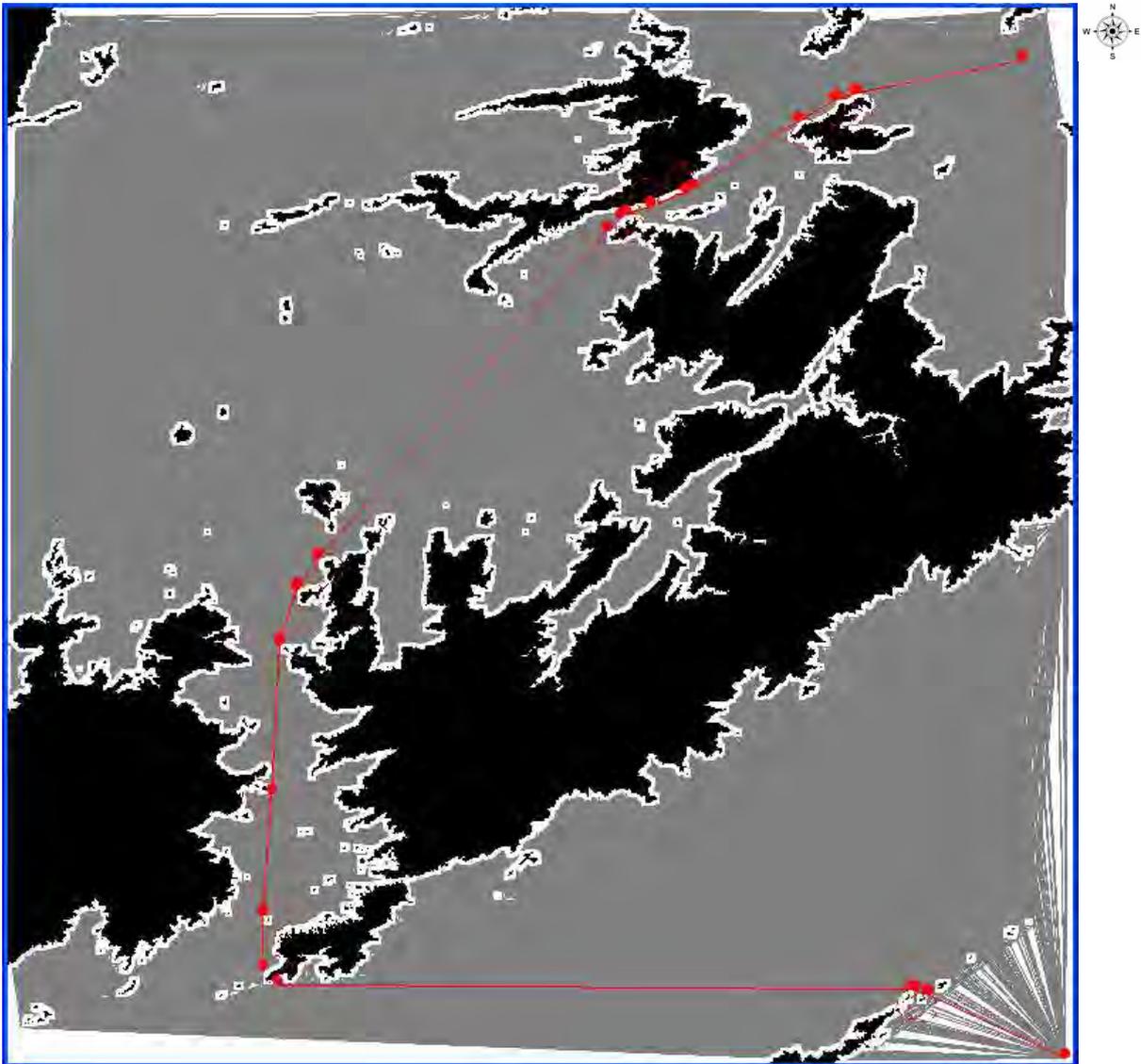


Figura 5.18 – Trajetória planejada pelo ADE entre as coordenadas $w'_o = (latitude_o, longitude_o) = (-22.99, -45.01)$ e $w'_d = (latitude_d, longitude_d) = (-22.05, -45.05)$ com um grafo de visibilidade de 3937 nós e 490707 arestas e com $r_c = 1500$ m.

Tabela 5.10 – Informações referentes à 1-trajetória apresentada na Figura 5.18.

Extensão	222237.612 m
Tempo para planejamento	7.5 s
r_c	1500 m
n_t	34
latitude	longitude
-22.990000	-45.010000
-22.929995	-45.137650
-22.929327	-45.139369
-22.926058	-45.149847
-22.925447	-45.153750
-22.920606	-45.746999
-22.907132	-45.760364
-22.855594	-45.760364
-22.854611	-45.760328
-22.741834	-45.752082
-22.741557	-45.752065
-22.600977	-45.744696
-22.597728	-45.744122
-22.550058	-45.729488
-22.546233	-45.727609
-22.520421	-45.709359
-22.519324	-45.708495
-22.211428	-45.438944
-22.210768	-45.438325
-22.198482	-45.426019
-22.195192	-45.420630
-22.188069	-45.398512
-22.187712	-45.397535
-22.174855	-45.366142
-22.174442	-45.365231
-22.171675	-45.359690
-22.170966	-45.358439
-22.107737	-45.259735
-22.107499	-45.259351
-22.087710	-45.226050
-22.086452	-45.223246
-22.081353	-45.207193
-22.080979	-45.205743
-22.050000	-45.050000

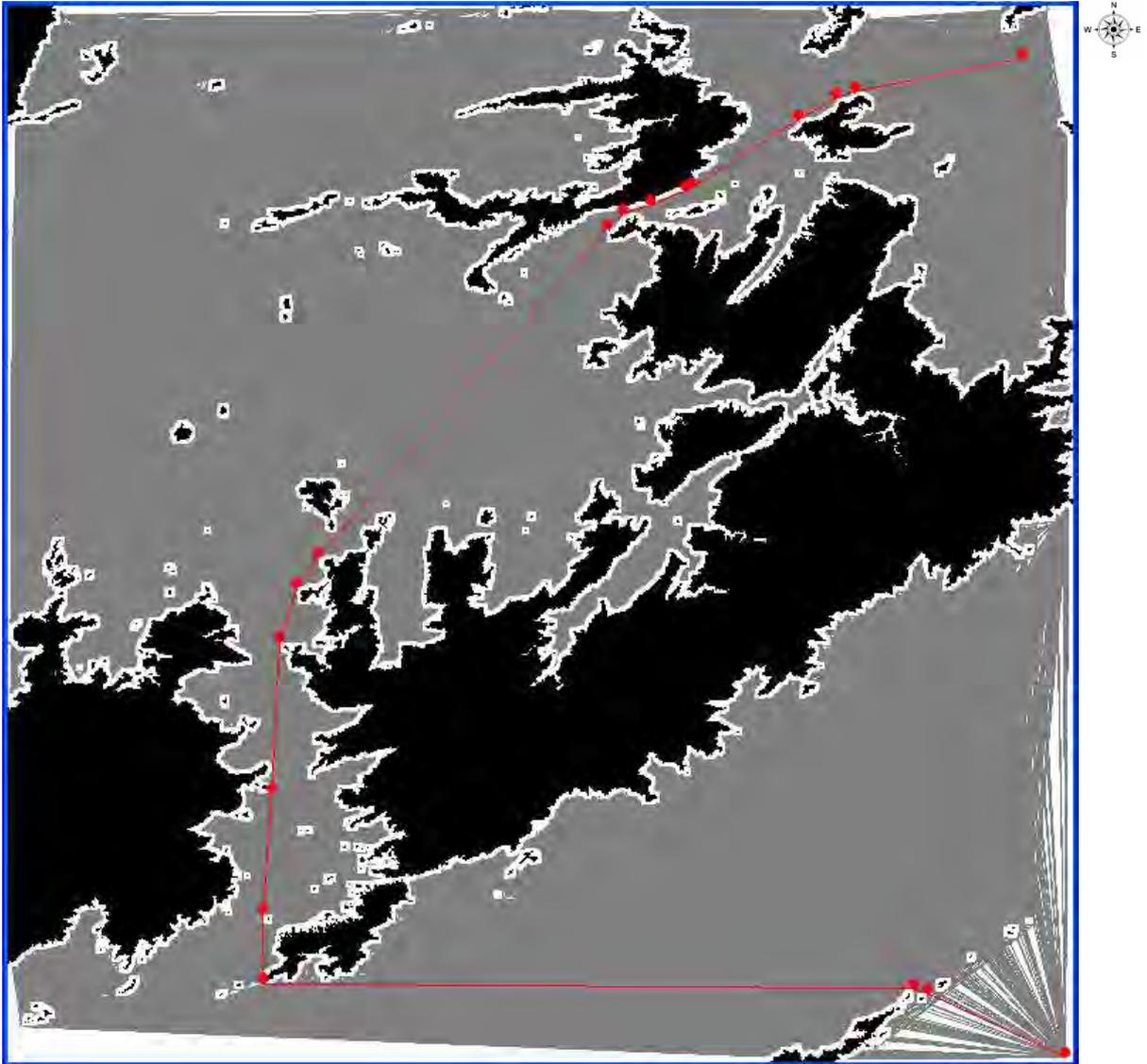


Figura 5.19 – Trajetória planejada pelo ADE entre as coordenadas $w'_o = (latitude_o, longitude_o) = (-22.99, -45.01)$ e $w'_d = (latitude_d, longitude_d) = (-22.05, -45.05)$ com um grafo de visibilidade de 3937 nós e 490707 arestas e com $r_c = 100$ m.

Tabela 5.11 – Informações referentes à 1-trajetória apresentada na Figura 5.19.

Extensão	222818.53 m
Tempo para planejamento	6.4 s
r_c	100 m
n_t	34
latitude	longitude
-22.990000	-45.010000
-22.929628	-45.138430
-22.929584	-45.138545
-22.925503	-45.151626
-22.925462	-45.151886
-22.920504	-45.759473
-22.919606	-45.760364
-22.855135	-45.760364
-22.855069	-45.760361
-22.741705	-45.752073
-22.741686	-45.752071
-22.599428	-45.744615
-22.599211	-45.744577
-22.548132	-45.728897
-22.547877	-45.728772
-22.519888	-45.708982
-22.519815	-45.708924
-22.211110	-45.438666
-22.211066	-45.438624
-22.196340	-45.423874
-22.196121	-45.423515
-22.187920	-45.398049
-22.187896	-45.397984
-22.174678	-45.365710
-22.174650	-45.365649
-22.171375	-45.359088
-22.171328	-45.359005
-22.107623	-45.259558
-22.107607	-45.259532
-22.086973	-45.224809
-22.086889	-45.224622
-22.081141	-45.206526
-22.081116	-45.206429
-22.050000	-45.050000

Tabela 5.12 – Informações referentes à 1-trajetória apresentada na Figura 5.20.

Extensão	193443.819 m
Tempo para planejamento	5.9 s
r_c	200 m
n_t	52
latitude	longitude
-22.500000	-45.050000
-22.506905	-45.038998
-22.508427	-45.038157
-22.515286	-45.038157
-22.516091	-45.038347
-22.516941	-45.038772
-22.517631	-45.039383
-22.749925	-45.387993
-22.750015	-45.388144
-22.833554	-45.544731
-22.833608	-45.544826
-22.913005	-45.674063
-22.913082	-45.674201
-22.914614	-45.677270
-22.914750	-45.677638
-22.919619	-45.697147
-22.919672	-45.697553
-22.920496	-45.748741
-22.920497	-45.748770
-22.920497	-45.758567
-22.918700	-45.760364
-22.855167	-45.760364
-22.855036	-45.760359
-22.741714	-45.752073
-22.741677	-45.752071
-22.599538	-45.744621
-22.599105	-45.744544
-22.548270	-45.728940
-22.547760	-45.728689
-22.520067	-45.709108
-22.519687	-45.708746
-22.478651	-45.656132
-22.478363	-45.655595
-22.475183	-45.646039
-22.475128	-45.645834
-22.464400	-45.593753
-22.464381	-45.593647
-22.443713	-45.450382
-22.443694	-45.450125
-22.443694	-45.440974
-22.444133	-45.439798
-22.481235	-45.396978
-22.482593	-45.396358
-22.491281	-45.396358
-22.492085	-45.396548
-22.499776	-45.400400
-22.500158	-45.400657
-22.540466	-45.436089
-22.540615	-45.436236
-22.571117	-45.470092
-22.571222	-45.470221
-22.580000	-45.482000

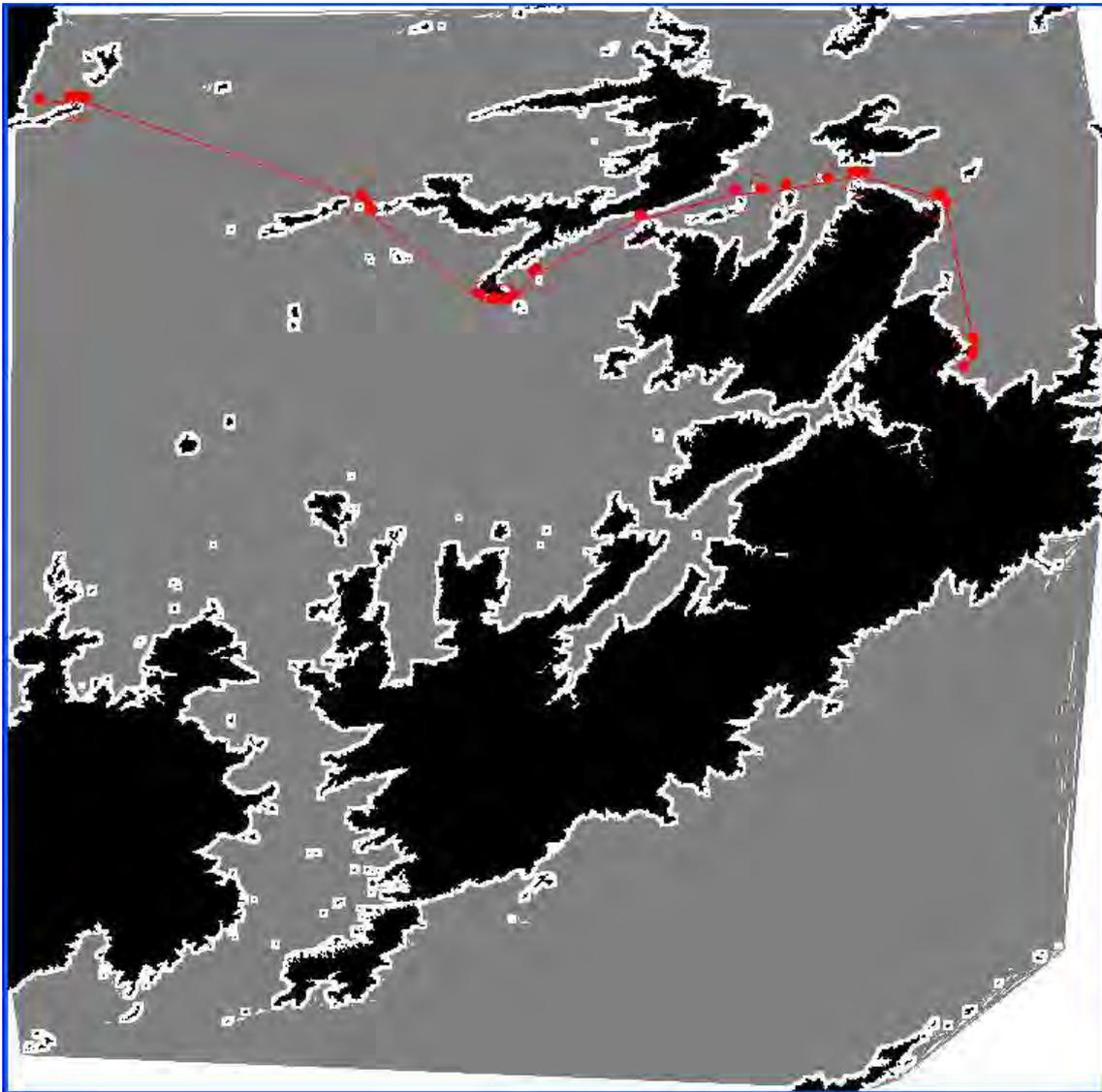


Figura 5.21 – Trajetória planejada pelo ADE entre as coordenadas $w'_o = (latitude_o, longitude_o) = (-22.09, -45.97)$ e $w'_d = (latitude_d, longitude_d) = (-22.335, -45.13)$ com um grafo de visibilidade de 3937 nós e 490707 arestas e com $r_c = 1000$ m.

Tabela 5.13 – Informações referentes à 1-trajetória apresentada na Figura 5.21.

Extensão	118001.805 m
Tempo para planejamento	6.6s
r_c	1000 m
n_t	44
latitude longitude -22.090000 -45.970000 -22.087779 -45.943151 -22.087748 -45.942410 -22.087748 -45.934402 -22.087870 -45.932928 -22.088444 -45.929479 -22.088847 -45.927928 -22.178314 -45.677815 -22.183228 -45.672586 -22.189124 -45.670055 -22.192634 -45.667363 -22.268005 -45.571787 -22.269526 -45.568900 -22.272763 -45.558524 -22.273171 -45.555848 -22.273171 -45.547694 -22.272827 -45.545230 -22.272274 -45.543293 -22.269392 -45.538861 -22.248051 -45.521047 -22.245853 -45.518326 -22.196645 -45.424593 -22.195929 -45.422770 -22.173282 -45.339320 -22.172974 -45.337276 -22.172205 -45.314938 -22.172060 -45.313614 -22.168069 -45.292024 -22.168021 -45.291740 -22.162306 -45.254122 -22.162158 -45.253366 -22.156704 -45.230732 -22.156454 -45.228628 -22.156454 -45.222740 -22.156843 -45.220124 -22.176803 -45.154545 -22.177776 -45.152406 -22.179238 -45.150063 -22.185083 -45.146013 -22.307887 -45.121251 -22.309662 -45.121074 -22.319895 -45.121074 -22.325210 -45.122815 -22.335000 -45.130000	

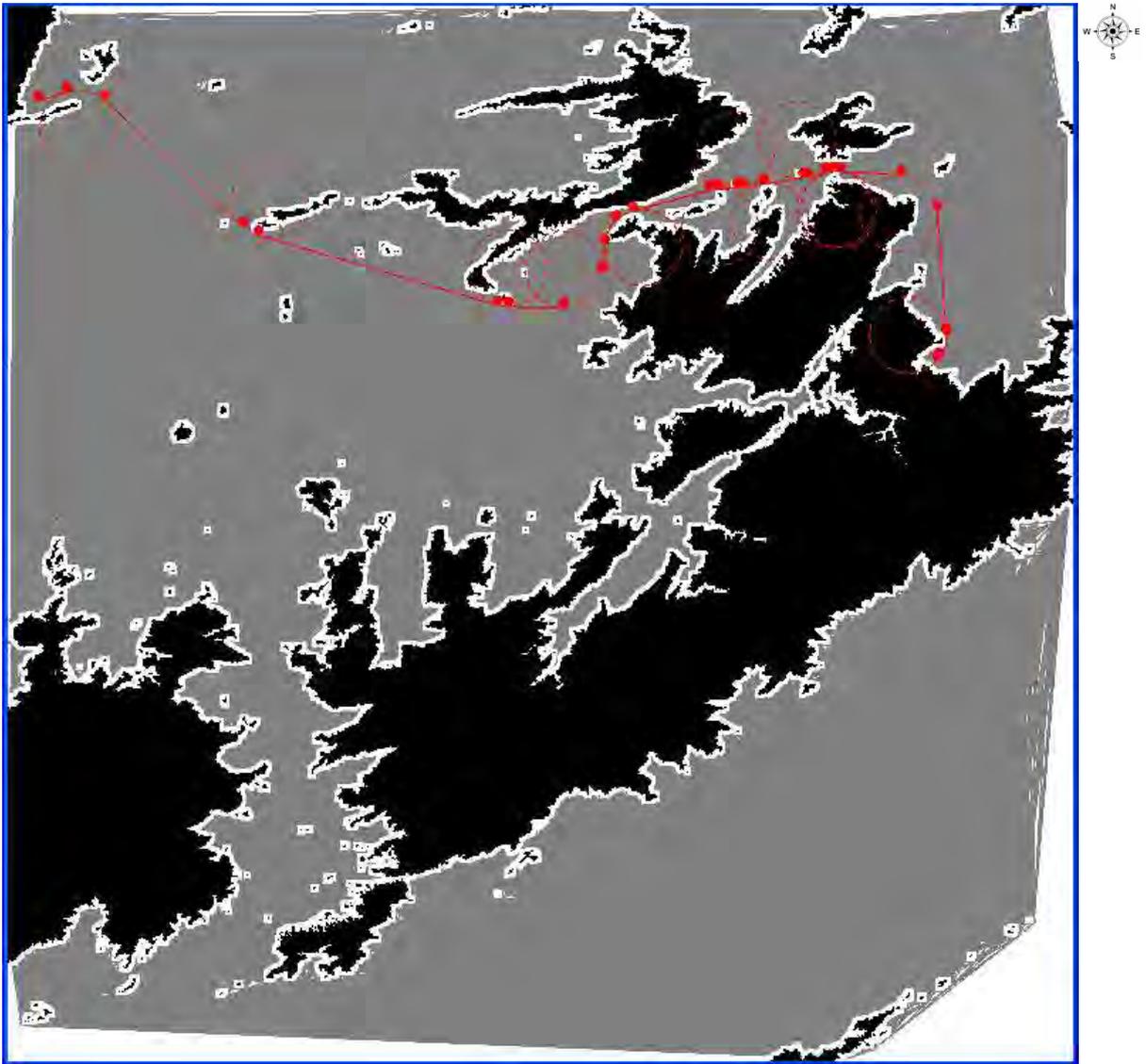


Figura 5.22 – Trajetória planejada pelo ADE entre as coordenadas $w'_o = (latitude_o, longitude_o) = (-22.09, -45.97)$ e $w'_d = (latitude_d, longitude_d) = (-22.335, -45.13)$ com um grafo de visibilidade de 3937 nós e 490707 arestas e com $r_c = 4000$ m.

Tabela 5.14 – Informações referentes à 1-trajetória apresentada na Figura 5.22.

Extensão	125070.48 m
Tempo para planejamento	8.2 s
r_c	4000 m
n_t	32
latitude	longitude
-22.090000	-45.970000
-22.081609	-45.943521
-22.089373	-45.908389
-22.208769	-45.778114
-22.216750	-45.763979
-22.282519	-45.540462
-22.283979	-45.530652
-22.284454	-45.479221
-22.251782	-45.443105
-22.224885	-45.440656
-22.202714	-45.430259
-22.202935	-45.430479
-22.193686	-45.414504
-22.174099	-45.342332
-22.172866	-45.334157
-22.172274	-45.316940
-22.171696	-45.311644
-22.168147	-45.292449
-22.167956	-45.291313
-22.162480	-45.255264
-22.161887	-45.252243
-22.157454	-45.233844
-22.156454	-45.225427
-22.156454	-45.222575
-22.156530	-45.220234
-22.160222	-45.163704
-22.193240	-45.130227
-22.307349	-45.121186
-22.310187	-45.121074
-22.311067	-45.121074
-22.332327	-45.128038
-22.335000	-45.130000

Os resultados obtidos comprovam que o ADE planeja 1-trajetórias livres de colisão para VANTs através de grafos de visibilidade, considerando a restrição raio de curva.

O algoritmo possui uma heurística de busca local para verificar situações de colisão por meio de grades regulares que representam as regiões não navegáveis de um ambiente de navegação.

Nos experimentos com o ADE, observou-se que situações de colisão eram detectadas pela heurística devido a um problema que ocorre com as 1-

trajetórias. A utilização de uma 1-trajetória calculada através dos nós w'_{i-1} , w'_i e w'_{i+1} gera o problema de situações de colisão com a própria região não navegável que define o nó w'_i . Este problema ocorreu nas aplicações do ADE com a redução do ângulo β e com o aumento do raio de curva. Portanto, como trabalho futuro, espera-se solucionar este problema, adaptando o ADE com a capacidade de planejar outros tipos de k -trajetórias ou até mesmo uma combinação destas.

Outro trabalho futuro seria a comparação entre a heurística com discretização do arco de trajetória com outros métodos de verificação de colisão, como o método baseado em OBBTrees (GOTTSCHALK, 1996) ou o método I-Collide proposto em (COHEN et al., 1995).

5.5. Simulações de Navegações por Trajetórias Planejadas pelo ADE

Algumas simulações de VANTs navegando por 1-trajetórias planejadas pelo ADE foram feitas através do simulador de voo FlightGear. Estas simulações foram realizadas no laboratório do projeto Processamento de Imagens em Tempo Real (PITER), em desenvolvimento no Instituto de Estudos Avançados (IEAv), localizado em São José dos Campos, no estado de São Paulo. A aeronave utilizada como VANT nas simulações é apresentada na Figura 5.23.

As principais especificações desta aeronave são apresentadas na Tabela 5.15 (WIKIPEDIA, 2011b).

Tabela 5.15 – Especificações da aeronave simulada.

Especificações	Piper J3 Cub
e_v	10.74 m
c_v	6.83 m
v_v	121 km/h
m_v	550 kg



Figura 5.23 – Aeronave simulada: Piper J3 Cub.

Algumas simplificações foram feitas para a execução das simulações. No planejamento das trajetórias e nas simulações não foram considerados o consumo de combustível e o alcance da aeronave. O objetivo das simulações era analisar apenas o comportamento cinemático da aeronave navegando pelas 1-trajetórias planejadas pelo ADE. No planejamento das trajetórias foram considerados o raio de curva mínimo ($r_{cmín}$) e a velocidade para $r_{cmín}$ ($v_{r_{cmín}}$) desta aeronave. A velocidade $v_{r_{cmín}}$ é aquela que permite a navegação com o raio de curva mínimo, mantendo a sustentação da aeronave. Outra simplificação foi realizada, o valor de $v_{r_{cmín}}$ foi especificado como sendo a velocidade de cruzeiro do Piper J3 Cub. A Equação 5.33 foi utilizada para o cálculo simplificado do $r_{cmín}$ (ANDERSON, 1999). Neste caso, os valores de $v_{r_{cmín}}$ e $r_{cmín}$ são apresentados na Tabela 5.16.

$$r_c = \frac{v_v^2}{g \operatorname{tg}(\lambda_v)} \quad (5.33)$$

Em que: g é a aceleração da gravidade.

Tabela 5.16 – Cálculo de $r_{cmín}$ para a aeronave Piper J3 Cub.

VANT	$v_{r_{cmín}}$	λ_v	$r_{cmín}$
Piper J3 Cub	121 km/h	45°	115.16 m

Nas simulações foi possível constatar que as 1-trajetórias planejadas pelo ADE permitiram a navegação segura das aeronaves entre as coordenadas de origem e de destino especificadas. Entretanto, é importante ser mencionado que o problema de descontinuidade das curvas de Dubins não foi abordado neste trabalho. Apesar de alguns recentes trabalhos utilizarem primitivas de Dubins para o planejamento de trajetórias para VANTs de asa fixa (MANATHARA e Ghose 2011; Bry e Roy, 2011), como apresentado em (TSOURDOS et al., 2011), este problema de descontinuidade deve ser considerado, pois consiste no desvio de um VANT de asa fixa da trajetória planejada devido à manobra do mesmo para sair de um segmento de reta e entrar em um arco de trajetória de Dubins ou vice-versa, ou, ainda, para sair de um arco de trajetória e entrar em outro arco com direção oposta (TSOURDOS et al., 2011; Neto, 2008). Por exemplo, supondo uma trajetória de Dubins composta por um arco que corresponde a uma curva à direita, um segmento de reta e um arco que corresponde a uma curva à esquerda, tal trajetória possui dois pontos de descontinuidade. A solução apresentada em (TSOURDOS et al., 2011) para este problema é a construção de trajetórias de curvatura contínua denominadas curvas de hodógrafos de Pitágoras, que substituem as trajetórias geradas pelas conexões de diferentes primitivas de movimento de Dubins. As curvas de hodógrafos de Pitágoras podem ser construídas através de curvas de Bézier. Devido ao problema de descontinuidade das curvas de Dubins, pode-se concluir que não há garantia que as 1-trajetórias planejadas pelo ADE para VANTs de asa fixa sejam sempre dinamicamente viáveis, como apresentado em (TSOURDOS et al., 2011). Entretanto, o ADE pode ser aplicado ao planejamento de trajetórias para VANTs que não sofram influências significativas do problema causado pela capacidade de manobra dos mesmos (TSOURDOS et al., 2011). Portanto, um trabalho futuro seria o estudo da possibilidade de utilização de curvas de hodógrafos de Pitágoras no ADE.

Informações referentes às implementações de algoritmos desenvolvidos ou utilizados neste trabalho são apresentadas no Apêndice D.

6 CONCLUSÕES

Neste trabalho pesquisou-se o problema de planejamento automático de trajetórias para Veículos Aéreos Não Tripulados (VANTs) usando modelagem computacional de ambientes de navegação através de grafos de visibilidade, modelos digitais de elevação e distribuições espaciais de análises de efeitos causados pela queda de tais veículos. Ao longo desta pesquisa, foram elaboradas as contribuições listadas a seguir:

- a) dois algoritmos desenvolvidos para a construção de grafos de visibilidade, visando ao planejamento de trajetórias para VANTs. O primeiro algoritmo determina os nós de grafos de visibilidade através de uma avaliação do modelo digital de elevação de um ambiente de navegação. O outro algoritmo conecta tais nós por meio de arestas que não interceptam regiões do modelo digital de elevação, consideradas obstáculos para um VANT devido à envergadura e à altitude de navegação do veículo;
- b) um lema proposto para a verificação de possíveis situações de colisão de um VANT com elevações da superfície de um ambiente de navegação, considerando o veículo em movimento retilíneo. O lema considera que o comprimento do segmento de reta que descreve o movimento do VANT é inferior ao menor lado de uma célula da grade regular de um modelo digital de elevações. Assim, o lema afirma que a verificação de toda possível situação de colisão neste segmento de reta resume-se em verificar se as seguintes células são obstáculos: as células que contêm os pontos limitantes do segmento, ou uma terceira célula, quando tais pontos limitantes pertencem a células com linhas e colunas diferentes. Os casos em que os pontos limitantes pertencem às bordas das células são tratados separadamente;
- c) um algoritmo proposto para a solução do problema do elevado número de nós e de arestas em grafos de visibilidade construídos a partir de

modelos digitais de elevação. O algoritmo proposto permite a simplificação de grafos de visibilidade baseado em áreas de invisibilidade geradas por vértices das representações poligonais de obstáculos definidos por modelos digitais de elevação;

- d) o Algoritmo Dijkstra baseado em Elevações (ADE), que é uma adaptação do Algoritmo Dijkstra Modificado (ADM), e que permite o planejamento de trajetórias dinamicamente viáveis e livres de colisão para VANTs, considerando o ambiente de navegação representado por grafos de visibilidade e por modelos digitais de elevação. O ADE apresenta três principais distinções em relação ao ADM. A primeira distinção é que o ADE soluciona o problema do planejamento da trajetória entre duas coordenadas e o ADM soluciona o problema de planejamento de trajetória entre a coordenada de destino e os demais nós do grafo de visibilidade. A segunda distinção é que o ADE planeja trajetórias da classe k -trajetórias. O método do ADM para verificações de situações de colisão em arcos de trajetórias sendo planejadas pelo algoritmo depende do número de obstáculos do ambiente de navegação. O problema é que ambientes de navegação definidos por modelos digitais de elevação apresentam elevado número de obstáculos. Então, a última distinção é que o ADE possui uma heurística para verificar situações de colisão nos arcos das trajetórias planejadas pelo algoritmo e que é invariante ao número de obstáculos do ambiente de navegação. Esta heurística baseia-se em buscas locais em uma grade regular e apresenta complexidade computacional $O(n_p)$ no pior caso, onde n_p é o número de verificações de colisão;
- e) um lema que afirma que toda possível situação de colisão em uma trajetória curvilínea de um VANT pode ser verificada através de uma discretização de tal trajetória em seções, do encapsulamento destas seções em pentágonos e através da aplicação do outro lema

mencionado anteriormente, para a detecção de colisões nos segmentos de reta que delimitam tais pentágonos; e

- f) um método proposto para gerar modelos computacionais de ambientes de navegação definidos através de modelos digitais de elevação e através de uma métrica para a avaliação dos efeitos da queda de VANTs. Cada modelo computacional gerado pelo método proposto é uma grade regular binária, que corresponde à fusão de diferentes conjuntos de informações: especificações de um VANT, especificações de voo deste VANT, informações geográficas do ambiente de navegação, e distribuição espacial de uma estimativa do risco à população na ocorrência de queda do VANT especificado. Estas grades representam as regiões navegáveis e não navegáveis de um ambiente de navegação. Os grafos de visibilidade construídos através destas grades regulares binárias permitem o planejamento de trajetórias baseados em uma estimativa do número esperado de fatalidades em possíveis acidentes com queda de VANTs.

Assim, analisando as contribuições apresentadas, observa-se que foram alcançados os seguintes objetivos especificados:

- a) o estudo do problema de simplificação de grafos de visibilidade através da simplificação das representações poligonais dos obstáculos do ambiente de navegação ou através da eliminação direta de nós ou de arestas de tais grafos. Este objetivo foi alcançado por meio da terceira contribuição apresentada;
- b) a proposta de um novo algoritmo para planejamento de trajetórias para VANTs, que consiste em uma adaptação do algoritmo ADM para aplicação em ambientes de navegação que apresentam regiões não navegáveis definidas por modelos digitais de elevação. Este objetivo foi alcançado por meio da quarta contribuição apresentada; e

- c) a união do problema de planejamento de trajetórias com uma métrica para a avaliação de riscos à população em acidentes com queda de VANTs, visando à redução de possíveis fatalidades nestes tipos de acidentes. Este objetivo foi alcançado por meio da última contribuição apresentada.

Como trabalhos futuros, as seguintes propostas são sugeridas:

- a) a utilização de ordenação crescente de áreas de invisibilidade no algoritmo proposto para simplificação de grafos de visibilidade, o que supostamente ocasionará uma melhoria deste algoritmo;
- b) a adaptação do ADE, tornando-o capaz de planejar outros tipos de k -trajetórias ou até mesmo uma combinação destas; e
- c) a utilização de curvas de hodógrafos de Pitágoras no ADE, visando à solução do problema de descontinuidade das curvas de Dubins para VANTs de asa fixa.

REFERÊNCIAS BIBLIOGRÁFICAS

- AHN, C. W.; RAMAKRISHNA, R. S. A genetic algorithm for shortest path routing problem and the sizing of populations. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 6, p. 566-579, 2002.
- AHN, C. W.; RAMAKRISHNA, R. S.; KANG, C. G., CHOI, I. C. Shortest path routing algorithm using Hopfield neural network. **IEEE Electronics Letters**, v. 37, n. 19, p. 1176-1178, 2001.
- ANDERSON, J. D. J. **Aircraft performance and design**. Boston: McGraw-Hill, 1999. 600 p.
- ANDERSON, E. P.; BEARD, R. W.; MCLAIN, T. W. Real-time dynamic trajectory smoothing for unmanned air vehicles. **IEEE Transactions on Control Systems Technology**, v. 13, n. 3, p. 471-477, 2005.
- ARAÚJO, F.; RIBEIRO, B.; RODRIGUES, L. A neural network for shortest path computation. **IEEE Transactions on Neural Networks**, v. 12, n. 5, p. 1067-1073, 2001.
- BAN, T. R.; SEN, D. Graph based analysis of tessellated surfaces. In: ACM SYMPOSIUM ON SOLID MODELING AND APPLICATIONS, 8., 2003, New York, USA. **Proceedings...** New York: ACM, 2003. p. 274-279.
- BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. **Linear programming and network flows**. New York: John Wiley & Sons Press, 1990. 704 p.
- BELLINGHAM, J.; TILLERSON, M.; RICHARDS, A.; HOW, J. P. Multi-task allocation and path planning for cooperating UAVs. In: BUTENKO, S.; MURPHEY, R.; PARDALOS, P. (Ed.). **Cooperative control: models, applications and algorithms**. Dordrecht/Boston/London: Springer , 2003. cap. 2, p. 23-42.

BERGAMINI, D. **O Universo**. Rio de Janeiro: Livraria José Olympio, 1970. 192 p.

BHATTACHARYA, P.; GAVRILOVA, M. L. Roadmap-based path planning: using the Voronoi diagram for a clearance-based shortest path. **IEEE Robotics and Automation Magazine**, v. 15, n. 2, p. 58-66, 2008.

BOURKE, P. **Determining if a point lies on the interior of a polygon**. 1987. Disponível em: <<http://paulbourke.net/geometry/insidepoly/>>. Acesso em: 16 nov. 2011.

BRY, A.; ROY, N. Rapidly-exploring random belief trees for motion planning under uncertainty. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 26., 2011, Shanghai, China. **Proceedings...** Shanghai: IEEE, 2011. p. 723-730.

BYKAT, A. Convex hull of a finite set of points in two dimensions. **Information Processing Letters**, v. 7, n. 1, p. 296–298, 1978.

CHITSAZ, H.; LAVALLE, S. M. Time-optimal paths for a Dubins airplane. In: IEEE CONFERENCE ON DECISION AND CONTROL, 46., 2007, New Orleans, USA. **Proceedings...** New Orleans: IEEE, 2007. p. 2379-2384.

CHOSSET, H.; LYNCH, K. M.; HUTCHINSON, S.; KANTOR, G.; BURGARD, W.; KAVRAKI, L. E.; THRUN, S. **Principles of robot motion: theory, algorithms and implementations**. Boston: MIT press, 2005. 625 p.

COHEN, J.; LIN, M.; MANOCHA, D.; PONAMGI, M. I-collide: an interactive and exact collision detection system for large scale environments. In: ACM INTERACTIVE 3D GRAPHICS CONFERENCE, 1., 1995, Monterey, USA. **Proceedings...** Monterey: ACM, 1995. p. 189-196.

COXETER, H. S. M. **Regular polytopes**. New York: Dover Publications, 1973. 321 p.

DALAMAGKIDIS, K.; VALAVANIS, K. P.; PIEGL, L. A. Current status and future perspectives for unmanned aircraft system operations in the US. **Journal of Intelligent and Robotic Systems**, v. 52, n. 2, p. 313–329, 2008a.

DALAMAGKIDIS, K.; VALAVANIS, K. P.; PIEGL, L. A. Evaluating the risk of unmanned aircraft ground impacts. In: IEEE MEDITERRANEAN CONFERENCE ON CONTROL AND AUTOMATION, 16., 2008b, Ajaccio-Corsica, France. **Proceedings...** Ajaccio-Corsica : IEEE, 2008. p. 709-716.

DALAMAGKIDIS, K.; VALAVANIS, K. P.; PIEGL, L. A. **On integrating unmanned aircraft systems into the national airspace system: issues, challenges, operational restrictions, certification, and recommendations.** New York: Springer, 2008c. 220 p.

DANTZIG, G. B. Programming in a linear structure. **Econometrica**, v. 17, n. 1, p. 73–74, 1949.

DASKIN, M. S. **Network and discrete location: models, algorithms and applications.** New York: John Wiley & Sons Press, 1995. 498 p.

DAVIS JUNIOR, C. A.; QUEIROZ, G. R. Algoritmos geométricos e relacionamentos topológicos. In: CASANOVA, M. A.; CÂMARA, G.; DAVIS JUNIOR, C. A.; VINHAS, L.; QUEIROZ, G. R. (Ed.). **Bancos de dados geográficos.** Curitiba: EspaçoGeo, 2005. cap. 2, p. 53-92.

DE BERG, M.; VAN KREVELD, M.; OVERMANS, M.; SCHWARZKOPF, O. Visibility graphs: finding the shortest route. In: _____ (Ed.) **Computational geometry: algorithms and applications.** Berlin: Springer, 2000. cap. 15, p. 307-317.

DE FLORIANI, L.; MAGILLO, P. Algorithms for visibility on terrains: a survey. **Environment and Planning B: Planning and Design**, v. 30, n. 5, p. 709-728, 2003.

DIJKSTRA, E. W. A Note on two problems in connection with graphs.

Numerische Mathematik, v. 1, n. 1, p. 269–271, 1959.

DING, X. C.; RAHMANI, A. R.; EGERSTEDT, M. Multi-UAV convoy protection: an optimal approach to path planning and coordination. **IEEE Transactions on Robotics**, v. 26, n. 2, p. 256-268, 2010.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. The ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 26, n. 1, p. 29-41, 1996.

DORIGO, M.; DI CARO, G.; GAMBARDELLA, L. M. Ant algorithms for discrete optimization. **Artificial Life**, v. 5, n. 2, p. 137-172, 1999.

DOUGLAS, D. H.; PEUCKER, T. K. Algorithms for the Reduction of the Number of Points Required to Represent a Line or its Caricature. **The Canadian Cartographer**, v. 10, n. 2, p. 112-122, 1973.

DUBINS, L. E. On Curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. **American Journal of Mathematics**, v. 79, n. 3, p. 497-516, 1957.

EDDY, W. A new convex hull algorithm for planar sets. **ACM Transactions on Mathematical Software**, v. 3, n. 4, p. 398–403, 1977.

ENG, P.; MEJIAS, L.; WALKER, R.; FITZGERALD, D. Path planning and control for a UAV-forced landing. **IEEE Robotics and Automation Magazine**, v. 17, n. 2, p. 90-98, 2010.

SULSOFT Environment for Visualizing Images (ENVI) 4.3. Porto Alegre, Brasil. Disponível em: <<http://www.envi.com.br/>>. Acesso em 20 ago. 2010.

FERGUSON, D.; LIKHACHEV, M.; STENTZ, A. A guide to heuristic-based path planning. In: AAAI INTERNATIONAL CONFERENCE ON AUTOMATED

PLANNING AND SCHEDULING, 15., 2005, Monterey, USA. **Proceedings...** Monterey: AAAI , 2005. p. 9-18.

FERGUSON, D.; STENTZ, A. Anytime RRT's. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 19., 2006, Beijing, China. **Proceedings...** Beijing:IEEE/RSJ, 2006. p. 5369-5375.

FINKEL, R.; BENTLEY, J. L. Quad trees: a data structure for retrieval on composite keys. **Acta Informatica**, v. 4, n. 1, p. 1-9, 1974.

FLIGHTGEAR PROJECT. **Sophisticated, professional, open-source flight simulation**. 0.9.10. Disponível em: <<http://www.flightgear.org/>>. Acesso em: 12 dez. 2011.

GHOSH, S. K.; GOSWAMI, P. P. Unsolved problems in visibility graphs of points, segments and polygons. In: INDIA-TAIWAN CONFERENCE ON DISCRETE MATHEMATICS, 1., 2009, Taipei, China. **Proceedings...** Taipei: 2009. p. 44-54.

GILBERT, A. C.; LEVCHENKO, K. Compressing network graphs. In: ACM CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 10., 2004, Seattle, USA. **Proceedings...** Seattle: ACM, 2004.

GOERZEN, C.; KONG, Z.; METTLER, B. A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance. **Journal of Intelligent and Robotic Systems**, v. 57, n. 1, p. 65-100, 2010.

GOTTSCHALK, S., LIN, M. C., MANOCHA, D. OBBTree: a hierarchical structure for rapid interference detection. In: ACM CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 23., 1996, New Orleans, USA. **Proceedings...** New Orleans: ACM, 1996. p. 171-180.

GRAHAM, R. L. An efficient algorithm for determining the convex hull of a finite planar set. **Information Processing Letters**, v. 1, n. 4,. p. 132-133, 1972.

GREEN, P.; SILVERMAN, B. Constructing the convex hull of a set of points in the plane. **The Computer Journal**, v. 22, n. 3, p. 262–266, 1979.

GROCHOLSKY, B.; KELLER, J.; KUMAR, V.; PAPPAS, G. Cooperative air and ground surveillance: a scalable approach to the detection and localization of targets by a network of UAV's and UGV's. **IEEE Robotics & Automation Magazine**, v.13, n. 3, p. 16-26, 2006.

HART, P. E.; NILSSON, N. J.; RAPHAEL, B. A formal basis for the heuristic determination of minimum cost paths. **IEEE Transactions on Systems Science and Cybernetics**, v. 4, n. 2, p. 100–107, 1968.

HENNESSEY, D.; BROOKS, D.; FRIDMAN, A.; BREEN, D. (2008) A simplification algorithm for visualizing the structure of complex graphs. In: IEEE INTERNATIONAL CONFERENCE ON INFORMATION VISUALIZATION, 12., 2008, London, UK. **Proceedings...** London: IEEE, 2008. p. 616-625.

HOLLAND, J. H. **Adaptation in natural and artificial systems**. Ann Arbor: University of Michigan Press, 1975. 206 p.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **Malha municipal digital 2005**. Disponível em:
<http://www.ibge.gov.br/home/geociencias/cartografia/territ_doc1a.shtm>.
Acesso em: 10 set. 2009.

JARVIS, R. A. On the identification of the convex hull of a finite set of points in the plane. **Information Processing Letters**, v. 2, n. 3, p. 18–21, 1973.

JENKS, G. F. Lines, computers and human frailties. **Annals of the Association of American Geographers**, v. 71, n. 1, p. 1-10, 1981.

JIA, D.; VAGNERS, J. Parallel Evolutionary Algorithms for UAV Path Planning. In: AIAA INTELLIGENT SYSTEMS TECHNICAL CONFERENCE, 1., 2004, Chicago, USA. **Proceedings...** Chicago: AIAA, 2004. p. 1-12.

JUN, M.; D'ANDREA, R. Path planning for unmanned aerial vehicles in uncertain and adversarial environments. In: BUTENKO, S.; MURPHEY, R.; PARDALOS, P. (Ed.). **Cooperative control: models, applications and algorithms**. Dordrecht/Boston/London: Springer, 2003. cap. 6, p. 95-108.

KANAYA, T.; TESHIMA, Y.; KOBORI, K.; NISHIO, K. A topology-preserving polygonal simplification using vertex clustering. In: ACM INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 3., 2005, Dunedin, New Zeland. **Proceedings...** Dunedin: ACM, 2005. p. 117-120.

KAVRAKI, L. E.; SVESTKA, P.; LATOMBE, J.-C.; OVERMARS, M. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. **IEEE Transactions on Robotics and Automation**, v. 12, n. 4, p. 566-580, 1996.

KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. **International Journal of Robotics Research**, v. 5, n. 1, p. 500-505, 1986.

KONG, Z.; e METTLER, B. Evaluation of guidance performance in urban terrains for different UAV types and performance criteria using CTG maps. **Journal of Intelligent and Robotic Systems**, v. 61, n. 1, p. 135-156, 2011.

KUWATA, Y. **Real-time trajectory design for unmanned aerial vehicles using receding horizon control**. 2003. 151 p. Thesis (Master of Science in Aeronautics and Astronautics) - Massachusetts Institute of Technology (MIT), Cambridge, 2003.

KUWATA, Y.; HOW, J. P. Stable trajectory design for highly constrained environments using receding horizon control. In: IEEE AMERICAN CONTROL CONFERENCE, 22., 2004, Boston, USA. **Proceedings...** Boston: IEEE, 2004. p. 902-907.

LANG, T. Rules for robot draughtsmen. **The Geographical Magazine**, v. 42, n. 1, p. 50-51, 1969.

LATOMBE, J. **Robot motion planning**. Norwell: Kluwer, 1991. 651 p.

LAVALLE, S. M. **Planning algorithms**. New York: Cambridge University Press, 2006. 842 p.

LEE, J.; KIM, H. J. Trajectory generation for rendezvous of unmanned aerial vehicles with kinematic constraints. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 24., 2007, Rome, Italy. **Proceedings...** Rome: IEEE, 2007. p. 1056-1061.

LOZANO-PÉREZ, T.; WESLEY, M. A. An algorithm for planning collision-free paths among polyhedral obstacles. **ACM Communications of the Association for Computing Machinery**, v. 22, n. 10, p. 560-570, 1979.

LUDINGTON, B.; JOHNSON, E.; VACHTSEVANOS, G. Augmenting UAV autonomy: vision-based navigation and target tracking for unmanned aerial vehicles. **IEEE Robotics & Automation Magazine**, v. 13, n. 3, p. 63-71, 2006.

MANATHARA, F. G.; GHOSE, D. Reactive collision avoidance of multiple realistic UAVs. **Aircraft Engineering and Aerospace Technology**, v. 83, n. 6, p. 388-396, 2011.

MEDEIROS, F. L. L.; SHIGUEMORI, E. H.; MONTEIRO, M. V. T.; DOMICIANO, M. A. P.; MARTINS, M. P. Verificação automática de situações de colisão na navegação de veículos aéreos não tripulados. In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL, 6., 2007, Rio de Janeiro, Brasil. **Proceedings...** Rio de Janeiro: SBC, 2007. p. 932-941.

MEDEIROS, F. L. L.; SILVA, J. D. S. Grafos de visibilidade aplicados à representação computacional de ambientes de navegação aérea. In: SIMPÓSIO DE APLICAÇÕES OPERACIONAIS EM ÁREAS DE DEFESA, 10., 2008. São José dos Campos, Brasil. **Proceedings...** São José dos Campos: ITA, 2008. p. 318-320.

MEDEIROS, F. L. L.; SILVA, J. D. S. Computational modeling for automatic Path planning based on evaluations of the effects of impacts of UAVs on the ground. **Journal of Intelligent and Robotic Systems**, v. 61, n. 1, p. 181-202, 2010a.

MEDEIROS, F. L. L.; SILVA, J. D. S. A Dijkstra algorithm for fixed-wing UAV motion planning based on terrain elevation. In: ROCHA COSTA, A. C.; VICARI, R. M.; TONIDANDEL, F. (Ed.). **Advances in artificial intelligence**. Heidelberg: Springer, 2010b. cap. 5, p. 213-222.

MIRANDA, E. E.; GOMES, E. G.; GUIMARÃES, M. Mapeamento e estimativa da área urbanizada do Brasil com base em imagens orbitais e modelos estatísticos. In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO, 12., 2005, Goiânia, Brasil. **Proceedings...** Goiânia: INPE, 2005. p. 3813-3820.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA). **Shuttle radar topography mission (SRTM)**. Disponível em: <<http://www2.jpl.nasa.gov/srtm/>>. Acesso em: 26 mar. 2012.

NETO, A. A. **Geração de trajetórias para veículos aéreos autônomos não tripulados**. 2008. 123 p. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, 2008.

NETO, A. A.; MACHARET, D. G.; CAMPOS, M. F. M. On the generation of trajectories for multiple UAVs in environments with obstacles. **Journal of Intelligent and Robotic Systems**, v. 57, n. 4, p. 123-141, 2010.

NIKOLOS, I. K.; VALAVANIS, K. P.; TSOURVELOUDIS, N. C.; KOSTARAS, A. Evolutionary algorithm based offline online path planner for UAV navigation. **IEEE Transactions on Systems, Man and Cybernetics**, v. 33, n. 6, p. 898-912, 2003.

NILSSON, N. J. A mobile automaton: an application of artificial intelligence techniques. In: ACM INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL

INTELLIGENCE, 1., 1969, Washington, USA. **Proceedings...** Washington: ACM, 1969. p. 509-520.

OLIVEIRA, C. P.; CUGNASCA, P. S.; CAMARGO JUNIOR, J. B.; TRINDADE JUNIOR, O. Analysis of security aspects for the utilization of unmanned aerial vehicles in frontiers. In: SIMPÓSIO DE TRANSPORTE AÉREO, 8., 2009, São Paulo, Brasil. **Proceedings...** São Paulo: RIDITA, 2009. p. 76-88.

OPHEIM, H. Smoothing a digitized curve by data reduction methods. In: EUROGRAPHICS, 2., 1981, Darmstadt, Germany. **Proceedings...** Darmstadt: Eurographics Association, 1981. p. 127-135.

PAPADIMITRIOU, C. H., STEIGLITZ, K. **Combinatorial optimization: algorithms and complexity.** New York: Dover Publications, 1998. 496 p.

PARK, W.; YU, K. Hybrid line simplification for cartographic generalization. **Elsevier Pattern Recognition Letters**, v. 32, n. 9, p. 1267-1273, 2011.

PINTO, M. J. **Algumas contribuições à resolução do problema de corte integrado ao problema de seqüenciamento dos padrões.** 2004. 194 p. (INPE-11826-TDI/964). Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2004.

RAMER, U. An iterative procedure for the polygonal approximation of plane curves. **Computer Graphics and Image Processing**, v. 1, n. 3, p. 244-256, 1972.

RANGAYYAN, R. M.; GULIATO, D.; de CARVALHO, J. D.; SANTIAGO, S. A. Polygonal approximation of contours based on the turning angle functions. **Journal of Electronic Imaging**, v. 17, n. 2, p. 023016, 2008.

REUMANN, K.; WITKAM, A. P. M. Optimizing curve segmentation in computer graphics. In: ELSEVIER INTERNATIONAL COMPUTING SYMPOSIUM, 1., 1973, Davos, Switzerland. **Proceedings...** Davos: Elsevier, 1973. p. 467-472.

- RICHARDS, J. A.; JIA, X. **Remote sensing digital image analysis: an introduction**. Heidelberg: Springer, 2006. 439 p.
- ROBINSON, A. H.; MORRISON, J.; MUEHRCKE, P. C.; KIMERLING, A. J.; GUPTILL, S. C. **Elements of cartography**. New York: John Wiley & Sons, 1995. 688 p.
- SABIDUSSI, G. The centrality index of a graph. **Psychometrika**, v. 31, n. 4, p. 581-603, 1966.
- SATHYARAJ, B. M., JAIN, L.C., FINN, A., DRAKE, S. Multiple UAVs path planning algorithms: a comparative study. **Fuzzy Optimization and Decision Making**, v. 7, n. 3, p. 257-267, 2008.
- SATO, A. **The RMAX helicopter UAV**. Iwata: Aeronautic Operations Yamaha Motor CO., 2003. 11 p.
- SHAEFFER, S. E. Graph clustering. **Elsevier Computer Science Review**, v. 1, n. 1, p. 27-64, 2007.
- PREPARATA, F. P.; SHAMOS, M. I. **Computational geometry: an introduction**. New York: Springer, 1985. 434 p.
- SHAPIRA, A. **Visibility and terrain labeling**. 1990. 168 p. Thesis (Master of Science) – Faculty of Rensselaer Polytechnic Institute, New York, 1990.
- STENTZ, A. The focussed D* algorithm for real-time replanning. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 14., 1995, San Francisco, USA. **Proceedings...** San Francisco: Morgan Kaufmann, 1995. p. 1652-1659.
- TANG, Z.; ZHU, J.; HE, F.; FENG, L.; YANG, G.; HAN, G. Adaptive polygon simplification basing on Delaunay triangulation and its application in high speed PCBs and IC packages simulation. In: IEEE INTERNATIONAL CONFERENCE ON MICROWAVE TECHNOLOGY & COMPUTATIONAL

ELECTROMAGNETICS, 1., 2011, Beijing, China. **Proceedings...** Beijing: IEEE, 2011. p. 253-256.

THOMAS, J.; BLAIR, A.; Barnes, N. Towards an efficient optimal trajectory planner for multiple mobile robots. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 16., Las Vegas, USA. **Proceedings...** Las Vegas: IEEE/RSJ, 2003. p. 2291-2296.

TIN, C. **Robust multi-UAV planning in dynamic and uncertain environments.** 2004. 110 p. Thesis (Master of Science in Mechanical Engineering) - Massachusetts Institute of Technology (MIT), Cambridge, 2004.

TOBLER, W. R. **An experiment in the computer generalization of maps.** Ann Arbor: University of Michigan, 1964. 43 p.

TOZOUR, P. Search space representations. In: RABIN, S. (Ed.). **Game programming wisdom 2.** Independence: Charles River Media, 2003. cap. 8, p. 85-102.

TSOURDOS, A.; WHITE, B. A.; SHANMUGAVEL, M. **Cooperative path planning of unmanned aerial vehicles.** New York: John Wiley & Sons Press, 2011. 214 p.

VAN DEN BOOMGARD, R; VAN BALEN, R. Methods for fast morphological image transforms using bitmapped images. **Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing**, v. 54, n. 3, p. 254-258, 1992.

VORONOI, G. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. **Journal für die Reine und Angewandte Mathematik**, v. 1908, n. 133, p. 97-178, 1908.

VISVALINGAM, M.; WHYATT, J. D. Line generalisation by repeated elimination of points. **Cartographic Journal**, v. 30, n. 1, p. 46-51, 1993.

WANG, X.; YADAV, V.; BALAKRISHNAN, S. N. Cooperative UAV formation flying with obstacle/collision avoidance. **IEEE Transactions on Control Systems Technology**, v. 15, n. 4, p. 672-679, 2007.

WENZHUN, H., YOUNGSHENG, W., XIANGYANG, Y. Studies on novel anti-jamming technique of unmanned aerial vehicle data link. **Elsevier Chinese Journal of Aeronautics**, v. 21, n. 2, p. 141-148, 2008.

XIAO-WEI, F.; ZHONG, L.; XIAO-GUANG, G. Path planning for UAV in radar network area. In: WRI GLOBAL CONGRESS ON INTELLIGENT SYSTEMS, 2., 2010, Wuhan, China. **Proceedings...** Wuhan: IEEE, 2010. p. 260-263.

WIKIPEDIA. **Apoena 3000**. Disponível em: <<http://en.wikipedia.org/wiki/Apoena>>. Acesso em: 11 nov. 2009a.

WIKIPEDIA. **KZO**. Disponível em: <[http://en.wikipedia.org/wiki/KZO_\(aircraft\)](http://en.wikipedia.org/wiki/KZO_(aircraft))>. Acesso em: 11 nov. 2009b.

WIKIPEDIA. **MQ-1 Predator**. Disponível em: <http://en.wikipedia.org/wiki/MQ-1_Predator>. Acesso em: 11 nov. 2009c.

WIKIPEDIA. **Pedro Reinel**. Disponível em: <http://pt.wikipedia.org/wiki/Pedro_Reinel>. Acesso em: 20 out. 2011a.

WIKIPEDIA. **Piper J3 Cub**. Disponível em: <http://en.wikipedia.org/Piper_J-3_Cub>. Acesso em: 23 set. 2011b.

WIKIPEDIA. **RQ-4 Global Hawk**. Disponível em: <http://en.wikipedia.org/wiki/RQ-4_Global_Hawk>. Acesso em: 11 nov. 2009d.

WIKIPEDIA. **ZALA 421-08**. Disponível em: <http://en.wikipedia.org/wiki/ZALA_421-08>. Acesso em: 11 nov. 2009e.

ZALOTI JUNIOR, O. D. **Avaliação do modelo digital do terreno extraído de dados SAR interferométricos na banda X do SAR R-99B**. 2007. 142 p.

(INPE-15312-TDI/1357). Dissertação (Mestrado em Sensoriamento Remoto) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2007.

ZHAO, Z.; SAALFELD, A. Linear-time sleeve-fitting polyline simplification algorithms. In: AUTO-CARTO, 13., 1997, Seattle, USA. **Proceedings...** Seattle: Cartography and Geographic Information Society, 1997. p. 214-223.

APÊNDICE A - DEFINIÇÃO FORMAL DE UMA TRAJETÓRIA DE NAVEGAÇÃO DINAMICAMENTE VIÁVEL

Considerando um VANT navegando a altitude constante e considerando que a dinâmica do VANT sujeita ao seu piloto automático para controle de velocidade e de proa é de primeira ordem

$$\left\{ \begin{array}{l} \dot{x}_v = v_v \cos \theta_v \\ \dot{y}_v = v_v \operatorname{sen} \theta_v \\ \dot{\theta}_v = u_{\theta_v} (\theta_{ve} - \theta_v) \\ \dot{v}_v = u_{v_v} (v_{ve} - v_v) \\ -\frac{v_v}{r_c} \leq \dot{\theta}_v \leq \frac{v_v}{r_c} \\ 0 < v_{v\min} \leq v_v \leq v_{v\max} \end{array} \right. \quad (\text{A.1})$$

Em que: (x_v, y_v) é a posição do VANT; v_v é a velocidade do VANT; θ_v é o ângulo formado pelo eixo x e o eixo longitudinal do VANT; u_{θ_v} e u_{v_v} são constantes que dependem do projeto do piloto automático; r_c é o raio de curva da trajetória do VANT; θ_{ve} e v_{ve} são entradas do sistema; e $v_{v\min}$ e $v_{v\max}$ são as velocidades mínima e máxima de navegação, respectivamente.

Uma trajetória $z_v(t) = (x_v(t), y_v(t))$ é dinamicamente viável se existem entradas θ_{ve} e v_{ve} tal que a Equação A.1 é satisfeita para todo $t \geq 0$.

APÊNDICE B - PROVA DA PRINCIPAL PROPRIEDADE DE GRAFOS DE VISIBILIDADE

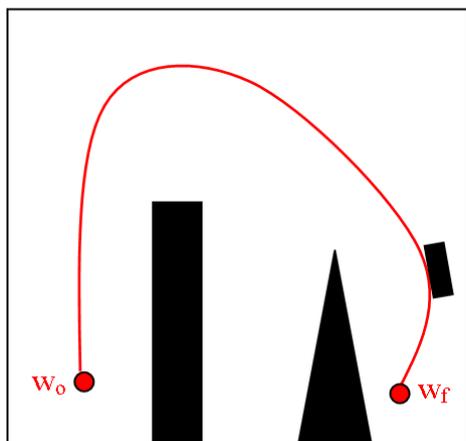
Grafos de visibilidade possuem a propriedade de permitirem o planejamento de rotas e trajetórias com a menor extensão possível se os obstáculos do ambiente de navegação forem representados por polígonos. Uma prova desta propriedade é descrita em (de Berg et al., 2000) e é apresentada a seguir de uma forma mais detalhada. Nesta prova, considera-se que as bordas de um obstáculo, isto é, os lados de um obstáculo são navegáveis. A Figura B.1 auxilia na compreensão desta prova.

Lema 1. Qualquer rota entre duas coordenadas w_i e w_f entre um conjunto C_o de obstáculos poligonais disjuntos é uma rota poligonal cujos vértices são vértices intermediários convexos de C_o . Uma rota poligonal é uma sequência de vértices, que são coordenadas de navegação conectadas por segmentos de reta.

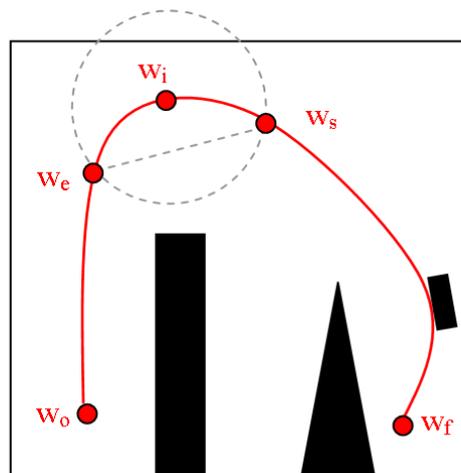
Prova 1. Suponha por contradição que uma rota W mais curta não seja poligonal. Como os obstáculos são poligonais, isto significa que há um ponto w_i em W que pertence a alguma região navegável com a propriedade que nenhum segmento de reta contendo w_i está contido em W . Como w_i está em uma região navegável, há uma circunferência com centro w_i , que não intercepta qualquer obstáculo e que intercepta a rota W em w_e e w_s . Observa-se então que o pedaço da trajetória dentro da circunferência pode ser substituído pelo segmento de reta $\overline{w_e w_s}$ tornando a trajetória total mais curta. O que pode ser verificado na Figura B.1b. Isto contradiz a suposição inicial, uma vez que qualquer rota mais curta deve ser localmente mais curta, isto é, qualquer pedaço de rota conectando duas coordenadas w_p e w_q deve ser a rota mais curta entre w_p e w_q . Então, foi provado que toda rota mais curta deve ser poligonal.

Considerando que a rota W é a rota mais curta, um vértice w_v desta rota não pode pertencer a alguma região navegável diferente da borda, pois seria possível especificar uma circunferência com centro em alguma coordenada w_i , tal que o fragmento da rota no interior da circunferência seria substituído pelo segmento de reta $\overline{w_e w_s}$. Isto é uma contradição, pois a rota W deixaria de ser a mais curta. O que pode ser observado na Figura B.1d. De modo análogo, se o vértice w_v pertencesse à borda do obstáculo, seria possível especificar uma circunferência com centro em w_v tal que, se o segmento de reta $\overline{w_e w_s}$ não intercepta qualquer obstáculo, o fragmento da rota no interior da circunferência seria substituído pelo segmento de reta $\overline{w_e w_s}$. Este fato pode ser observado na Figura B.1e. O que novamente é uma contradição para a definição da rota W mais curta. Portanto, a única possibilidade é que o vértice w_v seja um vértice intermediário convexo de C_o e que o segmento de reta $\overline{w_e w_s}$, definido pela circunferência de centro em w_v , intercepte o obstáculo que define w_v . Isto poder verificado através de uma análise da Figura B.1f. Então, foi provado que toda rota mais curta é uma rota poligonal cujos vértices são vértices intermediários convexos de C_o .

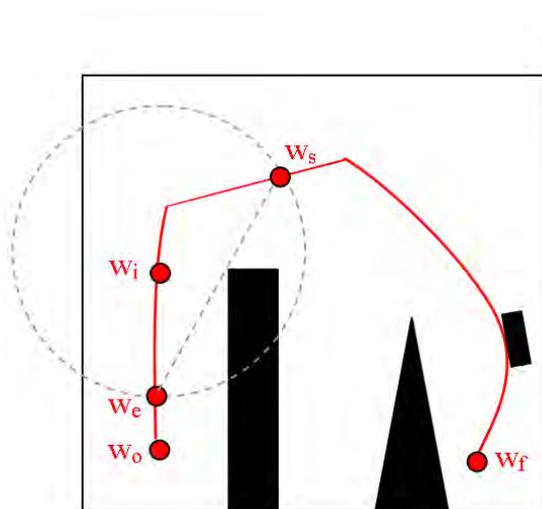
Portanto, como um grafo de visibilidade é um conjunto de nós definidos pelos vértices intermediários convexos de C_o e conectados por arestas que não interceptam obstáculos de C_o , ele permite o planejamento de rotas de menor extensão possível entre quaisquer duas coordenadas navegáveis, desde que tais coordenadas sejam adicionadas ao grafo de visibilidade, conectando-as com todos os nós visíveis deste grafo.



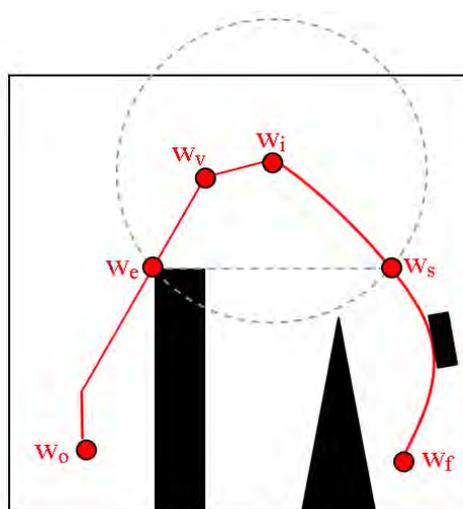
(a)



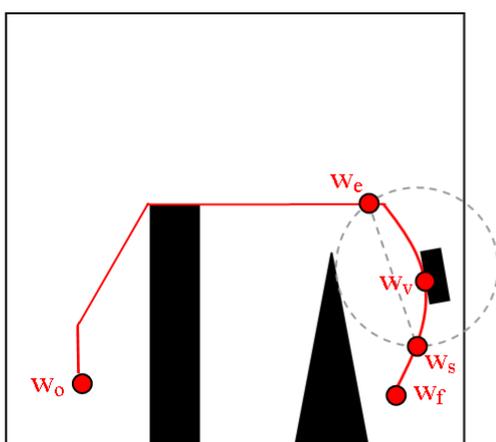
(b)



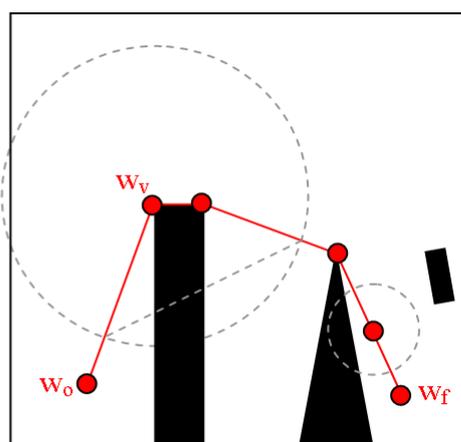
(c)



(d)



(e)



(f)

Figura B.1 – Exemplo de uma rota mais curta, que é poligonal e cujos vértices são vértices intermediários convexos de C_o .

APÊNDICE C - DEFINIÇÃO FORMAL DE UMA TRAJETÓRIA DE DUBINS

Considerando o movimento de um objeto descrito pela Equação 5.1, com módulo constante da velocidade e com o mínimo raio de curva, uma trajetória de Dubins (Dubins, 1957) é aquela composta por três primitivas de movimento: mover-se em linha reta, descrevendo um segmento de reta (S); mover-se descrevendo uma curva à esquerda (E); e mover-se descrevendo uma curva à direita (D). Como provado em (Dubins, 1957), a trajetória de menor extensão entre duas coordenadas quaisquer sempre pode ser expressa como uma combinação das primitivas S , E e D (LaValle, 2006). Há seis classes de trajetórias de Dubins: $E_{\psi_1}D_{\psi_2}E_{\psi_3}$; $D_{\psi_1}E_{\psi_2}D_{\psi_3}$; $E_{\psi_1}S_dE_{\psi_3}$; $E_{\psi_1}S_dD_{\psi_3}$; $D_{\psi_1}S_dE_{\psi_3}$; e $D_{\psi_1}S_dD_{\psi_3}$, onde $\psi_1, \psi_3 \in [0, 2\pi)$, $\psi_2 \in (\pi, 2\pi)$ e $d \geq 0$. Exemplos destas classes de trajetórias são apresentados na Figura C.1.

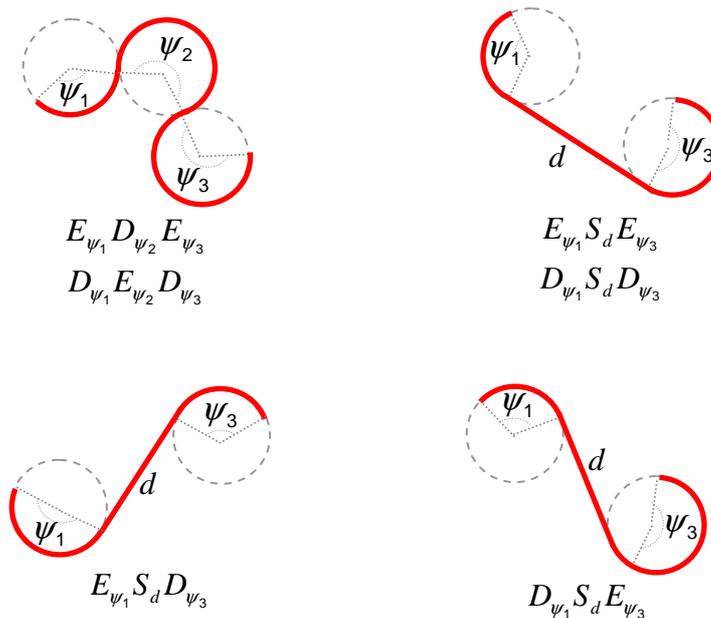


Figura C.1 – Exemplos das seis classes de trajetórias de Dubins.

APÊNDICE D - INFORMAÇÕES SOBRE IMPLEMENTAÇÕES COMPUTACIONAIS DE ALGORITMOS

Este apêndice apresenta as linguagens de programação, os ambientes de programação e os aplicativos computacionais utilizados para a implementação dos principais algoritmos propostos ou utilizados neste trabalho, como pode ser verificado na Tabela D.1.

Tabela D.1 – Informações referentes às implementações dos principais algoritmos propostos ou utilizados.

Algoritmos	Linguagem de Programação / Ambiente de Programação / Aplicativo Computacional
Algoritmo para aquisição da matriz de elevações (E)	Matlab
Algoritmo para verificação bidimensional de colisões	C++
Algoritmos para a construção de grafos de visibilidade	C++
Algoritmo de máxima verossimilhança para classificação de imagens	ENVI 4.3
Algoritmo para aplicação dos operadores morfológicos	Matlab
Algoritmo para criação da matriz de áreas urbanas (U)	Matlab
Algoritmo para a criação da matriz de densidades (D)	C
Algoritmo para a criação da matriz de distribuição espacial da métrica t_m	C
Algoritmo para a criação da grade binária (B)	C
Algoritmo para conversão de grade regular em representação poligonal	C
Algoritmo varredura de Graham	Matlab
Algoritmo Visvalingam-Whyatt	C
Algoritmo Ramer-Douglas-Peucker	C
Algoritmo para simplificação de polígonos baseado em áreas de invisibilidade entre vértices intermediários convexos	C
Algoritmo para determinação do vértice intermediário convexo v_2	C
Algoritmo para verificação da visibilidade entre os vértices v_1 e v_2	C
Algoritmo para cálculo da área de invisibilidade entre os vértices v_1 e v_2	C
Algoritmo Dijkstra baseado em Elevações (ADE)	C++
Heurística de Busca Local Baseada em Grade Regular (h_{cs})	C++
Representação gráfica dos ambientes de navegação, grafos de visibilidade, trajetórias e rotas de navegação	OpenGL

Todos os experimentos, com exceção das simulações, foram realizados em um computador com um processador de 1.73 GHz e com 1.0 GB de memória RAM. A especificação do computador utilizado é uma referência para a análise dos tempos de execução apresentados ao longo desta monografia.