

Universidade Federal do Rio Grande - FURG



Matheus Machado dos Santos

**Descrição e detecção de regiões subaquáticas
parcialmente estruturadas em imagens acústicas
adquiridas por um sonar de imageamento frontal**

Rio Grande

Abril-2016



Matheus Machado dos Santos

**Descrição e detecção de regiões subaquáticas
parcialmente estruturadas em imagens acústicas
adquiridas por um sonar de imageamento frontal**

Dissertação de Mestrado apresentado ao Programa de Pós Graduação em Computação da Universidade Federal do Rio Grande, como requisito para a obtenção do grau de Mestre em Engenharia de Computação.

Universidade Federal do Rio Grande – FURG

Centro de Ciências Computacionais – C3

Programa de Pós Graduação em Computação – PPGComp

Curso de Mestrado em Engenharia de Computação

Orientador: Dr.^a Silvia Silva da Costa Botelho

Coorientador: Dr. Paulo Lilles Jorge Drews Júnior

Rio Grande

Abril-2016

Ficha catalográfica

S237d Santos, Matheus Machado dos.

Descrição e detecção de regiões subaquáticas parcialmente estruturadas em imagens acústicas adquiridas por um sonar de imageamento frontal / Matheus Machados dos Santos. – 2016. 116 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-graduação em Engenharia de Computação, Rio Grande/RS, 2016.

Orientadora: Dr^a. Silvia Silva da Costa Botelho.

Coorientador: Dr. Paulo Lilles Jorge Drews Júnior.

1. Sonar 2. Imageamento 3. Robótica móvel 4. Levantamento subaquático 5 I. Botelho, Silvia Silva da Costa II. Drews Júnior Paulo Lilles Jorge III. Título.

CDU 004.896

UNIVERSIDADE FEDERAL DO RIO GRANDE
Centro de Ciências Computacionais
Programa da Pós-Graduação em Computação
Curso de Mestrado em Engenharia de Computação

DISSERTAÇÃO DE MESTRADO

Descrição e detecção de regiões subaquáticas parcialmente estruturadas em imagens acústicas adquiridas por um sonar de imageamento frontal.

Matheus Machado dos Santos


Banca examinadora:



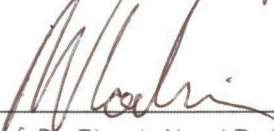
Prof. Dr. Fernando Santos Osório



Prof. Dr. Thiago de Castro Martins



Prof.^a Dr.^a Cátia Maria dos Santos Machado



Prof. Dr. Ricardo Nagel Rodrigues



Prof.^a Dr.^a Sílvia Silva da Costa Botelho
Orientadora



Prof. Dr. Paulo Lilles Jorge Drews Júnior
Coorientador

Agradecimentos

Gostaria de agradecer a professora Dr.^a Silvia Botelho e ao professor Dr. Paulo Drews que acreditaram e possibilitaram a realização deste trabalho aceitando a tarefa de orientar e coorientar, muito obrigado pela paciência, pelas discussões e pelas ideias que tivemos nas diversas reuniões realizadas durante esta jornada.

Agradeço a todos os colegas de laboratório do Núcleo de Automação e Computação NAUTEC-FURG e aos professores que de uma forma ou de outra contribuíram para a coleta de dados e para a realização deste trabalho, Luan Silveira, Felipe Moraes, Felipe Guth, Pedro Ballester, Tiago Giacomelli, Guilherme Zaffari, Amanda Duarte, Cristiano Steffens, Mateus Saquetti, Ronaldo Cozza e aos professores Dr. Vagner Rosa e Dr. Glauber Gonçalves. Aos demais colegas pela agradável convivência diária Telmo Santos, Eduardo Santos, Alex Telocken, Ricardo Rocha, Daniel Fernandes, Everson Siqueira, Jusoan Mór, Sidnei da Silva e Joel Gaya. A esses colegas que de certa forma deixam de ser colegas para serem grandes amigos, muito obrigado.

Agradeço aos membros da banca por terem aceitado participar da avaliação deste trabalho revisando e contribuindo para a seu aperfeiçoamento, muito obrigado aos professores Dr. Fernando Osório, Dr. Thiago Martin, Dr.^a Catia Machado e Dr. Ricardo Nagel.

Gostaria de agradecer ao apoio financeiro da Agência Nacional do Petróleo, Gás Natural e Biocombustível - ANP, da Financiadora de Estudos e Projetos – FINEP – e do Ministério de Ciências e Tecnologia - MCT por meio do Programa de Recursos Humanos da ANP para o Setor Petróleo e Gás - PRH-ANP/MCT. Gostaria de agradecer aos professores responsáveis pelo Programa de Recursos Humanos da FURG – PRH-27–, professora Dr.^a Maria Isabel e ao professor Gilberto Griep pelo empenho e dedicação para manter o programa e oferecer as melhores condições possíveis para os estudantes do programa concluírem os seus estudos.

Finalmente gostaria de fazer um agradecimento muito especial a minha família, principalmente a minha mãe, ao meu pai e a minha irmã que deram o incentivo e todo o apoio que foi necessário durante esta jornada. Com certeza sem esse apoio nada disto teria acontecido. A vocês MUITO OBRIGADO.

“To have a great idea, have a lot of them”
(Thomas A. Edison)

Resumo

O mar é uma importante fonte de recursos minerais e biológica que ainda pouco foi explorada. A crescente utilização de robôs tanto para a pesquisa quanto para aplicações comerciais é eminente. Devido as limitações impostas pelo ambiente, um dos principais desafios da robótica móvel subaquática é a localização. Este trabalho propõe um método de descrição e reconhecimento de regiões subaquáticas em ambientes parcialmente estruturados a partir de imagens de um sonar de imageamento frontal. O método utiliza o modelo probabilístico Gaussiano para a descrição dos objetos detectados e um grafo para representar a topologia da região. As regiões são detectadas através da comparação dos grafos. O trabalho apresentará as principais características do sonar de imageamento frontal, entre elas, a capacidade de operação em ambientes com pouca visibilidade. Ao final, cada etapa do método proposto é avaliada utilizando dados reais capturados em uma marina.

Palavras-chave: Sonar de imageamento frontal. Detecção de regiões semelhantes. Navegação autônoma

Abstract

The sea is an important source of minerals and biological resources which is still not largely explored. The increasing use of robots for both commercial and research applications is imminent. Because of the limitations imposed by the environment, one of the main challenges of underwater mobile robotics is the self localisation. This dissertation proposes a method to describe and recognize underwater regions in partially structured environments using acoustic images from a forward looking sonar. The method uses a Gaussian probabilistic function to describe detected objects and a graph to represent its topology. The regions are detected by comparing the graphs. The dissertation will present the main features of the forward looking sonar, among them, its ability to operate in low visibility condition environments. At the end, each stage of the proposed method is evaluated using a real data captured in a harbor.

Keywords: Forward Looking Sonar. Loop Closure Detection. Autonomous Navigation.

Lista de ilustrações

Figura 1 – Representação do problema de localização e mapeamento simultâneo (SLAM). Em verde a trajetória verdadeira realizada pelo robô e desconhecida pelo sistema de navegação. Em vermelho a trajetória estimada pelo sistema de navegação. Em azul as marcas visuais do ambiente. Em laranja as marcas visuais percebidas pelo sistema de navegação. Os triângulos indicam a pose no instante em que uma percepção visual é realizada.	30
Figura 2 – Dataset Aracati 2014 disponibilizado por (SILVEIRA et al., 2015).	32
Figura 3 – Etapas do método descrição e identificação de imagens acústicas.	34
Figura 4 – Representação de uma imagem acústica de um sonar de imageamento frontal.	40
Figura 5 – Imagens acústicas de um sonar de imageamento frontal extraído do dataset ARACATI 2014 (SILVEIRA et al., 2015).	41
Figura 6 – Imagem apresentada em (HURTÓZ, 2014) que ilustra como uma imagem acústica é formada. Um modelo 3D representa a cena observada, a pose do sonar e a forma de propagação das ondas acústicas. Ao fundo existe um paralelepípedo. As ondas se propagam para a frente varrendo os arcos azimute θ e de inclinação ϕ até interceptarem o fundo, também representado na imagem. Na parte superior da imagem é mostrada a imagem acústica resultante capturado pelo sonar nesta cena.	42
Figura 7 – Estas imagens foram apresentadas por Aykin, M. D. e Negahdaripour em (AYKIN; NEGAHDARIPOUR, 2012), onde é proposto um método de reconstrução 3D utilizando sombras acústicas. Foi utilizado o sonar de imageamento frontal DIDSON de 1.8 MHz.	45
Figura 8 – Etapas de processamento de imagem segundo (GONZALEZ; WOODS, 2002).	47
Figura 9 – Grafo de Königsberg por Euler. Cada ilha é um vértice, cada ponte é uma aresta. Imagem modificada de (SEVEN...,)	49
Figura 10 – Exemplo de um grafo (a) não dirigido, (b) dirigido (dígrafo) e (c) dirigido e ponderado.	49
Figura 11 – Exemplo de um grafo G em (a) e o seu subgrafo $G_s(G)$ em (b).	50
Figura 12 – Exemplo de um grafo bipartido em (a) e a sua representação em forma matricial em (b).	50
Figura 13 – Exemplo de dois grafos isomórficos em (a), a sua função de mapeamento $f : V(G_1) \leftrightarrow V(G_2)$ em (b) e a matriz de correspondência de vértices X em (c).	51

Figura 14 – Representação de um grafo por matriz de adjacência, em (a) o grafo e em (b) a sua representação em matriz de adjacência.	51
Figura 15 – Representação de grafo por lista de adjacência.	52
Figura 16 – Representação de um grafo não dirigido por matriz de incidência.	53
Figura 17 – Representação de grafo dirigido por matriz de incidência.	53
Figura 18 – Problema de alocação linear.	54
Figura 19 – Representação dos pixel de vizinhança 4 em (a) e os pixels de vizinhança 8 em (b). Assim como numa matriz, o número de linha cresce para baixo e o número de colunas cresce para a direita.	60
Figura 20 – Exemplo dos métodos de extração de segmentos aplicado em (a), um recorte de uma imagem acústica. As figuras (b),(c) e (d) mostram a extração de segmentos com base na distância. As figuras (e) e (f) mostram o método de extração de borda de segmento e as figuras (g) e (h) mostram a extração de segmento completa. Cada segmento recebe uma cor diferente e segmentos com a mesma cor são considerados o mesmo segmento.	62
Figura 21 – Variáveis utilizadas pelo método proposto. O gráfico representa um perfil de intensidade de um <i>beam</i> acústico, o eixo horizontal representa os <i>bins</i> e o eixo vertical a intensidade de cada <i>bin</i>	64
Figura 22 – Fluxograma do método proposto.	65
Figura 23 – Modelo de descrição das imagens acústicas. Duas Gaussianas GA_1 e GA_2 são representadas como vértices de um grafo. Em vermelho o ângulo de inclinação $\theta_g(GA_1)$ entre o eixo vertical da imagem(em vermelho) e o eixo de maior dispersão da elipse. Em verde e em azul, respectivamente, o maior e o segundo maior eixo de dispersão da Gaussiana GA_1 . Em amarelo uma aresta que liga duas Gaussianas GA_1 e GA_2 composta pela inclinação $\theta_e(e_1)$ relativa a inclinação da Gaussiana de origem $\theta_g(GA_1)$ e a distância ρ_e entre o centro das duas Gaussianas GA_1 e GA_2	67
Figura 24 – Encontrando a semelhança entre vértices.	70
Figura 25 – Encontrando o alinhamento dos vértices através da geometria dos objetos extraídos da imagem. Em (a) uma cena observada de dois pontos de vista distintos, em (b) e (c) as imagens capturadas de cada ponto de vista. As linhas em vermelho representam a orientação das Gaussianas θ_g extraídas da imagem.	71
Figura 26 – Cálculo do erro de posição final de um par arestas.	74
Figura 27 – Problema de ambiguidade na orientação das Gaussianas θ_g	75
Figura 28 – Distorção acústica na forma dos objetos extraídos.	77
Figura 29 – Distorção acústica e o efeito na orientação das arestas θ_e	78
Figura 30 – Efeito de achatamento nas imagens acústicas.	78

Figura 31 – Distorção acústica e a variação angular das arestas.	79
Figura 32 – Combinações de testes para o alinhamento de vértices baseado em topologia(força bruta).	79
Figura 33 – Alinhamento guloso de vértice com base em arestas adjacentes.	81
Figura 34 – Diagrama UML do código em C++ gerado pelo programa Umbrello. Em laranja as classes relacionadas a segmentação, em vermelho as classes relacionadas com a descrição e em verde as classes relacionadas ao cálculo de semelhança de grafos.	83
Figura 35 – Foto do LBV300-5 com o sonar P900-130 em (a). Em (b) um exemplo de imagem acústica capturado e em (c) uma imagem óptica do ambiente capturada aproximadamente na mesma região da imagem acústica de (b). Em vermelho estão marcados os postes e em verde os cascos dos barcos identificados de forma empírica nas duas imagens.	86
Figura 36 – Imagens de satélite obtidas pelo software Google Earth e os pontos de GPS coletados. A trajetória percorrida começa no ponto em azul e terminando no ponto em verde.	87
Figura 37 – Padrão de insonificação do sonar de imageamento frontal Blue View P900-130, utilizado no dataset ARACATI 2014, obtido através da média de 443 imagens.	87
Figura 38 – Resultado da aplicação do padrão de insonificação da figura 37 em um par de imagens.	88
Figura 39 – Resultados do método de busca por regiões de interesse quando alterado os parâmetros ρ_{linear} e $\rho_{recursivo}$	90
Figura 40 – Comparação do método de extração de segmentos baseado em bordas com o de extração de segmentos relativo a distância.	92
Figura 41 – Imagens utilizadas para o estudo do comportamento dos parâmetros $\pi_{recursivo}$, π_{fim} e H_{busca} . Em (a) o perfil de intensidade do <i>beam</i> representado pela linha azul na figura (b). O eixo horizontal representa os <i>bins</i> e o eixo vertical representa a intensidade de cada <i>bin</i>	94
Figura 42 – Comportamento do parâmetro $\pi_{recursivo}$	94
Figura 43 – Gráfico do comportamento a busca de segmentos ao variar o parâmetro $\pi_{recursivo}$	95
Figura 44 – Comportamento do parâmetro π_{fim}	96
Figura 45 – Gráfico do comportamento a busca de segmentos ao variar o parâmetro π_{fim}	96
Figura 46 – Comportamento do parâmetro H_{busca}	97
Figura 47 – Gráfico do comportamento a busca de segmentos ao variar o parâmetro H_{busca}	97

Figura 48 – Análise de picos de 25 <i>beams</i> de uma imagem acústica utilizando os parâmetros $\pi_{fim} = 0.6$, $\pi_{recursivo} = 0.3$ e $H_{busca} = 165$	98
Figura 49 – Imagem após a etapa de segmentação. Os segmentos estão marcados por pixels coloridos.	98
Figura 50 – Comportamento do algoritmo de descrição de imagens acústica quando alterado o parâmetro de distância R_{lig} aplicado a imagem segmentada da Figura 49.	100
Figura 51 – Resultados obtidos pelo algoritmo de comparação de grafo.	101
Figura 52 – Demonstração da ferramenta para marcação manual das regiões semelhantes entre pares de imagens. As linhas em vermelho indicam regiões semelhantes definidas pelo usuário, as linhas em azul indicam regiões semelhantes sugeridas pela ferramenta através do ajuste de uma transformação afim utilizando as regiões semelhantes definidas pelo usuário.	102
Figura 53 – Comparação dos resultados obtidos pelo método proposto com o <i>Ground Truth</i>	103
Figura 54 – Imagens geradas após a comparação dos resultados obtidos com o <i>Ground Truth</i> . As elipses representam as regiões identificadas, em vermelho pela ferramenta de <i>Ground Truth</i> e em amarelo automaticamente pelo método proposto.	104
Figura 55 – Comparação dos resultados obtidos pelo método proposto com o <i>Ground Truth</i>	106

Lista de tabelas

Tabela 1 – Sensores utilizados em AUVs para a navegação	28
Tabela 2 – Modelos de sonar de imageamento frontal (<i>forward looking sonar</i> - FLS) (HURTÓS, 2014)	43
Tabela 3 – Resultados quando alterando os parâmetros ρ_{linear} e $\rho_{recursivo}$	89
Tabela 4 – Resultados obtidos na etapa de construção do grafo.	99
Tabela 5 – Parâmetros adotados para obter os resultados do experimento de com- paração de grafos.	99
Tabela 6 – Parâmetros adotados para obter os resultados do experimento.	105
Tabela 7 – Parâmetros adotados para obter os resultados do experimento utilizando o <i>framework</i> de (ZHOU; TORRE, 2015).	106

Siglas

AUV robô subaquático autônomo, do inglês *Autonomous Underwater Robot*. 19, 20, 25, 46, 47, 99, 100

BFS Algoritmo de busca em largura, do inglês *Breadth First Search*. 50, 51, 81

CSV Do inglês *Comma-separated values*, é um padrão de arquivo de texto geralmente utilizados para representar matriz ou uma planilha de dados onde os valores são separados por vírgula. 74

DGPS Do inglês *Differential Global Positioning System*, é um sistema de posicionamento global (GPS) capaz de obter precisões superiores ao do GPS tradicional, por utilizar múltiplos GPS para a correção do erro de posição. 75

DIDSON *Dual-frequency, IDentification SONar*. 47, 48

DVL do inglês *Doppler Velocity Log*. 20, 46, 47

EKF Filtro de Kalman Extendido do inglês *Extended Kalman Filter*. 47

FLS Sonar de imageamento frontal do inglês *Forwarding Look Sonar*. 23, 25, 27, 46–48, 53, 74, 99

GPS Sistema de Posicionamento Global, do inglês *Global Position System*. 20

HAN *Hungarian Algorithm* - Um algoritmo para a solução do problema de alocação linear. 44

LBL do inglês *Long Baseline*. 20

LBV Do inglês *Little Benthic Vehicle*, é um mine veículo subaquático controlado remotamente, também conhecido como mini-ROV. 7, 75, 76

MSIS do inglês *Mechanical Scanning Image sonar*. 46, 47

NDT *Normal Distribution Transform*. 23, 47

NP Do inglês *Nondeterministic Polynomial time*, é a classe dos problemas de decisões que podem ser verificadas em tempo polinomial. Inclui problemas que não possuem soluções determinísticas com complexidade de tempo polinomial **NP-Completo** e problemas que possuem soluções polinomiais **P**. 11

NP-Completo Do inglês *Nondeterministic Polynomial Complete time*, são os problemas mais difíceis de serem resolvidos da classe **NP**, pois não possuem soluções eficientes conhecidas.. 11, 59

NP-Difícil Do inglês *Nondeterministic Polynomial Hard time*, é a classe dos problemas que podem ser transformados, em tempo polinomial, em problema **NP-Completo**.. 43, 45, 96

P Classe dos problemas de decisão que possuem soluções determinísticas em tempo polinomial.. 44

PDI Processamento Digital de Imagem.. 25, 49

QAP do inglês *Quadratic Assignment Problem*, problema de alocação quadrático. 44, 45, 74, 96, 97, 99

RANSAC *Random Sample Consensus*. 46, 97, 99

ROV robô operado remotamente, do inglês *Remotely Operated Vehicle*. 19, 75

SBL do inglês *Short Baseline*. 20

SIFT *Scale-invariant feature transform*. 22, 47

SLAM do inglês *Simultaneous Localization and Mapping*. 5, 21, 22, 25, 27, 46, 47, 49, 74, 97

SURF *Speeded up robust features*. 22

SVD Decomposição em valor singular, do inglês *Singular Value Decomposition*. 46, 58

TVG Ganho em função da variação do tempo, do inglês *Time Varying Gain*. 34

UML Do inglês *Unified Modeling Language*, é uma linguagem que permite a representação gráfica de um sistema de forma padronizada. 72, 74

USBL do inglês *Ultra Short Baseline*. 20

Lista de Símbolos e Variáveis

Δ_{θ_e} - Variação angular entre as inclinações θ_e de duas arestas adjacentes de um vértice. 14, 70

ϵ_{esc} - Erro de distância ρ_{esc} média entre arestas utilizado como medida de similaridade pelo algoritmo de erro escaleno. 64

θ_e - Ângulo de inclinação de uma aresta com relação ao eixo de maior dispersão da sua gaussiana de origem GA_{orig} . 6, 13, 14, 57–62, 64–66, 68

θ_g - Ângulo de inclinação do eixo de maior dispersão de uma gaussiana com relação ao eixo vertical da imagem. 6, 57–61, 65, 66

μI - A intensidade média de uma amostra de pixel de uma imagem. 57, 60

μX - A média de uma amostra de pixel no eixo X. 57, 58, 60

μY - A média de uma amostra de pixel no eixo Y. 57, 58, 60

π_{fim} - Taxa de variação de intensidade utilizada para calcular H_{fim} . 7, 8, 54, 56, 83–88, 95

$\pi_{recursivo}$ - Taxa de variação de intensidade utilizada para calcular $\rho_{recursivo}$. 7, 8, 54, 56, 83–89, 95

ρ_e - Comprimento de uma aresta com relação ao centro das duas gaussianas envolvidas (GA_{orig}, GA_{dest}). 6, 57–62, 64, 66, 72

ρ_{esc} - Distância entre as posições finais de duas arestas utilizado como métrica para similaridade entre arestas. 13, 15, 63, 64

ρ_{linear} - Limiar de intensidade utilizado para iniciar a extração de um segmento da imagem. 7, 9, 18, 52, 66, 79–81, 83, 89

$\rho_{recursivo}$ - Limiar de intensidade utilizado na extração de um segmento para definir se um pixel pertence ou não ao segmento. 7, 9, 18, 50–54, 56, 79–83, 85, 88, 89

ρ_{ref} - Parâmetro do algoritmo de alinhamento de vértices guloso que determina a diferença de comprimento mínimo entre duas arestas para que sejam consideradas arestas de referência. 70

$\rho_{similar}$ - Restringe o número mínimo de arestas adjacentes semelhantes entre um par de vértices para serem considerados a sua semelhança. 72, 89, 95, 96

$\Sigma Acerto$ - Contagem de arestas semelhantes, considerado no cálculo do erro entre dois vértices. 61–64, 70, 71

σI - O desvio padrão das intensidades de uma amostra de pixel de uma imagem. 57, 60

σX - O desvio padrão de uma amostra de pixel de uma imagem no eixo X. 57, 58, 60

σY - O desvio padrão de uma amostra de pixel de uma imagem no eixo Y. 57, 58, 60

$\epsilon_{ambiguo}$ - Distância mínima entre a similaridade de pares de vértices para que um par de vértices seja considerado semelhantes. Utilizado para eliminar caso em que pares de vértices possuem valores de similaridade muito próximo, pelo algoritmo de cálculo de semelhança entre grafos baseado em heurística. sort. 72, 89, 95, 96

ϵ_{θ_e} - Erro entre variações angulares de arestas adjacentes Δ_{θ_e} utilizado no algoritmo guloso de alinhamento de vértices, $\epsilon_{\theta_e} = |\theta_{e1} - \theta_{e2}|$ sort. 70

ϵ_{gu} - Erro entre variações angulares médio de arestas adjacentes Δ_{θ_e} calculado pelo algoritmo guloso de alinhamento de vértices, utilizado como métrica de similaridade sort. 70

ϵ_w - Erro médio entre vértices calculado pelo algoritmo de erro ponderado, utilizado como métrica de similaridade sort. 61–64

A - Matriz de adjacência de um grafo. 41

D_{seg} - Parâmetro de distância utilizado no algoritmo de extração de segmentos baseado na distância relativa ao segmento válido mais próximo. Determina a quantidade de pixels inválidos explorados. 51, 52, 83, 95

E - Conjunto de arestas de um grafo. 39, 40, 42, 58, 81

E_{adj} - Conjunto das arestas adjacentes, são arestas que parte de um vértice incomum, e.g. $E_{adj}(v_i)$ são as arestas que partem do vértice i . 14, 39, 62, 64, 66

G - Utilizado para representar um grafo. 5, 39–41, 44, 58, 59, 61, 70, 71

GA - Representa uma função gaussiana utilizada como descritor do formato de um segmento. 6, 57–59, 61–64, 66, 67, 70, 72, 73

GA_{dest} - Gaussiana de destino de uma aresta. 13, 58

GA_{orig} - Gaussiana de origem de uma aresta. 13, 58, 59

G_b - Representa um grafo bipartido, um grafo no qual existem dois grupos de vértices U e V onde existem arestas E ligando apenas os vértices entre U e V . 15, 40, 44, 71, 72

G_s - Utilizado para representar um subgrafo. 5, 40, 44

H_{atual} - Variação de intensidade do bin atualmente analisado pela segmentação com base na busca de picos em $beams$ acústicos. 53, 56

H_{busca} - Parâmetro utilizado na segmentação por busca de picos em $beams$ acústicos para determinar quando um pico é utilizado para extrair um segmento. 7, 8, 53, 56, 83–88, 95

H_{fim} - Variação de intensidade mínima de um bin para encerrar a análise de um pico e começar a análise do próximo. 53, 54, 56

H_{total} - Variação de intensidade de um pico detectado na segmentação com base na busca de picos em $beams$ acústicos. 53, 54, 56, 83, 85, 88

I_{max} - Maior intensidade de um pico detectado, utilizado na segmentação com base na busca de picos em $beams$ acústicos. 53, 54, 56

I_{min} - Menor intensidade de um pico detectado, utilizado na segmentação com base na busca de picos em $beams$ acústicos. 53, 54, 56

M_{viz} - Matriz binária utilizada para marcar os pixels segmentados na etapa de detecção de regiões para segmentação. 52

$mx\epsilon_{esc}$ - Distância ρ_{esc} máxima permitida para que duas arestas sejam contabilizadas como similar. 63, 64, 89

$mx\rho$ - Máxima diferença de comprimento permitido entre duas arestas. 62

$mx\theta$ - Máxima diferença de inclinação permitido entre duas arestas. 62

N_g - O número de pixels de uma amostra representado por uma gaussiana. 57, 60

R_{lig} - Parâmetro de distância utilizado pelo algoritmo de construção do grafo topológico para descrição de uma imagem acústica, estabelece a distância máxima entre dois vértices para que uma aresta de ligação seja criada. 8, 59, 88–90, 95

U_b - Primeiro conjunto de vértices de um grafo bipartido G_b . 40, 44

V - Conjunto de vértices de um grafo. 5, 39–42, 58

V_{adj} - Conjunto dos vértices adjacentes, são os vértices que possuem uma ligação por uma única aresta com um vértice incomum, e.g. $V_{adj}(v_i)$ são os vértices que ligam com o vértice i por uma única aresta. 15, 39, 42

V_b - Segundo conjunto de vértices de um grafo bipartido G_b . 40, 44

X - Matriz de correspondência de vértices, é uma matriz binária que estabelece a relação entre dois conjuntos de vértices V_1 e V_2 , sendo que V_1 é representado pelas linhas e V_2 pelas colunas de X , de forma que o elemento $a_{ij} = 1$ da matriz X indica uma relação do vértice i do conjunto V_1 com o vértice j do conjunto V_2 . 5, 40, 41, 44, 45, 92, 93

Sumário

1	INTRODUÇÃO	27
1.1	Descrição do Problema	29
1.2	Motivação	31
1.3	Objetivo Geral	33
1.4	Objetivo Específico	34
1.5	Organização do Trabalho	34
2	REFERENCIAL TEÓRICO	37
2.1	Imagens Acústicas de um Sonar de Imageamento Frontal	37
2.1.1	Formação da Imagem Acústica	37
2.1.2	Principais Desafios	44
2.2	Etapas de um método de processamento de imagem	46
2.3	Grafos	48
2.3.1	Conceitos	49
2.3.2	Formas de representar um Grafo	51
2.3.3	Semelhança entre Grafos	53
2.3.3.1	Problema de alocação linear	54
2.3.3.2	Problema de alocação quadrático	54
2.3.3.3	<i>Factorized Graph Matching</i> e o <i>framework</i> para encontrar semelhança entre grafos	55
2.4	Trabalhos Correlatos	56
3	UMA ARQUITETURA PARA A DESCRIÇÃO E DETECÇÃO DE LOOP EM IMAGENS ACÚSTICAS	59
3.1	Processamento digital de imagem	59
3.2	Segmentação	59
3.2.1	Extração de Segmentos	60
3.2.1.1	Segmentação Completa	61
3.2.1.2	Segmentação de Borda	61
3.2.1.3	Segmentação Relativa a Distância	61
3.2.2	Busca de regiões para segmentação	62
3.2.2.1	Segmentação linear utilizando limiar de intensidade fixa	62
3.2.2.2	Segmentação baseada na busca de picos em <i>beams</i> acústicos	63
3.3	Descritor	66
3.3.1	Representação dos Segmentos	66
3.3.2	Estabelecendo as relações topológicas	68
3.3.2.1	Construção do grafo	69

3.4	Encontrando Imagens Acústicas Semelhantes	69
3.4.1	Similaridade	70
3.4.1.1	Alinhamento Utilizando a Forma Geométrica	71
3.4.1.2	Utilizando topologia para o alinhamento	75
3.4.1.3	Considerações finas	80
3.4.2	Encontrando a semelhança entre grafos	81
3.4.2.1	Semelhança entre grafos baseado em heurística	81
3.4.2.2	Semelhança entre grafos utilizando o <i>framework</i> FGM	82
3.5	Ferramentas	82
3.6	Conclusões Finais	84
4	AVALIAÇÃO DOS MÉTODOS	85
4.1	Dataset ARACATI 2014	85
4.2	Processamento digital de imagem	86
4.3	Segmentação	89
4.3.1	Sensibilidade aos Parâmetros ρ_{linear} e $\rho_{recursivo}$	89
4.3.2	Segmentação de borda vs segmentação relativa a distância e o problema de múltiplas segmentações	92
4.3.3	Segmentação baseada na busca de picos em <i>beams</i> acústicos	93
4.4	Resultados do método de descrição de imagens acústicas	98
4.5	Comparação	99
4.5.1	Teste de Detecção de Similaridade	101
4.5.1.1	Informações do Ground Truth	101
4.5.1.2	Comparação	102
4.6	Considerações finais	106
5	CONCLUSÃO	109
	REFERÊNCIAS	111

1 Introdução

O mar é uma importante fonte de riquezas minerais e biológicas que exerce influências não apenas na economia mas, também na qualidade de vida na terra e da sociedade humana. O mar ocupa mais de 70 % da superfície da Terra e contém cerca de 97 % de toda água do planeta. Estima-se que apenas 5 % dessa área já foi explorada pelo homem (HOW... , 2014).

A robótica subaquática tem servido como uma importante ferramenta para a exploração dos oceanos. Dentre as aplicações pode-se citar o uso de robôs na geologia, biologia, oceanografia, em operações militares e também na indústria de óleo e gás, onde os veículos subaquáticos atuam na exploração de campos de petróleo *offshore* e monitoram operações subaquáticas tais como implantação, manutenção e inspeção de estruturas subaquáticas — tubulações (*pipelines*), plataformas, árvores de natal, etc.(NEW... , 2015; WARNER, 2015; OIL... , 2010; SUSTAITA, 2015; HUAN, 2015).

Os robôs controlados remotamente, conhecidos pela sigla **ROV** (do inglês *Remotely Operated Vehicle*), foram os primeiros robôs subaquáticos a serem utilizados na exploração de águas ultraprofundas. Todas as informações sensoriais, o vídeo, os sinais de controle e a alimentação elétrica de um **ROV** trafegam por um cabo umbilical que o interliga à sua unidade de controle localizada na superfície. A presença do cabo umbilical limita a área de operação do robô e aumenta a complexidade das missões executadas. A utilização de **ROVs** requer operadores, também conhecidos como pilotos de **ROVs**, experientes e o uso de embarcações de suporte. O custo de operação é tipicamente alto (ANTONELLI, 2014; CHRIST; WERNLI, 2014).

Com a evolução dos sistemas de computação, e com o advento de novas tecnologias de sensoriamento subaquático, tornou-se possível o desenvolvimento de robôs subaquáticos autônomos cujo controle passa a ser realizado por sistemas embarcados inteligentes e pré-programados para executar determinadas tarefas. Tais robôs, conhecidos pela sigla **AUV** (do inglês *Autonomous Underwater Vehicle*), são independentes de uma embarcação ou de um piloto durante a sua operação. A ausência de um cabo umbilical amplia consideravelmente a sua área de operação, entretanto as custas de um maior comprometimento quanto ao consumo de energia na qual é armazenada em bancos de baterias recarregáveis. A operação de **AUVs** tipicamente custa menos que a operação de **ROVs**, mas a complexidade da programação dos sistemas de computação embarcados necessários à execução de missões autônomas bem sucedidas é expressiva (ANTONELLI, 2014; GRIFFITHS, 2002).

Soluções robóticas autônomas têm sido desenvolvidas ao longo das últimas décadas para desempenhar diversos papéis importantes na exploração, monitoramento e inspeção

subaquáticas, incluindo a observação de habitats bentônicos, bancos de recifes de corais, mangues e marinas (WILLIAMS; MAHON, 2004). A operação de tais sistemas robóticos envolve inúmeros desafios devido principalmente às características do meio aquoso, que limita o uso e a capacidade de sensores, dificultando a percepção ampla do ambiente. Por exemplo, a rápida absorção das ondas eletromagnéticas de alta frequência pela coluna d'água impede a propagação de ondas de rádio dificultando a comunicação sem fio e impossibilitando a utilização da tecnologia GPS embaixo d'água. Os sensores ópticos também são afetados; o alcance de sensores laser e de imageamento são limitados devido à turbidez da água. Os sensores acústicos são menos sensíveis à turbidez da água e geralmente possuem um grande campo de visão, porém seus dados são de baixa resolução, mais suscetível ao ruído e, muitas vezes, difíceis de serem interpretados (HURTÓ; CUFÍ; SALVI, 2010).

Um dos desafios mais importantes na robótica subaquática autônoma é a navegação. Devido a ausência de sensores proprioceptivos¹, comumente são utilizados sistemas com sensores exteroceptivos² redundantes, que através da fusão dos dados sensoriais, são obtidas informações com maior precisão da posição do robô. A Tabela 1 mostra alguns dos sensores utilizados por AUVs para a navegação classificados em sensores de percepção visual, que capturam imagens do ambiente, e sensores de odometria, que medem o deslocamento do robô com relação ao ambiente.

Tabela 1: Sensores utilizados em AUVs para a navegação

Sensores de Odometria	
Sensor	Variável Medida
Sistema Inercial	Acelerações lineares e angulares
Pressão	Profundidade
Bússola	Campo magnético da terra/Orientação
Sonar Vertical (Altímetro)	Distância até o fundo.
Sonar Horizontal (Varredura de 360°)	Distância até obstáculos.
GPS	Posição absoluta (apenas na superfície).
LBL,SBL,USBL ³	Posição absoluta (limitado a região de operação).
Doppler Velocity Log DVL	Velocidade relativa.
Sensores de Percepção Visual	
Sensor	Variável Medida
Sonar de Imageamento	Posição relativa.
Câmera Óptica	Posição relativa.

¹ Sensores proprioceptivos são sensores que medem informações próprias do robô como o torque dos motores, nível de bateria, etc.

² Sensores exteroceptivos medem informações relativas ao ambiente em que o robô está inserido como distância até os obstáculos, posição absoluta, direção do campo magnético da terra, etc.

³ *Baseline* são sensores acústicos utilizados para rastrear objetos em baixo d'água baseado na distância entre um ou mais transdutores através de ondas acústicas, existem três tipos de *Baseline*: *Long Baseline* (LBL), *Short Baseline* (SBL) e *Ultra Short Baseline* (USBL)

1.1 Descrição do Problema

A capacidade do robô, através de leituras sensoriais, de se auto localizar e mapear o ambiente desconhecido no qual está navegando é chamada de Mapeamento e Localização Simultânea (do inglês *Simultaneous Localization and Mapping (SLAM)*) (DURRANT-WHYTE; BAILEY, 2006). A Figura 1 ilustra o problema de SLAM, um robô autônomo navega em um ambiente desconhecido, a trajetória estimada pelo seu sistema de navegação é representada pela linha em vermelho. A trajetória verdadeira⁴ percorrida pelo robô é representada pela linha verde e é desconhecida pelo seu sistema de navegação. Os triângulos em verde são as suas poses⁵ verdadeiras enquanto que os triângulos em vermelho são as poses estimadas pelo seu sistema de navegação, a cada percepção visual ao longo do tempo. Em azul são representadas as marcações visuais do ambiente.

Inicialmente a trajetória do robô é estimada apenas pelos sensores de odometria, embora as informações visuais também sejam registradas pelo sistema. Entre cada informação visual capturada, várias medições de odometria são realizadas. Em cada uma delas, um erro de estimativa de posição é introduzido e acumulado durante o percurso do robô. A cada informação visual capturada, são extraídas as informações relevantes para demarcar o lugar que o robô está, a percepção dessas marcações visuais pelo sistema de navegação são representadas pelas linhas em laranja.

No instante T-13/P-13 uma região anteriormente visitada é reconhecida pelo sistema de visão, nesse instante o erro acumulado pelos sensores de odometria é estimado e corrigido a partir da percepção visual realizada. Este tipo de detecção é conhecido como detecção de *loop*, ou do inglês *loop closure detection*.

Essas detecções geralmente são realizadas utilizando sensores visuais como câmeras, lasers, ou sonar, sendo que o sonar é amplamente utilizado em ambiente subaquáticos devido a limitação da propagação da luz. A detecção de *loop* consiste basicamente na identificação de um lugar anteriormente visitado, utilizando informações visuais capturadas pelos sensores em tempos e posições diferentes.

Os lugares visitados são descritos pelas estruturas que os compõem, por exemplo, a presença de pedras ou objetos construídos pelo homem, ambos com características marcantes, como as formas ou as cores contrastantes com o resto do ambiente. Essas estruturas são utilizadas como marcadores e são comparadas com as demais marcações dos lugares que serão visitados.

Grande parte dos trabalhos em SLAM terrestre encontrados na literatura (MADDERN; MILFORD; WYETH, 2012), (SIBLEY et al., 2010), (MILFORD; WYETH, 2008),

⁴ Também conhecido pelo termo em inglês *Ground Truth*, é a trajetória confiável do robô, utilizada para a avaliação de métodos estimadores de posição.

⁵ Pose é a combinação da posição e a orientação de um robô com referência a um determinado sistema de coordenadas.

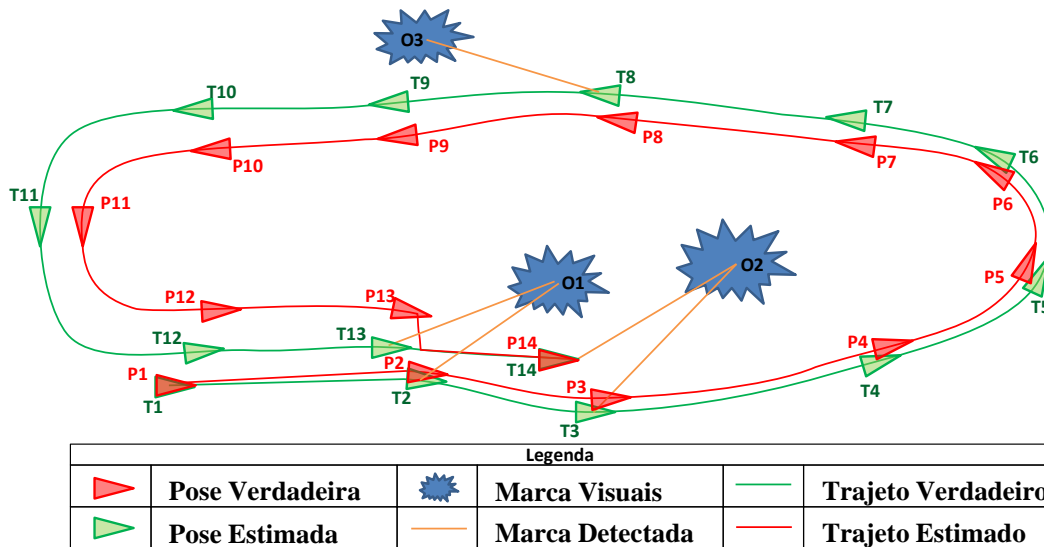


Figura 1: Representação do problema de localização e mapeamento simultâneo (SLAM). Em verde a trajetória verdadeira realizada pelo robô e desconhecida pelo sistema de navegação. Em vermelho a trajetória estimada pelo sistema de navegação. Em azul as marcas visuais do ambiente. Em laranja as marcas visuais percebidas pelo sistema de navegação. Os triângulos indicam a pose no instante em que uma percepção visual é realizada.

(KONOLIGE; AGRAWAL, 2008), fazem uso de imagens ópticas para a detecção de *loop*. Por exemplo, uma descrição simplificada do processo de detecção de *loop* inicia-se através da extração de regiões que contém bordas ou cantos dos objetos presentes na imagem. Em seguida essas regiões podem ser descritas utilizando descritores locais, como o SIFT (LOWE, 2004) ou SURF (BAY; TUYTELAARS; GOOL, 2006), que são invariantes a escala, rotação e translação. Posteriormente calcula-se a semelhança entre os descritores de cada imagem utilizando uma métrica que, nos casos mais simples, pode ser a distância Euclidiana, para definir quais as imagens que representam os mesmos lugares.

Apesar destas abordagens funcionarem muito bem em ambientes terrestres, seu uso é restrito no meio subaquático. A turbidez da água reduz o contraste e a capacidade de extração dos cantos dos objetos em imagens ópticas (GARCIA; GRACIAS, 2011), (DREWS et al., 2013), (CODEVILLA et al., 2015) dificultando a utilização de abordagens do SLAM terrestre no SLAM subaquático. Apesar dos sensores de imageamento acústico não sofrerem com o efeito da turbidez, eles não são capazes de capturar detalhes em alta resolução do ambiente como os sensores de imageamento ópticos. Por esse motivo as abordagens tradicionais aplicadas em imagens ópticas para o SLAM terrestre não apresentam bons resultados quando aplicados em imagens acústicas (HURTÓS et al., 2013).

Portanto alguns trabalhos buscaram novas maneiras de descrever imagens acústicas

considerando a forma dos objetos detectados (RIBAS et al., 2006) (JOHANSSON et al., 2010) (AYKIN; NEGAHDARIPOUR, 2013) ou utilizando as imagens acústicas no domínio da frequência (HURTÓŠ et al., 2012).

1.2 Motivação

Algumas das soluções de navegação autônoma representam o ambiente percebido pelos sensores visuais através de um mapa tridimensional discretizado em um *grid* de células, onde cada célula indica a probabilidade da presença de um objeto no espaço (HORNUNG et al., 2013; KLÄSS; STÜCKLER; BEHNKE, 2012; EINHORN; SCHRÖTER; GROSS, 2011; THRUN; BURGARD; FOX, 2000). Este tipo de representação, em função da granularidade, pode oferecer um alto nível de detalhe do ambiente, sendo ideal para ambientes complexos com uma alta densidade de obstáculos que precisam ser considerados pelo sistema de navegação, como é o caso por exemplo, da navegação para drones e para carros autônomos. Entretanto, este tipo de representação exige uma quantidade expressiva de memória que acaba limitando a capacidade de mapeamento de ambientes amplos.

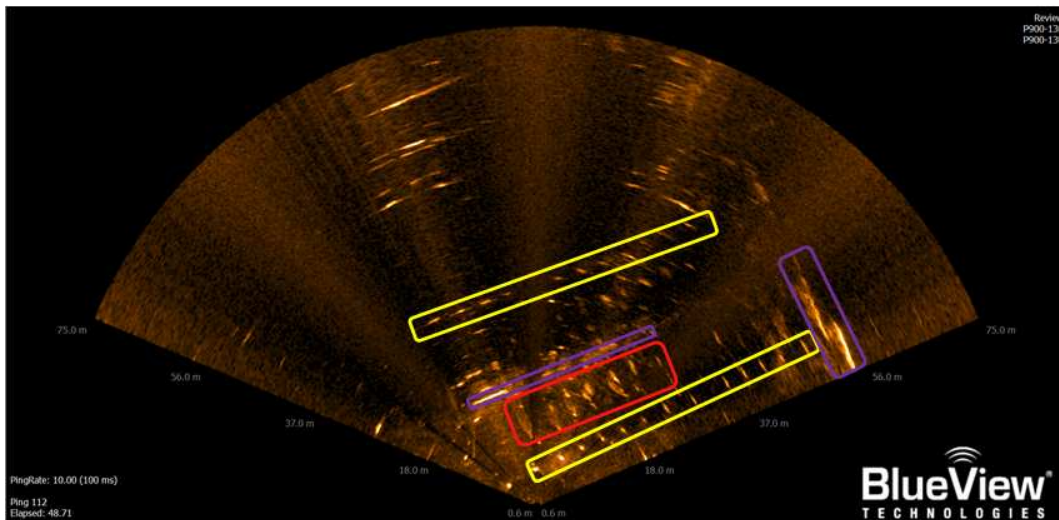
Outras soluções mapeiam o ambiente em um *grid* 2D, onde cada célula é uma função Gaussiana ajustada aos objetos da cena utilizando o algoritmo *Normal Distribution Transform* (NDT) (BIBER; STRASSER, 2003) como em (JOHANSSON et al., 2010). Este tipo de representação oferece a vantagem de ser uma forma genérica de descrever o formato de qualquer objeto, uma vez que um conjunto de funções Gaussianas pode ser ajustado, de forma aproximada, para representar qualquer forma geométrica. Entretanto esse tipo de mapeamento ainda utiliza um *grid* para a representação do ambiente.

No caso dos sonares de imageamento frontal (FLS - Do inglês *Forwarding Look Sonar*), as informações capturadas se resumem em informações de distância e da forma dos objetos presentes no ambiente. Entretanto ocorrem diversos fenômenos relacionados ao seu princípio de funcionamento e ao meio aquoso.

A reverberação de ondas acústicas provoca a presença de objetos fantasma na imagem, as sombras acústicas ou o ângulo de incidência das ondas na superfície dos objetos com quinas muito agudas causam a oclusão de parte da cena, e devido à tecnologia de fabricação do sonar, as imagens acústicas sofrem com o problemas de distorção.

Estes fenômenos dificultam o reconhecimento de assinaturas acústicas próprias de cada marca ou objeto de relevância no mapa, conduzido a um complexo problema de detecção de *loops* invariantes a transformações da imagem, tais como: rotação, translação e escala.

Uma vez que a representação dos objetos de uma cena apresenta-se como um desafio devido aos fenômenos e às informações ruidosas do sonar (FLS). E também pela



(a) Imagem acústica capturada na marina do Yatch Clube de Rio Grande. Nesta imagem é possível identificar cascos de barcos, em vermelho, os postes utilizados para atar os barcos, em amarelo, e o cais em roxo.



(b) Foto da marina do Yatch Clube de Rio Grande.

(c) Imagem de satélite adquirida pelo Google Earth.

Figura 2: Dataset Aracati 2014 disponibilizado por (SILVEIRA et al., 2015).

possibilidade de existirem mais de um objeto com a mesma forma geométrica, busca-se estudar outras formas de extrair mapas de cenários visitados a partir de dados acústicos que permitam a detecção de passagem por locais já visitados.

Por exemplo, mais do que retratar fielmente um determinado objeto percebido, pode-se verificar a presença dos objetos ao seu entorno, utilizando a detecção e registro deste conjunto como uma marcação do ambiente. Essas marcações representariam as relações topológicas entre os elementos presentes na cena e que, devido a configuração de dispersão dos elementos que povoam o ambiente, seria uma característica única daquele lugar.

A Figura 2(a) mostra uma imagem acústica capturada na marina do Yacht Clube de Rio Grande-RS. Nesta imagem é possível identificar vários retornos acústicos de alta intensidade que representam os cascos dos barcos, os postes utilizados para atar os barcos, o cais e algumas pedras no fundo. As figuras 2(b) e 2(c) descrevem o local. Como pode ser visto na imagem acústica, existem diversos objetos cuja forma geométrica é similar,

principalmente objetos pequenos como os postes, e neste caso, uma representação baseada apenas na forma geométrica seria ambígua. Porém, caso as relações topológicas entre elas forem consideradas, o problema de ambiguidade pode ser minimizado, sem a necessidade de discretizar o ambiente em um *grid*, utilizando, por exemplo, um grafo para representar essas relações de topologia.

O laboratório NAUTEC da FURG realiza pesquisas sobre o problema de localização e mapeamento simultâneo (SLAM) para veículos autônomos subaquáticos desde 2003 (BOTELHO et al., 2003; BOTELHO et al., 2004; VARGAS; MADSEN; BOTELHO, 2004; BOTELHO; NEVES; TADDEI, 2005; DREWS et al., 2006; BOTELHO; DREWS; LEIVAS, 2008; DREWS; BOTELHO; GOMES, 2008; BOTELHO et al., 2009; LEIVAS; DREWS; BOTELHO, 2009) utilizando câmeras ópticas como principal sensor de percepção visual. Nos últimos anos foram desenvolvidos trabalhos utilizando sensores acústicos para a percepção visual do ambiente como em (GUTH et al., 2014) onde foi utilizado o sonar mecânico de varredura de 360° MSIS e em (SILVEIRA et al., 2015), onde foi proposto um sistema bio-inspirado híbrido que pode utilizar sensores acústicos ou ópticos para percepção visual. O presente trabalho é o primeiro a tratar especificamente o problema de identificação do fechamento de *loop* utilizando um FLS.

1.3 Objetivo Geral

Devido a importância da exploração subaquática e buscando melhorar os sistemas de navegação de robôs autônomos, AUVs, este trabalho propõe um novo método de descrição e comparação de imagens acústicas de um sonar de imageamento frontal (FLS), que considera as formas dos objetos e as suas relações topológicas visando a representação de descritores mais robustos para a realização de SLAM visual em ambientes subaquáticos. Propõe-se utilizar o modelo Gaussiano para representação da forma dos objetos e grafos para representação das relações topológicas entre os objetos detectados.

A Figura 3 mostra em resumo as etapas do método proposto, as imagens acústicas adquiridas passam pela etapa de processamento digital de imagem (PDI) onde são aplicadas técnicas de melhoramento de imagem. Logo após é realizada a segmentação da imagem, onde são extraídos os grupos de pixels relevantes para a descrição do ambiente onde a imagem foi capturada. Logo após a segmentação, é realizada a descrição dos segmentos utilizando o modelo probabilístico Gaussiano, uma elipse é ajustada a cada segmento extraído e um grafo é construído para representar as relações geométricas entre cada elipse. Na última etapa é realizada a comparação dos descritores, onde são identificadas as imagens que representam a mesma região, ou seja, a detecção de *loop*. A etapa 6 sugere a utilização dos resultados em um sistema de Mapeamento e Localização Simultâneo (SLAM).

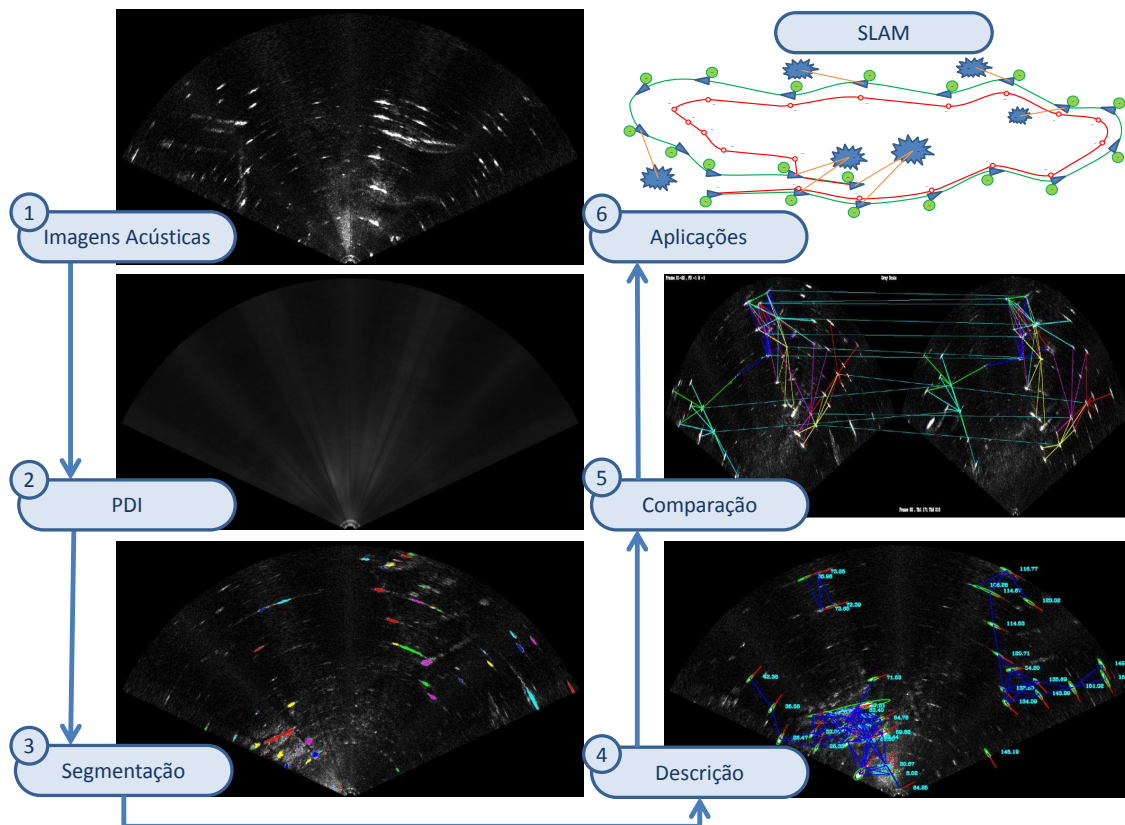


Figura 3: Etapas do método descrição e identificação de imagens acústicas.

1.4 Objetivo Específico

De forma mais precisa o trabalho apresenta os seguintes objetivos específicos:

- Estudar o funcionamento do sonar de imageamento frontal (FLS) e os principais métodos relacionados com a detecção de *loop* utilizando imagens acústicas.
- Propor um método de representação do ambiente baseado na forma geométrica dos objetos detectados e as suas relações topológicas.
- Utilizar grafos como pistas visuais para a detecção de *loop*.
- Aplicar e validar os métodos propostos utilizando datasets de imagens acústicas como o dataset Aracati 2014 disponibilizado por (SILVEIRA et al., 2015).

1.5 Organização do Trabalho

No próximo capítulo será apresentado o referencial teórico utilizado neste trabalho, onde serão tratados assuntos como a formação de uma imagem acústica em um sonar de imageamento frontal, os métodos utilizados para melhoramento e segmentação de

imagens e os trabalhos relacionados. No capítulo 3 será apresentada a arquitetura do sistema proposto, onde cada etapa do método será descrita. No capítulo 4 será avaliado o método proposto mostrando os resultados obtidos em cada etapa. Finalmente no capítulo 5 apresenta-se as conclusões e trabalhos futuros desta dissertação.

2 Referencial Teórico

Neste capítulo será apresentado o referencial teórico necessário para a compreensão deste trabalho. Na seção 2.1 será apresentado o processo de formação de imagens acústicas e as principais características de um sonar de imageamento frontal (FLS). Na seção 2.2 será apresentada as etapas de uma solução de visão computacional proposta por (GONZALEZ; WOODS, 2002) e quais são utilizadas por este trabalho. A seção 2.3 apresentará os principais conceitos de grafos e uma introdução ao problema de encontrar a semelhança entre grafos, e finalmente na seção 2.4 será apresentado os principais trabalhos relacionados a identificação de regiões e alinhamento de imagens acústicas.

2.1 Imagens Acústicas de um Sonar de Imageamento Frontal

Os sonares de imageamento frontal, também conhecidos como *Forward Looking Sonar* (FLS) são dispositivos ativos que utilizam ondas sonoras para perceber o ambiente. Uma das principais vantagens do sonar de imageamento frontal com relação a um sensor de imageamento óptico é a menor sensibilidade às condições de turbidez da água. Entretanto, eles sofrem com outros problemas que não são enfrentados pelos dispositivos de imageamento óptico.

Com relação aos tipos de sonar existentes, os sonares de imageamento frontal merecem destaque principalmente por fornecerem imagens de alta resolução com uma alta taxa de atualização e sofrerem pouca distorção à medida que se movimentam, permitindo a aquisição de dados acústicos em movimento, sendo ideal para ser utilizado em inspeção de equipamento, rastreamento de objetos, desvio de obstáculos e para SLAM.

Nesta seção será apresentada uma breve descrição do funcionamento de um sonar de imageamento frontal, formação das imagens acústicas e os principais desafios de operação.

2.1.1 Formação da Imagem Acústica

Apesar de existir diversos modelos de sonar de imageamento frontal, os quais apresentam diferentes frequências de operação, taxas de atualização, área de cobertura e tecnologias de formação de ondas acústicas (do inglês *beamforming*), veja a Tabela 2, o princípio de funcionamento de todos eles é semelhante.

Esses dispositivos produzem ondas acústicas que se propagam para frente do dispositivo cobrindo um campo de visão em forma de leque, onde existe uma variação angular no sentido azimute (θ) e uma elevação (ϕ), veja na Figura 6. As ondas acústicas se espalham pelo meio até colidir com algum obstáculo ou até serem completamente

absorvidas. Ao colidir com um obstáculo, parte da sua energia é refletida de volta para o sonar. As ondas retornadas são capturadas por um *array* de hidrofones, este processo de ida e volta da onda acústica é conhecido como *ping*.

Os hidrofones são dispositivos que através de um transdutor transformam as variações de pressão provocadas pelas ondas acústicas em sinais elétricos. Geralmente os transdutores são projetados para serem estruturas ressonantes, o que significa que são equipamentos projetados para ser sensível apenas a frequência de operação do sonar.

O sinal produzido durante o processamento de um pulso de onda acústica recebido pelos hidrofones é dividido em diversas partes que são chamadas de *bins*. Conhecendo a velocidade do som na água e medindo o tempo de retorno da onda acústica em cada hidrofone é possível determinar a distância ρ e a direção θ do objeto refletor com relação ao sonar. Desta forma, cada *bin* indica a intensidade da onda acústica em função da distância ρ e direção θ . O conjunto de *bins* numa mesma direção é chamado de *beam*, veja na Figura 4(a). As intensidades acústicas registradas pelo sonar durante um intervalo de tempo Δt são representadas inicialmente por uma imagem bruta em coordenadas polares $I(\theta, \rho)$ onde a resolução depende do número de *beams* do sonar e do número *bins* registrados no intervalo de tempo Δt .

O intervalo de tempo Δt , que determina o tempo de espera entre o envio de cada onda acústica, afeta o número de imagens geradas por segundo (a taxa de atualização), e o alcance da imagem acústica. Quanto maior for o intervalo de tempo Δt , mais tempo a onda tem para se propagar pelo meio, permitindo a captura de ondas refletidas de distâncias maiores.

O número de *bins* registrados durante o intervalo de tempo Δt está relacionado com a frequência das ondas transmitidas. Quanto maior a frequência de transmissão, maior a resolução da imagem e maior a atenuação da onda pela água. Por esse motivo os sonares de alta resolução não possuem um longo alcance.

Conhecendo a resolução angular de cada *beam* e a resolução de distância de cada *bin* é possível converter a imagem crua na forma polar $I(\theta, \rho)$ para uma imagem com significado geométrico, na forma cartesiana $I(X, Y)$. Onde a posição de cada pixel representa a posição do objeto refletor da onda em um espaço 2D, veja as figuras 4(a) e 4(b).

O processo de conversão da imagem polar para a imagem cartesiana gera um problema de resolução não homogênea, onde os *bins* próximos ao sonar são sobrepostos e os *bins* distantes são interpolados. A relação de pixels por *bins* varia de acordo com a distância ρ do *bin*. Quanto maior for a distância ρ mais pixels são utilizados para representar um único *bin*, enquanto que em distâncias ρ pequenas um pixel pode representar mais de um *bin*, veja as figuras 4(a) e 4(b).

A Figura 4 mostra como as imagens acústicas são compostas. Na Figura 4(a) é

mostrada a composição da imagem na forma polar $I(\theta, \rho)$, onde os *beams* são organizados em colunas e os *bins* em linhas, neste tipo de imagem cada pixel é equivalente a um *bin*. A Figura 4(b) mostra a composição da imagem na forma cartesiana $I(X, Y)$. Neste formato de imagem é considerada a cobertura de um *bin* em termos de variação angular $\Delta\theta$ e de distância $\Delta\rho$ resultando em uma imagem em forma de leque onde os *bins* representam regiões de tamanhos diferentes dependendo da sua distância ρ com relação a origem do sonar.

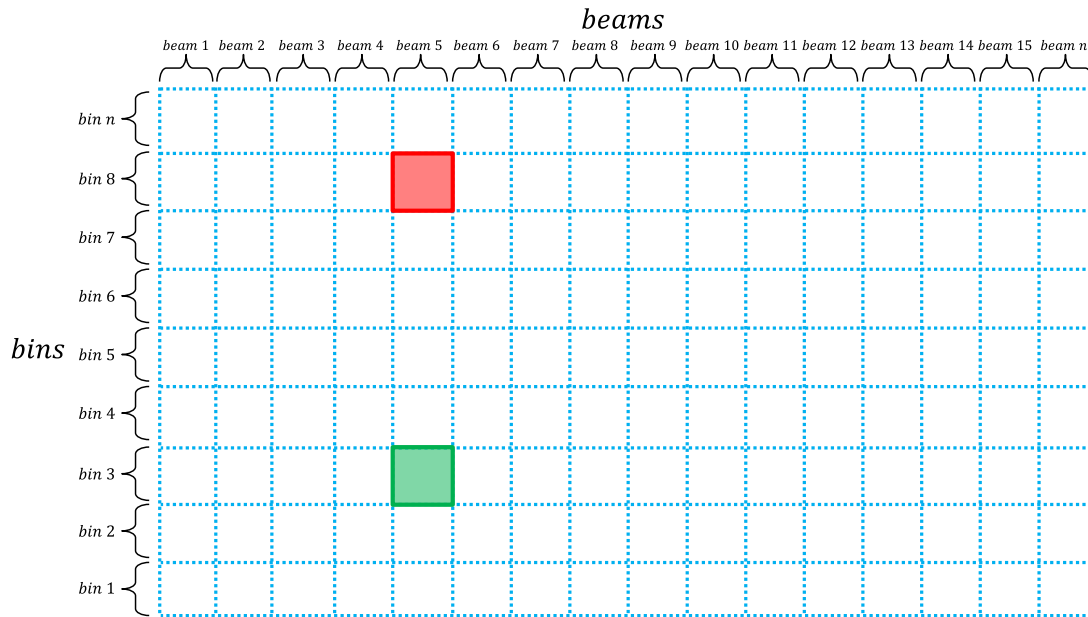
A Figura 5 mostra as duas formas de uma imagem acústica, a forma polar, Figura 5(a), e a forma cartesiana, Figura 5(b), que foram extraídas do dataset ARACATI 2014 disponibilizado por (SILVEIRA et al., 2015). Neste dataset foi utilizado o sonar de imageamento frontal BlueView P900-130 fabricado pela Teledyne Blueview e configurado para cobrir uma distância de 30 metros. Como é mostrado na Tabela 2, este dispositivo é composto por 768 *beams* dispostos a uma distância 0.18° entre eles, cobrindo um leque de 130° de abertura. Portanto a imagem 5(a) possui 768 colunas, uma para cada *beam* e 760 linhas uma para cada *bin* registrando aproximadamente um *bin* a cada 4 cm.

A Figura 6 foi apresentada por (HURTÓ, 2014) e mostra a direção de propagação das ondas transmitidas pelo sonar de imageamento frontal em um cenário onde existe um paralelepípedo no solo. As ondas se propagam em forma de leque varrendo as direções definidas pelo ângulo θ no sentido horizontal e pelo ângulo de inclinação ϕ . A imagem acústica produzida na cena é exibida na parte superior da imagem. Próximo a origem do sonar existe uma região preta que foi desprezada. Logo após é possível notar uma região escura que representa a coluna d'água. Os primeiros retornos acústicos de alta intensidade, representados pelas regiões claras, começam a ser registrados quando o leque acústico intersepta o plano do solo. Nesta região da imagem é possível identificar a forma do paralelepípedo e uma textura que foi formada devido ao relevo do fundo.

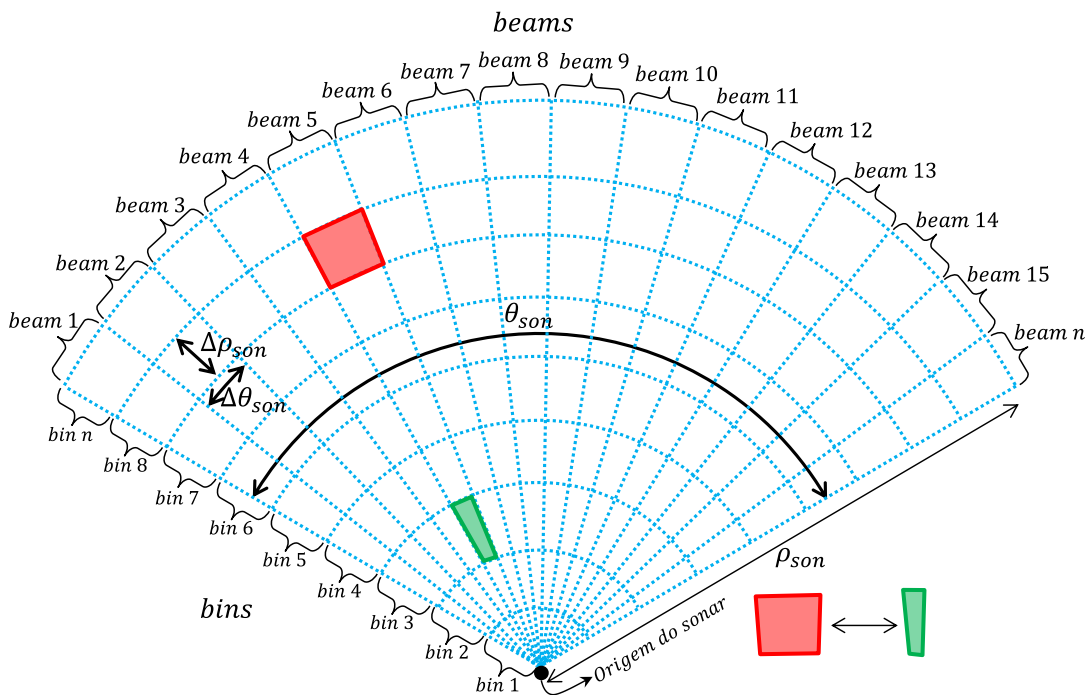
Esta imagem mostra a dificuldade de diferenciar a inclinação ϕ , representada pelo modelo 3D de propagação da onda, nos retornos acústicos registrados pelo sonar. Ou seja, embora algumas ondas acústicas foram refletidas pelo paralelepípedo a uma altura diferente das ondas refletidas pelo solo. Ambos os retornos são representados da mesma forma na imagem acústica. Portanto uma imagem acústica de um sonar de imageamento frontal é uma projeção 2D do ambiente observado, um *bin* na posição (θ, ρ) pode estar em qualquer ponto do arco de inclinação ϕ .

Uma vez que a inclinação ϕ de um *bin* é desconhecida, a sua posição verdadeira é incerta, podendo ocorrer redundância de *bins* provocados por retornos acústicos de diferentes inclinações ou pela variação da distância do sonar até o fundo. Como o arco ϕ aumenta conforme a distância ρ , quanto maior for a distância ρ de um *bin* mais incerta é a sua posição.

Embora essa incerteza exista, ela é desprezada por alguns trabalhos que assumem

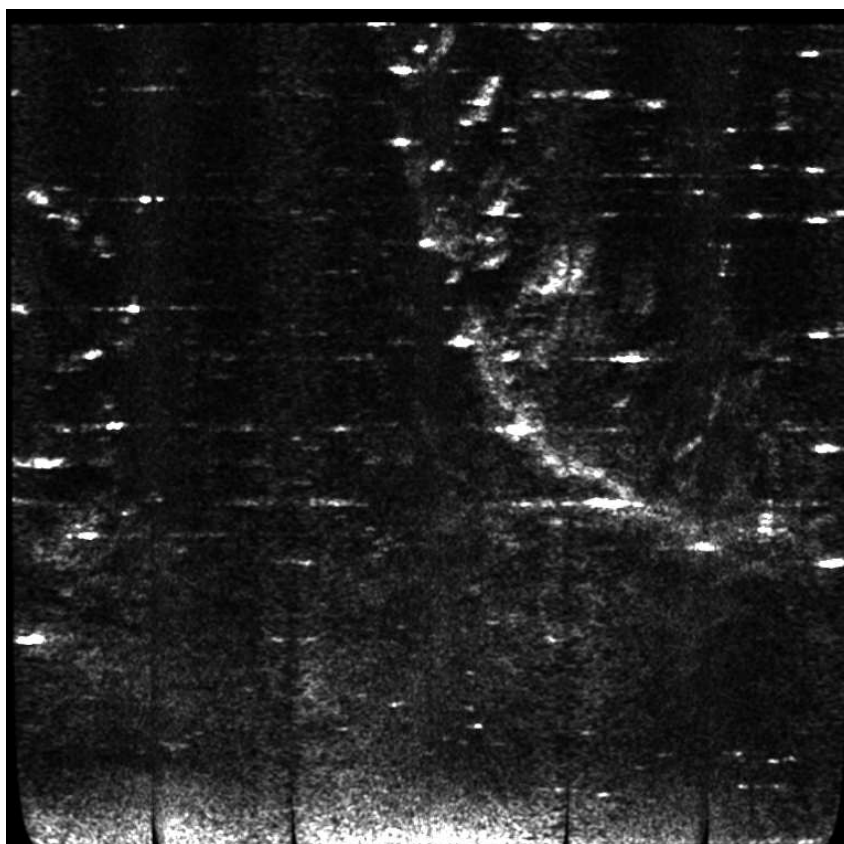


(a) Representação dos *beams* e *bins* em uma imagem acústica de um sonar de imageamento frontal na forma crua, polar.

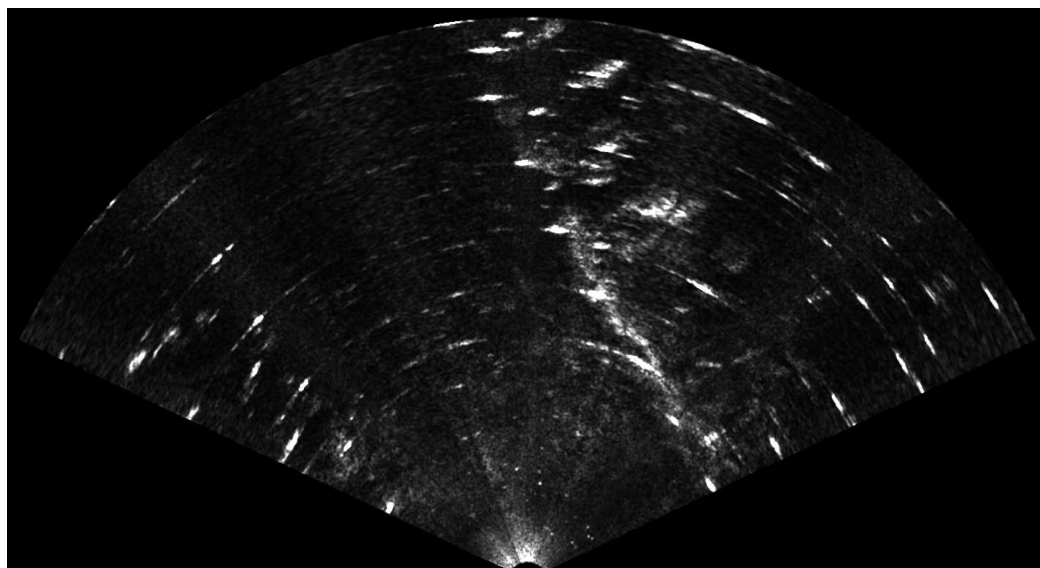


(b) Representação dos *bins* e *beams* em uma imagem acústica na forma cartesiana. Um *bin* cobre uma região $\Delta\theta$ por $\Delta\rho$ e representa a intensidade da onda que foi refletida na posição (θ, ρ) . A área de cobertura de um *bin* varia conforme a sua distância ρ do sonar, este efeito pode ser observado nos *bins* pintados de vermelho e de verde.

Figura 4: Representação de uma imagem acústica de um sonar de imageamento frontal.



(a) Exemplo de imagem acústica crua, na forma polar (*beamxbin*).



(b) Exemplo da mesma imagem convertida para a forma cartesiana.

Figura 5: Imagens acústicas de um sonar de imageamento frontal extraído do dataset ARACATI 2014 ([SILVEIRA et al., 2015](#)).

que a distância do sonar com relação ao solo permanece constante, como é o caso de (JOHANSSON et al., 2010) e (HURTÓS, 2014). Entretanto outros trabalhos, como (AYKIN; NEGAHDARIPOUR, 2013), tentam estimar a inclinação ϕ através da reconstrução 3D da cena utilizando o tamanho da sombra acústica provocada pelos objetos e considerando o solo plano.

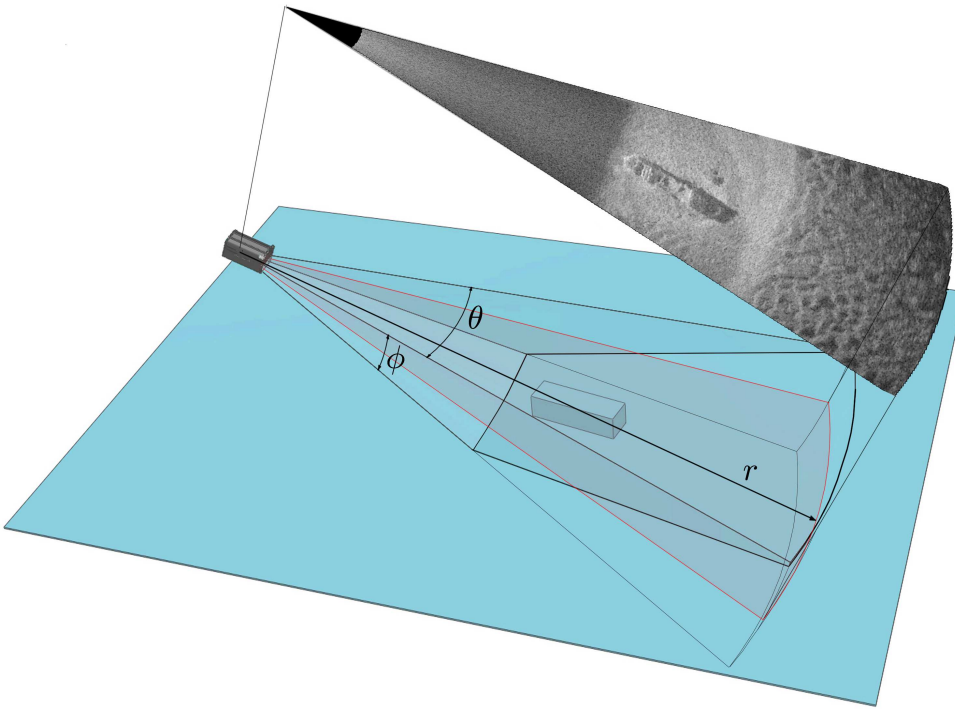


Figura 6: Imagem apresentada em (HURTÓS, 2014) que ilustra como uma imagem acústica é formada. Um modelo 3D representa a cena observada, a pose do sonar e a forma de propagação das ondas acústicas. Ao fundo existe um paralelepípedo. As ondas se propagam para a frente varrendo os arcos azimute θ e de inclinação ϕ até interceptarem o fundo, também representado na imagem. Na parte superior da imagem é mostrada a imagem acústica resultante capturado pelo sonar nesta cena.

Tabela 2: Modelos de sonar de imageamento frontal (*forward looking sonar* - FLS) (HURTÓŠ, 2014)

Modelo	Freq.	Alcance Máximo	Cobertura Angular	Número de <i>Beam</i>	Tamanho do <i>Beam</i> ($\theta \times \phi$)	Taxa Atualiz.	Resolução Distância	Resolução Angular
ARIS Explorer 3000	1.8/3 MHz	0.5-20 m	30°	128	0.25°x14°	15 Hz	0.003 m	0.25°
BlueView P900-130	900 kHz	1-100 m	130°	768	1°x20°	15 Hz	0.025 m	0.18°
BlueView P900-45	900 kHz	1-100 m	45°	256	1°x20°	15 Hz	0.025 m	0.18°
DIDSON	1.1/1.8 MHz	0.4-40 m	29°	96	0.3°x14°	21 Hz	0.0025 m	0.3°
Tritech Gemini i720	720 KHz	0.2-120 m	120°	256	1°x20°	30 Hz	0.008 m	0.5°
Imagenex 965	260 KHz	0.5-300 m	120°	480	1.5°x20°	10 Hz	0.0025 m	0.25°
Kongsberg M3	500 KHz	0.2-150 m	140°	256	0.95°x30°	40 Hz	0.01 m	0.95°
R2Sonar Sonic2024	200-400, 700 KHz	1-400 m	160°	256	1°x20°	60 Hz	0.0125 m	0.5°
RESON Seabat 7128	200/400 KHz	1-500 m	128°	256	0.54°x31°	50 Hz	0.025 m	0.5°

2.1.2 Principais Desafios

Baixa Resolução

Apesar dos sonares de imageamento frontal serem considerados sonares de alta resolução, a sua resolução é baixa quando comparada a resolução das câmeras ópticas. O ângulo de cobertura de um sonar é limitado ao número de transdutores presentes no equipamento enquanto que a resolução depende da frequência de operação do sonar. Um dos sonares de imageamento frontal com maior resolução disponível no mercado é o Sound Metrics ARIS 2013 que opera a uma frequência máxima de 3 MHz, possui uma resolução angular $\Delta\theta$ de 0.25°, resolução no sentido da propagação da onda $\Delta\rho$ de 3 mm com alcance máximo de 10 metros. Um *bin* a 10 metros de distância possui uma resolução de aproximadamente 3 cm por 3 mm. Além da baixa resolução, existe o problema da resolução não homogênea, onde a resolução do *bin* varia conforme a distância, como explicado na seção 2.1.1.

Ruído

Devido à baixa intensidade do sinal os sonares de imageamento frontal sofrem com ruídos, a principal fonte de ruído é a interferência entre as ondas transmitidas que causam a perda da qualidade da imagem. Porém, interferências também podem ser geradas pela vibração dos motores próximos ao sonar ou até mesmo por animais marinhos.

Distribuição Sonora não Uniforme

Os sonares de imageamento frontal sofrem com o efeito de distribuição não uniforme do som, a onda transmitida pelo sonar perde energia a medida que se propaga pelo meio, de forma com que as ondas refletidas de longas distâncias retornam com menos energia do que as ondas refletidas de curtas distâncias. Tipicamente os sonares de imageamento frontal aplicam um ganho no sinal capturado em função do tempo de retorno da onda, este ganho é conhecido por **TVG** (do inglês - *Time Varying Gain*).

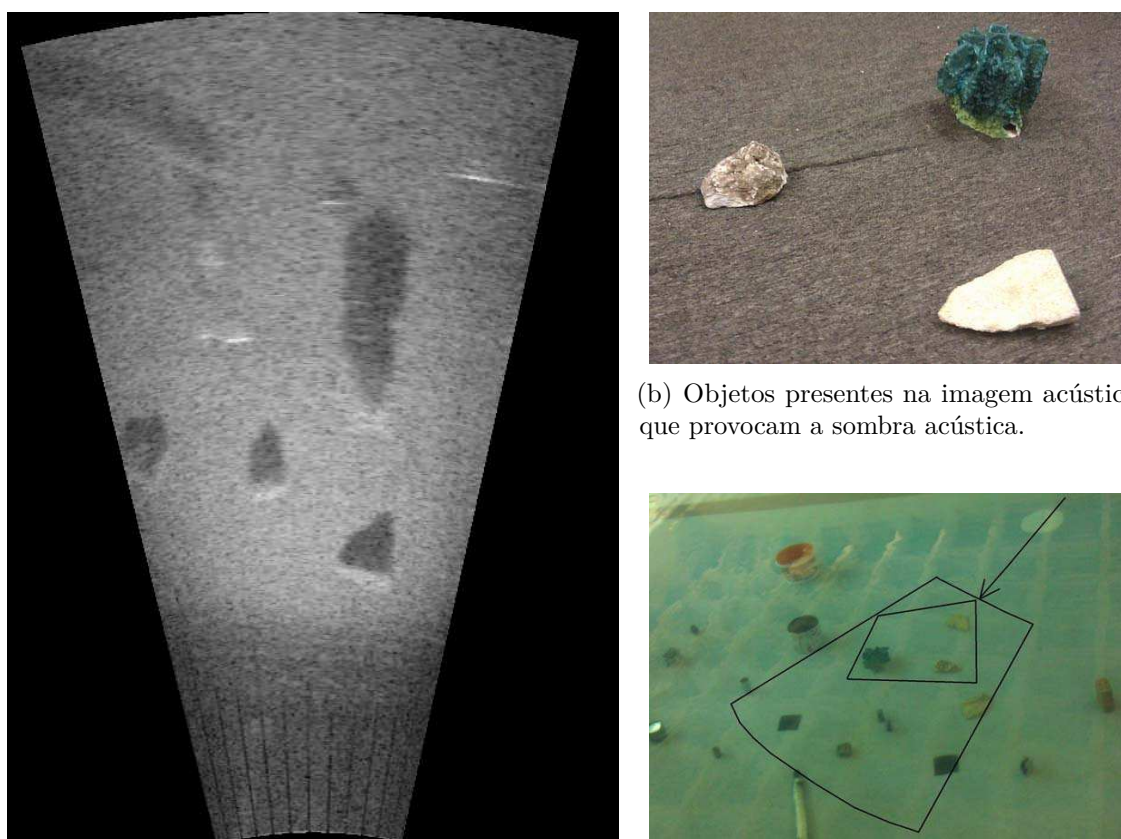
Uma distribuição não uniforme das ondas sonoras também pode ser provocada pela mudança da inclinação do sonar, alterando o ângulo de incidência das ondas acústicas na superfície dos objetos, ou pela diferença de sensibilidade dos transdutores (NEGAHDARI-POUR; FIROOZFAM; SABZMEYDANI, 2005).

O efeito de distribuição sonora não homogênea pode ser reduzido através do levantamento do padrão de distribuição sonora realizando a média de várias imagens consecutivas ou através da equalização da imagem ou ainda utilizando um método de regularização da imagem (KIM; NERETTI; INTRATOR, 2005).

Alteração do Ponto de Vista

A mudança no ponto de vista do sonar causa o movimento das sombras acústicas provocando a oclusão de objetos ou o surgimento de novos objetos que anteriormente estavam cobertos por ela.

A sombra acústica é um fenômeno causado pelo bloqueio da propagação da onda acústica por um obstáculo, causando uma região sem retorno acústico caracterizada por uma mancha escura imediatamente após o obstáculo. A Figura 7 mostra um exemplo de sombra acústica. Elas também podem ser causadas por irregularidades no terreno, como fendas ou crateras. Nesses casos, como não existe um obstáculo bloqueando as ondas, a mancha escura aparece sem retornos acústicos de alta intensidade próximos.



(a) A região escura na imagem é um exemplo de sombra acústica.

(b) Objetos presentes na imagem acústica que provocam a sombra acústica.

(c) Imagem do cenário de teste e a região de cobertura da imagem acústica.

Figura 7: Estas imagens foram apresentadas por Aykin, M. D. e Negahdaripour em (AYKIN; NEGAHDARIPOUR, 2012), onde é proposto um método de reconstrução 3D utilizando sombras acústicas. Foi utilizado o sonar de imageamento frontal DIDSON de 1.8 MHz.

A mudança do ponto de vista do sonar também provoca a alteração no ângulo de incidência da onda na superfície dos objetos alterando a sua forma, tornando os seus retornos mais fracos ou até mesmo provocando o seu desaparecimento por completo, caso o ângulo de incidência seja muito grande.

Outros Efeitos

Outros efeitos podem ocorrer como a reverberação de ondas acústica, quando a onda refletida por um objeto próximo ao sonar, colide no sonar ou no veículo, retorna para o mesmo objeto que reflete novamente para o sonar, causando o registro de múltiplos retornos acústicos do mesmo objeto em tempos diferente durante o registro de uma imagem. Isto causa a replicação do mesmo objeto na imagem em intervalos de distâncias iguais. Este efeito também é conhecido como efeito fantasma e é comum em ambientes com dimensões reduzidas.

Outro fenômeno que também ocorre é a reflexão das ondas na superfície da água. Este efeito ocorre quando o sonar está próximo a superfície e faz com que apareça o relevo da superfície da água nas imagens acústica.

2.2 Etapas de um método de processamento de imagem

Segundo (GONZALEZ; WOODS, 2002) existem duas categorias principais de processamento de imagem, os métodos que recebem como entrada uma imagem, e produzem como saída uma imagem processada, e métodos que recebem como entrada uma imagem e produzem como saída atributos da imagem. A Figura 8 mostra o diagrama proposto por (GONZALEZ; WOODS, 2002) das etapas fundamentais envolvida no processamento digital de imagem.

O módulo de aquisição de imagem refere-se aos métodos utilizados para transformar um sinal analógico em uma imagem digital. A aquisição de imagem também pode ser o simples fato de carregar uma imagem que já esta no formato digital. Este módulo geralmente é o primeiro a ser utilizado no processamento de imagens.

Melhoramento de imagem

No módulo de melhoramento de imagem a imagem é manipulada de forma que ela se torne mais fácil de ser tratada do que a imagem original em um problema específico de processamento de imagem.

Restauração de imagem

O módulo de restauração de imagem também manipula a aparência da imagem, porém diferentemente do módulo de melhoramento de imagem, onde as técnicas aplicadas são subjetivas, a restauração é baseada no que é melhor para resolver um problema em específico. Nesta etapa, as imagens são manipuladas de forma objetiva, geralmente a partir de um modelo matemático ou probabilístico da degradação da imagem.

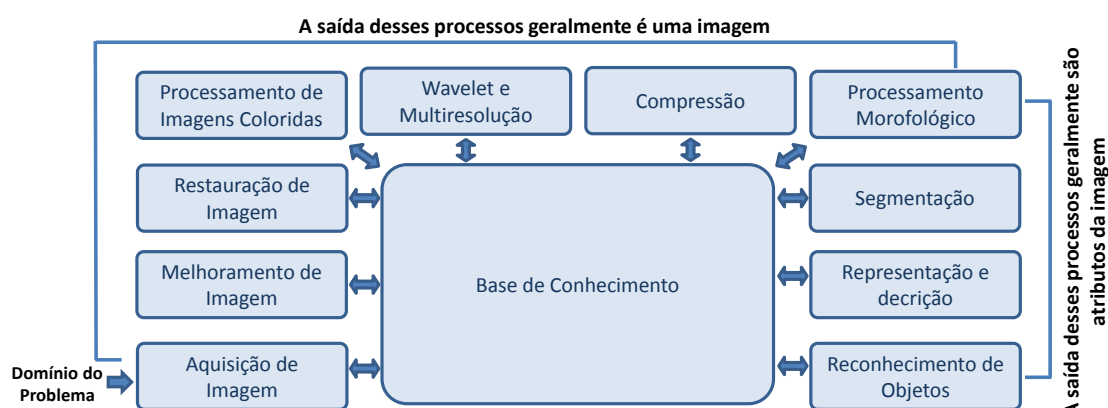


Figura 8: Etapas de processamento de imagem segundo (GONZALEZ; WOODS, 2002).

Processamento de imagens coloridas

O módulo de processamento de imagem colorida trata dos modelos de representação de cores, e é a base para alguns métodos de extração de características de uma imagem baseado na sua textura.

Wavelets e multirresolução

No módulo *wavelets* e multirresolução, são tratadas das técnicas de representação de imagem em várias níveis de resolução. Este módulo geralmente é a base para a compressão de imagem.

Compressão

O módulo de compressão trata das técnicas de redução da quantidade de memória necessária para armazenar uma imagem. Bastante utilizada para otimizar a utilização do espaço em disco e a banda necessária para a transmissão de imagens.

Processos morfológicos

O módulo de processamento morfológico trata dos operadores morfológicos que auxiliam no processo de extração de componentes da imagem e na identificação da sua forma geométrica, utilizados para a descrição de imagens.

Segmentação

No módulo de segmentação são delimitadas as regiões da imagem que representam os objetos que a compõe. A saída desta etapa são os segmentos da imagem que representam

um objeto que compõe a imagem separadamente. Um dos maiores desafios da segmentação de imagem automática é garantir a segmentação de um objeto por um único segmento.

Representação e Descrição

O módulo de representação e descrição geralmente utiliza as regiões segmentadas na etapa de segmentação como entrada. Basicamente este módulo é responsável por transformar uma região segmentada da imagem em uma informação única que resume o segmento como um todo. Por exemplo essa informação pode ser relativa a forma geométrica do segmento ou relacionada com a sua textura.

Reconhecimento de objetos

No módulo de reconhecimento de objetos, as descrições de uma imagem realizada na etapa de representação e descrição, são classificadas e nomeadas. Por exemplo, é reconhecido se um determinado segmento de uma imagem representa um carro, uma casa ou um animal.

A base de conhecimento determina a interação entre os módulos de processamento de imagem dependendo do problema que está sendo tratado, fazendo com que alguns módulos possam ser utilizados mais de uma vez ou fazendo com que alguns módulos não sejam utilizados.

No caso da detecção de *loop* utilizando imagens acústicas proposta neste trabalhos, os módulos de aquisição de imagem, melhoramento de imagem, segmentação e de representação e descrição são utilizados.

2.3 Grafos

Grafo é uma estrutura de dados que foi utilizada inicialmente pelo matemático Suíço Leonard Euler em 1735 para solucionar o problema que ficou conhecido como "O problema das sete pontes de Königsberg". O desafio deste problema era encontrar um caminho que passasse uma única vez por cada uma das 7 pontes que ligavam as 4 ilhas da cidade, a Figura 9 ilustra o problema. Euler modelou o problema utilizando um grafo para representar a topologia das pontes e provou matematicamente que o problema não tem solução (EULER, 1741). Este trabalho é considerado o marco zero dos estudos da teoria dos grafos.

Atualmente os grafos são utilizados para solucionar problemas em diversas áreas como na engenharia, na química, na logística, na computação, etc. Neste trabalho os grafos são utilizados para descrever as relações topológicas das imagens acústicas.

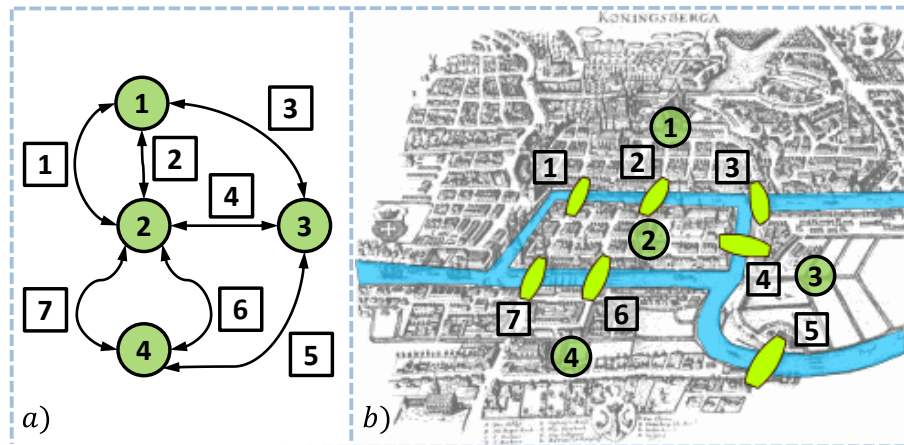


Figura 9: Grafo de Königsberg por Euler. Cada ilha é um vértice, cada ponte é uma aresta. Imagem modificada de (SEVEN...,)

2.3.1 Conceitos

Um grafo é definido como $G = (V(G), E(G))$, onde $V(G)$ é o conjunto não vazio de vértices de G e $E(G)$ é um conjunto de arestas de G . A notação pode ser simplificada para $V(G) = V$ e $E(G) = E$. As arestas estabelecem as relações entre os vértices. Na forma mais básica os elementos de V são número inteiros $\{1, \dots, |V|\}$ e os elementos de E são duplas de inteiros (i, j) sendo que $i, j \in V$ e $|V|$ é o número total de vértices e $|E|$ é o número total de arestas.

As aresta de um grafo podem ser dirigidas ou não dirigidas, ponderadas ou não ponderadas. Graficamente um grafo pode ser representado por um diagrama onde os vértices são representados por círculos e as arestas podem ser representadas por setas, se for um grafo dirigido, ou por linhas, se for um grafo não dirigido. O peso de uma aresta é indicado pelo número ao seu lado. A Figura 10 mostra alguns exemplos de grafos.

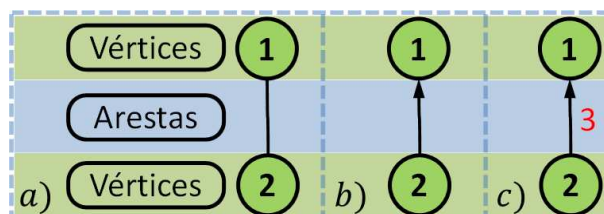


Figura 10: Exemplo de um grafo (a) não dirigido, (b) dirigido (dígrafo) e (c) dirigido e ponderado.

Todas as arestas que partem de um vértice v_i são ditas arestas adjacentes de v_i , $E_{adj}(v_i)$, assim como todos os vértices ligados por arestas adjacentes de v_i são ditos vértices adjacentes de v_i , $V_{adj}(v_i)$. Um grafo é dito simétrico quando, para cada aresta que liga os vértices (u_i, v_j) , existe uma aresta oposta que liga (v_j, u_i) . Um grafo é geométrico quando

os seus vértices e as suas arestas estão associadas a um objeto geométrico.

Um subgrafo G_s de G é um grafo cuja o conjunto de vértices $V(G_s)$ é um subconjunto de $V(G)$, isto é $V(G_s) \subset V(G)$, e o conjunto de arestas $E(G_s)$ é um subconjunto de $E(G)$, isto é $E(G_s) \subset E(G)$. A Figura 11 mostra um subgrafo.

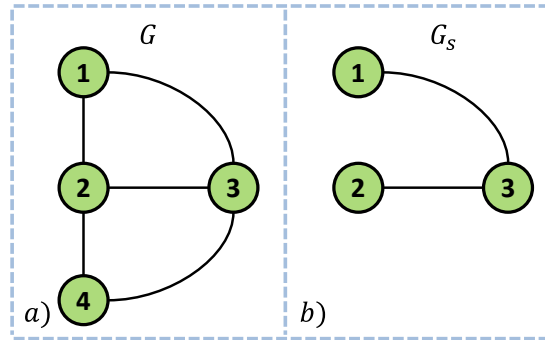


Figura 11: Exemplo de um grafo G em (a) e o seu subgrafo $G_s(G)$ em (b).

Um grafo bipartido $G_b = (U_b, V_b, E)$, é um grafo que pode ser dividido em dois conjuntos disjuntos de vértices U_b e V_b , isto é, U_b e V_b são conjuntos independentes, de forma que cada aresta liga sempre um vértice do conjunto U_b com um vértice do conjunto V_b ou vice versa.

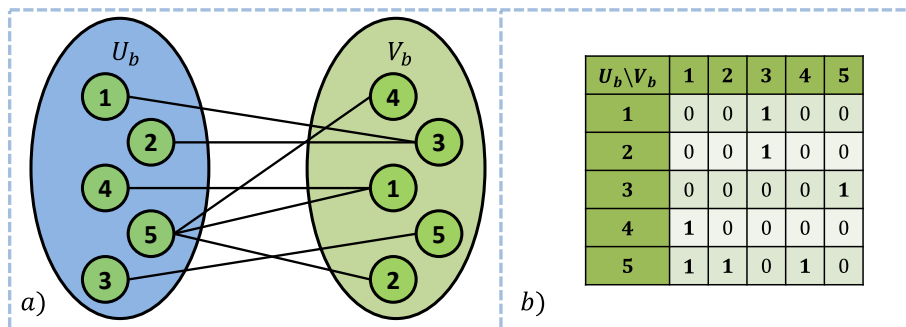


Figura 12: Exemplo de um grafo bipartido em (a) e a sua representação em forma matricial em (b).

Dois grafos G_1 e G_2 são isomórficos quando é possível criar uma função de mapeamento bijetiva $f : V(G_1) \leftrightarrow V(G_2)$ tal que $(v_x, v_y) \in E(G_1)$ somente se $(f(v_x), f(v_y)) \in E(G_2)$. A função de mapeamento pode ser representada por uma matriz de correspondência de vértices X , onde as linhas representam os vértices de G_1 e as colunas os vértices de G_2 , sendo uma matriz binária onde o elemento $a_{ij} = 1$ indica a relação do vértice i de G_1 com o vértice j de G_2 . Um exemplo de dois grafos isomórficos é apresentado pela Figura 13.

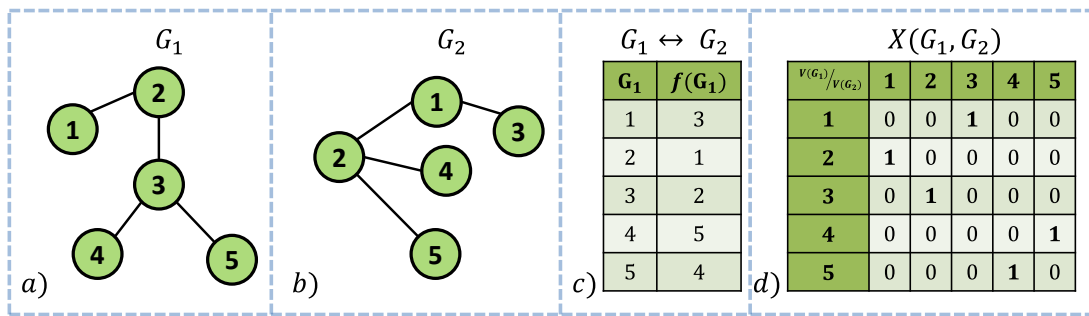


Figura 13: Exemplo de dois grafos isomórficos em (a), a sua função de mapeamento $f : V(G_1) \leftrightarrow V(G_2)$ em (b) e a matriz de correspondência de vértices X em (c).

2.3.2 Formas de representar um Grafo

Além da forma gráfica, que facilita a sua visualização, os grafos também podem ser representados matematicamente, facilitando a sua análise pelo computador. Cada forma de representação possui uma vantagem, a escolha sua escolha depende da aplicação.

Matriz de Adjacência

A representação de um grafo G por matriz de adjacência utiliza uma matriz quadrada A de ordem $|V(G)|$ onde o elemento a_{ij} representa a existência de uma aresta ligando o vértice i ao vértice j do grafo. Em um grafo não ponderado a matriz A é binária, onde 1 indica a presença de aresta e 0 a ausência. No caso da representação de grafos ponderados, a ponderação das arestas é dado pelo valor do elemento da matriz. A Figura 14 mostra a representação por matriz de adjacência de um grafo não ponderado.

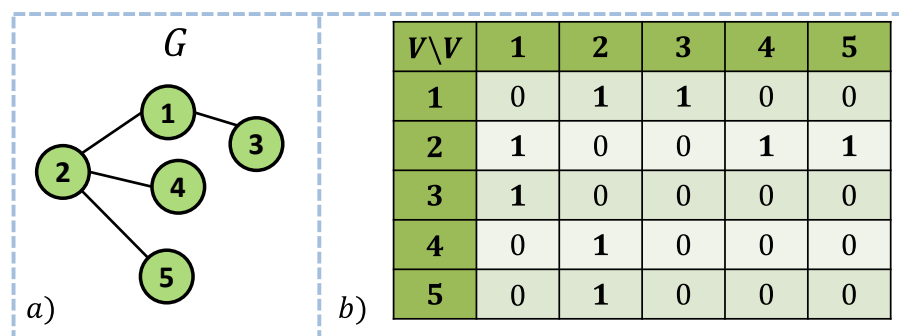


Figura 14: Representação de um grafo por matriz de adjacência, em (a) o grafo e em (b) a sua representação em matriz de adjacência.

Computacionalmente, uma vez que os seus vértices estão indexados em forma de linha e coluna na matriz. A representação de grafos por matriz de adjacência apresenta a

vantagem de ser possível a realização de consultas de arestas em tempo constante, ou seja, com complexidade $O(1)$. Facilitando a edição e a manipulação do grafo. Entretanto, esta representação não é indicada para a representação de grafos esparsos, devido ao consumo de memória ser quadrático em função do número de vértices. Outra desvantagem desta representação é a busca por vértices adjacentes V_{adj} ser linear em função do número de vértices do grafo, complexidade $O(|V|)$.

Listas de Adjacências

A representação de grafos por listas de adjacências utiliza uma lista encadeada simples para cada vértice do grafo. As arestas adjacentes de cada vértice é armazenada na sua respectiva lista encadeada.

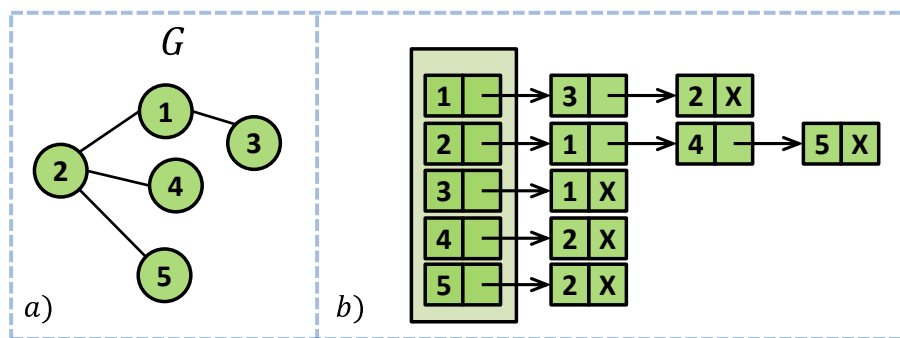


Figura 15: Representação de grafo por lista de adjacência.

Esta representação é mais indicada para grafos esparsos, uma vez que a utilização de memória é linear em função do número de arestas, $O(|E|)$. A busca pelas arestas adjacentes a um determinado vértice é mais eficiente do que na representação por matriz de adjacência. Uma vez que apenas as arestas adjacentes são armazenadas na lista encadeada de cada vértice, a sua complexidade de busca é linear em função do número de arestas adjacentes. Entretanto, a consulta por arestas existentes é menos eficiente.

Matriz de Incidência

A representação de um grafo não dirigido por matriz de incidência utiliza uma matriz $Gn \times m$ onde n é o número de vértices $|V|$ e m é o número de arestas $|E|$ do grafo. Cada coluna representa uma aresta e possui apenas dois valores 1, um em cada linha que representa os vértices de origem e de destino da aresta. A Figura 16 mostra a representação de um grafo não dirigido por matriz de incidência.

Os grafos dirigidos são representados por duas matrizes de incidência G e H , com a diferença que a matriz G descreve apenas os vértices de origem das arestas e a matriz H representa apenas os vértices de destino das arestas. A Figura 17 mostra a representação

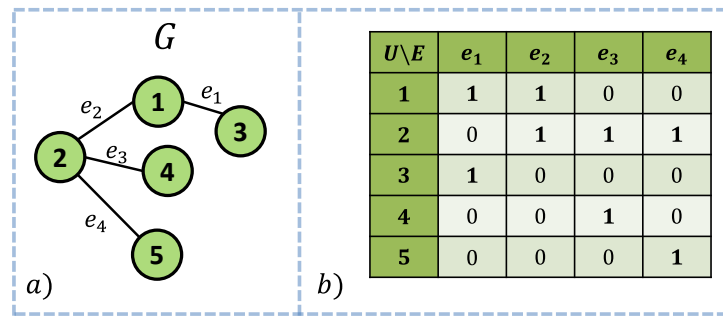


Figura 16: Representação de um grafo não dirigido por matriz de incidência.

de um grafo dirigido por matriz de incidência. Esta representação foi utilizada por (ZHOU; TORRE, 2013; ZHOU; TORRE, 2015) no seu método de encontrar semelhança entre grafos utilizando fatoração.

Alguns autores utilizam a representação de matriz de incidência para grafos dirigidos utilizando apenas uma matriz, neste caso é realizado a subtração da matriz G pela matriz H . Os valores negativos indicam os vértices de destino das arestas enquanto que os valores positivos indicam os vértices de origem das arestas.

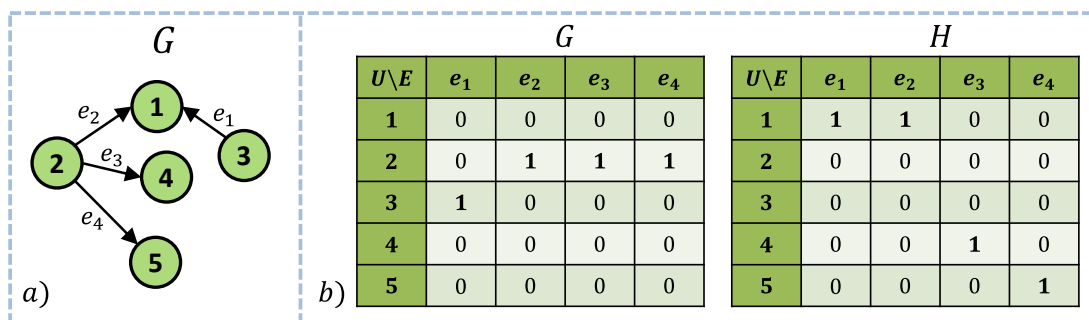


Figura 17: Representação de grafo dirigido por matriz de incidência.

2.3.3 Semelhança entre Grafos

O problema de similaridade entre grafos pode ser considerado uma aproximação para o problema de isomorfismo de grafos, com a diferença que é considerado que dois grafos podem ser aproximadamente iguais. Apesar do problema de encontrar a semelhança entre grafos ser NP-Difícil (ZHOU; TORRE, 2015), existem diversas abordagens propostas na literatura para tratar o problema.

2.3.3.1 Problema de alocação linear

O problema de encontrar a similaridade entre grafos pode ser simplificado para o problema de alocação linear, também conhecido do inglês como *Assignment Problem*, é um problema de otimização que minimiza a alocação de tarefas.

Tipicamente, o problema de alocação linear é representado por um grafo bipartido G_b onde as tarefas pertencem ao conjunto de vértices U_b e as alocações ao conjunto de vértices V_b . O peso das arestas entre os vértices U_b, V_b estabelecem os custos das alocações de cada tarefa. A Figura 18 ilustra o problema. Os vértices do conjunto U_b podem representar as tarefas a serem realizadas, e os vértices do conjunto V_b podem representar os trabalhadores aptos a realizar as tarefas. O custo das arestas indica o valor que cada trabalhador cobra para realizar cada tarefa. A alocação linear minimiza o custo para a realização das 3 tarefas alocando um trabalhador em cada tarefa. A solução do problema está representada pelas arestas em vermelho, o menor custo encontrado para realizar as três tarefas é $4 + 6 + 4 = 14$ e a alocação de cada trabalhador é representado pela matriz $X(U_b, V_b)$.

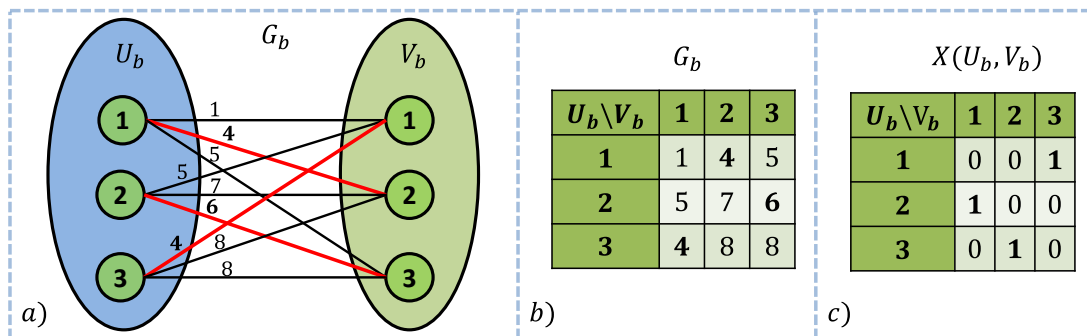


Figura 18: Problema de alocação linear.

Para encontrar a similaridade entre dois grafos G_1, G_2 , basta criar um grafo bipartido ponderado e completo G_s onde os vértices de G_1 são representados pelo conjunto de vértices $U(G_s)$ e os vértices de G_2 são representados pelo conjunto de vértices $V(G_s)$. O peso das arestas representa a semelhança entre o par de vértice dos dois grafos.

O problema de alocação linear pertence a classe dos problemas **P** cuja a solução pode ser encontrada em tempo polinomial determinístico pelo *Hungarian Algorithm* (HAN) (KUHN, 1955) que em sua versão otimizada tem uma complexidade de tempo $O(|V|^3)$.

2.3.3.2 Problema de alocação quadrático

O problema de encontrar a semelhança entre grafos também pode ser modelado como um problema de alocação quadrática QAP (ZHOU; TORRE, 2013), do inglês

Quadratic Assignment Problem, no qual pertence a classe dos problemas NP-Difícil, porém existem diversas soluções aproximadas propostas na literatura (ZHOU; TORRE, 2015).

Diferente do problema de alocação linear que considera apenas a similaridade de primeira ordem, entre vértices, o problema de alocação quadrático QAP considera as semelhanças de segunda ordem, entre vértices e entre arestas.

O problema de similaridade de grafos foi modelado por Lawler como QAP através da Equação 2.1 (LAWLER, 1963). Onde K é a matriz que estabelece a similaridade entre todos os vértices e entre todas as arestas dos dois grafos comparados. A matriz X^1 determina a correspondência entre os vértices dos dois grafos comparados. A maximização da equação resulta numa matriz X , cuja a seleção dos vértices correspondentes possuem a máxima similaridade.

$$\begin{aligned} \max \quad & \text{vec}(X)^T K \text{vec}(X) \\ X \in \Pi \end{aligned} \quad (2.1)$$

O problema de similaridade entre grafos também foi modelado como QAP por Koopmans-Beckmann utilizando a equação 2.2. De forma similar a equação 2.1, a matriz X^2 determina a correspondência entre os vértices, a matriz K_p estabelece a similaridade entre vértices e as matrizes de adjacência ponderadas A_1 e A_2 estabelecem a topologia dos dois grafos.

$$\begin{aligned} \max \quad & \text{tr}(K_p^T X) + (A_1 X A_2 X^T) \\ X \in \Pi \end{aligned} \quad (2.2)$$

Diversas estratégias para otimizar o problema de QAP foram propostas utilizando relaxamento espectral, *semidefinite programming* SDP e *Doubly-stochastic Relaxation*, uma revisão sobre os principais métodos para a solução do problema e as formulações que foram propostas foi feita em (LOIOLA; ABREU; NETTO, 2004; LOIOLA et al., 2007).

2.3.3.3 Factorized Graph Matching e o framework para encontrar semelhança entre grafos

Em (ZHOU; TORRE, 2015) foi proposto o método *Factorized Graph Matching* (FGM) no qual é apresentado um método de fatoração das matrizes da equação de Lawler que modela o problema de semelhança de grafos como um QAP. Neste trabalho também foi desenvolvido um *framework* em Matlab que reúne a implementação de algumas das principais abordagens para o cálculo de similaridade entre grafos como o *Hungarian algorithm* para resolver o problema de alocação linear, *Graduated assignment* (GA)(GOLD; RANGARAJAN, 1996), *Spectral Matching* (LEORDEANU; HEBERT, 2005), *Spectral*

¹ $\text{vec}(X)$ é a vetorização da matriz X

² $\text{tr}(X)$ é o traço da matriz X

Matching com transformação afim (COUR; SRINIVASAN; SHI, 2007), *Probabilistic Matching* (ZASS; SHASHUA, 2008), *Integer Projected Fixed Point Method* (LEORDEANU; HEBERT; SUKTHANKAR, 2009), *Re-Weighted Random Walk Matching* (CHO; LEE; LEE, 2010) e o próprio *Factorized Graph Matching* (ZHOU; TORRE, 2013; ZHOU; TORRE, 2015).

O método de fatoração de matriz proposto em (ZHOU; TORRE, 2015) encontra a similaridade de grafos não dirigidos e de grafos dirigidos. A diferença é que as características que definem as arestas nos grafos dirigidos não são simétricas. Por esse motivo a representação proposta por Feng Zhou utiliza 6 matrizes. Sendo que as primeiras 4 são utilizadas para a representação da estrutura dos dois grafos comparados, duas matrizes de incidência binárias para cada grafo dirigido, são as matrizes G e H apresentadas na seção 2.3.2 e demonstrada na Figura 17. As outras duas matrizes definem a similaridade entre vértices, nomeada K_p e a similaridade entre arestas nomeada K_q .

As duas matrizes, K_p e K_q , são indexadas de forma que o valor do elemento na linha i e na coluna j indica a similaridade entre o vértice i do grafo 1 com o vértice j do grafo 2 para a matriz K_p e a similaridade da aresta i do grafo 1 com a aresta j do grafo 2 para a matriz K_q .

O *framework* de (ZHOU; TORRE, 2015) também contribuiu para a unificação das soluções propostas na literatura para encontrar semelhança entre grafos, definido as transformações do método de fatoração para as demais soluções de forma que definindo as 6 matrizes é possível utilizar todas as abordagens do implementadas no *framework*.

2.4 Trabalhos Correlatos

Nesta seção serão apresentados alguns trabalhos relacionados com o registro e a identificação de imagens acústicas semelhantes.

Em (NEGAHDARIPOUR; FIROOZFAM; SABZMEYDANI, 2005) é apresentado as principais características e desafios dos FLSs, como corrigir a distorção das imagens do sonar DIDSON 1.8 MHz. Também é apresentado um método de registro de imagens acústicas baseado na extração de cantos utilizando o detector de cantos Harris (HARRIS; STEPHENS, 1988) e um método baseado no RANSAC-SVD para calcular a homografia entre imagens. O método obtém resultados considerado instáveis pelo autor.

Em (KIM; NERETTI; INTRATOR, 2005) é utiliza o extrator de cantos Harris, usando o terceiro nível da piramide Gaussiana. As imagens são alinhadas através da correlação dos descritores junto com um algoritmo de mosaico.

Em (RIBAS et al., 2006) é apresentado um método de SLAM para AUVs em ambientes parcialmente estruturados equipados com um sensor de velocidade DVL e um sonar mecânico de varredura de 360° (do inglês *Mechanical Scanning Image sonar* (MSIS))

). As distorções das imagens acústicas geradas pelo movimento do robô são corrigidas pela estimativa de deslocamento de um filtro de Kalman Extendido (EKF) utilizando um sensor de velocidade DVL.

As imagens acústicas são descritas através da detecção de linhas usando a transformada de Hough (BALLARD, 1981). Antes de calcular a transformada de Hough é realizada a remoção de ruído das imagens eliminando os pixels de baixa intensidade e selecionando os máximos locais. As linhas são extraídas da imagem utilizando um esquema de votação. Devido ao atraso do sonar MSIS para realizar a varredura completa da cena, o SLAM precisa ser atrasado para que as correções visuais possam ser realizadas. Para determinar quando duas linhas de duas imagens acústicas consecutivas representam o mesmo ambiente na cena, é utilizado um teste de compatibilidade de linhas, onde uma equação é utilizada como métrica de comparação. Um mapa é construído com base nas linhas detectadas. Um experimento real é realizado utilizando o AUV GARBI da Universidade de Girona.

Em (JOHANSSON et al., 2010) é apresentado um método de SLAM para a vigilância de portos e inspeção de casco de navios autônoma. O método utiliza informações de odometria combinado com um sonar DIDSON de 1.8 MHz. As correções de posição pela visão é estimada através do alinhamento das imagens acústicas, onde é realizada uma densa extração de *features* das imagens.

Inicialmente é utilizado um filtro mediano para eliminar o ruído das imagens e afetando o mínimo possível as suas bordas. Logo após é calculado o gradiente da imagem. Os pixels com gradiente superior a um determinado limiar são considerados *features* da imagem. O limiar é adaptado automaticamente em função do número de *features* encontrado, de forma que sempre seja extraída a mesma quantidade de *features* de cada imagem. Depois que todas as *features* foram extraídas, é realizada uma etapa de eliminação de *features* ruins *clusterizando* todas as *features* e eliminando os *clusters* pequenos.

O alinhamento das imagens é encontrado utilizando as *features* extraídas no algoritmo *Normal Distribution Transform*(NDT)(BIBER; STRASSER, 2003).

Em (NEGAHDARIPOUR; AYKIN; SINNARAJAH, 2011) é mostrado os problemas enfrentados e a complexidade de encontrar a semelhança entre imagens acústicas. Foi demonstrado a dificuldade do detector SIFT (LOWE, 2004), considerado estado da arte para detecção de semelhanças entre imagens ópticas, para detectar semelhança entre imagens acústicas.

Em (AYKIN; NEGAHDARIPOUR, 2012) é apresentado um método de extração de *features* de imagens acústicas de um sonar de imageamento frontal (FLS) considerando o modelo de projeção 3D do sonar, ou seja, é estimado a elevação ϕ da imagem acústica de acordo com a inclinação do sonar e o tamanho das sombras acústicas dos objetos detectados na cena. Este método é aplicado em regiões planas.

Inicialmente é realizado uma compensação de sensibilidade dos hidrofones do sonar, esta compensação pode variar de acordo com o sonar utilizado, neste caso foi utilizado o sonar [DIDSON](#) 1.8 MHz. Logo após são selecionados os pixels de alta intensidade da imagem. Esses pixels são *clusterizados* utilizando o algoritmo K-Means adaptativo proposto pelo autor, os *clusters* pequenos são eliminados utilizando o *blob size filter*.

Basicamente o *K-Means* adaptativo utiliza dois parâmetros, a distância mínima entre dois agrupamentos T_d , e o tamanho máximo de um agrupamento T_σ . Esses parâmetros são utilizados durante as interações do algoritmo, quando a distância entre o centro de dois agrupamentos for menor que T_d os agrupamentos são agrupados, ou quando o tamanho de um agrupamento for maior que T_σ ele é dividido.

A detecção da sombra acústica de cada objeto é realizada primeiro determinando dois limiares de intensidade, um para determinar se um pixel pertence a sombra acústica ou se pertence ao fundo do cenário, e outro para determinar se um pixel pertence ao fundo do cenário, ou se representa um objeto da cena. Esses limiares são encontrados baseado na distribuição de intensidade dos pixels ao redor do objeto detectado. A análise dos pixels que representam a sombra acústica é realizada nos perfis de intensidade, suavizados, de cada *beam* na imagem crua $I(\theta, \rho)$.

Em ([HURTÓŠ et al., 2012](#)) é proposto um método de alinhamento de imagens acústicas de um ([FLS](#)) utilizando as imagens no domínio da frequência. É utilizado o algoritmo de correlação de fase ([REDDY; CHATTERJI, 1996; CASTRO; MORANDI, 1987](#)). Este método de correlação de fase demonstra-se ser robusto ao ruído, diferença de iluminação e oclusão presentes nas imagens acústicas.

Segundo os trabalhos apresentados, percebe-se que os métodos tradicionais para encontrar semelhança entre imagens ópticas apresentam limitações para imagens acústicas. Por este motivo diversas abordagens foram propostas, baseado no gradiente das imagens, no formato dos objetos detectados ou na correlação entre a fase das imagens no domínio da frequência. Essas abordagens são consideradas o estado da arte para o registro e o alinhamento de imagens acústicas.

3 Uma arquitetura para a descrição e detecção de *loop* em imagens acústicas

Nesta seção será apresentada a arquitetura proposta para a descrição e identificação de imagens acústicas semelhantes para a detecção de *loop*, mostrada na Figura 3. As imagens são adquiridas na primeira, logo após passam por 4 etapas de processamento, a Seção 3.1 descreve a etapa de PDI, onde é realizado o melhoramento das imagens. A Seção 3.2 descreve a etapa de segmentação, onde são extraídas as informações relevantes para a identificação das imagens, como por exemplo, os objetos que compõem a cena observada. Na Seção 3.3 é apresentado o método de descrição das imagens, as informações segmentadas são descritas pelo modelo probabilístico Gaussiano e utilizado para representar a forma dos objetos extraídos. Um grafo é criado para representar as relações topológicas entre as Gaussianas. A última etapa de processamento é a de comparação, descrita na Seção 3.4. Nesta etapa a descrição de cada imagem é comparada e são identificadas as imagens semelhantes. A última etapa prevista na figura 3 é a de aplicação onde o trabalho proposto é utilizado como detector de *loop* de um algoritmo de SLAM. A seção 3.5 descreve as ferramentas utilizadas para a implementação do método proposto e também mostra como o código foi organizado. Finalmente a seção 3.6 apresenta as considerações finais do capítulo.

3.1 Processamento digital de imagem

O único processo realizado para o melhoramento das imagens acústicas é o proposto em (KIM; NERETTI; INTRATOR, 2005), onde é realizado a média de várias imagens acústicas e é utilizado como padrão de insonificação do sonar. Este método reduz o problema de interferência acústica entre os *beams* e a atenua a perda de sinal dos retornos acústicos mais distantes.

3.2 Segmentação

Uma das etapas do método de descrição de imagens acústicas proposto neste trabalho é a de segmentação. Nesta etapa os objetos presentes na imagem acústica precisam ser identificados e separados do fundo da imagem e dos demais objetos da cena. O grande desafio nesta etapa é determinar os grupos de pixels que representam um objeto relevante para a identificação do cenário representado pela imagem acústica.

Como descrito na Seção 2.1.2, dependendo da posição do sonar, podem ocorrer variações de intensidade dos pixels fazendo com que um objeto deixe de ser detectado, ou

fazendo que dois objetos fiquem muito próximos conduzindo a uma segmentação única e prejudicando o reconhecimento do local representado pela imagem.

Neste trabalho o método de segmentação foi dividido em duas etapas, a etapa de localização das regiões para a segmentação e a etapa de extração de um segmento. A etapa de localização das regiões para a segmentação é executada uma única vez por imagem enquanto que a etapa de extração de segmento é executada para cada região de segmentação encontrada.

Nesta seção serão apresentados três métodos de extração de segmento e dois métodos de identificação de regiões para a segmentação adotados na proposta.

3.2.1 Extração de Segmentos

Os métodos de extração de segmentos tem como base o algoritmo clássico de busca em largura (BFS). Este algoritmo tem como característica principal a busca por informações respeitando uma hierarquia de forma que todas as informações de um nível superior sejam exploradas antes das informações de um nível inferior.

No caso do problema em específico, as informações que são buscadas são os pixels da imagem que pertencem a um segmento. A hierarquia entre os pixels é estabelecida com base no pixel inicial da busca e a sua vizinhança, que pode ser os 4 pixels vizinhos, ou os 8 pixels vizinhos, como é mostrado na Figura 19. Estes pixels também podem ser chamados de pixels adjacentes, ou conectados.

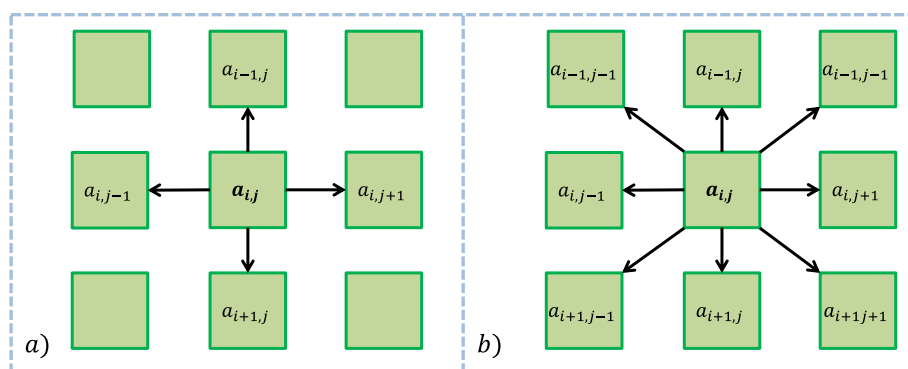


Figura 19: Representação dos pixel de vizinhança 4 em (a) e os pixels de vizinhança 8 em (b). Assim como numa matriz, o número de linha cresce para baixo e o número de colunas cresce para a direita.

A decisão se um pixel é válido para pertencer ao segmento é tomada com base na sua intensidade e no limiar de intensidade $\rho_{precurso}$. Se a sua intensidade for superior a $\rho_{precurso}$, o pixel é válido e é inserido no segmento.

3.2.1.1 Segmentação Completa

Na segmentação completa é utilizado o algoritmo **BFS** sem alterações, todos os pixels válidos são adicionados ao segmento. Desta forma um segmento é composto por todos os pixels conectados com intensidade superior a $\rho_{\text{recursivo}}$.

As figuras 20-g e 20-h) mostram os segmentos extraídos por completo.

3.2.1.2 Segmentação de Borda

Na segmentação de borda é inserido uma nova restrição no algoritmo **BFS**. Para que um pixel seja inserido em um segmento, ele precisa ser válido e os seus pixels vizinhos, que ainda não foram explorados, precisam ser inválidos, ou seja, ter intensidade inferior a $\rho_{\text{recursivo}}$.

O resultado deste método pode ser visualizado na Figura 20(e) e 20(f). Apenas as bordas dos segmentos são extraídas resultando em segmentos com menos pixels do que os extraídos pela segmentação completa e ao mesmo tempo mantendo a sua forma.

3.2.1.3 Segmentação Relativa a Distância

Nos dois métodos de extração de segmentos anteriores um segmento é formado sempre por um conjunto de pixels conectados. Entretanto dependendo do limiar de intensidade $\rho_{\text{recursivo}}$ utilizado, um objeto pode ser dividido em dois segmentos. Isto pode ocorrer porque parte dos seus retornos acústicos não atingiram uma intensidade superior a $\rho_{\text{recursivo}}$. Este caso pode ser observado nos segmentos das figuras 20-g) e 20-e). O segmento verde e o segmento azul representam o mesmo objeto na cena.

A segmentação relativa a distância foi criada para unir segmentos próximos que não estão conectados e possam representar um único objeto.

O algoritmo **BFS** foi alterado para continuar explorando os pixels inválidos desde que a sua distância até o último pixel válido seja menor que uma distância D_{seg} estipulada como parâmetro do algoritmo. Uma vez que a **BFS** possui a característica de explorar os pixels obedecendo a ordem hierárquica dos vizinhos, a distância entre o pixel inválido que está sendo explorado e o pixel válido mais próximo pode ser calculada facilmente através da diferença entre os níveis hierárquicos dos dois pixels.

Uma fila extra foi adicionada ao algoritmo para que os pixels válidos sejam explorados com maior prioridade que os pixels inválidos. Desta forma a fila dos pixels inválidos é acessada apenas quando a fila de pixels válidos estiver vazia.

O resultado da extração de segmentos com base na distância pode ser visualizado nas figuras 20-(b,c e d). A Figura 20(b) mostra como os pixels foram explorados, em azul estão representados os pixels válidos, em verde os pixels pertencentes a borda do segmento

e em vermelho os pixels inválidos que foram explorados respeitando a distância D_{seg} . As figuras 20-c) e d) mostram os segmentos extraídos. Comparando com os resultados obtidos pelos métodos de extração de segmentos anteriores pode-se observar que os segmentos em verde e em azul das figuras 20-(h,f) foram unidos e estão representados pela cor verde na Figura 20(d). Isto ocorre porque ao explorar os pixels inválidos representados em vermelho na Figura 20(b) encontra-se os dois segmentos.

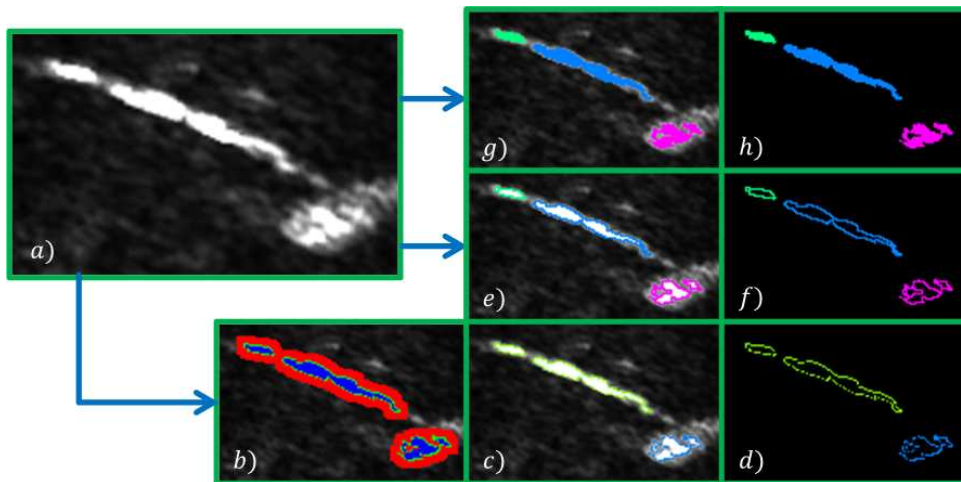


Figura 20: Exemplo dos métodos de extração de segmentos aplicado em (a), um recorte de uma imagem acústica. As figuras (b),(c) e (d) mostram a extração de segmentos com base na distância. As figuras (e) e (f) mostram o método de extração de borda de segmento e as figuras (g) e (h) mostram a extração de segmento completa. Cada segmento recebe uma cor diferente e segmentos com a mesma cor são considerados o mesmo segmento.

3.2.2 Busca de regiões para segmentação

Esta etapa de segmentação é responsável por encontrar as regiões da imagem que devem ser segmentadas. Basicamente esta etapa encontra os pixels iniciais utilizados na etapa de extração de segmento. Durante todo o processo é mantida uma matriz binária M_{viz} , com o mesmo tamanho da imagem processada, para indicar os pixels extraídos e impedir a extração múltipla de um pixel entre as operações de extração de segmento.

Dois métodos foram desenvolvidos, segmentação linear utilizando limiares fixos e segmentação baseada na busca de picos em *beams* acústicos.

3.2.2.1 Segmentação linear utilizando limiar de intensidade fixa

Nesta abordagem propõe-se uma busca linear pixel a pixel ignorando os pixels já extraídos, indicado pela matriz M_{viz} . Os pixels com intensidade superior ao limiar ρ_{linear} são utilizados como pixels de partida para a extração de segmentos. A intensidade ρ_{linear} é superior a intensidade $\rho_{recursivo}$ utilizada na extração de segmento. Desta forma um

segmento é extraído de uma imagem a partir de um pixel de alta intensidade localizado pela busca linear e os seus pixels vizinhos de menor intensidade através da etapa de extração de segmento.

3.2.2.2 Segmentação baseada na busca de picos em *beams* acústicos

Esta abordagem encontra o parâmetro $\rho_{\text{recursivo}}$ de forma automática. Cada *beam* acústico é analisado individualmente e busca-se os picos de intensidade. Dependendo da variação de intensidade de cada pico, um novo segmento pode ser extraído. O pixel que representa o *bin* de maior intensidade do pico é utilizado como pixel de inicial na etapa de extração de segmento. O limiar $\rho_{\text{recursivo}}$ é calculado em função da variação de intensidade do pico.

Como descrito na Seção 2.1 existem pelo menos duas formas de representar imagens acústicas de um FLS, a imagem crua na forma polar $I(\theta, \rho)$ e a imagem na forma cartesiana $I(X, Y)$. Tradicionalmente são utilizadas as imagens na forma cartesiana $I(X, Y)$ por representar os retornos acústicos considerando a geometria da cena observada, facilitando a sua análise e interpretação, permitindo inclusive a mensuração de objetos.

Entretanto as imagens cruas $I(\theta, \rho)$ seriam mais indicadas para serem utilizados nesta abordagem. Uma vez que um *beam* é uma coluna na imagem $I(\theta, \rho)$, a busca de picos seria realizada coluna a coluna e não existiria o problema de resolução gerado no processo de transformação da imagem $I(\theta, \rho)$ em $I(X, Y)$. Porém, como as imagens $I(\theta, \rho)$ nem sempre são disponibilizadas pelos softwares de aquisição de dados dos sonares e para que o método seja independente deste requisito, as análises de perfis de intensidades de cada *beam* são realizadas a partir das imagens cartesianas $I(X, Y)$ varrendo todas as direções θ da imagem.

O processo de identificação dos picos de intensidade de cada *beam* é representado pelo fluxograma da Figura 22. A ideia do método é extrair segmentos com base nos picos com variações de intensidade superiores ao parâmetro H_{busca} . Pois os picos com grandes variações de intensidade caracterizam a presença de um objeto relevante na cena. Para realizar essa seleção o método utiliza algumas variáveis de controle que estão representadas graficamente na Figura 21.

As variáveis I_{min} e I_{max} armazenam respectivamente a menor e a maior intensidade encontrada no pico. A variável H_{total} indica a variação de intensidade do pico. A variável H_{atual} indica a variação de intensidade do último *bin* extraído. Durante a análise de um *beam*, mais de um pico pode ser encontrado, a variável H_{fim} determina a intensidade mínima de um *bin* para encerrar a análise de um pico e começar a análise do próximo na sequência. Quando esta troca acontece as variáveis I_{min} e I_{max} são reiniciadas.

A intensidade $\rho_{\text{recursivo}}$ é o parâmetro utilizado na etapa de extração de segmento.

As variáveis H_{fim} e $\rho_{recursivo}$ são calculadas em função da variação total H_{total} do pico encontrado utilizando as equações 3.1 e 3.2.

$$H_{fim} = H_{total} \times \pi_{fim} \quad (3.1)$$

$$\rho_{recursivo} = I_{min} + H_{total} \times \pi_{recursivo} \quad (3.2)$$

As variáveis π_{fim} e $\pi_{recursivo}$ são as taxas de variação de intensidade com relação a cada pico encontrado utilizado para calcular os respectivos valores de H_{fim} e $\rho_{recursivo}$. As taxas π_{fim} e $\pi_{recursivo}$ são parâmetros pré-determinados do algoritmo e permitem que as variáveis H_{fim} e $\rho_{recursivo}$ sejam ajustadas automaticamente dependendo do pico encontrado.

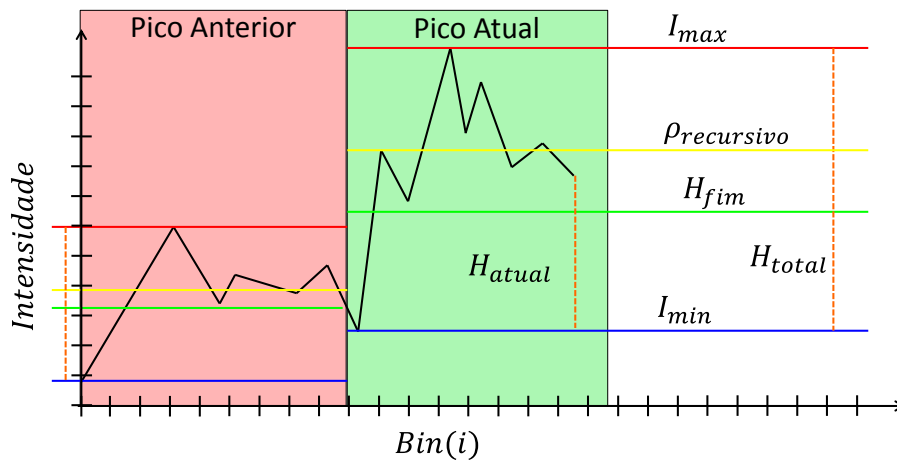


Figura 21: Variáveis utilizadas pelo método proposto. O gráfico representa um perfil de intensidade de um *beam* acústico, o eixo horizontal representa os *bins* e o eixo vertical a intensidade de cada *bin*.

O fluxograma da Figura 22 representa o processo de análise de picos de um *beam*. A variável I_{bin} representa a intensidade do último *bin* extraído.

Durante o processo as variáveis I_{min} e I_{max} são atualizadas pelos respectivos estados 4 e 5 conforme a intensidade I_{bin} .

Toda vez que um *bin* com intensidade inferior a I_{min} ou quando a análise de um pico é encerrada, as variáveis I_{min} e I_{max} são reiniciadas para começar a análise do próximo pico da sequência. Esta tarefa pode ser realizada pelo estado 4 ou 12 do fluxograma.

Um pico começa a ser analisado após a definição da sua intensidade base I_{min} , o elemento de decisão 7 avalia essa definição verificando se $I_{max} > I_{min}$, ou seja, se o pico possui uma variação de intensidade superior a zero.

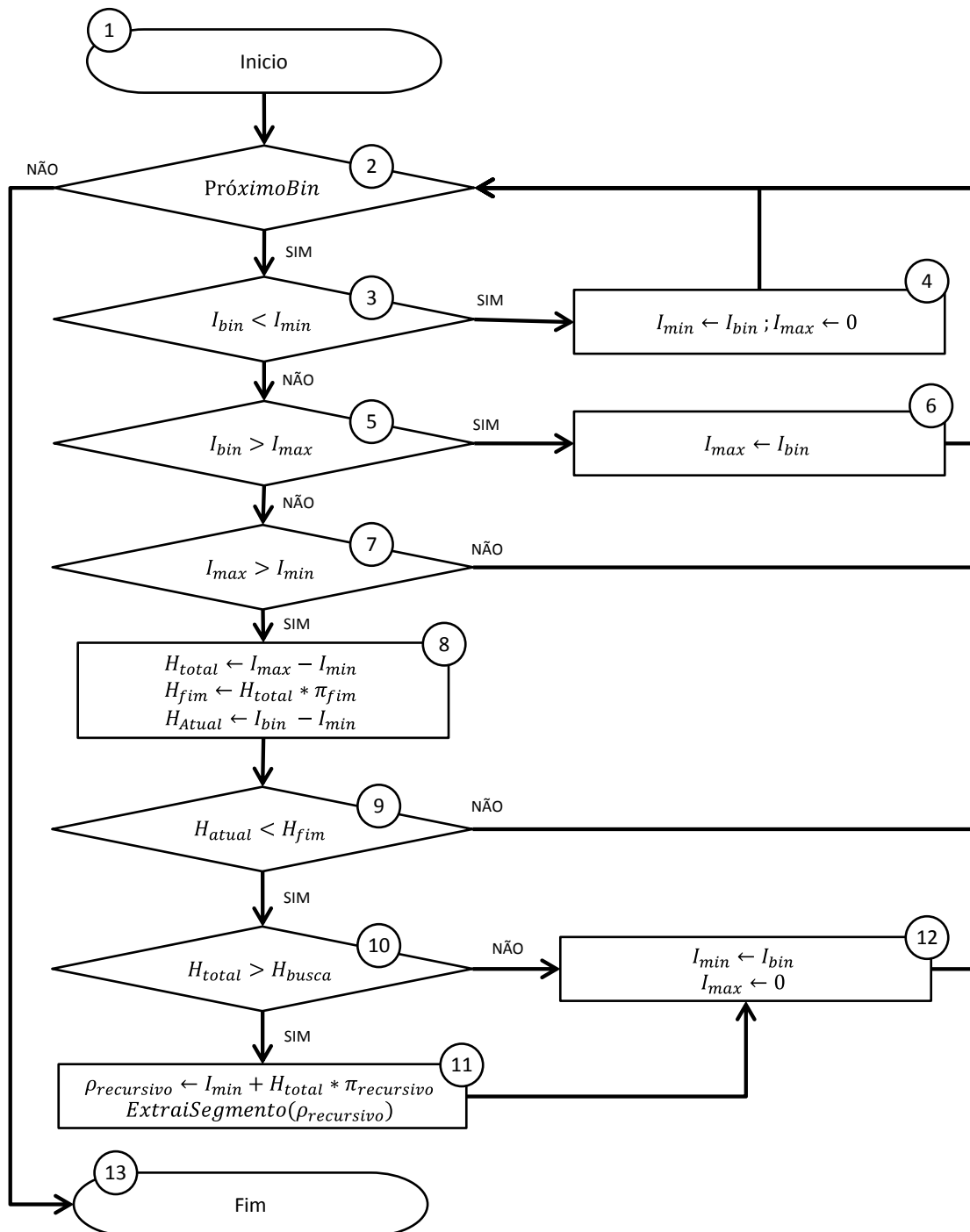


Figura 22: Fluxograma do método proposto.

Quando a intensidade do *bin* atual I_{bin} está entre a intensidade mínima I_{min} e a máxima I_{max} do pico, as variações de intensidade H_{total} , H_{fim} e H_{atual} são atualizadas no estado 8.

A todo momento da análise as intensidades máximas e mínimas e as variações de um pico são atualizadas. O elemento de decisão 9 verifica quando um pico termina e um novo pico começa a ser analisado na sequência. A análise de um pico é finalizada quando a variação de intensidade do último *bin* I_{bin} é menor que a variação de finalização do pico H_{fim} .

Quando a análise de um pico é encerrada, o elemento de decisão 10 decide se o pico encontrado será utilizado para a extração de um segmento. Para que um pico seja aceito, a sua variação H_{total} deve ser superior a variação H_{busca} passada por parâmetro para o algoritmo.

O processo 11 realiza a extração de um segmento utilizando o pico detectado, o limiar de intensidade $\rho_{recursivo}$ é calculado em função da variação total do pico H_{total} e a taxa de variação $\pi_{recursivo}$ pela Equação 3.2. O pixel inicial para busca utilizado na extração de segmento é o pixel que representa o *bin* de maior intensidade do pico I_{max} .

Após o termino da análise de um pico os limites de intensidade I_{min} e I_{max} são reiniciados para que o próximo pico possa ser analisado.

Esta abordagem de segmentação necessita dos parâmetros π_{fim} , $\pi_{recursivo}$ e H_{busca} , anteriormente apresentados.

3.3 Descritor

As regiões segmentadas em cada imagem são representadas por um modelo que considera o formato de cada segmento e as suas relações topológicas. Para a representação da forma de um segmento é utilizado o modelo probabilístico Gaussiano e para representar as relações de distância e inclinação entre os segmentos é utilizado um grafo geométrico. Desta forma cada imagem acústica é descrita por um grafo.

3.3.1 Representação dos Segmentos

Cada segmento é um conjunto de pixel que compõem uma amostra da imagem, onde cada pixel possui três valores, referente a sua posição, a linha Y e a coluna X e referente a sua intensidade I que podem ser representados por vetores coluna onde cada

linha é a informação de um pixel da amostra.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix}; Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}; I = \begin{bmatrix} i_1 \\ i_2 \\ \dots \\ i_N \end{bmatrix}. \quad (3.3)$$

Cada amostra é representado por uma Gaussiana definida como $GA = (\mu X, \mu Y, \mu I, \sigma X, \sigma Y, \sigma I, \theta_g, N_g)$, onde $\mu X, \mu Y, \mu I$ são as respectivas médias aritméticas das variáveis X, Y e I . O valor de σI é o desvio padrão da variável I , o valor de σX é o desvio padrão no eixo de maior dispersão espacial da amostra, o valor σY é o desvio padrão no segundo maior eixo de dispersão espacial da amostra, θ_g é o ângulo de rotação da gaussiana com relação ao eixo vertical da imagem e N_g é o tamanho da amostra em pixels. Veja a Figura 23.

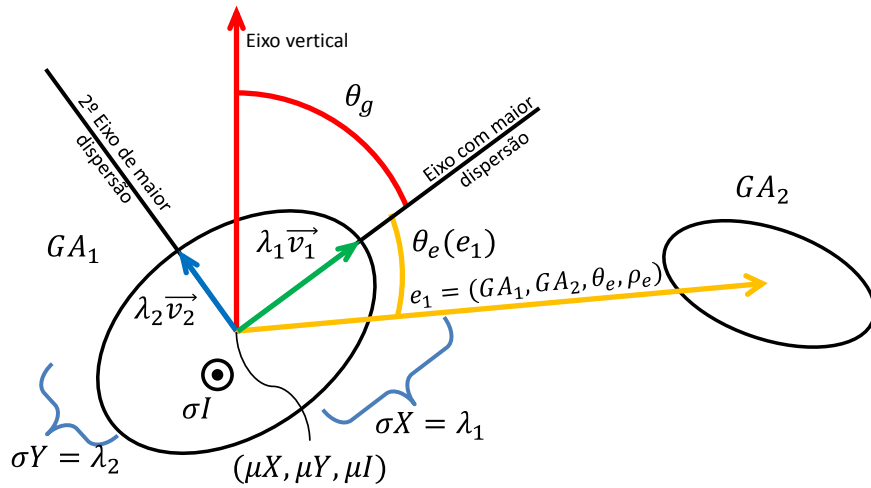


Figura 23: Modelo de descrição das imagens acústicas. Duas Gaussianas GA_1 e GA_2 são representadas como vértices de um grafo. Em vermelho o ângulo de inclinação $\theta_g(GA_1)$ entre o eixo vertical da imagem(em vermelho) e o eixo de maior dispersão da elipse. Em verde e em azul, respectivamente, o maior e o segundo maior eixo de dispersão da Gaussiana GA_1 . Em amarelo uma aresta que liga duas Gaussianas GA_1 e GA_2 composta pela inclinação $\theta_e(e_1)$ relativa a inclinação da Gaussiana de origem $\theta_g(GA_1)$ e a distância ρ_e entre o centro das duas Gaussianas GA_1 e GA_2 .

O valor σI é calculado aplicando a Equação do desvio padrão 3.4.

$$\sigma I = \frac{1}{N_g - 1} \sum_{i=1}^{N_g} (i_i - \mu I)^2. \quad (3.4)$$

Os valores de $\sigma X, \sigma Y$ são os autovalores da matriz de covariância Σ da amostra. A matriz de covariância Σ de uma amostra é calculada pela Equação 3.5.

$$\begin{aligned} \sigma(x, y) &= \mathbb{E}[(x - \mathbb{E}(x))(y - \mathbb{E}(y))] \\ \Sigma &= \begin{bmatrix} \sigma(x, x) & \sigma(x, y) \\ \sigma(y, x) & \sigma(y, y) \end{bmatrix} \end{aligned} \quad (3.5)$$

Os autovalores λ e os autovetores \vec{v} são encontrados resolvendo a Equação 3.6 utilizando a decomposição em valores singulares SVD.

$$\begin{aligned} \Sigma \vec{v} &= \lambda \vec{v} \\ \begin{bmatrix} \sigma(x, x) & \sigma(x, y) \\ \sigma(y, x) & \sigma(y, y) \end{bmatrix} \begin{bmatrix} vx \\ vy \end{bmatrix} &= \lambda \begin{bmatrix} vx \\ vy \end{bmatrix} \end{aligned} \quad (3.6)$$

Resolvendo o sistema de equações 3.6, são encontrados dois autovalores λ_1, λ_2 e dois autovetores \vec{v}_1, \vec{v}_2 , onde \vec{v}_1 aponta para a direção de maior dispersão, λ_1 é o comprimento de \vec{v}_1 e $\sigma X = \lambda_1$. \vec{v}_2 aponta em direção da segunda maior dispersão, λ_2 é o comprimento de \vec{v}_2 e $\lambda_2 = \sigma Y$. A inclinação da gaussiana θ_g é a tangente do autovetor \vec{v}_1 , $\theta_g = \text{atan2}(v_{1x}, v_{1y})$, veja na Figura 23.

3.3.2 Estabelecendo as relações topológicas

Após a descrição dos segmentos da imagem utilizando o modelo Gaussiano, um grafo G é criado para estabelecer as relações geométricas entre os segmentos. Cada Gaussiana GA que descreve um segmento é um vértice de G , onde G é um grafo dirigido e simétrico definido como $G = (V, E)$ onde $V = \{GA_1, GA_2, \dots, GA_N\}$ e $E = \{E_1, E_2, \dots, E_M\}$. Onde cada aresta $E_i = (GA_{orig}, GA_{dest}, \theta_e, \rho_e)$ conecta duas GA s qualquer $\{GA_{orig}, GA_{dest} \in V / GA_{orig} \neq GA_{dest}\}$, onde θ_e é a inclinação da aresta com relação ao eixo de maior dispersão da Gaussiana de origem $\theta_g(GA_{orig})$ calculado pela Equação 3.7 e ρ_e é o comprimento da aresta que liga GA_{orig} a GA_{dest} calculado pela distância euclidiana entre as médias das Gaussianas, Equação 3.8. Veja a Figura 23.

$$\theta_e = \text{atan2}(\mu Y(GA_{dest}) - \mu Y(GA_{orig}), \mu X(GA_{dest}) - \mu X(GA_{orig})) - \theta_g(GA_{orig}) \quad (3.7)$$

$$\rho_e = \sqrt{(\mu X(GA_{orig}) - \mu X(GA_{dest}))^2 + (\mu Y(GA_{orig}) - \mu Y(GA_{dest}))^2} \quad (3.8)$$

Desta forma, se for considerado que a posição e a forma geométrica dos segmentos não variam ao longo do tempo e nem com a alteração do ponto de vista do sonar, a inclinação

θ_e e o comprimento ρ_e das arestas também não variam. Pois a inclinação θ_e depende da orientação θ_g da gaussiana, que depende da forma do segmento e o comprimento ρ_e depende da posição das *GAs* envolvidas. E portanto o problema de encontrar a semelhança de imagens através de grafos estaria praticamente resolvido. Porém na prática a forma dos segmentos são afetadas pela mudança do ponto de vista e pela distorção acústica do sonar, e nem todos os objetos da cena são estáticos, os cardumes ou o casco de um barco podem estar em movimento, o que torna mais complexa a tarefa de encontrar a similaridade entre as imagens.

3.3.2.1 Construção do grafo

Todos os elementos segmentados são descritos por Gaussianas e inseridos como um vértice no grafo topológico que descreve uma imagem. As arestas são criadas em função da distância entre os vértices e o parâmetro R_{lig} que determina o comprimento máximo de uma aresta no grafo. Para cada par de vértices (GA_1, GA_2) com distância inferior a R_{lig} são criadas duas arestas, uma de ida $GA_1 \rightarrow GA_2$ e uma de volta $GA_2 \rightarrow GA_1$. Embora o comprimento ρ_e seja simétrico, igual nas duas arestas, a inclinação θ_e não é, pois depende da inclinação da gaussiana de origem $\theta_g(GA_{orig})$, portanto sendo necessário o uso das duas arestas.

3.4 Encontrando Imagens Acústicas Semelhantes

Neste capítulo será apresentada a etapa de comparação dos grafos que descrevem as imagens acústicas, buscando encontrar as imagens que representam um mesmo local. O problema de determinar se dois grafos G_1 e G_2 são iguais, ou se existe um sub grafo de G_1 que é igual a um sub grafo de G_2 é conhecido como isomorfismo de grafos ou isomorfismos de sub-grafos e são problemas com complexidade computacional **NP-Completo** (SCHÖNING, 1988). Ou seja, não existe uma solução determinística que resolva o problema em complexidade de tempo polinomial. Entretanto, como os grafos tratados neste trabalho possuem informações extraídas da imagem, é possível encontrar uma solução aproximada através da similaridade entre grafos, como descrito na Seção 2.3.3, explorando as informações extraídas da imagem.

As abordagens utilizadas para encontrar a semelhança entre grafos serão apresentadas em duas etapas. A etapa onde a similaridade entre os vértices e entre as arestas são computadas considerando as informações extraídas da imagem e a etapa onde os vértices semelhantes entre os dois grafos são selecionados.

3.4.1 Similaridade

Conforme descrito na Seção 3.3.1, os vértices possuem informações relacionadas a sua forma e tamanho ($\sigma X, \sigma Y, \theta_g, N_g$), a sua posição ($\mu X, \mu Y$) e a sua intensidade acústica ($\mu I, \sigma I$). As arestas relacionam os vértice com as informações de topologia (θ_e, ρ_e).

A Figura 24 ilustra o problema de encontrar a semelhança entre dois vértices. A representação de duas imagens acústicas e os grafos que as descrevem são exibidas pelas figuras 24(a) e 24(b). Os seus respectivos grafos são exibidos em maiores detalhes nas figuras 24(d) e 24(e). Um método para encontrar a semelhança entre dois vértices deve ser capaz de lidar com problemas como o de alinhamento dos grafos e o de oclusão de vértices. Uma abordagem baseada nas informações das arestas adjacentes precisa inicialmente encontrar a rotação entre os dois vértices comparados. Este problema é ilustrado na Figura 24(f) que mostra o grafo das duas imagens centralizadas no vértice que está sendo comparado. A Figura 24(g) mostra algumas das possíveis comparações que podem ser realizadas rotacionando o grafo da Figura 24(e), considerando que nem todas as arestas possuem o seu par semelhante devido a possibilidade de oclusão de vértices, a quarta rotação é a que possui a melhor semelhança entre as arestas adjacentes dos dois vértices comparados. A Figura 24(c) mostra a imagem acústica da Figura 24(b) alinhada com a Figura 24(a) utilizando o alinhamento encontrado na Figura 24(g4).

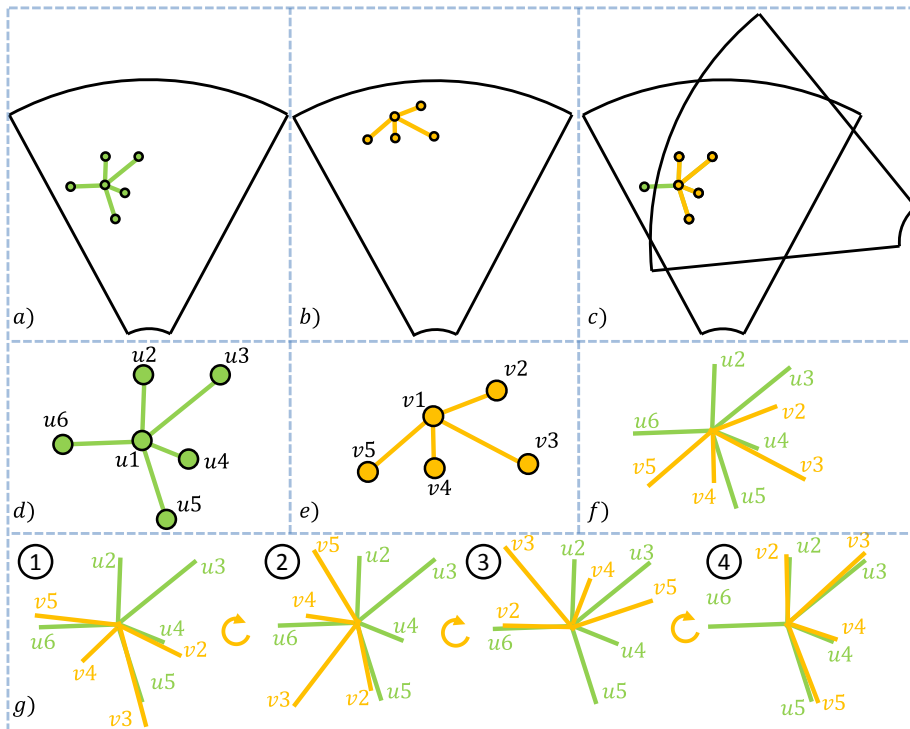


Figura 24: Encontrando a semelhança entre vértices.

3.4.1.1 Alinhamento Utilizando a Forma Geométrica

Esta abordagem considera que a forma geométrica dos objetos detectados nas imagens acústicas não variam com a alteração do ponto de vista do sonar. E por este motivo, a forma dos objetos detectados são utilizadas como referência de orientação para a comparação dos vértices, reduzindo o problema de encontrar o alinhamento entre eles.

A Figura 25(a) ilustra uma cena capturada em tempos e pontos de vista diferentes. As figuras 25(b) e 25(c) mostram as imagens capturadas considerando que não houve alteração na forma geométrica dos objetos extraídos da cena. As linhas em vermelho representam a orientação θ_g de cada Gaussianas. Apesar da orientação das Gaussianas θ_g sofrer alterações ao mudar o ponto de vista, as inclinações das arestas $\theta_{e1}, \theta_{e2}, \theta_{e3}$ se mantem as mesmas, isto porque, como descrito na Seção 3.3.2, a inclinação θ_e de cada aresta é calculada com referência a inclinação θ_g da forma geométrica de origem.

O cálculo da semelhança entre vértices considera as arestas adjacentes dos vértices comparados, utilizando a inclinação θ_e e o comprimento ρ_e de cada aresta.

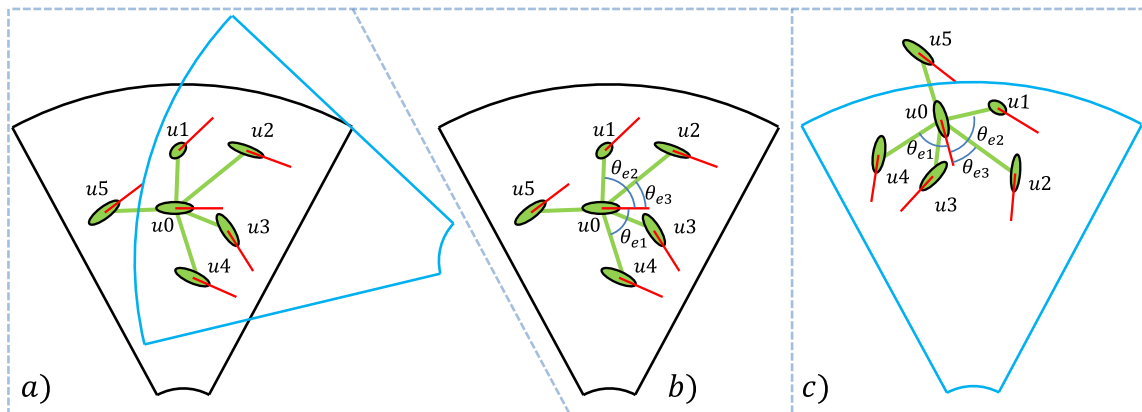


Figura 25: Encontrando o alinhamento dos vértices através da geometria dos objetos extraídos da imagem. Em (a) uma cena observada de dois pontos de vista distintos, em (b) e (c) as imagens capturadas de cada ponto de vista. As linhas em vermelho representam a orientação das Gaussianas θ_g extraídas da imagem.

Erro Ponderado

Uma função $cmp(GA_x, GA_y)$ calcula a semelhança entre um par de vértices GA_m, GA_n de dois grafos G_k, G_l distintos e retorna o número de pares de arestas consideradas semelhantes, chamado de $\Sigma Acerto$, e o erro entre os dois vértices chamado ϵ_w utilizado como métrica de semelhança.

O Algoritmo 1 mostra como a função $cmp(GA_x, GA_y)$ compara as arestas. A estrutura de repetição *enquanto* da linha 3 percorre todas as arestas adjacentes dos dois vértices em ordem crescente de inclinação. As linhas 4 e 5 calculam, respectivamente, as

diferenças de comprimento em pixel Δ_ρ e de inclinação em graus Δ_θ do par de arestas atual.

Algorithm 1 Compara vértices ponderado

```

1: função CMP( $GA_x, GA_y$ )
2:    $\Sigma Acerto \leftarrow 0$ 
3:   enquanto  $e_x \neq fim$  e  $e_y \neq fim$  faça
4:      $\Delta_\theta \leftarrow \|\theta_e(e_x) - \theta_e(e_y)\|$ 
5:      $\Delta_\rho \leftarrow \|\rho_e(e_x) - \rho_e(e_y)\|$ 
6:     se  $\Delta_\rho < mx\rho$  e  $\Delta_\theta < mx\theta$  então
7:        $\epsilon\theta \leftarrow \epsilon\theta + \Delta_\theta$ 
8:        $\epsilon\rho \leftarrow \epsilon\rho + \Delta_\rho$ 
9:        $\Sigma Acerto \leftarrow \Sigma Acerto + 1$ 
10:       $e_x \leftarrow proxima(e_x, E_{adj}(GA_x))$ 
11:       $e_y \leftarrow proxima(e_y, E_{adj}(GA_y))$ 
12:    senão
13:      se  $\theta_e(e_x) < \theta_e(e_y)$  então
14:         $e_x \leftarrow proxima(e_x, E_{adj}(GA_x))$ 
15:      senão
16:         $e_y \leftarrow proxima(e_y, E_{adj}(GA_y))$ 
17:    fim se
18:  fim se
19:  fim enquanto
20:  se  $\Sigma Acerto > 0$  então
21:     $\epsilon\theta \leftarrow \epsilon\theta / \Sigma Acerto$ 
22:     $\epsilon\rho \leftarrow \epsilon\rho / \Sigma Acerto$ 
23:  fim se
24:   $\epsilon_w \leftarrow w\theta\epsilon\theta + w\rho\epsilon\rho$ 
25:  devolve  $\epsilon_w, \Sigma Acerto$ 
26: fim função

```

A estrutura condicional *se* da linha 6 determina se o par de aresta atual é semelhante, verificando se os erros de inclinação Δ_θ e de comprimento Δ_ρ são menores que os respectivos erros de comprimento máximo permitido $mx\theta$ e inclinação máxima permitida $mx\rho$. Onde os valores de $mx\rho$ e de $mx\theta$ são parâmetros predeterminados do algoritmo e definem a sensibilidade do método para comparação dos vértices semelhantes.

Nas linhas 7 e 8 do algoritmo são atualizados, respectivamente, o erro de inclinação acumulada do vértice $\epsilon\theta$, e o erro de comprimento acumulado do vértice $\epsilon\rho$. Na linha 9 é incrementado o contador de arestas semelhantes $\Sigma Acerto$ e nas linhas 10 e 11 são acessadas as próximas arestas dos respectivos vértices GA_x e GA_y em ordem crescente de inclinação utilizando a função $proximo(e, E_{adj}(GA_i))$, onde e é a aresta atual e $E_{adj}(GA_i)$ é a próxima aresta adjacente a GA_i .

Caso um par de aresta não seja semelhante, a aresta com menor inclinação é descartada da análise. Esse comportamento guloso foi criado para tratar o problema

de oclusão. Onde objetos são detectados em apenas uma das imagens acústicas. Nessas situações, as arestas que não obtiveram seus pares semelhantes são descartadas e não são consideradas no cálculo da semelhança dos vértices. Esse tratamento é realizado nas linhas 13-17 do Algoritmo 1.

As linhas 20-23 calculam os erros médios de inclinação e de comprimento dos pares de arestas considerados semelhantes. Na linha 24 é realizada uma soma ponderada ϵ_w considerando as diferenças médias de inclinação $\epsilon\theta$ e de tamanho $\epsilon\rho$ entre as arestas comparadas. Os pesos $w\theta$ e $w\rho$ são parâmetros predeterminados que definem, respectivamente, a importância do erro angular e do erro de comprimento das arestas.

A soma ponderada ϵ_w é o erro das arestas adjacentes dos dois vértices comparados utilizado como indicador da semelhança. Neste caso como se trata do erro, quanto menor for ϵ_w , maior é a semelhança do par de vértices comparado. A contagem $\Sigma Acerto$ indica o número de pares de arestas considerados no cálculo da semelhança entre os vértices. Ambos os valores ϵ_w e $\Sigma Acerto$ são retornados pela função de comparação $cmp(GA_x, GA_y)$.

Erro escaleno

Uma alternativa ao método da soma ponderada, que considera os erros de inclinação médio $\epsilon\theta$, comprimento médio $\epsilon\rho$ e os pesos $w\theta$ e $w\rho$ que devem ser determinados *a priori*, é a substituição pelo erro calculado a partir das distâncias entre as posições finais dos pares de arestas comparados, chamado ρ_{esc} .

A Figura 26 ilustra como a distância entre as posições finais de um par de arestas adjacentes ρ_{esc} pode ser calculada. As figuras 26(a) e 26(b) mostram os dois grafos que estão sendo comparados. Na Figura 26(c) é exibido os dois grafos centrados nos vértice GA_u, GA_v que estão sendo comparados. Como é mostrado na Figura 26(d), o cálculo das distâncias entre as posições finais de um par de aresta pode ser generalizado ao problema de encontrar o comprimento de um dos lados de um triângulo escaleno, onde é conhecido o comprimento de dois lados ρ_u e ρ_v e a variação angular entre eles $\Delta\theta$. A distância ρ_{esc} pode ser encontrada pela Equação 3.9 que é obtida através da lei dos cossenos e do teorema de Pitágoras.

$$\rho_{esc}^2 = \rho_u^2 * \rho_v^2 - 2 * \rho_u * \rho_v * \cos(\Delta\theta) \quad (3.9)$$

O Algoritmo 2 mostra a função de comparação de vértices utilizando a distância entre as posições finais de cada par de aresta ρ_{esc} . Com relação ao Algoritmo 1, o novo método altera a forma de testar se um par de aresta é semelhante substituindo a comparação dos erros de inclinação $\Delta\theta$ e comprimento $\Delta\rho$ com os respectivos limiares $max\theta$ e $max\rho$, pela comparação do erro das posições finais entre arestas ρ_{esc} com um limiar $mx\epsilon_{esc}$

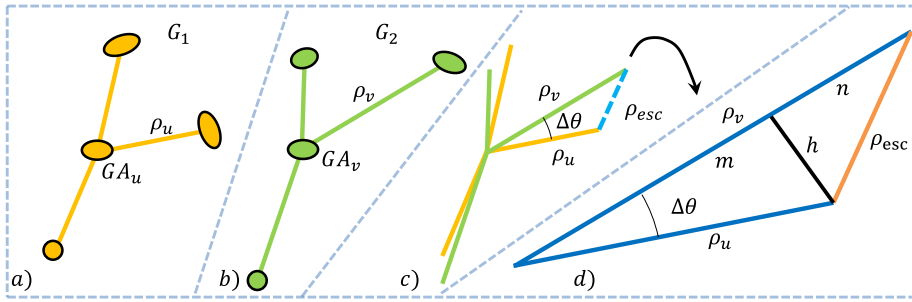


Figura 26: Cálculo do erro de posição final de um par arestas.

predeterminado. A métrica de similaridade ϵ_w é substituída pelo erro de posição final médio de todos pares de arestas semelhantes, chamado ϵ_{esc} .

Algorithm 2 Comparação de vértices escaleno.

```

1: função CMP( $GA_x, GA_y$ )
2:    $\Sigma Acerto \leftarrow 0$ 
3:   enquanto  $e_x \neq fim$  e  $e_y \neq fim$  faça
4:      $\Delta\theta \leftarrow \|\theta_e(e_x) - \theta_e(e_y)\|$ 
5:      $\rho_{esc} \leftarrow \sqrt{\rho_e(e_x)^2 + \rho_e(e_y)^2 - 2 * \rho_e(e_x) * \rho_e(e_y) * \cos(\Delta\theta)}$ 
6:     se  $\rho_{esc} < mx\epsilon_{esc}$  então
7:        $\epsilon_{esc} \leftarrow \epsilon_{esc} + \rho_{esc}$ 
8:        $\Sigma Acerto \leftarrow \Sigma Acerto + 1$ 
9:        $e_x \leftarrow proxima(e_x, E_{adj}(GA_x))$ 
10:       $e_y \leftarrow proxima(e_y, E_{adj}(GA_y))$ 
11:     senão
12:       se  $\theta_e(e_x) < \theta_e(e_y)$  então
13:          $e_x \leftarrow proxima(e_x, E_{adj}(GA_x))$ 
14:       senão
15:          $e_y \leftarrow proxima(e_y, E_{adj}(GA_y))$ 
16:     fim se
17:   fim se
18:   fim enquanto
19:   se  $\Sigma Acerto > 0$  então
20:      $\epsilon_{esc} \leftarrow \epsilon_{esc} / \Sigma Acerto$ 
21:   fim se
22:   devolve  $\epsilon_{esc}, \Sigma Acerto$ 
23: fim função

```

Desta forma o único parâmetro necessário neste método de cálculo de similaridade é o erro máximo permitido $mx\epsilon_{esc}$ em pixel.

Ambiguidade de orientação

Um dos problemas de utilizar as formas geométricas como referência para a orientação dos vértices é a ambiguidade. Para cada orientação existem dois ângulos possíveis que

devem ser considerados. A Figura 27 ilustra uma situação hipotética onde é considerado que não existe distorção acústica dos objetos da cena quando capturados de pontos de vistas diferentes.

A Figura 27(a) mostra o problema de ambiguidade devido a simetria da gaussiana. A orientação da gaussiana θ_{g_i} representada pelas linhas em vermelho, é calculada com relação ao eixo vertical da imagem e varia entre 0° e 180° positivos. As linhas em laranja mostram que a orientação é ambígua, sendo um ângulo com 180° de diferença com relação a orientação calculada, equivalente ao lado negativo, a esquerda do eixo vertical da imagem.

As figuras 27(b) e 27(c) mostram as descrições da mesma cena hipotética capturada de pontos de vista diferentes. Na Figura 27(b) a orientação das Gaussianas θ_g , assumidas para este ponto de vista, são representadas pelas linhas em vermelho. Em azul é representado o campo de cobertura da segunda captura, descrita na Figura 27(c).

Na Figura 27(c) as linhas em vermelho representam as orientações das Gaussianas θ_g assumidas pelo ponto de vista da Figura 27(b) e as linhas em azul representam as orientações das Gaussianas θ_g assumidas pelo ponto de vista atual.

É possível concluir que o problema da ambiguidade das orientação θ_g pode ocorrer quando imagens de diferentes pontos de vistas são capturadas. Esse problema afeta as inclinações das arestas θ_e utilizadas no cálculo de similaridade. No caso do exemplo, os vértices u_2, u_3 e u_4 foram afetados pelo problema da ambiguidade. No caso específico do vértice u_2 , é representado o efeito causado na inclinação θ_{e1} pelos diferentes pontos de vista, que são complementos de 180° .

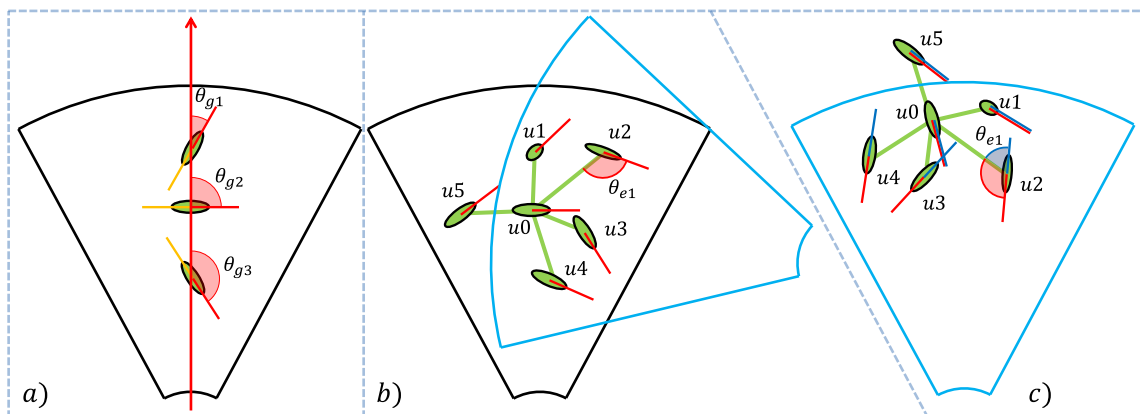


Figura 27: Problema de ambiguidade na orientação das Gaussianas θ_g .

3.4.1.2 Utilizando topologia para o alinhamento

Na prática o alinhamento de vértices utilizando as formas extraídas da imagem sofre com o problema da distorção acústicas, que afeta principalmente objetos pequenos ou com

formas arredondadas ao ponto de modificar a sua inclinação θ_g e afetar a inclinação das arestas θ_e . Uma solução seria encontrar o alinhamento dos vértices através das informações topológicas, considerando a posição dos vértices adjacentes. Entretanto, encontrar o alinhamento utilizando apenas a topologia exige um custo computacional maior.

A Figura 28(a) ilustra a distorção acústica causada em duas imagens capturadas na mesma cena, porém em pontos de vista diferentes, com cerca de 180° de rotação entre as visualizações. As duas imagens foram segmentadas utilizando a busca linear e descritas utilizando o modelo gaussiano. Os números em verde indicam a inclinação θ_g de cada gaussiana com relação ao eixo vertical da imagem.

Uma região que cobre o mesmo lugar na cena foi recortada em cada imagem e ampliada nas figuras 28(b) e 28(c). No recorte da Figura 28(b) 6 Gaussianas foram detectadas, enquanto que no recorte da Figura 28(c) foram detectadas apenas 5. As GAs foram nomeadas conforme a suas relações com a cena real, a gaussiana GA_6 foi detectada apenas na Figura 28(b) porque os seus retornos acústicos na segunda captura, 28-(c), foram de baixa intensidade ficando abaixo da intensidade ρ_{linear} .

Para o estudo do efeito da distorção acústica na forma das GAs e por consequência na inclinação das arestas θ_e , uma representação dos recortes das figuras 28(b) e 28(c) foram criados e podem ser visualizados nas figuras 28(d), 28(e), 29 e 31.

Nas figuras 29(a) e 29(b), as inclinações θ_e das arestas adjacentes $E_{adj}(GA_2)$ representadas por θ_{e1}, θ_{e2} e θ_{e3} se mantêm aproximadamente iguais nos dois pontos de vista. Enquanto que nas figuras 29(c) e 29(d) as inclinações das arestas adjacentes $E_{adj}(GA_3)$, em específico as inclinações θ_{e2} e θ_{e3} , são diferentes. Acredita-se que o efeito da distorção acústica é mais aparente na GA_3 do que na GA_2 devido ao formato menor e mais arredondado da GA_3 . As demais GAs também sofrem com a distorção, basta observar que a inclinação θ_g , representado pelas linhas em vermelhos, ficam alinhadas para a mesma direção nas duas imagens, exceto $\theta_g(GA_2)$.

Esta distorção acústica causa o achatamento da forma no sentido ortogonal de propagação das ondas acústicas. A Figura 30 mostra uma representação hipotética desse achatamento. A Figura 30(a) mostra duas Gaussianas, uma pequena com a forma mais arredondada GA_1 e outra maior GA_2 com uma forma mais achatada, ambas sem o efeito da distorção, na Figura 30(b) as forma são alongadas no sentido da distorção indicada pelas setas, causando uma alteração $\Delta\theta_a$ na inclinação θ_g .

Uma solução para este problema seria o alinhamento com base nas informações de topologia sem considerar a inclinação θ_g e considerando as variações angulares e o comprimento das arestas adjacentes. A Figura 31 representa a mesma situação da Figura 29, porém utilizando as variações angulares entre as arestas, representadas pelas letras $\Delta\theta_e$. Apesar da distorção acústica, as variações angulares $\Delta\theta_e$ e os comprimentos ρ_e entre

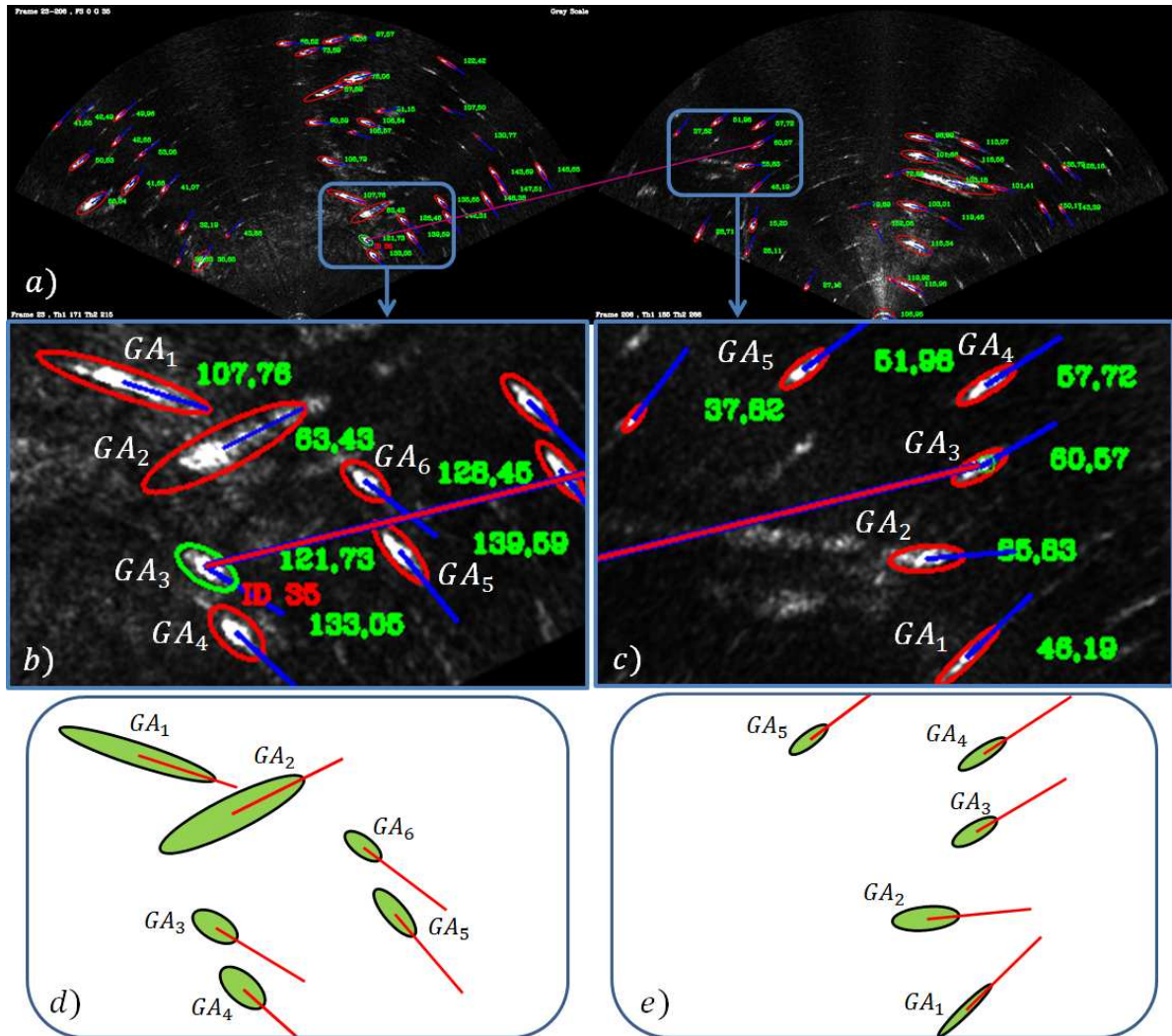


Figura 28: Distorção acústica na forma dos objetos extraídos.

as arestas adjacentes sofrem menos alterações. Como pode ser visualizado nas figuras 31(a) e 31(b) referentes a GA_2 e nas figuras 31(c) e 31(d) referentes ao vértice GA_3 , em ambos os casos as variações angulares $\Delta\theta_{e1}$, $\Delta\theta_{e2}$ e $\Delta\theta_{e3}$ se mantêm aproximadamente constante com a alteração do ponto de vista.

Alinhamento por força bruta

Uma solução baseada em força bruta para solucionar o problema de alinhamento através das arestas adjacentes deve considerar todos os testes possíveis incluindo a oclusão de vértices. A Figura 32 ilustra as combinações para um grafo com 4 vértices.

Alinhamento guloso

Para realizar o alinhamento de vértices utilizando as informações de variação angular e de comprimento das arestas adjacentes é necessário que os dois vértices analisados tenham pelo menos duas arestas adjacentes.

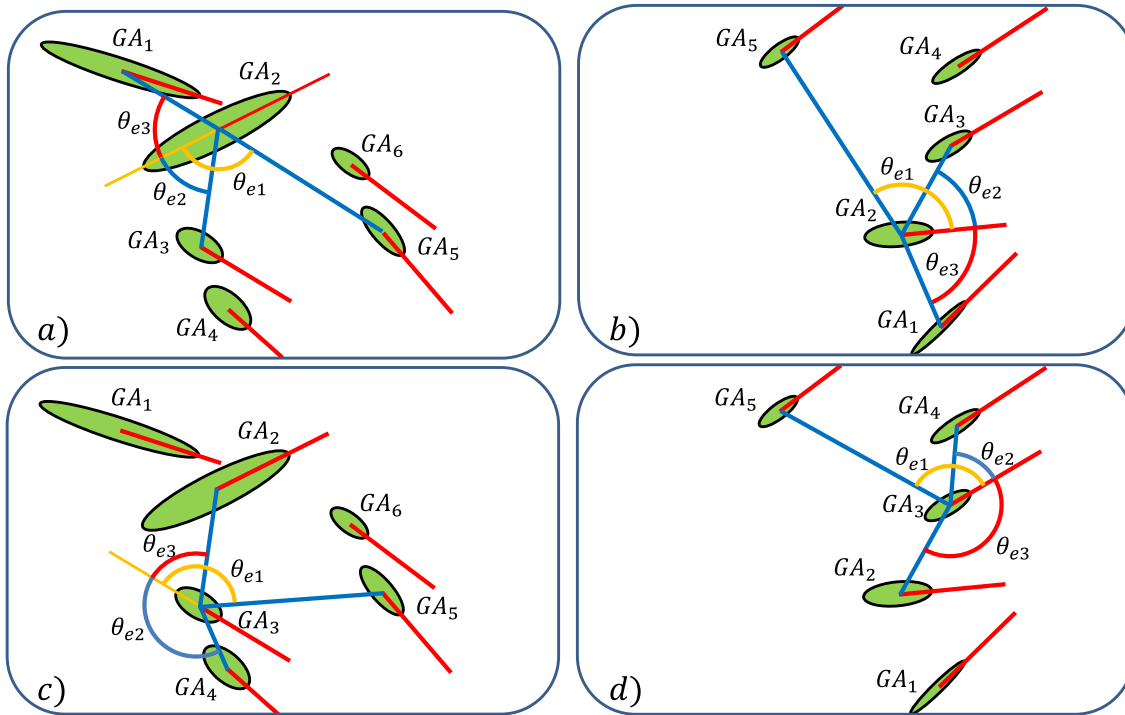


Figura 29: Distorção acústica e o efeito na orientação das arestas θ_e .

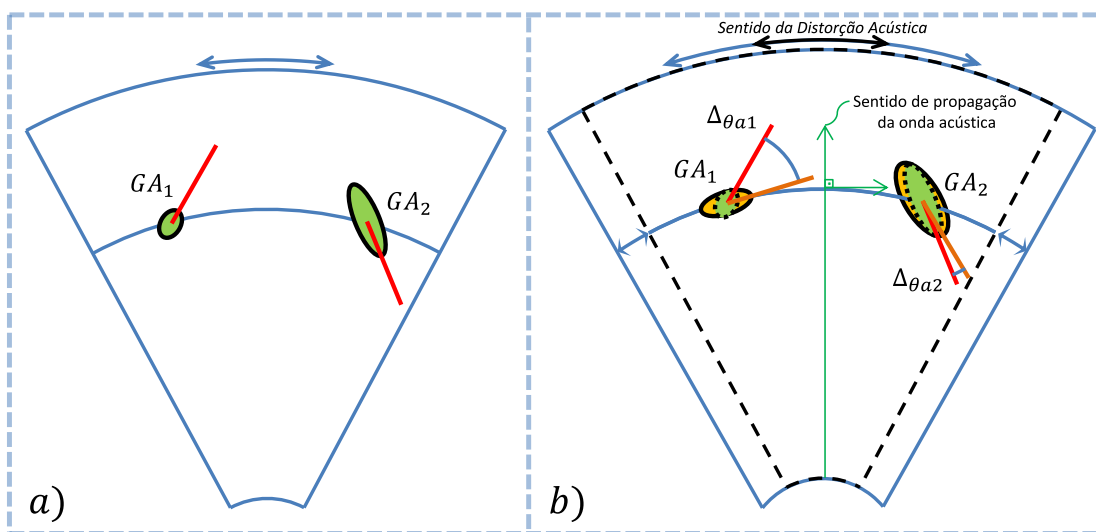


Figura 30: Efeito de achatamento nas imagens acústicas.

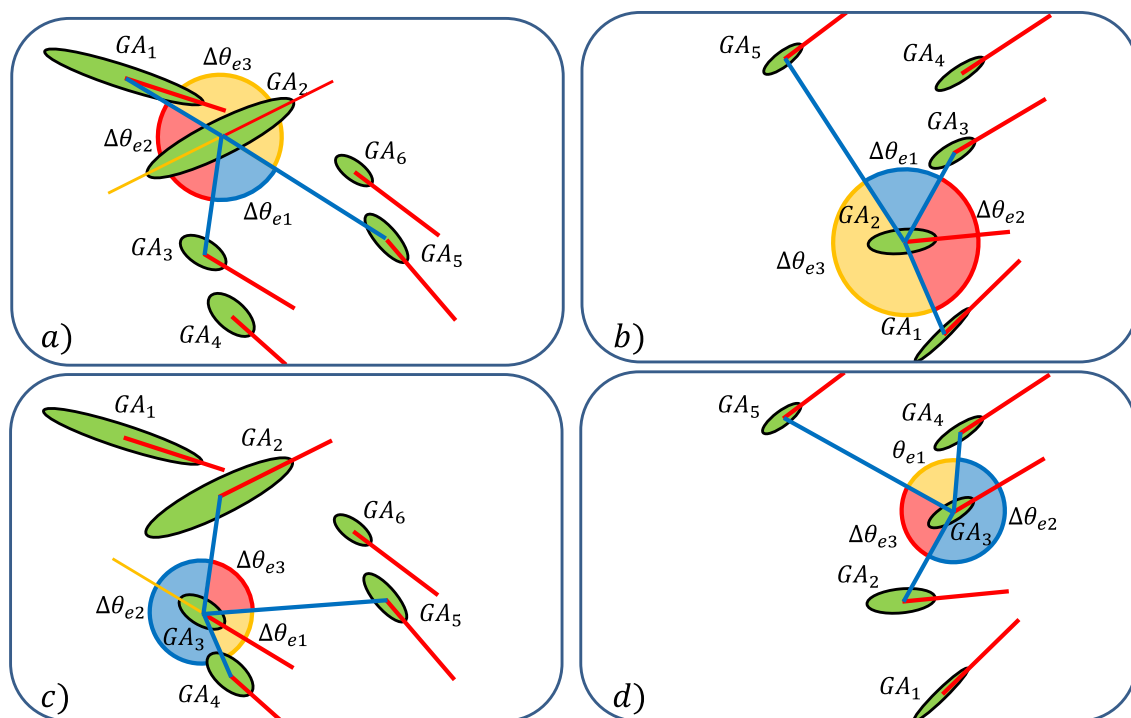


Figura 31: Distorção acústica e a variação angular das arestas.

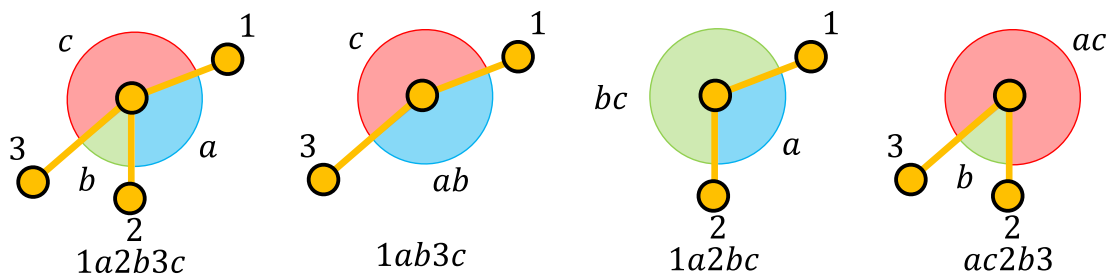


Figura 32: Combinações de testes para o alinhamento de vértices baseado em topologia(força bruta).

Inicialmente são identificados os pares de arestas de referência, sendo uma aresta de cada vértice, utilizada para calcular a primeira diferença angular Δ_{θ_e} entre a aresta de referência e a próxima aresta adjacente. As arestas de referência são identificadas através da semelhança dos seus comprimentos. Quando a diferença de comprimento de um par de arestas é inferior ao comprimento ρ_{ref} predeterminado, o par de aresta é selecionado como referência. Mais de um par de aresta de referência pode ser encontrado, neste caso todos os pares encontrados são utilizados e a solução que obtiver no menor erro é utilizada.

Após determinar todos os pares de referência, comparando todas as arestas dos dois vértices que estão sendo alinhados, é calculado o erro de variação angular ϵ_{θ_e} para cada par de aresta começando pela aresta de referência.

A Figura 33 mostra dois grafos G_1 e G_2 onde as GAs são representadas por letras maiúsculas e as variações angulares Δ_{θ_e} por letras em minúsculo, ambas seguidas pelo número que representa o seu grafo de origem, G_1 ou G_2 . As GAs A_1 e A_2 estão sendo alinhadas e o par de aresta em vermelho, $A_1 \leftrightarrow C_1$ e $A_2 \leftrightarrow B_2$, são as arestas de referência.

Na Figura 33(c) as arestas de referência, em vermelho, estão alinhadas e a diferença ϵ_{θ_e} é calculada entre os dois vértices, como é mostrado em vermelho na Figura 33(d). Neste caso as arestas de referência são comparadas com as próximas arestas adjacentes, $A_1 \leftrightarrow B_1$ e $A_2 \leftrightarrow C_2$. O erro final ϵ_{gu} é calculado pela Equação 3.9, do erro escaleno, considerando a diferença de variação angular ϵ_{θ_e} entre os dois pares de arestas e a diferença de comprimento entre as duas arestas finais $A_1 \leftrightarrow B_1 = \rho_u$ e $A_2 \leftrightarrow C_2 = \rho_v$.

Para que o erro ϵ_{gu} calculado seja aceito, o erro da próxima aresta, cuja a inclinação angular seja a mais próxima das arestas finais, mantendo as arestas de referência, precisa ser maior. Ou seja, para este exemplo, o erro encontrado trocando a aresta $A_2 \leftrightarrow C_2$ pela aresta $A_2 \leftrightarrow D_2$ deve ser maior. Caso contrario a aresta $A_2 \leftrightarrow B_2$ é descartada. O mesmo teste é realizado para as próximas arestas. Caso o erro seja aceito as novas arestas de referência passam a ser as últimas arestas testadas, neste caso $A_1 \leftrightarrow B_1$ e $A_2 \leftrightarrow C_2$. Este processo é repetido até que todas as arestas sejam analisadas.

No fim do processo, de forma semelhante aos métodos anteriores, um erro ϵ_{gu} médio é encontrado, e também o número de arestas aceitas para o cálculo do erro $\Sigma Acerto$.

3.4.1.3 Considerações finas

Os dois métodos de alinhamento com base na orientação das Gaussianas propostos são computacionalmente mais eficientes do que o baseado apenas na topologia, pois não precisam encontrar os pares de arestas de referência. Porém devido ao problema do achatamento causado pela distorção acústica, o último método proposto baseado apenas na topologia é mais indicado para calcular a similaridade entre vértices.

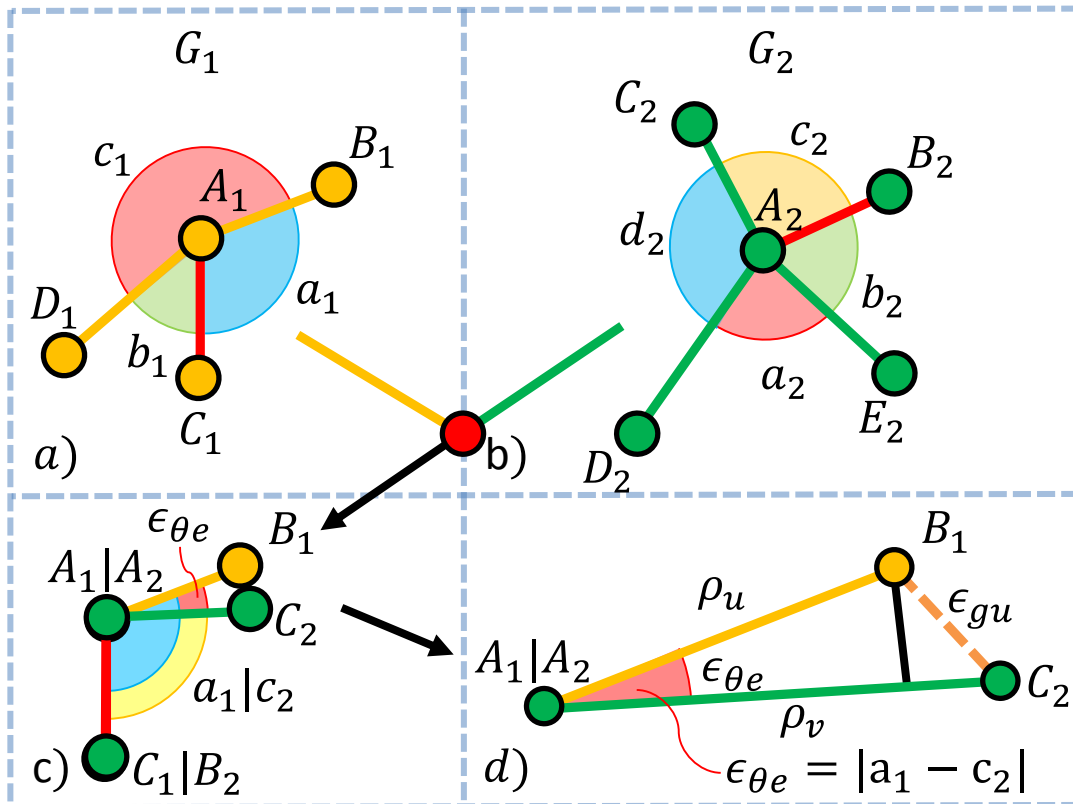


Figura 33: Alinhamento guloso de vértice com base em arestas adjacentes.

3.4.2 Encontrando a semelhança entre grafos

Nesta etapa a semelhança entre os vértices e/ou arestas são avaliadas e é definido quais os pares de vértices e arestas são semelhantes. Foram testadas duas abordagens, uma abordagem simples baseada no corte de pares de vértices com pouca semelhança. E outra utilizando o *framework* desenvolvido por Feng Zhou e Fernando De la Torre (ZHOU; TORRE, 2015) que implementa diversos métodos para encontrar a semelhança entre grafos como descrito na seção 2.3.3.3.

3.4.2.1 Semelhança entre grafos baseado em heurística

Para encontrar os pares de vértices semelhantes entre dois grafos G_k, G_l quaisquer, é preciso calcular a similaridade entre todos os vértices dos dois grafos. Esta abordagem trabalha com a minimização do erro entre vértices como métrica de similaridade. Um grafo bipartido G_b ponderado e completo é criado, onde as arestas estabelecem o erro ϵ e o número de arestas $\Sigma Acerto$ utilizadas para o cálculo do erro entre os dois vértices envolvidos.

Inicialmente as arestas com $\Sigma Acerto$ menor que o número mínimo de arestas

semelhantes definido pelo parâmetro $\rho_{similar}$ do algoritmo são eliminadas. Após, para cada vértice, é encontrado as duas arestas adjacentes com os menores erros ϵ_v . Se a diferença entre o erro das duas arestas encontradas for superior ao erro $\epsilon_{ambiguo}$ e o erro da aresta com o menor erro ϵ_v for inferior a $\epsilon_{v,max}$, ela é mantida e todas as demais arestas adjacentes são eliminadas.

No fim, as arestas restantes do grafo bipartido G_b definem os vértices considerados semelhantes.

3.4.2.2 Semelhança entre grafos utilizando o *framework* FGM

Como descrito na Seção 2.3.3.3, o *framework* desenvolvido por Feng Zhou e Fernando De la Torre (ZHOU; TORRE, 2015) reúne algumas das principais abordagens para encontrar as semelhanças entre grafos, dada a similaridade entre vértices e entre arestas. O *framework* baseia-se no cálculo das matrizes de similaridade entre vértices K_p utilizando uma das métricas descritas na Seção 3.4.1, ou através da diferença entre os parâmetros das GAs , e as matrizes de similaridade entre arestas K_q utilizando, por exemplo, a diferença entre os comprimentos ρ_e das arestas.

3.5 Ferramentas

Nesta seção será apresentada as ferramentas utilizadas neste trabalho. O trabalho foi desenvolvido utilizando a linguagem de programação C++ e a biblioteca de visão computacional *OpenCV*. A biblioteca *OpenCV* foi utilizada para realizar operações matriciais como o cálculo das matrizes de covariância e as decomposições em valores singulares descrita na seção 3.3, ajuste de transformações afins e operações sobre imagens envolvendo leitura, modificação e escrita de imagens.

A figura 34 mostra o diagrama UML simplificado do código desenvolvido.

As classes em laranja estão relacionadas com a segmentação, as classes em vermelho estão relacionados com a descrição e as classes em verde estão relacionadas com o cálculo da semelhança entre as imagens acústicas.

A classe *Sonar*, em azul, é a classe principal. Nela são definidas operações como *newImage*, que recebe uma imagem acústica e retorna a sua descrição em forma de grafo e a operação *similarity* que recebe duas imagens acústicas e retorna a sua similaridade.

A classe *Segmentation* realiza a segmentação de imagens acústicas com o auxílio das classes bases *SegmentSearch* e *SegmentExtractor* e da classe *Segment*, utilizada para a representação de segmentos. A classe *SegmentSearch* busca os segmentos conforme descrito na seção 3.2.2 e é derivada pela classe *LinearSegmentSearcher* ou pela classe *ThetaRhoSegmentSearcher* dependendo do método utilizado. A classe *SegmentExtractor*

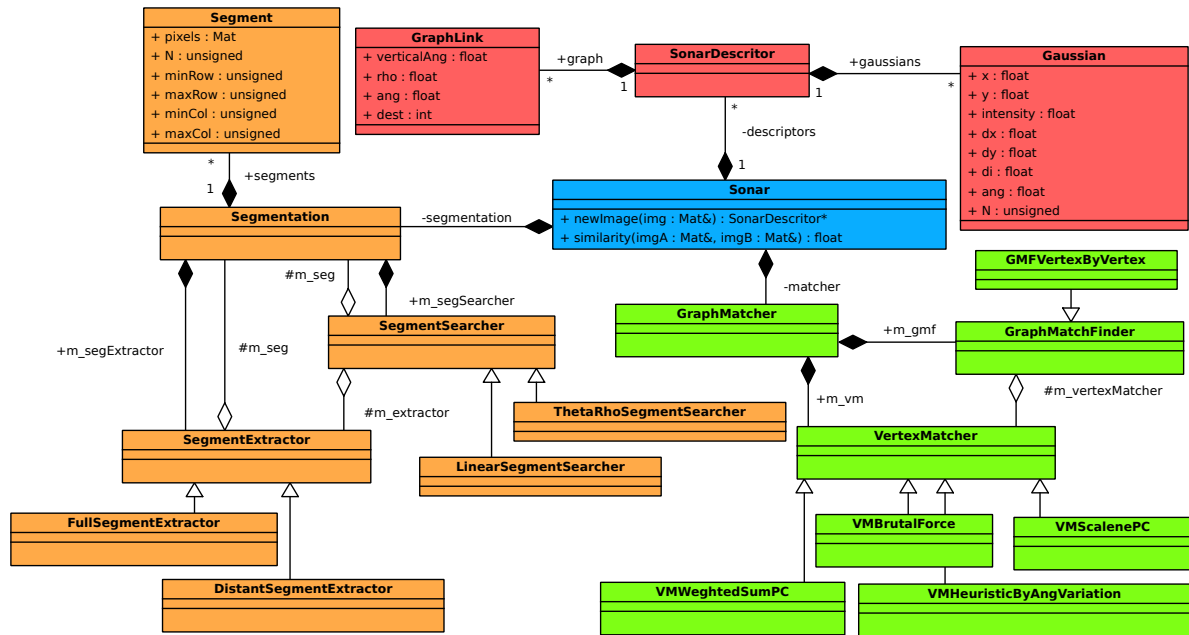


Figura 34: Diagrama UML do código em C++ gerado pelo programa Umbrello. Em laranja as classes relacionadas a segmentação, em vermelho as classes relacionadas com a descrição e em verde as classes relacionadas ao cálculo de semelhança de grafos.

extrai um segmento utilizando um dos métodos descritos na seção 3.2.1 e é derivada por uma das classes *FullSegmentExtractor*, *RelativeSegmentExtractor* ou *DistSegmentExtractor* dependendo do método utilizado.

A classe *SonarDescriptor* realiza a descrição das imagens utilizando os segmentos extraídos conforme descrito na seção 3.3. A classe *GraphLink* representa uma aresta do grafo e armazena informações como inclinação relativa ao eixo vertical *verticalAng*, inclinação relativa a orientação da *GA* de origem *ang*, comprimento *rho* e o vértice de destino *dest*. Uma vez que os *GraphLinks* são indexados em forma de lista de adjacência conforme descrito na seção 2.3, não é necessário armazenar o vértice de origem. A classe *Gaussian* representa as *GAs* utilizadas como vértices do grafo e contém informações do segmento extraído como posição média *x* e *y*, intensidade média *intensity*, maior desvio padrão *dy* segundo maior desvio padrão *dx*, desvio padrão de intensidade dos pixels *di*, orientação *ang* e o número de pixels do segmento *N*.

A classe *GraphMatcher* compara a descrição de duas imagens acústicas com o auxílio das classes bases *VertexMatcher* e *GraphMatchFinder*. A classe *VertexMatcher* calcula a similaridade entre vértices e é derivada por uma das classes *VMBrutalForce*, *VMWightedSumPC*, *VMHeuriscByAngVariation* ou *VMScalenePC* dependendo do método de comparação desejado. De forma semelhante a classe base *GraphMatchFinder* realiza a comparação entre dois grafos e é derivada pela classe *GMFVertexByVertex* na qual realiza a comparação entre todos os vértices dos dois grafos.

Outras classes foram desenvolvidas para a integração com o *framework* disponibilizado por (ZHOU; TORRE, 2015), onde os grafos de descrição das imagens são convertidos em matrizes no formato CSV, e também para o desenvolvimento da ferramenta de visualização e criação de *Ground Truth* que será discutida no capítulo 4.

3.6 Conclusões Finais

Este capítulo apresentou a arquitetura para a descrição e detecção de imagens acústicas semelhantes que foi dividida em 6 etapas, sendo que a última é a aplicação do método como detector de *loop* em um sistema de SLAM. Foram propostos três métodos de segmentação baseado na busca de pixels conexos e dois métodos de busca de segmento, sendo um linear e outro baseado nas características do sonar FLS. Foi proposto um método de descrição de imagens, onde é utilizado um grafo para representar as relações de inclinação e distância entre os objetos detectados e o modelo probabilístico Gaussiano para representar a forma dos objetos detectados. Foram apresentadas duas formas de comparação dos descritores, uma utilizando uma heurística proposta de corte do grafo e outra utilizando o *framework* disponibilizado em (ZHOU; TORRE, 2015), onde é reunido os métodos considerados estado da arte para a comparação de grafos utilizando a abordagem QAP.

Também foram apresentadas as ferramentas utilizadas para construção deste trabalho e a organização do código desenvolvido através do digrama UML. O método proposto será avaliado no próximo capítulo utilizando o dataset ARACATI 2014 (SILVEIRA et al., 2015).

4 Avaliação dos Métodos

Neste capítulo são apresentados os resultados obtidos com esse trabalho. Para a validação foi utilizado o dataset ARACATI 2014 disponibilizado por (SILVEIRA et al., 2015) que é descrito em maiores detalhes na próxima seção 4.1. A Seção 4.2 mostra os resultados obtidos na etapa de processamento digital de imagem. A seção 4.3 mostra os resultados obtidos na etapa de segmentação de imagem, a seção 4.4 mostra os resultados obtidos na etapa de descrição de imagens, a seção 4.5 mostra os resultados obtidos pela etapa de comparação do método.

4.1 Dataset ARACATI 2014

O dataset ARACATI 2014 ,disponibilizado por (SILVEIRA et al., 2015), foi coletado utilizando um mini-ROV modelo LBV300-5 produzido pela Seabotix. O LBV300-5 foi equipado com um sonar de imageamento frontal BlueView P900-130 produzido pela Teledayne e foi amarrado à uma prancha de *Stand Up Paddle* na qual estava sendo rastreada por um DGPS. A Figura 35(a) mostra o LBV300-5 com o sonar P900-130. As especificações do sonar BlueView P900-130 podem ser conferidas na Tabela 2. Durante todo o experimento foi utilizado o sonar configurado para cobrir uma distância de 30 metros.

Os dados foram coletados em uma marina do Yacht Clube de Rio Grande, situado na cidade de Rio Grande, no estado do Rio Grande do Sul - Brasil. O mini-ROV percorreu cerca 776 metros durante 1 hora e 15 minutos. Durante todo o percurso, o ROV manteve-se próximo a superfície, devido ao uso da prancha e o rastreamento por um DGPS, a uma distância aproximada de 3 metros do fundo. Foram coletadas 9659 imagens acústicas e 4200 posições pelo DGPS. O trajeto percorrido pode ser visualizado na Figura 36.

A marina do Yacht Clube de Rio Grande pode ser classificada como um local parcialmente estruturado, pois são encontrados objetos como cascos de barcos, postes, o cais e algumas pedras no fundo na maioria das imagens acústicas. Estes elementos podem ser utilizados como marcação da imagem. Na Figura 35(b) e 35(c) são destacados alguns postes e cascos de barcos marcados de forma empírica. Outra característica da região é a alta turbidez da água que impossibilita a utilização de câmeras ópticas.

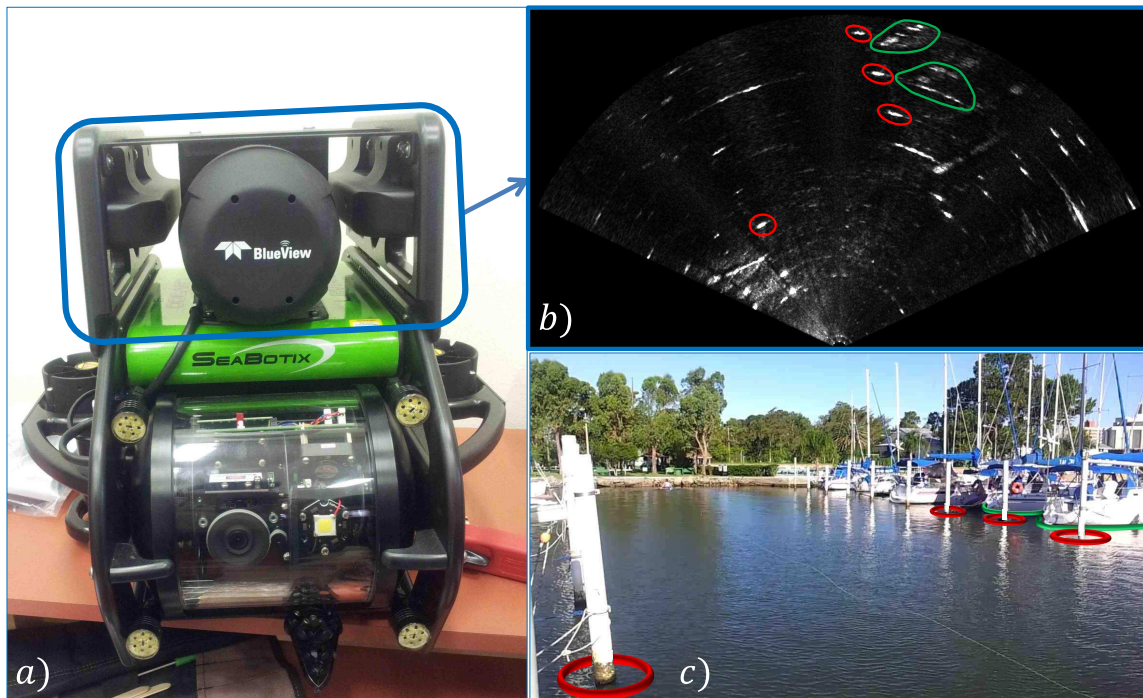


Figura 35: Foto do LBV300-5 com o sonar P900-130 em (a). Em (b) um exemplo de imagem acústica capturado e em (c) uma imagem óptica do ambiente capturada aproximadamente na mesma região da imagem acústica de (b). Em vermelho estão marcados os postes e em verde os cascos dos barcos identificados de forma empírica nas duas imagens.

4.2 Processamento digital de imagem

O método de correção de imagens acústicas descritos na seção 3.1 foi aplicado nas imagens do dataset ARACATI 2014. Foi calculada a média de 443 imagens acústicas e levantado o padrão de insonificação do sonar que pode ser visto na Figura 37.

O resultado da aplicação do padrão de insonificação em um par de imagens acústica pode ser visualizado na Figura 38. A principal diferença observável é a redução da interferência entre os *beams* acústicos. Esta interferência também é visível no padrão de insonificação da Figura 37. Entretanto essa correção não traz uma melhoria significativa para o método de descrição de imagens, pois ela não provoca uma alteração na intensidade dos pixels suficientemente grande para afetar as regiões extraídas na etapa de segmentação. As regiões segmentadas tipicamente possuem intensidades entre 200 a 2300 podendo atingir o valor máximo de 65535 da representação de 16 bits. A aplicação da correção nas imagens provoca alterações de intensidade nos pixels entorno de 30 para mais ou para menos.

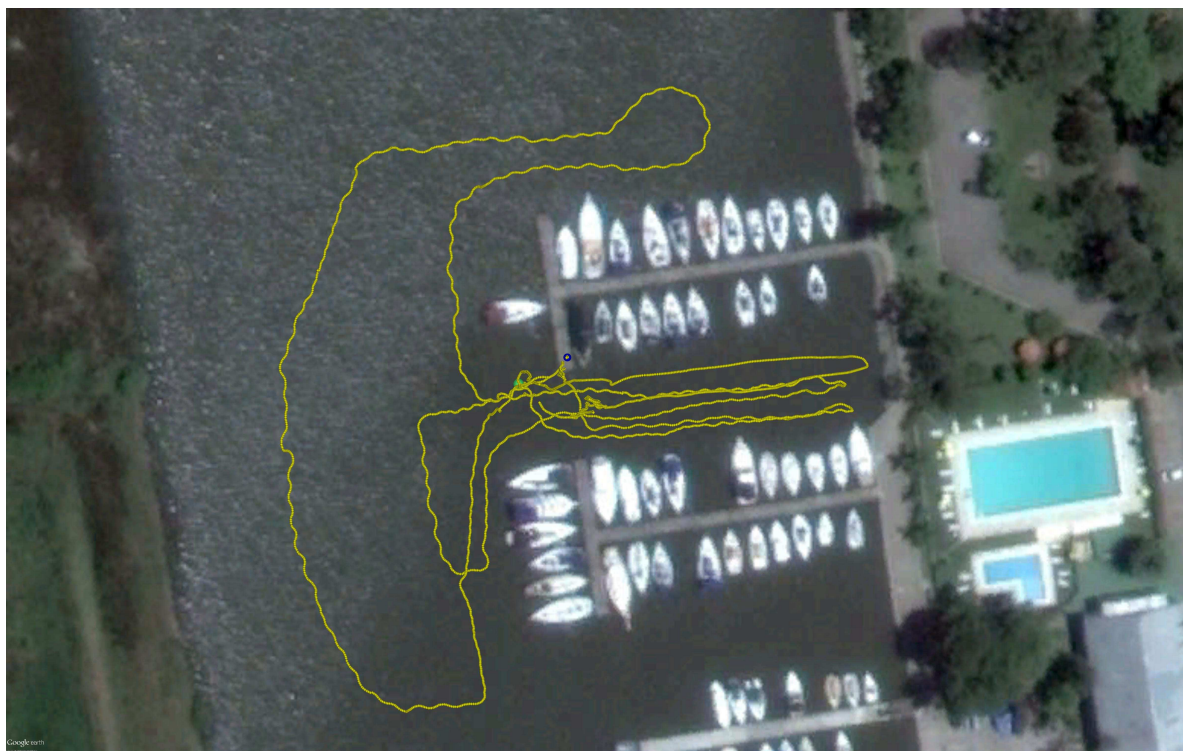


Figura 36: Imagens de satélite obtidas pelo software Google Earth e os pontos de GPS coletados. A trajetória percorrida começa no ponto em azul e terminando no ponto em verde.

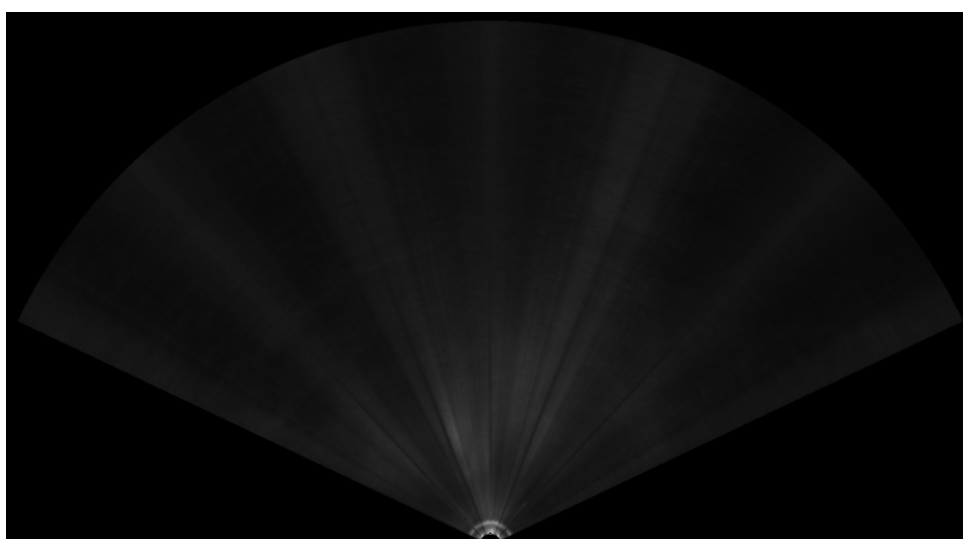
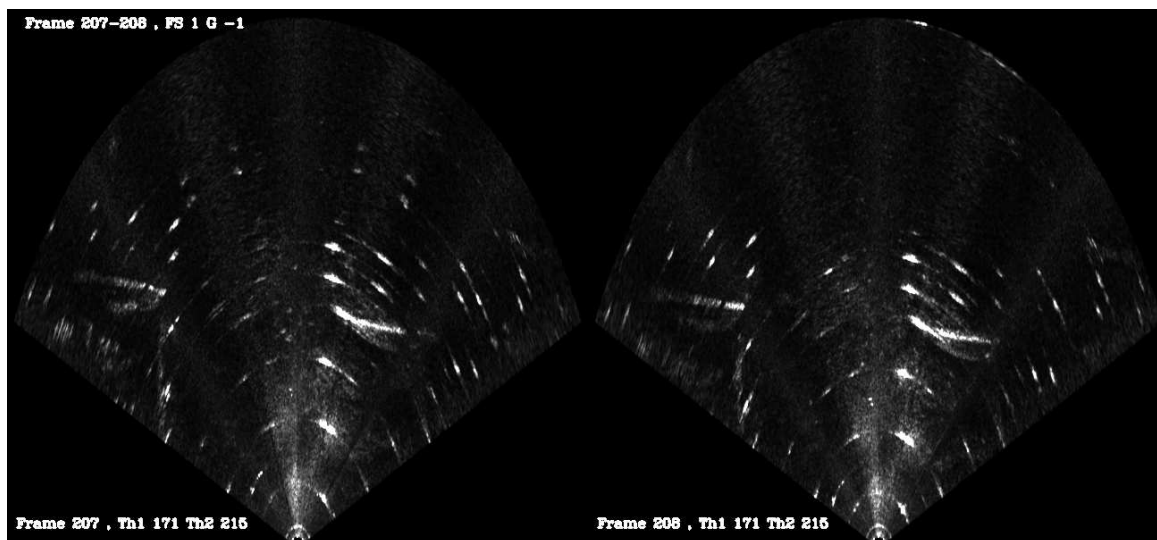
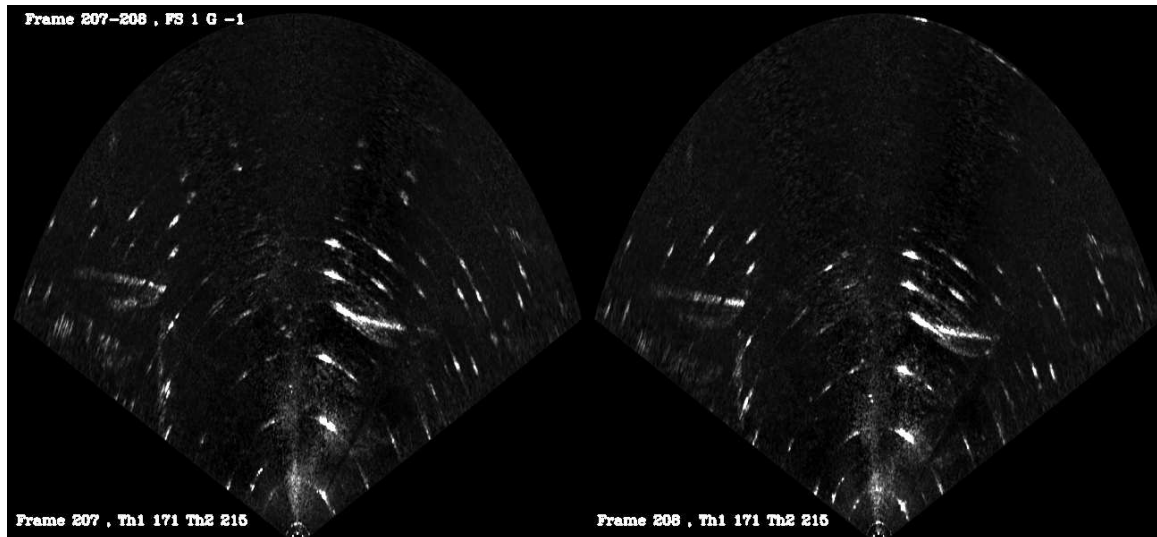


Figura 37: Padrão de insonificação do sonar de imageamento frontal Blue View P900-130, utilizado no dataset ARACATI 2014, obtido através da média de 443 imagens.



(a) Par de imagens acústicas originais.



(b) Par de imagens acústicas após a aplicação do padrão de insonificação.

Figura 38: Resultado da aplicação do padrão de insonificação da figura 37 em um par de imagens.

Tabela 3: Resultados quando alterando os parâmetros ρ_{linear} e $\rho_{recursivo}$

Figura	ρ_{linear}	$\rho_{recursivo}$	N. Reg.	> Reg.	< Reg.	Temp. Exec. Médio
39(b)	800	400	19	383	28	11506 μ seg
39(c)	800	200	17	499	72	11851 μ seg
39(d)	800	100	16	1892	44	14341 μ seg
39(e)	215	171	51	637	15	13039 μ seg
39(f)	200	199	46	500	15	12362 μ seg
39(g)	200	100	54	1892	22	10519 μ seg
39(h)	100	99	101	1937	15	16760 μ seg
39(i)	100	50	97	22545	18	20303 μ seg
39(j)	50	49	226	23132	15	23489 μ seg

4.3 Segmentação

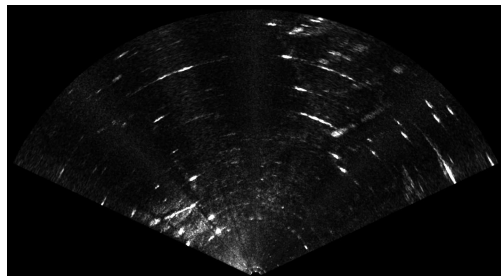
Nesta seção serão mostrados alguns resultados obtidos com os métodos de segmentação propostos.

4.3.1 Sensibilidade aos Parâmetros ρ_{linear} e $\rho_{recursivo}$

Os resultados obtidos utilizando a segmentação completa, descrita na seção 3.2.1.1, e a busca linear, descrita na seção 3.2.2.1, podem ser visualizados na Figura 39 e na Tabela 3. É mostrado o comportamento do método ao alterar os parâmetros ρ_{linear} e $\rho_{recursivo}$. Na Figura 39(a) é exibida a imagem acústica original. As figuras 39(b), 39(c), 39(d), 39(f), 39(g), 39(h), 39(i) e 39(j) mostram os resultados obtidos ao alterar os parâmetros ρ_{linear} e $\rho_{recursivo}$. Para ambos os testes foi utilizada a vizinhança 8.

Analisando os resultados obtidos, pode-se observar casos em que um único objeto foi separado em dois segmentos nas figuras 39(b), 39(c). Casos onde não houve problemas de segmentação, figuras 39(d), 39(e) e casos onde mais de um objeto está no mesmo segmento nas figuras 39(g), 39(h), 39(i) e 39(j). Entre os resultados apresentados, o mais próximos do esperado, onde cada objeto é segmentado separadamente, são os das figuras 39(d) e 39(e), porém em ambos os casos existem objetos que não foram segmentados.

A Tabela 3 mostra os resultados quantitativos dos testes realizados na Figura 39. A primeira coluna relaciona a imagem que mostra os segmentos extraídos em cada teste. A segunda e a terceira coluna indicam, respectivamente, os parâmetros ρ_{linear} e $\rho_{recursivo}$ utilizados. Na quarta coluna é mostrado o número de segmentos identificados em cada teste. A quinta coluna indica o tamanho em número de pixel do maior segmento. A sexta coluna mostra o tamanho em número de pixels do menor segmento e finalmente a sétima e última coluna mostra o tempo de execução para a segmentação em cada teste. O equipamento utilizado nos testes foi um notebook com processador Intel Core i3-2310M 2.1 GHz memória RAM 4GB 1333 MHz DDR3 e HD 5400 RPM SATA 2 de 3Gb/s.



(a) Imagem Acústica Original.

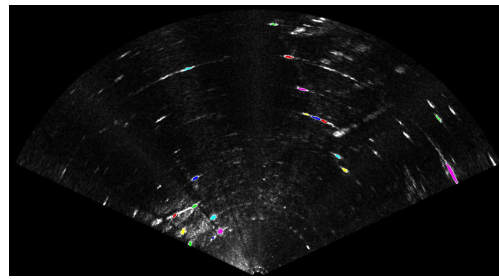
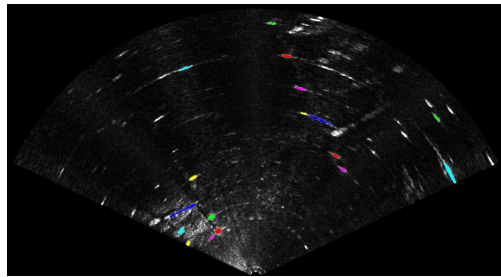
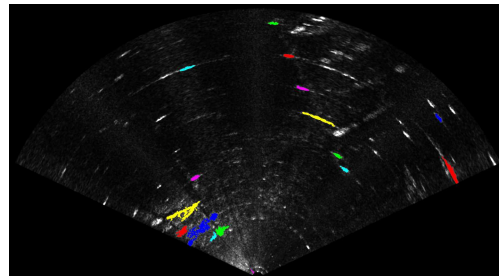
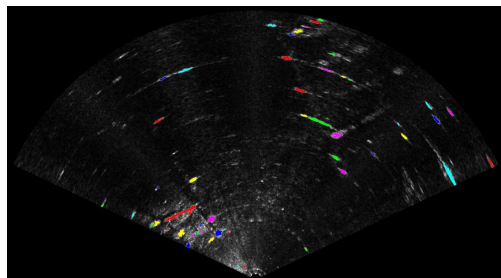
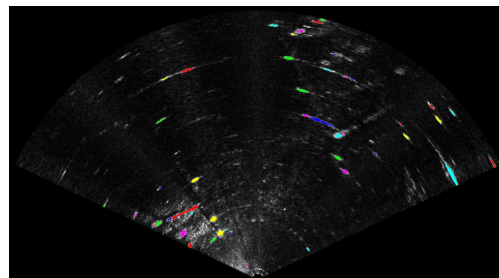
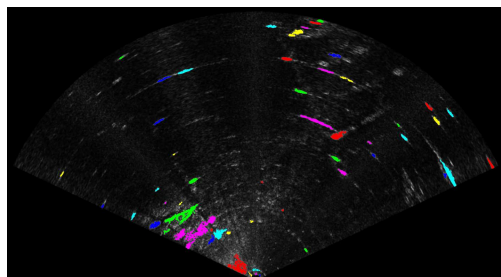
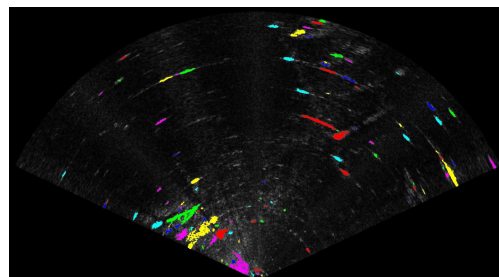
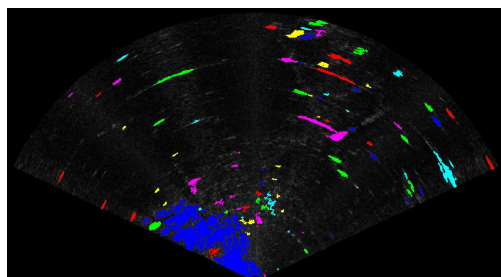
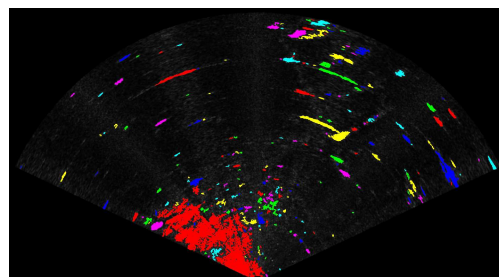
(b) $\rho_{linear} = 800$ e $\rho_{recurso} = 400$ (c) $\rho_{linear} = 800$ e $\rho_{recurso} = 200$ (d) $\rho_{linear} = 800$ e $\rho_{recurso} = 100$ (e) $\rho_{linear} = 215$ e $\rho_{recurso} = 171$ (f) $\rho_{linear} = 200$ e $\rho_{recurso} = 199$ (g) $\rho_{linear} = 200$ e $\rho_{recurso} = 100$ (h) $\rho_{linear} = 100$ e $\rho_{recurso} = 99$ (i) $\rho_{linear} = 100$ e $\rho_{recurso} = 50$ (j) $\rho_{linear} = 50$ e $\rho_{recurso} = 49$

Figura 39: Resultados do método de busca por regiões de interesse quando alterado os parâmetros ρ_{linear} e $\rho_{recurso}$

Uma desvantagem deste método é a sua sensibilidade aos parâmetros de busca ρ_{linear} e $\rho_{recursivo}$ que podem variar dependendo do ambiente que está sendo explorado. Como mostra a Tabela 3, quanto maior for ρ_{linear} menos segmentos são extraídos, quanto menor for $\rho_{recursivo}$ maior é o tamanho de cada segmento extraído. Ao utilizar valores de intensidade baixos, o ruído acústico acaba sendo segmentado em conjunto com os objetos provocando a segmentação errada da imagem, ao utilizar valores de intensidade altos, poucos objetos são segmentados.

Apesar do método não atingir o resultado ótimo esperado, onde todos os objetos da imagem seriam segmentados corretamente, a solução alcança bons resultados para os valores de ρ_{linear} e $\rho_{recursivo}$ da Figura 39(e), onde a maioria dos objetos são segmentados corretamente. Ressalta-se que problemas de oclusão serão tratados na etapa seguinte de comparação das imagens acústicas, uma vez que esta pode estar ocorrendo por outros motivos, como a alterações do ponto de vista do sonar, provocando o movimento das sombras acústicas e cobrindo outros objetos, ou alterando o ângulo de incidência das ondas acústicas na superfície de um objeto reduzindo os seus retornos acústicos ou em casos onde o ângulo de incidência é muito grande, causando o desaparecimento por completo do objeto.

Quanto a complexidade computacional do método de segmentação proposto, a etapa de extração de segmento, na qual utiliza uma busca em largura (BFS), possui uma complexidade de tempo linear $O(|E|)$ onde $|E|$ é o número de arestas do grafo da busca. Para o caso das imagens, cada pixel possui até 8 vizinhos, portanto a complexidade de tempo continua linear, $O(N)$, onde N é o número de pixels da imagem. A etapa de localização das regiões para a segmentação também possui complexidade linear $O(N)$, pois todos os pixels da imagem serão visitados uma única vez. Como é garantido que um pixel não é extraído mais de uma vez de uma imagem, a etapa de segmentação possui uma complexidade de tempo linear em função do número de pixels da imagem. As imagens acústicas do dataset ARACATI 2014 possuem a resolução de 1430x781 totalizando 1116830 pixels.

Quanto aos tempos de execução mostrado na Tabela 3, no qual possuem uma média de 14907 μseg ou 67Hz, são aceitáveis, uma vez que a taxa de atualização do sonar utilizado possuem uma taxa de atualização máxima de 15 Hz especificada pelo fabricante. Entretanto, na prática, esta taxa é alcançada apenas quando o equipamento é configurado para cobrir a menor distância possível de 4 metros. No caso médio, onde é configurado um alcance de 30 metros, a taxa de atualização fica entorno de 6 Hz.

Apesar de não existir uma diferença de tempo de execução significativa que dependa dos parâmetros ρ_{linear} e $\rho_{recursivo}$, a detecção de muitos segmentos pode afetar o desempenho das etapas de descrição e comparação das imagens acústicas.

4.3.2 Segmentação de borda vs segmentação relativa a distância e o problema de múltiplas segmentações

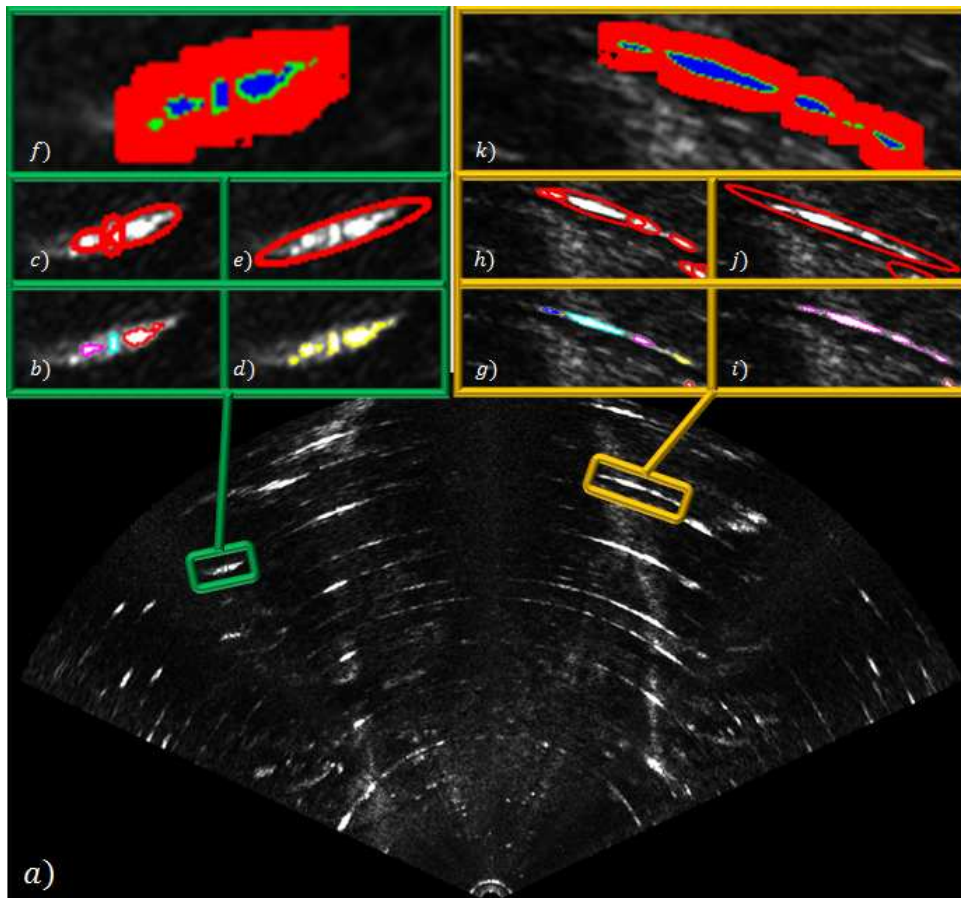


Figura 40: Comparação do método de extração de segmentos baseado em bordas com o de extração de segmentos relativo a distância.

Nesta seção é mostrada uma situação em que o método de segmentação baseado em distância resolve o problema de múltiplas segmentações de um mesmo objeto causado pela segmentação de borda. Nos experimentos realizados foram utilizados os mesmos valores de intensidade mínima para extração de segmento $\rho_{\text{precursivo}}$ para as duas abordagens.

A Figura 40(a) mostra uma imagem acústica e destaca as duas regiões estudadas. As figuras 40(b) e 40(g) mostram o resultado da segmentação de borda descrita na seção 3.2.1.2. Em ambas as imagens mais de um segmento foi extraído porque as regiões de alta intensidade não são conexas. Os resultados da descrição dos segmentos extraídos pelo método de extração de bordas pode ser visualizado nas figuras 40(c) e 40(h).

As figuras 40(f) e 40(k) descrevem o comportamento da extração de segmentos relativa a distância descrita na seção 3.2.1.3. Os pixels em azul representam as regiões válidas para a extração do segmento, cuja a intensidade está acima da intensidade mínima $\rho_{\text{precursivo}}$. Os pixels em verde são as bordas que delimitam os pixels válidos dos pixels inválidos para a segmentação. Os pixels em vermelho representam os pixels inválidos

que foram explorados porque a sua distância até o pixel verde mais próximo é menor que o parâmetro de distância D_{seg} utilizado para determinar o fim da busca. Durante a exploração dos pixels inválidos foram encontrados grupos de pixels válidos que foram incluídos no mesmo segmento. O resultado da descrição dos segmentos extraídos pelo método de extração de segmentos relativo a distância podem ser visualizados nas figuras 40(e) e 40(j).

O método de extração de segmentos relativo a distância eliminou o problema de múltiplos segmentos para o mesmo objeto causado pelo método de extração de borda de segmentos. O método de extração de segmentos relativo a distância é mais abrangente que o de segmentação de borda, uma vez que o método de segmentação de borda é o método de segmentação relativo a distância com $D_{seg} = 0$.

4.3.3 Segmentação baseada na busca de picos em *beams* acústicos

Esta abordagem de segmentação surgiu da necessidade de adaptar os parâmetros ρ_{linear} e $\rho_{recursivo}$ de forma automática dependendo das variações de intensidade das imagens acústicas. Como descrito na seção 3.2.2.2, a intensidade $\rho_{recursivo}$ é calculada conforme a variação de intensidade H_{total} de cada pico identificado nos *beams* acústicos. Para isso é necessária a definição das taxas π_{fim} , $\pi_{recursivo}$ e a intensidade H_{busca} utilizados como parâmetros do método.

Com o objetivo de apresentar o comportamento do algoritmo ao alterar os seus parâmetros, foram realizados testes utilizando o *beam* representado pela linha em azul na imagem acústica da Figura 41(b), o seu perfil de intensidade é mostrado na Figura 41(a). O eixo horizontal da Figura 41(a) representa os *bins* do *beam* analisado e o eixo vertical representa a intensidade de cada *bin*. Os comportamentos do algoritmo ao alterar os parâmetros $\pi_{recursivo}$, π_{fim} e H_{busca} são mostrados respectivamente nas figuras 42, 44 e 46. As linhas horizontais presentes nos perfis de intensidade indicam o início e o término da análise de um pico, as linhas horizontais vermelhas indicam que o pico não foi aceito para extração de segmento, porque não atingiu a variação de intensidade mínima H_{busca} , as linhas em azul indicam os picos aceitos para a extração de segmento e a sua altura com relação ao eixo horizontal indica a intensidade $\rho_{recursivo}$ encontrada para o pico detectado. Os círculos coloridos no topo de cada pico aceito representa o *bin* de maior intensidade que é utilizado como ponto de partida para extração de segmento. Um círculo com a mesma cor é desenhado na imagem acústica a direita, na posição correspondente ao *bin* do perfil de intensidade a esquerda.

A Figura 42 e o gráfico da figura 43 mostram que o parâmetro $\pi_{recursivo}$ influencia no tamanho dos segmentos extraídos, começando por um valor grande, $\pi_{recursivo} = 0.43$, nas figuras 42(a) e 42(b) resultando na extração de segmentos pequenos, um valor intermediário, $\pi_{recursivo} = 0.26$, nas figuras 42(c) e 42(d) resultando na extração de segmentos maiores

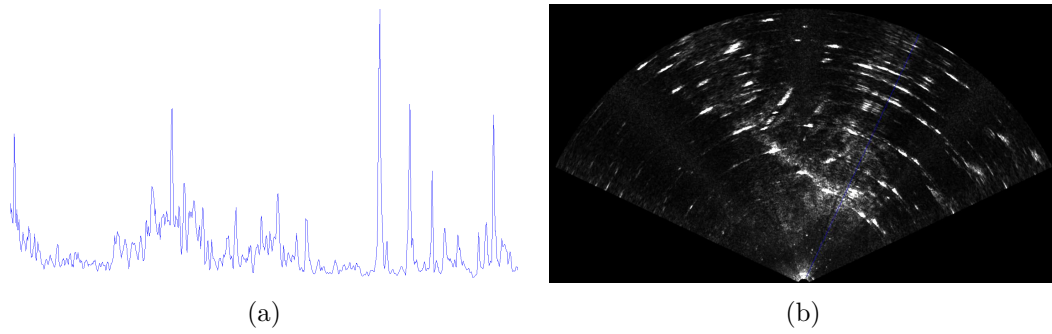


Figura 41: Imagens utilizadas para o estudo do comportamento dos parâmetros $\pi_{\text{recursivo}}$, π_{fim} e H_{busca} . Em (a) o perfil de intensidade do *beam* representado pela linha azul na figura (b). O eixo horizontal representa os *bins* e o eixo vertical representa a intensidade de cada *bin*.

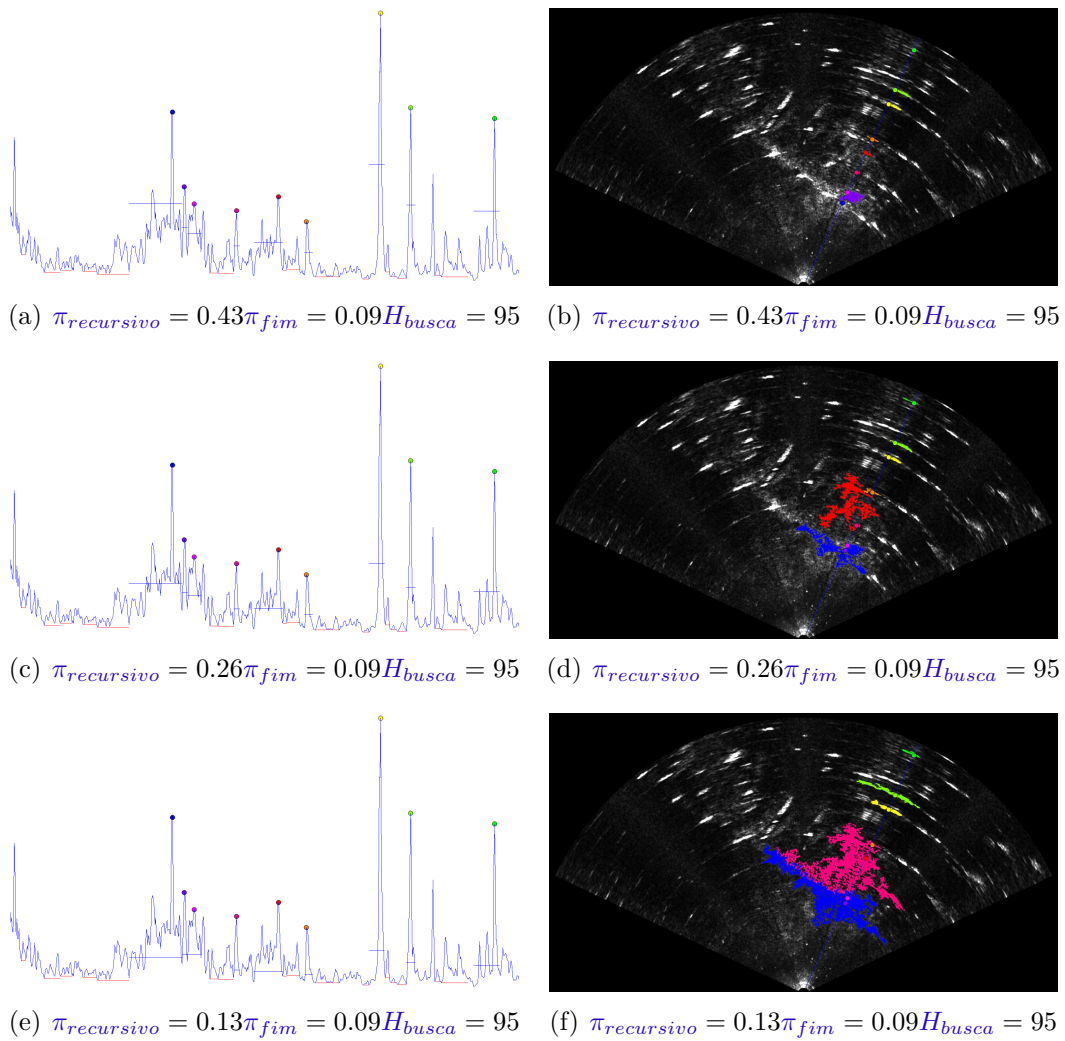


Figura 42: Comportamento do parâmetro $\pi_{\text{recursivo}}$.

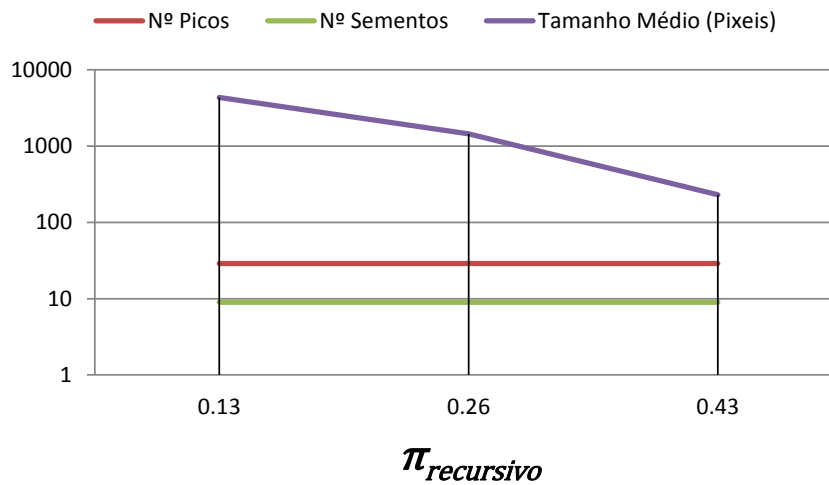


Figura 43: Gráfico do comportamento a busca de segmentos ao variar o parâmetro $\pi_{\text{recursivo}}$.

com um pouco de ruído, e um valor pequeno, $\pi_{\text{recursivo}} = 0.13$ nas figuras 42(e) e 42(f) resultando na extração de grandes segmentos com bastante ruído. O valor de $\rho_{\text{recursivo}}$, representado pela distância entre as linhas horizontais em azul até o eixo horizontal do perfil de intensidade, variam conforme o valor de $\pi_{\text{recursivo}}$ e a altura do pico detectado H_{total} .

Na Figura 44 é analisado o comportamento do parâmetro π_{fim} . A sua influência é no número de picos analisados, começando por um valor pequeno, $\pi_{\text{fim}} = 0.03$, como mostra nas figuras 44(a) e 44(b) onde foram analisados menos picos, porém com uma quantidade maior de *bins*, picos mais largos. Nas figuras 44(c) e 44(d) foi utilizado um valor intermediário, $\pi_{\text{fim}} = 0.1$, e nas figuras 44(e) e 44(f) foi utilizado um valor grande, $\pi_{\text{fim}} = 0.76$, resultando na análise de muitos picos, porém com poucos *bins*.

O gráfico da figura 45 mostra que o número de picos analisados aumenta conforme o aumento da taxa π_{fim} .

A Figura 46 mostra que o comportamento do parâmetro H_{busca} está relacionado com quantidade de segmentos extraídos. Este fato também pode ser visualizado na Gráfico da Figura 47.

As figuras 46(a) e 46(b) mostram que para um valor pequeno, $H_{\text{busca}} = 40$, muitos picos são aceitos resultando na extração de um número maior de segmentos. As figuras 46(c) e 46(d) mostram o resultado para um valor intermediário, $H_{\text{busca}} = 110$, e as figuras 46(e) e 46(f) mostram o resultado quando $H_{\text{busca}} = 160$, onde poucos picos são aceitos e consequentemente poucos segmentos são extraídos. A Figura 48 mostra a análise de picos em 25 *beams*, na prática são analisados os 768 *beams* do sonar *Blue View* P900-130.

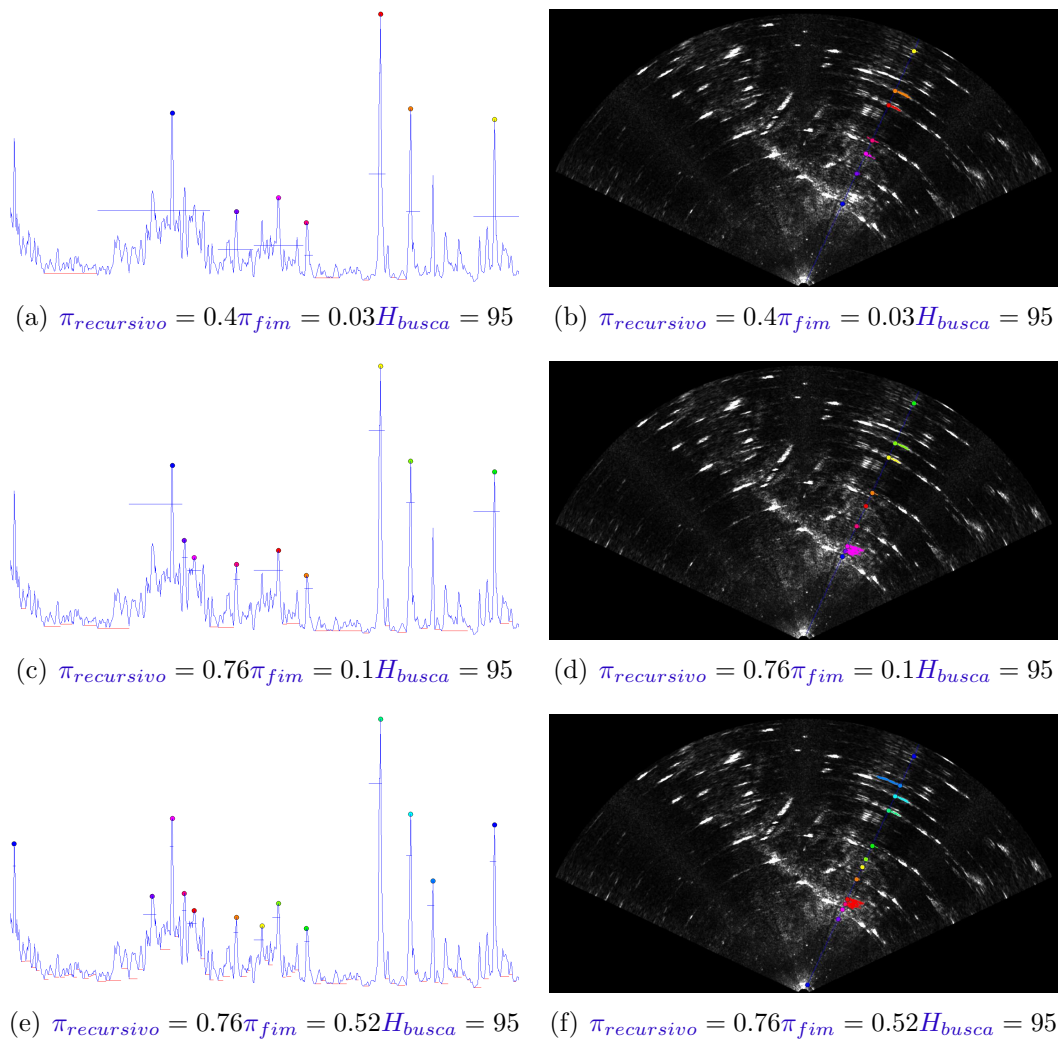


Figura 44: Comportamento do parâmetro π_{fim} .

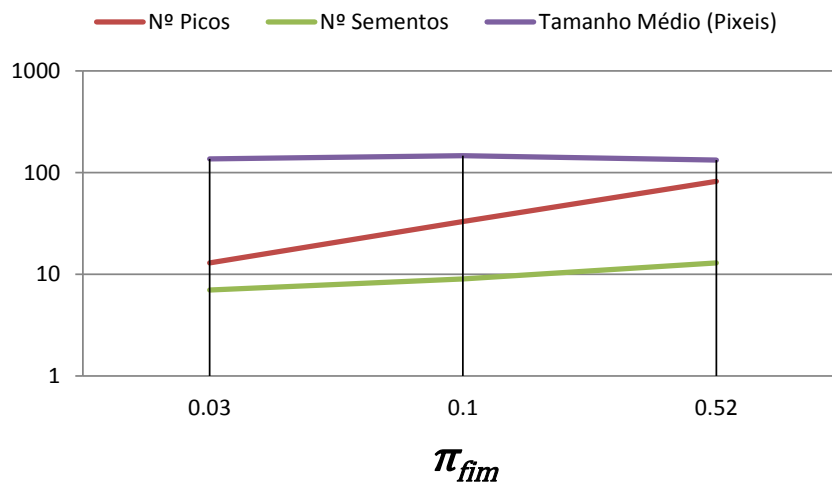


Figura 45: Gráfico do comportamento a busca de segmentos ao variar o parâmetro π_{fim} .

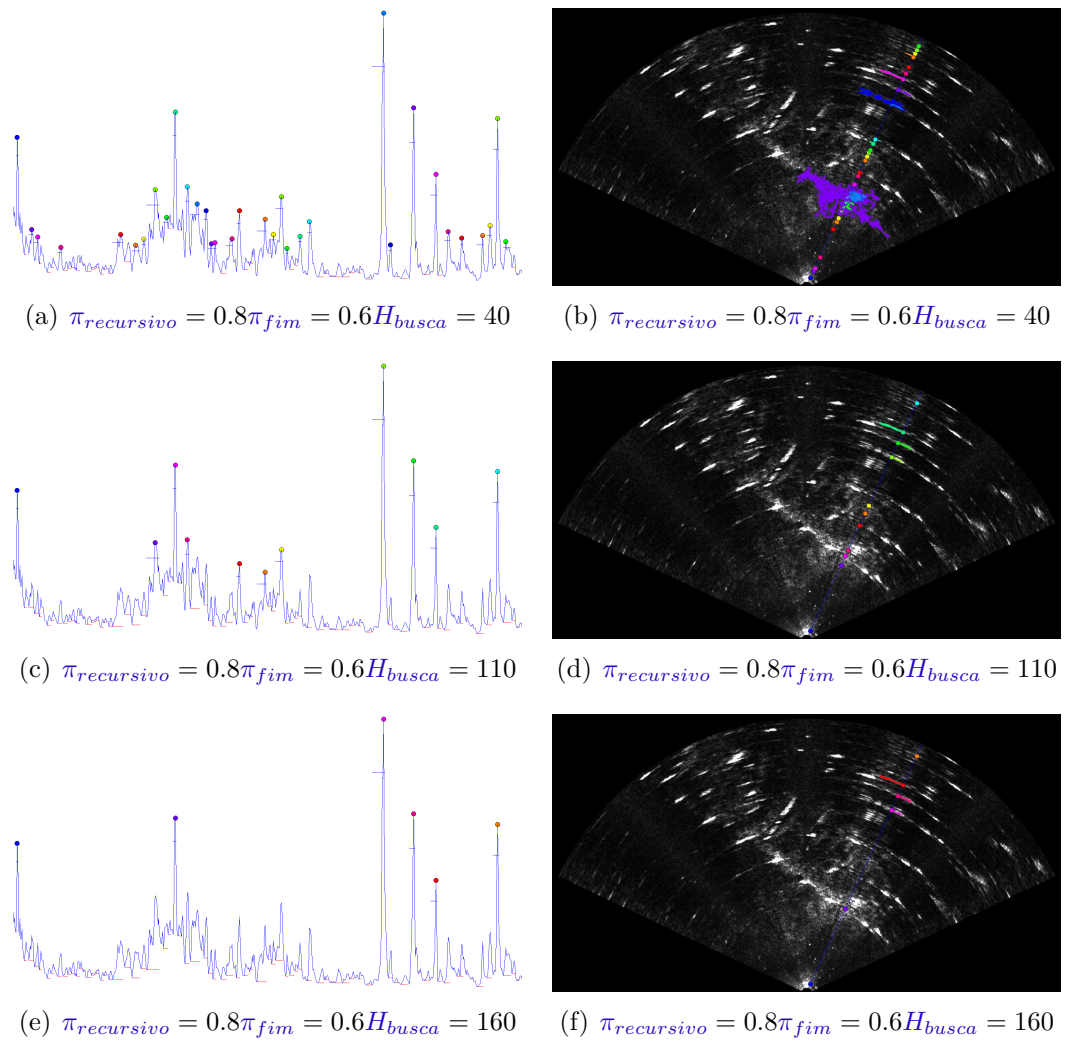


Figura 46: Comportamento do parâmetro H_{busca} .

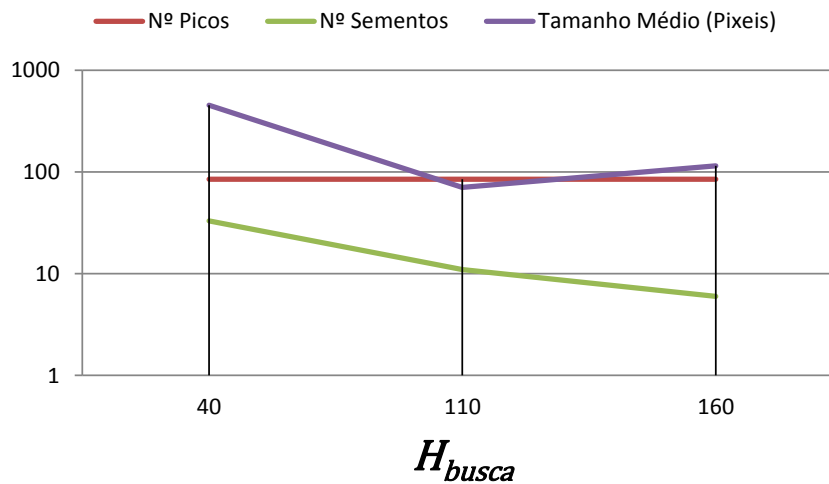


Figura 47: Gráfico do comportamento a busca de segmentos ao variar o parâmetro H_{busca} .

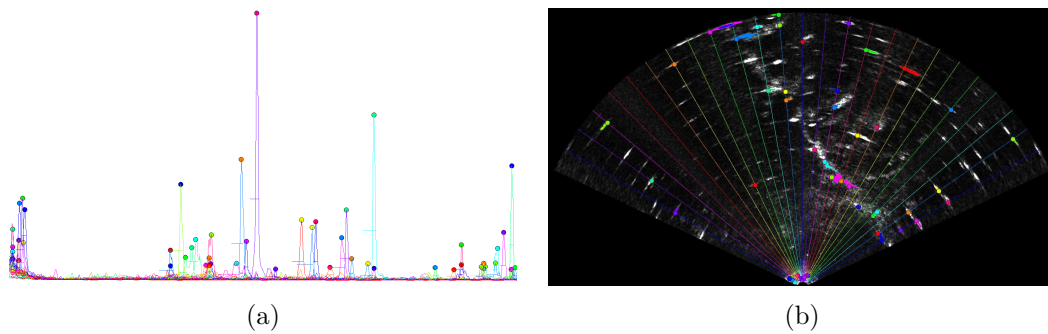


Figura 48: Análise de picos de 25 *beams* de uma imagem acústica utilizando os parâmetros $\pi_{fim} = 0.6$, $\pi_{recursivo} = 0.3$ e $H_{busca} = 165$.

A principal vantagem desta abordagem com relação a apresentada na seção 3.2.2.1 é que a busca por segmentos não é fixa a um determinado limiar de intensidade. Nesta abordagem a intensidade de busca $\rho_{recursivo}$ é ajustado de acordo com a variação de intensidade H_{total} de cada pico encontrado, permitindo que numa mesma imagem acústica sejam segmentadas regiões com pixels de baixa intensidade e regiões com pixels alta intensidade.

4.4 Resultados do método de descrição de imagens acústicas

Nesta seção será mostrado o comportamento do método de descrição de imagens acústicas. Nesta etapa a imagem segmentada é descrita utilizando o modelo Gaussiano para representar a forma dos objetos detectados e um grafo para estabelecer as suas relações topológicas. O parâmetro R_{lig} determina a distância máxima para que dois vértices sejam ligados por uma aresta do grafo.

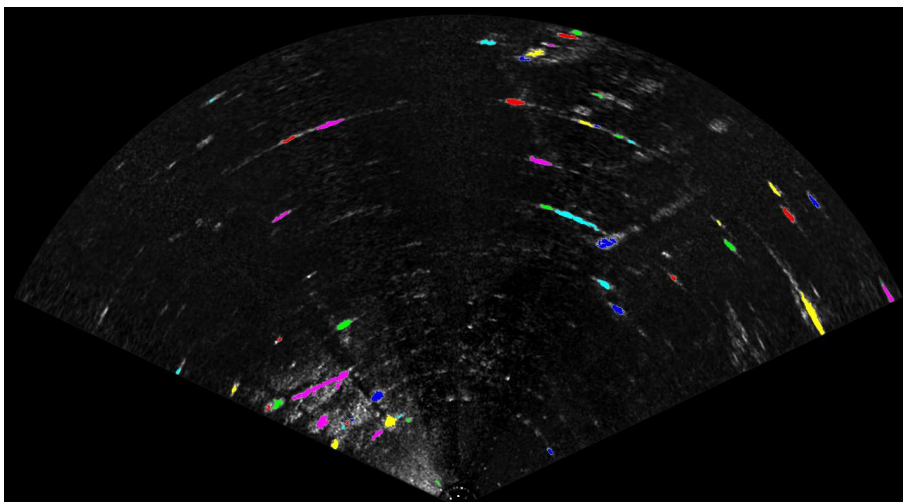


Figura 49: Imagem após a etapa de segmentação. Os segmentos estão marcados por pixels coloridos.

A Figura 50 mostra o comportamento do algoritmo para descrever a imagem segmentada da Figura 49 ao alterar o parâmetro R_{lig} . Como mostra a Tabela 4, quanto maior a distância máxima de ligação R_{lig} , mais denso é o grafo. A densidade do grafo está relacionada com o nível de detalhes da descrição da imagem. O nível de detalhes influencia na precisão e na capacidade de tratamento das oclusões do método de comparação dos descritores, uma vez que é fornecido mais informações para a comparação, entretanto o custo computacional em termos de processamento e consumo de memória para realizar as análises aumenta.

Tabela 4: Resultados obtidos na etapa de construção do grafo.

Parâmetro R_{lig}	Número de Arestas	Figura
300	326	50(c)
200	214	50(b)
110	90	50(a)

4.5 Comparação

Nesta seção serão mostrados os resultados da etapa de comparação entre os descritores de imagens acústicas propostos.

A Figura 51 mostra a comparação de duas imagens acústicas, Figura 51(a), o resultado da etapa de descrição das imagens é exibido na Figura 51(b) e as semelhanças encontradas pelo método de comparação é exibido na Figura 51(c). Para alcançar esses resultados foram utilizados o método de segmentação completo descrito na seção 3.2.1.1 a busca linear descrita na seção 3.2.2.1, semelhança entre vértices utilizando o algoritmo de erro escaleno descrito na seção 3.4.1.1 e o cálculo de similaridade entre grafos utilizando heurísticas descrito na seção 3.4.2.1 e foram utilizados os parâmetros estabelecidos na Tabela 5.

Tabela 5: Parâmetros adotados para obter os resultados do experimento de comparação de grafos.

Busca de segmentos linear	
ρ_{linear}	215
$\rho_{recursivo}$	171
Descrição	
R_{lig}	250
$\pi_{recursivo}$	0.98
Semelhança entre vértices com erro escaleno	
$m\mathcal{X}\epsilon_{esc}$	30
Semelhança entre grafos baseado em heurísticas	
$\rho_{similar}$	4.0
$\epsilon_{ambiguo}$	5.0

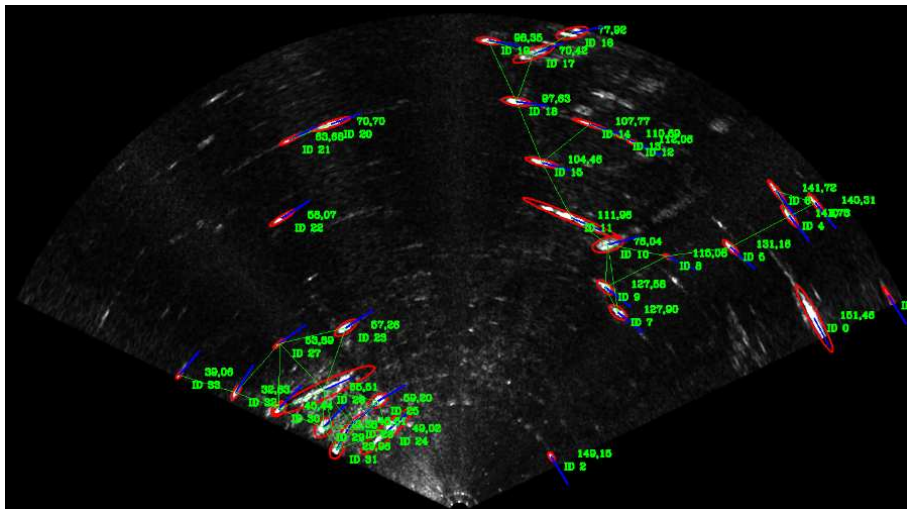
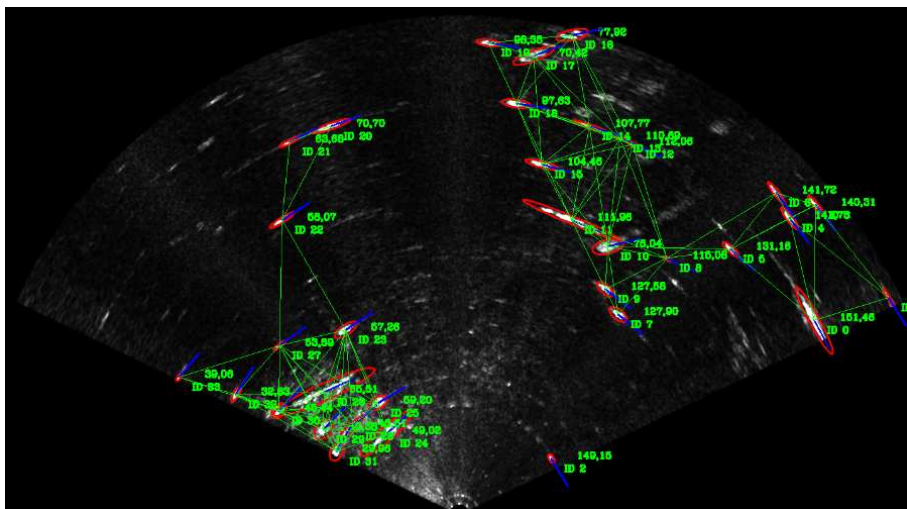
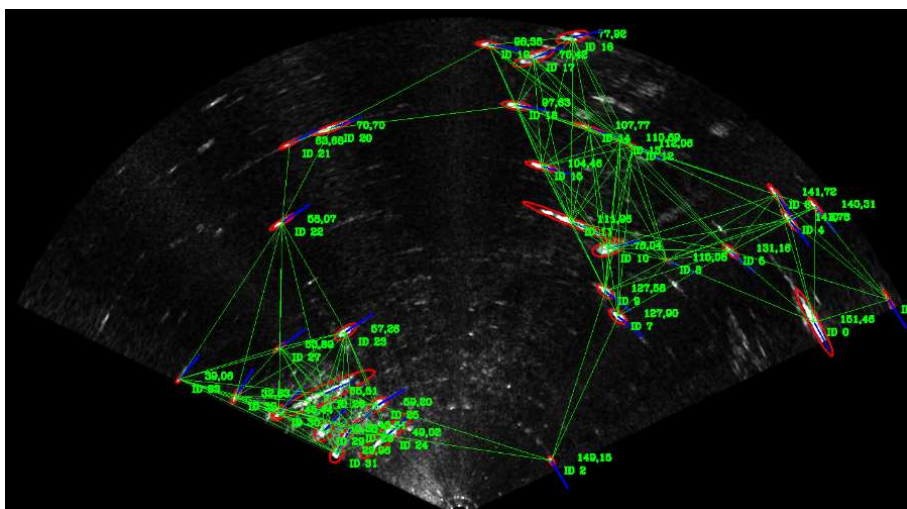
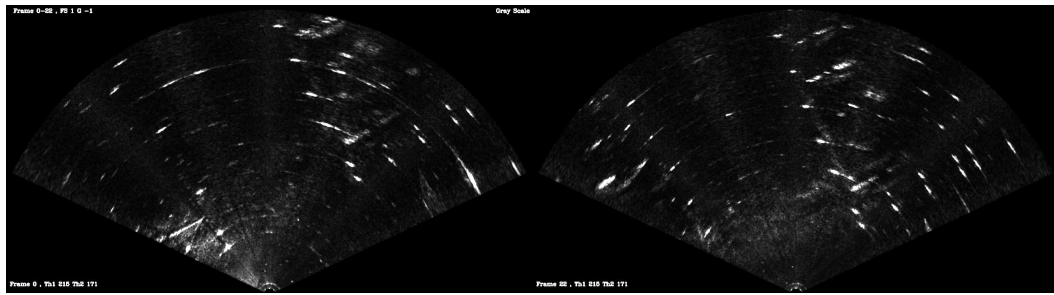
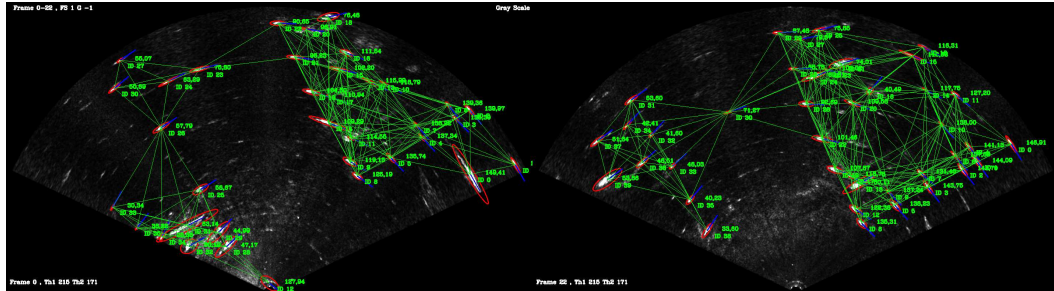
(a) Grafo criado quando $R_{lig} = 110$.(b) Grafo criado quando $R_{lig} = 200$.(c) Grafo criado quando $R_{lig} = 300$.

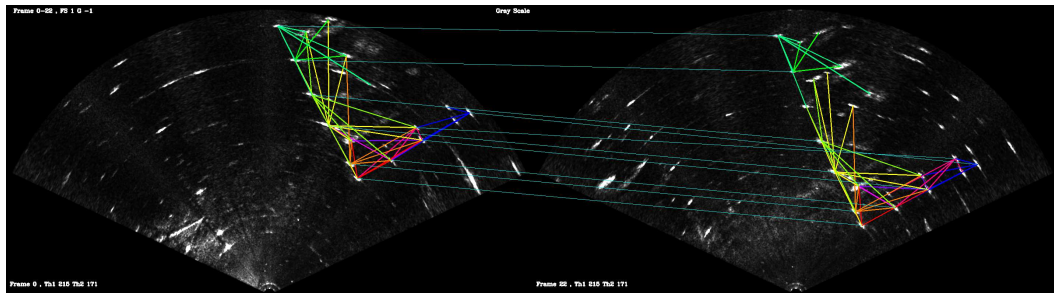
Figura 50: Comportamento do algoritmo de descrição de imagens acústica quando alterado o parâmetro de distância R_{lig} aplicado a imagem segmentada da Figura 49.



(a) Imagens originais comparadas.



(b) Imagens descritas.



(c) Resultado da etapa de comparação de descritores, em cores as arestas semelhantes de cada vértice, a linha fina em ciano indica os vértices semelhantes.

Figura 51: Resultados obtidos pelo algoritmo de comparação de grafo.

4.5.1 Teste de Detecção de Similaridade

Uma avaliação da detecção de imagens semelhantes foi realizada utilizando o dataset ARACATI 2014 (SILVEIRA et al., 2015). Neste experimento é possível avaliar quais as regiões foram corretamente identificadas como regiões semelhantes pelo método proposto.

4.5.1.1 Informações do Ground Truth

O *grond truth*, utilizado como base para a comparação dos resultados, consiste em 35 pares de imagens, cuja as regiões semelhantes foram definidas manualmente de forma empírica com o auxílio de um ferramenta desenvolvida especificamente para este propósito. Além de possibilitar a marcação manual das regiões semelhantes, a ferramenta também auxilia o usuário sugerindo possíveis regiões semelhantes, encontradas através do ajuste de transformações afim para as regiões já marcadas pelo usuário. A Figura 52 mostra a ferramenta, as linhas em vermelho indicam as regiões semelhantes marcadas manualmente

pelo usuário. As linhas em azul foram sugeridas pela ferramenta através do ajuste de uma transformação afim para as regiões similares marcadas pelo usuário, em vermelho.

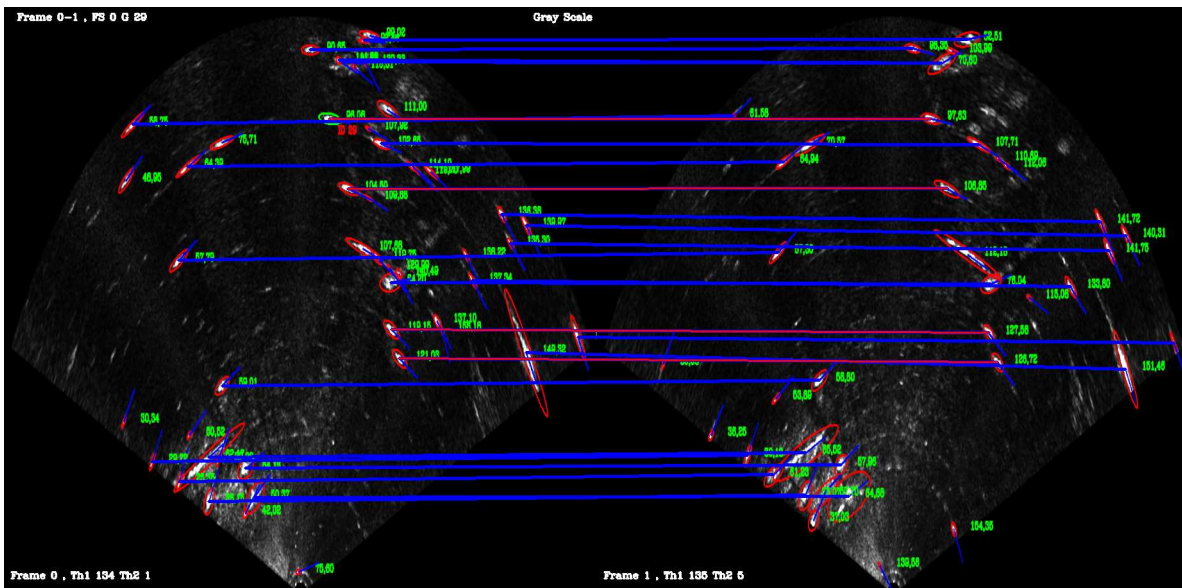


Figura 52: Demonstração da ferramenta para marcação manual das regiões semelhantes entre pares de imagens. As linhas em vermelho indicam regiões semelhantes definidas pelo usuário, as linhas em azul indicam regiões semelhantes sugeridas pela ferramenta através do ajuste de uma transformação afim utilizando as regiões semelhantes definidas pelo usuário.

As imagens são segmentadas e descritas automaticamente pela ferramenta utilizando um dos métodos de segmentação propostos na Seção 3.2 e o método de descrição proposto na Seção 3.3. É possível escolher entre os algoritmo de segmentação e definir os seus parâmetros de forma individual para cada imagem, de forma que o usuário possa escolher a melhor descrição para cada imagem.

Como resultado, a ferramenta produz um arquivo de texto para cada imagem, com a descrição dos objetos extraídos e um arquivo de texto para cada par de imagens, cuja a marcação das regiões semelhantes foi definida. As regiões semelhantes entre cada par de imagens é representada por uma matriz de correspondência X entre os objetos descritos nas duas imagens.

4.5.1.2 Comparação

A comparação das regiões semelhantes do *Ground Truth* com os resultados obtidos pelo método proposto são realizadas de forma automática. A primeira etapa da comparação é encontrar o mapeamento entre as descrições da mesma imagem geradas pelo *Ground Truth* e pelo método proposto. Este mapeamento é encontrado através das intersecção entre as Gaussianas das duas descrições.

Comparação com Ground Truth

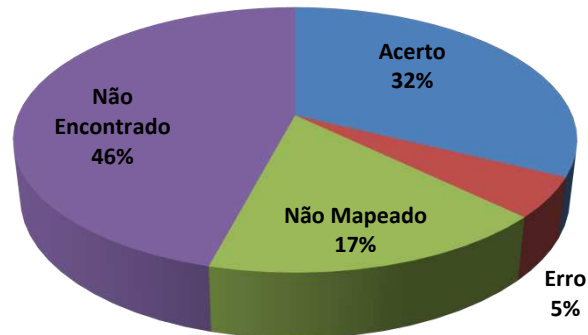
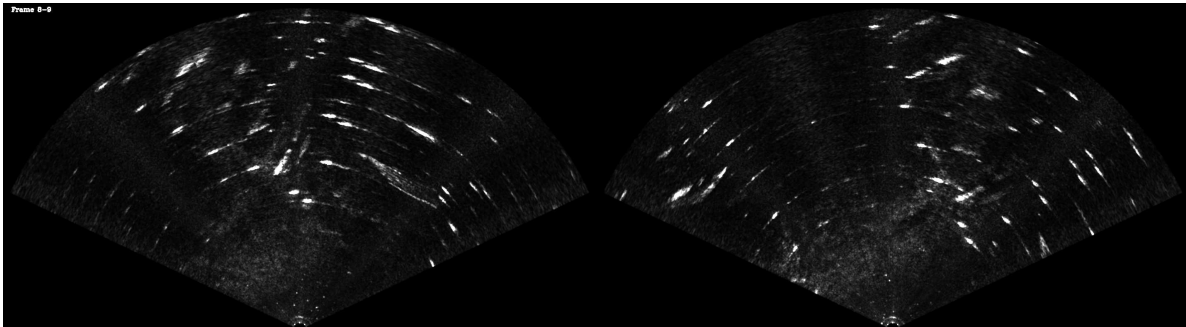


Figura 53: Comparação dos resultados obtidos pelo método proposto com o *Ground Truth*

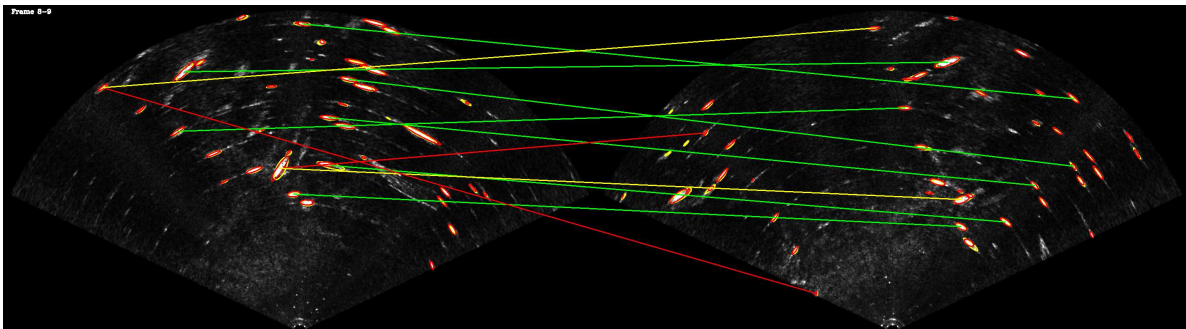
A segunda etapa é a comparação das matrizes de correspondência X gerada pela ferramenta e pelo método proposto, utilizando o mapeamento entre as descrições calculado na primeira etapa. Como resultado, são geradas informações estatísticas de acerto do algoritmo e imagens demonstrando o resultado da comparação com o *Ground Truth*.

Os resultados obtidos pelo método proposto utilizou a segmentação baseada em distância descrita na seção 3.2.1.3, a busca por picos em *beams* acústicos descrita na seção 3.2.2.2. Para o cálculo da similaridade entre vértices foi utilizado o alinhamento com base na topologia descrito na seção 3.4.1.2. Os parâmetros utilizados estão indicados na Tabela 6. A Figura 54 mostra o desempenho do algoritmo com relação ao *Ground Truth* para o par de imagens da Figura 54(a). Os erros e acertos do método proposto podem ser visualizados na Figura 54(b). A Figura 54(c) mostra as regiões similares que, ou não foram marcadas no *Ground Truth* e foram encontradas pelo método proposto, indicadas pelas linhas em rosa, ou foram marcadas no *Ground Truth* e não foram encontradas pelo método proposto, indicadas pelas linhas em azul.

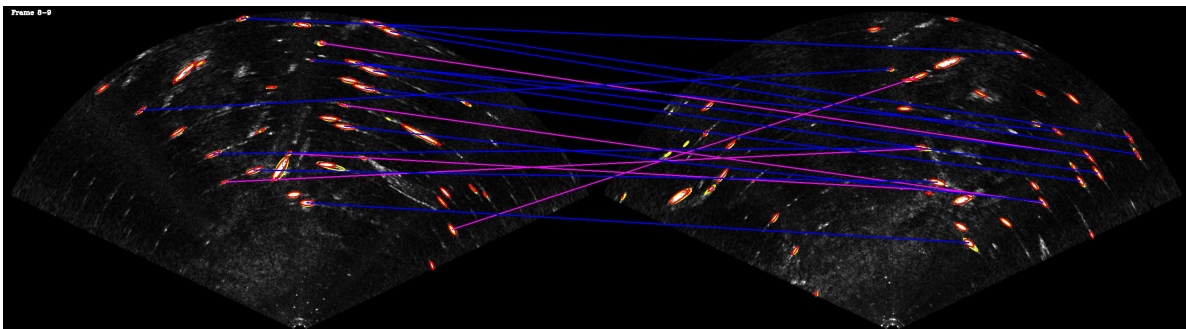
O gráfico da Figura 53 mostra o desempenho do algoritmo com relação aos 35 pares de imagem do *Ground Truth*, onde foram marcadas cerca de 30 regiões semelhantes para cada par de imagens, totalizando cerca de 1050 regiões semelhantes marcadas. Destas, o algoritmo proposto encontrou 37% , sendo que 32% estavam corretas e 5% estavam incorretas. Os 63% restantes são regiões que não possuem intersecção entre o *Ground Truth* e o método proposto, ou seja, são regiões semelhantes que foram marcadas no *Ground Truth* mas não foram encontradas pelo método proposto (46%) ou regiões encontradas pelo método proposto mas que não foram marcadas no *Ground Truth*(17%).



(a) Par de imagens original analisado



(b) Demonstração da comparação com o *Ground Truth*, as linhas em verde indicam as regiões semelhantes detectadas corretamente, as linhas em vermelho indicam as regiões semelhantes erradas pelo algoritmo e as linhas em amarelo indicam as correções dos erros a partir do *Ground Truth*.



(c) Regiões semelhantes que não foram avaliadas porque não foram definidas no *Ground Truth*, representadas pelas linhas em rosa, ou porque foram definidas no *Ground Truth* mas não foram detectadas pelo método proposto, representado pelas linhas em azul.

Figura 54: Imagens geradas após a comparação dos resultados obtidos com o *Ground Truth*. As elipses representam as regiões identificadas, em vermelho pela ferramenta de *Ground Truth* e em amarelo automaticamente pelo método proposto.

Tabela 6: Parâmetros adotados para obter os resultados do experimento.

Extração de segmento relativo a distância	
D_{seg}	5 pixels
Descrição	
R_{lig}	600 pixels
Busca de segmentos baseado em análise de picos	
H_{busca}	165
π_{fim}	0.6
$\pi_{recursivo}$	0.98
Semelhança entre grafos baseado em heurísticas	
$\rho_{similar}$	4.0
$\epsilon_{ambiguo}$	5.0

Como as regiões semelhantes do *Ground Truth* foram marcadas de forma manual, é difícil considerar que todas as regiões semelhantes existentes foram marcadas, mas considerando esta hipótese, os 17% referentes as regiões encontradas pelo método proposto mas que não foram mapeados no *Ground Truth* podem ser contabilizadas como erro, totalizando 22% de erro contra 32% de acerto.

Com as devidas considerações, na pior das hipóteses, o método proposto identifica corretamente 32% das regiões semelhantes, erra a identificação 22% das regiões semelhantes e deixa de identificar 46% das regiões semelhantes. Considerando a melhor das hipóteses, onde tiveram regiões semelhantes que não foram marcadas no *Ground Truth*, o método detecta corretamente 32% das regiões semelhantes, erra a detecção de 5% das regiões semelhantes e deixa de detectar 63% das regiões semelhantes.

Framework de Feng Zhou e Fernando De la Torre

Foram realizados testes utilizando o *framework* disponibilizado por (ZHOU; TORRE, 2015). Uma vez que está sendo analisado apenas a comparação dos grafos e a detecção das imagens semelhantes, foram utilizados as descrições dos objetos geradas pela ferramenta de *Ground Truth*. A partir destas descrições os grafos, as matrizes de similaridade entre vértices K_p e entre arestas K_q foram calculadas utilizando os parâmetros da tabela 7.

Entre os algoritmos disponíveis no *framework* citados na seção 2.3.3.3, o que obteve o melhor resultado foi o *Factorized Graph Matching*(FGM) proposto pelo criador do *framework*. Os resultados podem ser visualizados no gráfico da Figura 55. Acredita-se que resultados melhores podem ser alcançados explorando os parâmetros do *framework*. Para este experimento foram utilizados os parâmetros padrões dos algoritmos.

Um dos maiores problemas encontrados ao utilizar as abordagens de semelhança entre grafos, disponíveis no *framework*, nas quais modelam o problema de semelhança de grafos como um problema de alocação quadrática, onde deseja-se maximizar a semelhança

Tabela 7: Parâmetros adotados para obter os resultados do experimento utilizando o *framework* de (ZHOU; TORRE, 2015).

Segmentação	
Utilizado a descrição dos objetos detectados na ferramenta de <i>Ground Truth</i>	
Semelhança entre grafos baseado em heurísticas, cálculo de K_p	
$\rho_{similar}$	4.0
$\epsilon_{ambiguo}$	5.0
Semelhança entre arestas, cálculo de K_q	
Utilizado a diferença entre comprimento das arestas comparadas	

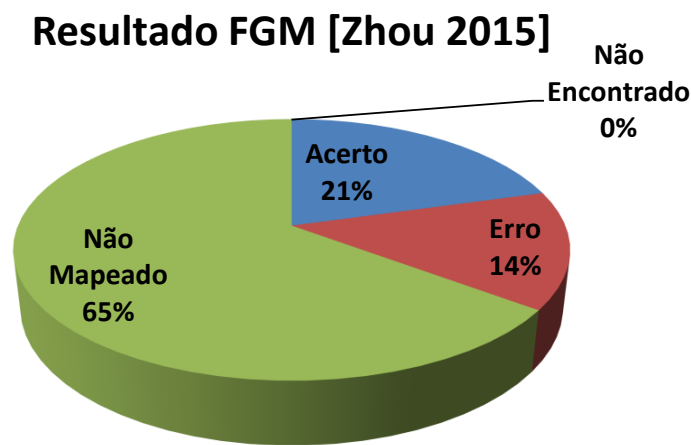


Figura 55: Comparação dos resultados obtidos pelo método proposto com o *Ground Truth*

entre os dois grafos comparados, foi o elevado tempo de processamento, podendo demorar várias horas para encontrar a semelhança entre um único par de grafos. Como descrito na Seção 2.3.3.2, este problema deve-se a complexidade para solucionar um QAP ser NP-Difícil. Os melhores algoritmos para solucionar este problema são aproximativos e dependem de muitas interações até a convergência para uma solução de qualidade.

4.6 Considerações finais

Foram apresentados um conjunto de métodos capazes de preprocessar, tratar e disponibilizar informações relativas a detecção de *loop* utilizando imagens acústicas de um sonar de imageamento frontal.

Quanto a segmentação de imagens, o método de extração de segmentos relativo a distância, descrito na seção 3.2.1.3, e a segmentação através da busca de picos em *beams* acústicos, descrita na seção 3.2.2.2, obtiveram os melhores resultados e possuem uma

maior flexibilidade para tratar os problemas de segmentação de imagens acústicas com relação aos demais métodos propostos. Portanto são os métodos mais indicados para a segmentação de imagens acústicas.

Quanto ao desempenho para detecção de regiões semelhantes, o método proposto foi capaz de identificar 32% das regiões semelhantes, tendo uma taxa de acerto maior que a taxa de erro 5%. Como para um algoritmo de [SLAM](#) a frequência de detecção de *loop* não precisa ser alta, porém é importante que sejam corretas. Uma taxa pequena de detecção de *loop* sem detecções erradas é aceitável. Acredita-se que os 5% de erro do método proposto pode ser reduzido através de uma etapa de verificação da consistência das semelhanças encontradas através de algoritmos de remoção de *outliers* como o [RANSAC](#) ou mesmo através da eliminação de pares de vértices semelhantes inconsistentes utilizando transformações afins ajustadas aos pares de vértices semelhantes encontrados entre os grafos comparados.

Quanto ao *framework* de de Feng Zhou e Fernando De la Torre ([ZHOU; TORRE, 2015](#)) que reúne os principais algoritmos de semelhança entre grafos (do inglês *Graph Matching*) modelados como [QAP](#), o custo computacional é um fator que impede a sua utilização em um sistema de localização e mapeamento em tempo real limitando sua utilização apenas em sistemas *offline*.

5 Conclusão

Este trabalho apresentou um método de descrição e detecção de regiões subaquáticas semelhantes para sistemas de localização de veículos subaquáticos autônomos [AUV](#) em ambientes parcialmente estruturados utilizando imagens acústicas de um sonar de imageamento frontal ([FLS](#)). As regiões são descritas por um grafo que considera a forma e a topologia dos objetos presentes na cena. O grafo é utilizado como marcação visual do ambiente e as regiões semelhantes são detectadas através da comparação dos grafos.

O trabalho apresentou as principais características de um sonar de imageamento frontal ([FLS](#)), as suas principais vantagens, relacionadas com a sua capacidade de operação em ambientes com pouca visibilidade, e as principais dificuldades de operação devido a problemas como a resolução não homogênea, insonificação não uniforme, presença de ruído devido a baixa razão sinal/ruído e os efeitos provocados pelas distorções acústicas, pelas sombras acústicas e pelas reverberações acústicas.

Foram propostos métodos para a segmentação de imagens acústicas baseado na busca de pixels conexos e o algoritmo clássico de busca em largura. Onde os segmentos são descritos pelo modelo probabilístico Gaussiano e as suas relações topológicas são estabelecidas por um grafo. Foram propostas métricas de comparação de vértices utilizando o modelo de descrição proposto e algoritmos de comparação de grafos baseado no corte de vértices utilizando heurísticas. Também foi apresentado o *framework* desenvolvido por ([ZHOU; TORRE, 2015](#)) no qual reúne algumas das principais abordagens para o cálculo da similaridade entre grafos, modelando o problema como um problema de alocação quadrático [QAP](#).

O método proposto foi validado utilizando dados reais do dataset ARACATI 2014 ([SILVEIRA et al., 2015](#)), onde foram apresentados resultados de cada etapa do método e também foi avaliado a sua capacidade de detectar regiões semelhantes.

Como trabalhos futuros pretende-se adicionar uma etapa de verificação da consistência da semelhança entre grafos calculada, eliminando as semelhanças inconsistentes utilizando algoritmos como o [RANSAC](#).

Pretende-se manter uma descrição global do ambiente de navegação de forma que ao invés de comparar as descrições entre duas imagens, as comparações sejam realizadas entre a descrição de uma imagem e a descrição global do ambiente. Nesta abordagem o grafo seria construído e relaxado dinamicamente conforme as similaridades com as imagens adquiridas são encontradas. Também seria levado em consideração a frequência de detecção dos objetos, de forma que objetos não estáticos sejam eliminados da descrição global do ambiente com o tempo.

Pretende-se estudar a aplicabilidade de métodos de classificação e aprendizagem de máquina, como o *Bag of Words*, para realizar a identificação de regiões semelhantes através dos grafos.

Pretende-se também avaliar o método proposto quando utilizado como um detector de *loop* em algoritmo de localização e mapeamento simultâneo para AUVs como por exemplo o Dolphin SLAM (SILVEIRA et al., 2015).

Referências

- ANTONELLI, G. *Underwater robots*. 3rd. ed. London, UK: Springer, 2014. v. 96. (STAR — Springer Tracts in Advanced Robotics, v. 96). Citado na página 27.
- AYKIN, M.; NEGAHDARIPOUR, S. On feature extraction and region matching for forward scan sonar imaging. In: IEEE. *Oceans, 2012*. [S.l.], 2012. p. 1–9. Citado 3 vezes nas páginas 13, 45 e 57.
- AYKIN, M. D.; NEGAHDARIPOUR, S. On feature matching and image registration for two-dimensional forward-scan sonar imaging. In: *J. Field Robotics*. [S.l.: s.n.], 2013. p. 602–623. ISSN 2153-0858. Citado 2 vezes nas páginas 31 e 42.
- BALLARD, D. H. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, Elsevier, v. 13, n. 2, p. 111–122, 1981. Citado na página 57.
- BAY, H.; TUYTELAARS, T.; GOOL, L. V. SURF: Speeded up robust features. In: *In ECCV*. [S.l.: s.n.], 2006. p. 404–417. Citado na página 30.
- BIBER, P.; STRASSER, W. The normal distributions transform: A new approach to laser scan matching. In: IEEE. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. [S.l.], 2003. v. 3, p. 2743–2748. Citado 2 vezes nas páginas 31 e 57.
- BOTELHO, S. S. C. et al. Lambdari um robô subaquático autônomo. In: *Simpósio Brasileiro de Automação Inteligentes, Anais do VI SBAI*. [S.l.: s.n.], 2003. p. 232–237. Citado na página 33.
- BOTELHO, S. S. C.; DREWS, P.; LEIVAS, G. Nlmap - visual-based self localization and mapping for autonomous underwater vehicles. In: *IEEE Oceans*. [S.l.: s.n.], 2008. p. 1–6. Citado na página 33.
- BOTELHO, S. S. C. et al. Multi-aUVs for visual mapping task. In: *Distributede Autonomous Robotic Systems 6*. [S.l.: s.n.], 2004. p. 54–59. Citado na página 33.
- BOTELHO, S. S. C.; NEVES, R.; TADDEI, L. Localization of a fleet of aUVs using visual maps. In: *IEEE Oceans*. [S.l.: s.n.], 2005. p. 1320–1325. Citado na página 33.
- BOTELHO, S. S. d. C. et al. Visual odometry and mapping for underwater autonomous vehicles. In: IEEE. *Robotics Symposium (LARS), 2009 6th Latin American*. [S.l.], 2009. p. 1–6. Citado na página 33.
- CASTRO, E. D.; MORANDI, C. Registration of translated and rotated images using finite fourier transforms. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, IEEE, n. 5, p. 700–703, 1987. Citado na página 58.
- CHO, M.; LEE, J.; LEE, K. M. Reweighted random walks for graph matching. In: *Computer Vision–ECCV 2010*. [S.l.]: Springer, 2010. p. 492–505. Citado na página 56.
- CHRIST, R. D.; WERNLI, R. L. *The ROV Manual: A User Guide for Remotely Operated Vehicles*. 2nd. ed. Oxford, UK: Butterworth-Heinemann, 2014. Citado na página 27.

- CODEVILLA, F. et al. Achieving turbidity robustness on underwater images local feature detection. In: *British Machine Vision Conference (BMVC)*. [S.l.: s.n.], 2015. Citado na página 30.
- COUR, T.; SRINIVASAN, P.; SHI, J. Balanced graph matching. *Advances in Neural Information Processing Systems*, MIT; 1998, v. 19, p. 313, 2007. Citado na página 56.
- DREWS, P.; BOTELHO, S. S. C.; GOMES, S. Slam in underwater environment using sift and topologic maps accepted. In: *Latin American Robotic System - LARS*. [S.l.: s.n.], 2008. p. 1–6. Citado na página 33.
- DREWS, P. et al. Using sift algorithm for visual-based localization in multi-auvs fleet. In: *Simpósio Brasileiro de Automação Inteligente SBAI*. [S.l.: s.n.], 2006. p. 1–6. Citado na página 33.
- DREWS, P. et al. Transmission estimation in underwater single images. In: *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. [S.l.: s.n.], 2013. p. 825–830. Citado na página 30.
- DURRANT-WHYTE, H.; BAILEY, T. Simultaneous localization and mapping: part i. *Robotics Automation Magazine, IEEE*, v. 13, n. 2, p. 99–110, June 2006. ISSN 1070-9932. Citado na página 29.
- EINHORN, E.; SCHRÖTER, C.; GROSS, H.-M. Finding the adequate resolution for grid mapping-cell sizes locally adapting on-the-fly. In: *IEEE. Robotics and Automation (ICRA), 2011 IEEE International Conference on*. [S.l.], 2011. p. 1843–1848. Citado na página 31.
- EULER, L. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, v. 8, p. 128–140, 1741. Citado na página 48.
- GARCIA, R.; GRACIAS, N. Detection of interest points in turbid underwater images. In: *OCEANS, 2011 IEEE - Spain*. [S.l.: s.n.], 2011. p. 1–9. Citado na página 30.
- GOLD, S.; RANGARAJAN, A. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 18, n. 4, p. 377–388, Apr 1996. ISSN 0162-8828. Citado na página 55.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing (2nd Edition)*. [S.l.]: Pearson, 2002. Citado 4 vezes nas páginas 13, 37, 46 e 47.
- GRIFFITHS, G. *Technology and Applications of Autonomous Underwater Vehicles (Ocean Science and Technology, Volume 2)*. London, UK: Taylor & Francis, 2002. ISSN 1561-5928. Citado na página 27.
- GUTH, F. et al. Underwater SLAM: Challenges, state of the art, algorithms and a new biologically-inspired approach. In: *Biomedical Robotics and Biomechatronics (2014 5th IEEE RAS EMBS International Conference on)*. [S.l.: s.n.], 2014. p. 981–986. ISSN 2155-1774. Citado na página 33.
- HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: *CITeseer. Alvey vision conference*. [S.l.], 1988. v. 15, p. 50. Citado na página 56.
- HORNUNG, A. et al. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, Springer, v. 34, n. 3, p. 189–206, 2013. Citado na página 31.

HOW much of the ocean have we explored? 2014. <<http://oceanservice.noaa.gov/facts/exploration.html>>. Acessado em 20-11-2015. Citado na página 27.

HUAN, E. *Robots Help Map England's Deepwater Canyons*. 2015. <<http://www.marinetechologynews.com/news/robots-englands-deepwater-canyons-521031>>. Acessado em 20-11-2015. Citado na página 27.

HURTÓS, N. et al. Fourier-based registrations for two-dimensional forward-looking sonar image mosaicing. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. [S.l.: s.n.], 2012. p. 5298–5305. ISSN 2153-0858. Citado 2 vezes nas páginas 31 e 58.

HURTÓS, N.; CUFÍ, X.; SALVI, J. Calibration of optical camera coupled to acoustic multibeam for underwater 3d scene reconstruction. In: *OCEANS 2010 IEEE - Sydney*. [S.l.: s.n.], 2010. p. 1–7. Citado na página 28.

HURTÓS, N. et al. Evaluation of registration methods on two-dimensional forward-looking sonar imagery. In: *OCEANS - Bergen, 2013 MTS/IEEE*. [S.l.: s.n.], 2013. p. 1–8. Citado na página 30.

HURTÓS, N. V. *Forward-Looking Sonar Mosaicing for Underwater Environments*. Tese (Doutorado) — Universitat de Girona, 2014. Citado 5 vezes nas páginas 13, 17, 39, 42 e 43.

JOHANSSON, H. et al. Imaging sonar-aided navigation for autonomous underwater harbor surveillance. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. [S.l.: s.n.], 2010. p. 4396–4403. ISSN 2153-0858. Citado 3 vezes nas páginas 31, 42 e 57.

KIM, K.; NERETTI, N.; INTRATOR, N. Mosaicing of acoustic camera images. *Radar, Sonar and Navigation, IEE Proceedings* -, v. 152, n. 4, p. 263–270, Aug 2005. ISSN 1350-2395. Citado 3 vezes nas páginas 44, 56 e 59.

KLÄSS, J.; STÜCKLER, J.; BEHNKE, S. Efficient mobile robot navigation using 3d surfel grid maps. In: *VDE. Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*. [S.l.], 2012. p. 1–4. Citado na página 31.

KONOLIGE, K.; AGRAWAL, M. FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, IEEE, v. 24, n. 5, p. 1066–1077, 2008. Citado na página 30.

KUHN, H. W. The hungarian method for the assignment problem. *Naval research logistics quarterly*, Wiley Online Library, v. 2, n. 1-2, p. 83–97, 1955. Citado na página 54.

LAWLER, E. L. The quadratic assignment problem. *Management science*, INFORMS, v. 9, n. 4, p. 586–599, 1963. Citado na página 55.

LEIVAS, G.; DREWS, P.; BOTELHO, S. S. C. A new approach for visual underwater mapping using topological shell maps. In: *IEEE Oceans*. [S.l.: s.n.], 2009. p. 1–8. Citado na página 33.

LEORDEANU, M.; HEBERT, M. A spectral technique for correspondence problems using pairwise constraints. In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. [S.l.: s.n.], 2005. v. 2, p. 1482–1489 Vol. 2. ISSN 1550-5499. Citado na página 55.

LEORDEANU, M.; HEBERT, M.; SUKTHANKAR, R. An integer projected fixed point method for graph matching and map inference. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2009. p. 1114–1122. Citado na página 56.

LOIOLA, E. M.; ABREU, N. M. M. d.; NETTO, P. O. B. Uma revisão comentada das abordagens do problema quadrático de alocação. *Pesquisa Operacional*, SciELO Brasil, v. 24, n. 1, p. 73–109, 2004. Citado na página 55.

LOIOLA, E. M. et al. A survey for the quadratic assignment problem. *European journal of operational research*, Elsevier, v. 176, n. 2, p. 657–690, 2007. Citado na página 55.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 60, n. 2, p. 91–110, 2004. ISSN 0920-5691. Citado 2 vezes nas páginas 30 e 57.

MADDERN, W.; MILFORD, M.; WYETH, G. CAT-SLAM: probabilistic localisation and mapping using a continuous appearance-based trajectory. *IJRR*, SAGE Publications, v. 31, n. 4, p. 429–451, 2012. Citado na página 29.

MILFORD, M.; WYETH, G. Mapping a suburb with a single camera using a biologically inspired SLAM system. *IEEE Transactions on Robotics*, v. 24, n. 5, p. 1038–1053, 2008. Citado na página 29.

NEGAHDARIPOUR, S.; AYKIN, M.; SINNARAJAH, S. Dynamic scene analysis and mosaicing of benthic habitats by fs sonar imaging-issues and complexities. In: IEEE. *OCEANS 2011*. [S.l.], 2011. p. 1–7. Citado na página 57.

NEGAHDARIPOUR, S.; FIROOZFAM, P.; SABZMEYDANI, P. On processing and registration of forward-scan acoustic video imagery. In: *Computer and Robot Vision, 2005. Proceedings. The 2nd Canadian Conference on*. [S.l.: s.n.], 2005. p. 452–459. Citado 2 vezes nas páginas 44 e 56.

NEW ROV support vessel Fugro Aquarius delivered in Brazil. 2015. <<http://www.worldoil.com/news/2015/11/06/new-rov-support-vessel-fugro-aquarius-delivered-in-brazil>>. World Oil - Acessado em 21-11-2015. Citado na página 27.

OIL slick spreads from sunken rig (Intrevista de vídeo). 2010. <<http://edition.cnn.com/2010/US/04/22/oil.rig.explosion/index.html>>. Acessado em 20-11-2015. Citado na página 27.

REDDY, B. S.; CHATTERJI, B. N. An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE transactions on image processing*, v. 5, n. 8, p. 1266–1271, 1996. Citado na página 58.

RIBAS, D. et al. SLAM using an imaging sonar for partially structured underwater environments. In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. [S.l.: s.n.], 2006. p. 5040–5045. Citado 2 vezes nas páginas 31 e 56.

- SCHÖNING, U. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences*, v. 37, n. 3, p. 312 – 323, 1988. ISSN 0022-0000. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0022000088900104>>. Citado na página 69.
- SEVEN Bridges of Königsberg. <https://en.wikipedia.org/w/index.php?title=Seven_Bridges_of_K%C3%B6nigsberg&oldid=709608933>. Acessado: 21-03-2016. Citado 2 vezes nas páginas 13 e 49.
- SIBLEY, G. et al. Vast-scale outdoor navigation using adaptive relative bundle adjustment. *The International Journal of Robotics Research*, SAGE Publications, 2010. Citado na página 29.
- SILVEIRA, L. et al. An open-source bio-inspired solution to underwater SLAM. In: *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles NGCUV*. [S.l.: s.n.], 2015. Citado 11 vezes nas páginas 13, 32, 33, 34, 39, 41, 84, 85, 101, 109 e 110.
- SIQUEIRA, E. B. et al. Algorithm to identification of parameters and automatic re-project of speed controller of bldc motor. *IFAC-PapersOnLine*, Elsevier, v. 48, n. 19, p. 256–261, 2015. Nenhuma citação no texto.
- SUSTAITA, M. *Fugro in Moho Nord subsea gig*. 2015. <<http://www.oedigital.com/subsea/equipment/item/8624-fugro-in-moho-nord-subsea-gig>>. Acessado em 21-11-2015. Citado na página 27.
- THRUN, S.; BURGARD, W.; FOX, D. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In: IEEE. *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*. [S.l.], 2000. v. 1, p. 321–328. Citado na página 31.
- VARGAS, A.; MADSEN, C. A.; BOTELHO, S. S. C. Navision - sistema de visão subaquático para navegação e montagem de mosaicos em auvs. In: *Seminário e Workshop em Eng. Oceânica*. [S.l.: s.n.], 2004. p. 12–17. Citado na página 33.
- WARNER, K. *Why are thousands of robots roaming oceans?* 2015. <<http://www.csmonitor.com/Science/2015/0824/Why-are-thousands-of-robots-roaming-oceans>>. Acessado em 20-11-2015. Citado na página 27.
- WILLIAMS, S.; MAHON, I. Simultaneous localisation and mapping on the great barrier reef. In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*. [S.l.: s.n.], 2004. v. 2, p. 1771–1776 Vol.2. ISSN 1050-4729. Citado na página 28.
- ZASS, R.; SHASHUA, A. Probabilistic graph and hypergraph matching. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. [S.l.: s.n.], 2008. p. 1–8. ISSN 1063-6919. Citado na página 56.
- ZHOU, F.; TORRE, F. Deformable graph matching. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2013. p. 2922–2929. Citado 3 vezes nas páginas 53, 54 e 56.
- ZHOU, F.; TORRE, F. D. la. Factorized graph matching. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*. [S.l.: s.n.], 2015. Citado 11 vezes nas páginas 17, 53, 55, 56, 81, 82, 84, 105, 106, 107 e 109.