

UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

BRUNO HENRIQUE JUSTO DA SILVA

**RECONSTRUÇÃO EM 3D DE SUPERFÍCIES EM AMBIENTES
SUBÁQUATICOS ATRAVÉS DE DADOS OBTIDOS COM SONAR
MSIS**

TRABALHO DE CONCLUSÃO DE CURSO

RIO GRANDE
2018

BRUNO HENRIQUE JUSTO DA SILVA

**RECONSTRUÇÃO EM 3D DE SUPERFÍCIES EM AMBIENTES
SUBÁQUATICOS ATRAVÉS DE DADOS OBTIDOS COM SONAR
MSIS**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Federal do Rio Grande, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Dr. Paulo Lilles Jorge Drews Junior
Centro de Ciências Computacionais – C3

Coorientador: Dr. Jorge Arigony Neto
Instituto de Oceanografia – IO

**RIO GRANDE
2018**

Altere este texto inserindo a dedicatória do seu trabalho.

AGRADECIMENTOS

Edite e coloque aqui os agradecimentos às pessoas e/ou instituições que contribuíram para a realização do trabalho.

É obrigatório o agradecimento às instituições de fomento à pesquisa que financiaram total ou parcialmente o trabalho, inclusive no que diz respeito à concessão de bolsas.

Eu denomino meu campo de Gestão do Conhecimento, mas você não pode gerenciar conhecimento. Ninguém pode. O que pode fazer - o que a empresa pode fazer - é gerenciar o ambiente que otimize o conhecimento. (PRUSAK, Laurence, 1997).

RESUMO

JUSTO, Bruno. Reconstrução em 3D de Superfícies em Ambientes Subáquaticos Através de Dados Obtidos com Sonar MSIS. 2018. 29 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Federal do Rio Grande. Rio Grande, 2018.

O aquecimento global é o processo de aumento da temperatura média da superfície terrestre e dos oceanos causado por uma grande quantidade de gases originados de atividades humanas e emitidos na atmosfera terrestre. Um dos problemas ocasionado pelo aumento da temperatura é a diminuição da área de geleiras. Devido a importância econômica, ao seu papel como indicador de mudança no clima, além de ser o maior reservatório de água doce sobre a Terra, o monitoramento das geleiras se torna necessário e indispensável para o planeta. Esse trabalho propõe uma metodologia através do uso de algoritmos para estimar variações espaciais de superfícies submersas através da reconstrução em 3D de superfícies a partir de dados coletados com sensor sonar MSIS. O trabalho servirá como base para ser aplicado nas superfícies submersas de geleiras com o propósito de obter uma melhor mensuração da ablação dessas superfícies.

Palavras-chave: Robótica Subaquática. Reconstrução 3D. *Point Cloud*, Sonar MSIS.

ABSTRACT

JUSTO, Bruno. 3D Surfaces Reconstruct in Underwater Environments Using Obtained Data From MSIS Sonar. 2018. 29 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Federal do Rio Grande. Rio Grande, 2018.

Elemento obrigatório em tese, dissertação, monografia e TCC. É a versão do resumo em português para o idioma de divulgação internacional. Deve ser antecedido pela referência do estudo. Deve aparecer em folha distinta do resumo em língua portuguesa e seguido das palavras representativas do conteúdo do estudo, isto é, das palavras-chave. Sugere-se a elaboração do resumo (Abstract) e das palavras-chave (Keywords) em inglês; para resumos em outras línguas, que não o inglês, consultar o departamento / curso de origem.

Keywords: Word. Second Word. Another word.

LISTA DE FIGURAS

Figura 1 – (A) Mudança na temperatura média da superfície terrestre (1986 – 2005). (B) Mudança do nível médio do mar (1986 – 2005).	1
Figura 2 – Aquisição de dados do sensoriamento remoto.	2
Figura 3 – Representação de um <i>bin</i> e um <i>beam</i> em uma imagem acústica	4
Figura 4 – Diferentes modelos de sonares.	6
Figura 5 – Imagem acústica sobreposta em uma imagem de satélite.	7
Figura 6 – Representação do processo de escaneamento de um MSIS.	7
Figura 7 – Relação entre tópico e nodo.	9
Figura 8 – Interface gráfica do simulador Gazebo	10
Figura 9 – Imagens correspondentes geradas pelo <i>GPU Sonar Simulator</i>	11
Figura 10 – Descrição gráfica do método utilizado no simulador do sonar	12
Figura 11 – Diferença entre nuvem de pontos (esquerda) e uma malha de polígonos (direita).	13
Figura 12 – Exemplo de utilização da PCL em dados adquiridos por um sensor a laser.	13
Figura 13 – Aplicação de um filtro em uma imagem de entrada e seu resultado na imagem de saída.	14
Figura 14 – Aplicação do Filtro Gaussiano. Imagem original à esquerda, e imagem filtrada à direita.	14
Figura 15 – Aplicação do Filtro de Mediana. Imagem com ruído à esquerda, e imagem filtrada à direita.	15
Figura 16 – Etapas do processo de desenvolvimento da aplicação.	17
Figura 17 – ROV utilizado no trabalho.	18
Figura 18 – Nuvem de pontos gerada a partir de dados coletados por um sonar MSIS.	18
Figura 19 – Remoção de ruído em uma nuvem de pontos.	19
Figura 20 – Filtros simples de remoção de <i>outliers</i> .	20
Figura 21 – Exemplo da aplicação do Filtro de Grade de <i>Voxels</i> .	21
Figura 22 – Exemplo do uso do SORF.	21
Figura 23 – Exemplo do uso do Filtro de Suavização.	22
Figura 24 – Reconstrução de um objeto em 3D a partir de uma nuvem de pontos.	23
Figura 25 – Exemplos de modelos de triangulação.	23
Figura 26 – Fecho convexo representado no plano (esquerda) e no espaço (direita).	24
Figura 27 – (a) Triangulação que não satisfaz o critério de Delaunay; (b) Triangulação de Delaunay.	24
Figura 28 – Regra do circuncírculo que define se um triângulo é ou não de Delaunay.	25
Figura 29 – Imagem acústica do sonar gerada em um ambiente de simulação.	26

LISTA DE TABELAS

Tabela 1 – Relação entre alcance e frequência da onda acústica	5
Tabela 2 – Velocidade do som em diferentes meios.	5

LISTA DE ABREVIATURAS E SIGLAS

C3	Centro de Ciências Computacionais
CPU	Unidade de Central de Processamento, do inglês <i>Central Process Unit</i>
FLS	Sonar de Imageamento Frontal, do inglês <i>Forwarding Look Sonar</i> .
FURG	Universidade Federal do Rio Grande
GPU	Unidade de Processamento Gráfico, do inglês <i>Graphics Processing Unit</i>
LACRIO	Laboratório de Monitoramento da Criosfera – FURG
MSIS	do inglês <i>Mechanical Scanning Image Sonar</i>
MLS	do inglês <i>Moving Least Squares</i>
NAUTEC	Grupo de Pesquisa em Automação e Robótica Inteligente
OSG	do inglês <i>Open Scene Graph</i>
P2P	do inglês <i>Peer-to-Peer</i>
PCL	Biblioteca de Nuvem de Pontos, do inglês <i>Point Cloud Library</i>
ROCK	do inglês <i>Robot Construction Kit</i>
ROS	Sistema Operacional de Robôs, do inglês <i>Robot Operating System</i>
ROV	Robô Operado Remotamente, do inglês <i>Remotely Operated Vehicle</i>
SONAR	Navegação e Determinação da Distância pelo Som, do inglês <i>Sound Navigation and Ranging</i>
SORF	Filtro Estatístico de Remoção de <i>Outliers</i> , do inglês <i>Statistical Outlier Removal Filter</i>
UUV	Veículo Submarino Não Tripulado, do inglês <i>Unmanned Underwater Vehicle</i>

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 Motivação	1
1.2 Objetivos	2
1.3 Organização do Trabalho	3
2 – FUNDAMENTAÇÃO TEÓRICA	4
2.1 Sensor Sonar	4
2.1.1 Princípio do Sonar	4
2.1.2 Sonar <i>Singlebeam</i> e <i>Multibeam</i>	6
2.1.3 Sonar MSIS	6
2.2 <i>Robot Operating System</i>	7
2.2.1 Grafos de computação	8
2.3 Simulador Gazebo	9
2.4 <i>GPU Sonar Simulator</i>	10
2.5 Nuvem de Pontos	12
2.6 Malha	12
2.7 <i>Point Cloud Library</i>	13
2.8 Filtros Espaciais	14
2.8.1 Filtro Gaussiano	14
2.8.2 Filtro de Mediana	15
3 – METODOLOGIA	16
3.1 Coleta de Dados	17
3.2 Pré-processamento	18
3.3 Filtragem	18
3.3.1 Filtro por Ângulo	19
3.3.2 Filtro por Distância Mínima	19
3.3.3 Filtro por Intensidade	20
3.3.4 Filtro de Grade de <i>Voxels</i>	20
3.3.5 Filtro Estatístico de Remoção de <i>Outliers</i>	21
3.3.6 Filtro de Suavização	22
3.4 Reconstrução	22
3.4.1 Triangulação de Delaunay	24
3.5 Comparação	25
4 – ANÁLISE E DISCUSSÃO DOS RESULTADOS	26

5 – CONCLUSÃO	27
5.1 Trabalhos Futuros	27
5.2 Considerações Finais	27
Referências	28

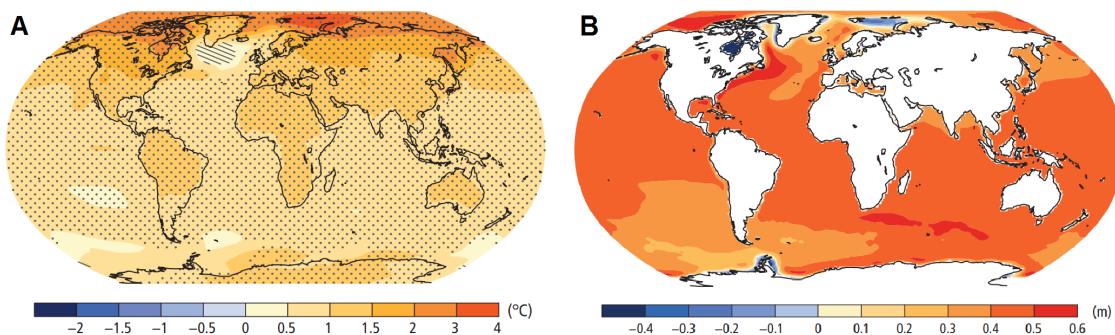
1 INTRODUÇÃO

1.1 Motivação

O aquecimento global é o processo de aumento da temperatura média da superfície terrestre e dos oceanos, causado por uma grande quantidade de gases, como o dióxido de carbono (CO_2) e o metano (CH_4), originados de atividades humanas e emitidos na atmosfera terrestre provocando grandes impactos ambientais negativos. Os principais impactos causados pelo aquecimento global são: aumento de eventos de extremo climático (ciclones, ondas de calor, chuvas torrenciais, etc), redução de biodiversidade, elevação do nível do mar e da temperatura na superfície terrestre, etc.

A temperatura da superfície global aumentou cerca de $0.2^{\circ}C$ em três décadas e $0.8^{\circ}C$ no último século ([HANSEN et al., 2006](#)). Para ilustrar esse retrato, a Figura 1-A mostra as mudanças na temperatura média da superfície terrestre enquanto a Figura 1-B exibe as mudanças no nível médio do mar. Essas mudanças ocorreram no intervalo de 1986 a 2005.

Figura 1 – (A) Mudança na temperatura média da superfície terrestre (1986 – 2005). (B) Mudança do nível médio do mar (1986 – 2005).



Fonte: Adaptado de ([PACHAURI et al., 2014](#)).

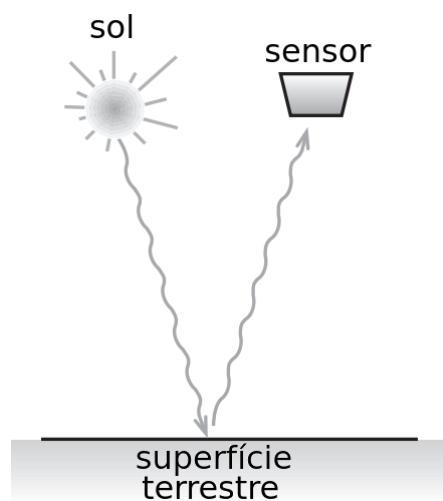
A redução da criosfera é um dos problemas ocasionado pelo aumento da temperatura e do nível do mar. A criosfera é constituída por regiões da superfície terrestre cobertas permanentemente por gelo e neve ou o solo é constituído por gelo, como por exemplo: lagos e rios congelados, mantos e calotas de gelo, neve sazonal e geleiras.

Em destaque, as geleiras do planeta estão perdendo massa, tornando uma das principais causas para o aumento no nível dos oceanos ([RIETBROEK et al., 2016](#)). Devido a importância econômica, ao seu papel como indicador de mudança no clima, além de ser o maior reservatório de água doce sobre a Terra, perdendo em volume total de água apenas para os oceanos ([COELHO; HAVENS, 2015](#)), o monitoramento das geleiras se torna indispensável para o planeta.

Entretanto, a maior parte das geleiras estão em locais remotos, dificultando o acesso

e a logística à essas áreas. Por consequência, os estudos com o sensoriamento remoto se tornou uma alternativa comumente utilizada para a estimativa de balanço de massa. Porém, os satélites não são eficazes para coletar informações de geleiras que possuem uma parcela submersa, visto que a tecnologia do sensoriamento remoto só captura informações da superfície terrestre, impedindo coletar informações a respeito do que está entre a superfície e o fundo de corpos d'água. A Figura 2 mostra o mecanismo de aquisição de dados do sensoriamento remoto. A obtenção de informações é realizada por meio do registro do retorno da interação da radiação eletromagnética provida do Sol com a superfície terrestre.

Figura 2 – Aquisição de dados do sensoriamento remoto.



Fonte: Adaptado de ([KERLE; JANSSEN; HUURNEMAN, 2004](#)).

Para coletar informações em ambientes submersos torna-se mais adequado utilizar sensores que operam com ondas sonoras, pois essas se propagam melhor em ambientes com maior densidade, como o fundo de oceanos e lagos.

1.2 Objetivos

O objetivo principal do trabalho é utilizar uma plataforma robótica do tipo ROV (Robô Operado Remotamente, do inglês *Remotely Operated Vehicle*) equipada com um sensor sonar MSIS (do inglês *Mechanical Scanning Image Sonar*) para estimar variações espaciais (balanço de massa) de superfícies submersas por meio da reconstrução em 3D de capturas temporalmente espaçadas de imagens acústicas. Para atingir o objetivo principal, será necessário alcançar tais objetivos:

- Estudar o mecanismo e eficiência da aquisição de imagens do sonar MSIS e os principais métodos para manipular imagens acústicas;
- Coletar dados em simulação e *in loco* utilizando robô subaquático acoplado com sonar e sensores auxiliares;

- Reconstruir as superfícies submersas utilizando algoritmos de reconstrução com o auxílio de bibliotecas de nuvem de pontos;
- Comparar modelos reconstruídos com o intuito de analisar a perda/ganho de volume entre eles.

Para a efetivação dos objetivos, o NAUTEC (Grupo de Pesquisa em Automação e Robótica Inteligente) junto ao C3 (Centro de Ciências Computacionais), disponibilizará a plataforma robótica, o sensor sonar e sensores auxiliares. Este trabalho é o inicio dos estudos para uma melhor precisão na estimativa da ablação¹ e o conhecimento de características da parte frontal submersa de geleiras, tais como configuração física e comportamento de degelo, que será uma contribuição para o LACRIO (Laboratório de Monitoramento da Criosfera). Além disso, o trabalho contribuirá para a consolidação da área da robótica subaquática na FURG (Universidade Federal do Rio Grande).

Embora o trabalho prático seja realizada em ambiente não glacial, a intenção do projeto é, posteriormente, ser executado em geleiras.

1.3 Organização do Trabalho

Primeiramente, após o capítulo de introdução, o Capítulo 2 apresenta assuntos fundamentais para a compreensão do mesmo, como a descrição do sensor e dos softwares utilizados. No terceiro capítulo, é abordado detalhadamente a metodologia utilizada e seus motivos. O Capítulo 4 apresenta os resultados que foram obtidos no experimento.

¹Ablação é a perda de massa/energia de um sistema, nessa circunstância, a fusão e sublimação do gelo.

2 FUNDAMENTAÇÃO TEÓRICA

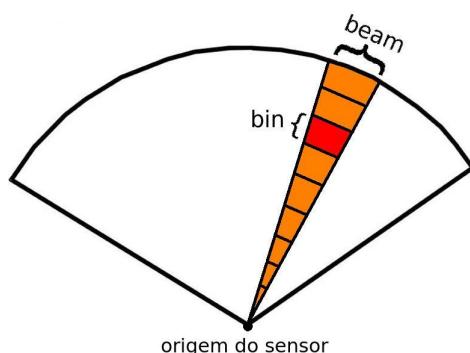
2.1 Sensor Sonar

2.1.1 Princípio do Sonar

O sonar é um sensor capaz de calcular a distância de objetos em um meio através de ondas sonoras. O sensor emite um pulso de onda que percorre o meio até colidir em um obstáculo/objeto ou ser absorvido completamente pelo meio. Após colidir, parte da onda é refletida e retorna para o sensor (a origem do pulso) com uma intensidade menor. Ao ser captada pelo sensor, a onda é decomposta em várias partes, onde cada parte é conhecida como *bin*. O conjunto de *bins* que compõe uma onda, é chamado de *beam*, como mostra a Figura 3.

De maneira geral, o sonar pode ser do tipo passivo ou ativo. O sonar passivo apenas escuta o meio com o intuito de filtrar os sinais sonoros de diferentes objetos do ruído do ambiente. Ao passo que, o sonar ativo tem a capacidade de emitir sinais acústicos e captar o retorno do pulso. O alcance de um pulso de um sonar ativo e a qualidade do sinal¹ retornado está relacionado com a frequência de operação do sensor (Tabela 1).

Figura 3 – Representação de um *bin* e um *beam* em uma imagem acústica



Sinais de alta frequência possuem um curto alcance, porém produz um sinal de retorno que carrega mais informações e com uma alta qualidade. Ao contrário, os sinais de baixa frequência são menos atenuados pelo meio, resultando em uma melhor propagação, mas em compensação, a quantidade de informação no sinal de retorno é menor.

¹Quanto maior a qualidade do sinal, maior será o nível de detalhamento e discriminação do alvo e menor será a quantidade de ruído.

Tabela 1 – Relação entre alcance e frequência da onda acústica

Frequência	Tamanho da onda	Alcance
100 Hz	15 m	1000 km ou mais
1 kHz	1.5 m	100 km ou mais
10 kHz	15 cm	10 km
25 kHz	6 cm	3 km
50 kHz	3 cm	1 km
100 kHz	1.5 cm	600 m
500 kHz	3 mm	150 m
1 MHz	1.5 mm	50 m

Fonte: ([CHRIST; SR, 2013](#))

Sabendo a velocidade de propagação do som no meio e o intervalo de tempo que a onda levou para retornar, é possível calcular a distância do obstáculo até o sensor. Definimos como *ping* a ída e retorno de um pulso de som emitido pelo sensor, e tempo de *ping* como o intervalo de tempo para realizar um *ping*. A velocidade de propagação do som é maior nos sólidos, pois as moléculas estão mais próximas, transmitindo a energia cinética da onda de umas para as outras com maior facilidade. Na Tabela 2, ([HALLIDAY JEARL WALKER, 2008](#)) comprovou esse fato, onde se encontra diferentes velocidades do som de acordo com o meio.

Tabela 2 – Velocidade do som em diferentes meios.

Meio ^a	Velocidade (m/s)
<i>Gases</i>	
Ar (0 ° C)	331
Ar (20 ° C)	343
Hélio	965
Hidrogênio	1284
<i>Líquidos</i>	
Água (0 ° C)	1402
Água (20 ° C)	1482
Água salgada ^b	1522
<i>Sólidos</i>	
Aço	5941
Alumínio	6420
Granito	6000

^a A 0 ° C e 1 atm de pressão, a menos que haja uma indicação em contrário.

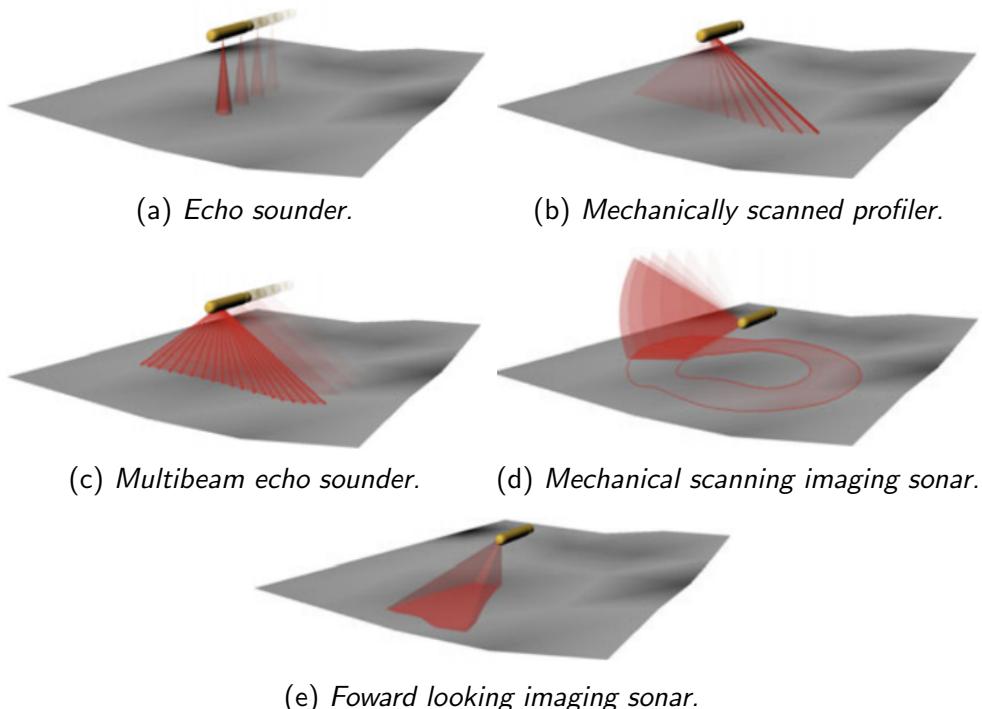
^b A 20 ° C e 3,5% de salinidade.

Fonte: ([HALLIDAY JEARL WALKER, 2008](#)).

2.1.2 Sonar *Singlebeam* e *Multibeam*

Além da classificação passiva e ativa, o sonar pode ser agrupado de diversas maneiras, uma delas referente a área de cobertura do sinal. O sonar *singlebeam* trabalha com apenas um único pulso de onda (Figura 4a). Geralmente ele é encontrado na parte inferior dos robôs e posicionado para baixo com o objetivo de encontrar a altitude do robô em relação ao fundo do corpo d'água em que está. Ainda pode ser posicionado nas partes laterais, frontal ou posterior para ser utilizado como um sensor de colisão.

Figura 4 – Diferentes modelos de sonares.



Fonte: Adaptado de ([RIBAS; RIDAO; NEIRA, 2010](#)).

Por outro lado, o sonar *multibeam* é mais complexo, composto por um conjunto de hidrofones² que emitem vários pulsos de onda de forma simultânea, que juntos possuem o formato de um leque (Figuras 4c e 4e). Esse tipo de sonar é comumente utilizado para fazer batimetria e imageamento de ambientes e superfícies submersas por possuir uma alta qualidade e velocidade de coleta de dados.

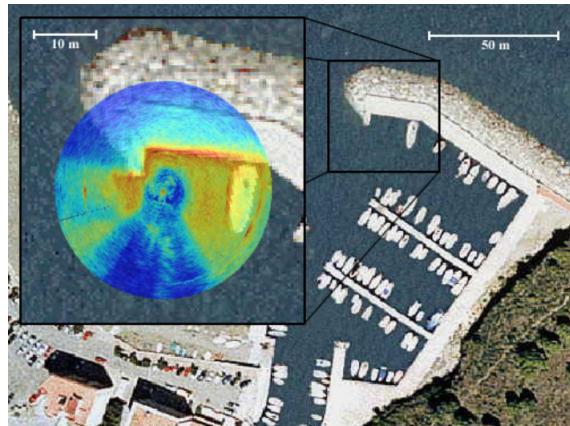
2.1.3 Sonar MSIS

O MSIS (do inglês *Mechanical Scanning Image Sonar*) é um sonar mecânico *singlebeam* de varredura de 360°, ou seja, possui um atuador no cabeçote do sensor para poder girar a

²Os hidrofones são transdutores de som para eletricidade que captam vibrações sonoras transmitidas no meio, reconhecendo na forma de frequência. Após reconhecer a frequência, o dispositivo transforma em um sinal elétrico.

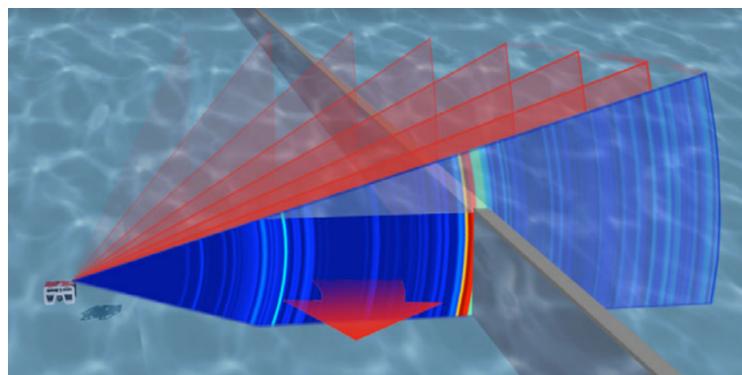
partir de passos pré-definidos e imagear o ambiente em seu entorno, ilustrado na Figura 4d. O sensor realiza o escaneamento no plano horizontal, onde o cabeçote gira em um incremento de um ângulo pré-estabelecido, efetuando uma leitura para cada incremento, como mostra a Figura 6. Para exemplificar, na Figura 5, apresenta uma imagem de satélite de uma doca com a sobreposição de uma imagem acústica obtida com um MSIS.

Figura 5 – Imagem acústica sobreposta em uma imagem de satélite.



Fonte: ([RIBAS et al., 2006](#))

Figura 6 – Representação do processo de escaneamento de um MSIS.



Fonte: ([RIBAS; RIDAO; NEIRA, 2010](#))

O processo de escaneamento pode ser vagaroso por conta do transdutor rotar os 360 graus³, porém seu baixo custo e peso torna o modelo viável para aplicações com alto risco de perda do equipamento como no caso das geleiras.

2.2 Robot Operating System

O ROS (Sistema Operacional de Robôs, do inglês *Robot Operating System*) é uma plataforma de desenvolvimento de softwares (*framework*) de código aberto direcionado para a

³Geralmente o equipamento possibilita a configuração para um range angular menor do que 360°, tornando menor o tempo de leitura de uma imagem completa.

área da robótica, que fornece a funcionalidade de um meta-sistema operacional, tais como, abstração de hardware, gerenciamento de pacotes, controle de dispositivos de baixo nível e troca de mensagens entre processos. O ROS atua como um *middleware* de comunicação entre aplicações robóticas e o *hardware*, oferecendo ferramentas e bibliotecas para auxiliar no desenvolvimento e execução de aplicações robóticas distribuídas. Os conjuntos de processos em execução do ROS são representados através de uma arquitetura de grafos orientados ou grafos de computação.

2.2.1 Grafos de computação

No ROS, os grafos de computação possuem a finalidade de representar sua rede de comunicação P2P entre os processos. O grafo computacional é composto pelos seguintes elementos:

- **Nodos:** Os nodos são os processos em execução no sistema e são responsáveis por realizar funções computacionais. Em um sistema ROS pode haver vários nodos responsáveis por controlar apenas um único robô, como por exemplo, um nodo responsável por controlar a localização do robô, outro nodo controla o sensor laser, outro faz o planejamento da rota, entre outros;
- **Mensagens:** As mensagens são estruturas de dados que podem incluir diversos tipos de dados, como inteiros, booleanos, strings, etc. Através das trocas de mensagens, os nodos fazem a comunicação entre si;
- **Tópicos:** Os tópicos são canais de comunicação que adotam a semântica de publicação/inscrição de mensagens. Os nodos utilizam essa semântica para publicar mensagens e/ou se inscreverem nos tópicos. Portanto, se um nodo publicar mensagens em um determinado tópico, todos os nodos inscritos nesse tópico receberão as mensagens. Além disso, podem haver múltiplos nodos inscritos e/ou publicando em um mesmo tópico, assim como um nodo pode publicar e se inscrever em múltiplos tópicos. A Figura 7 ilustra a relação entre nodo e tópico;
- **Serviços:** Diferente dos tópicos, os serviços possibilitam uma comunicação direta entre processos, através de interações requisição/resposta que é comumente empregado em sistemas distribuídos;
- **Bags:** O *bag* é um formato de arquivo usado pelo ROS para gravar e reproduzir dados gerados em um procedimento. Eles são extremamente importantes, pois proporcionam ao usuário gravar dados de sensores e atuadores para poder reproduzir futuramente;
- **Master:** O *master* permite que os nodos localizem uns aos outros para trocar mensagens ou invocar serviços. Uma vez que os nodos se localizaram através de consultas ao *master*, uma comunicação direta P2P entre os nodos é estabelecida. Além disso, ele

tem a tarefa de registrar nodos, tópicos, serviços e parâmetros da rede;

- Servidor de parâmetros: O servidor de parâmetros faz parte do *master*. Ele é responsável por permitir o compartilhamento distribuído de parâmetros em tempo de execução aos nodos.

Figura 7 – Relação entre tópico e nodo.

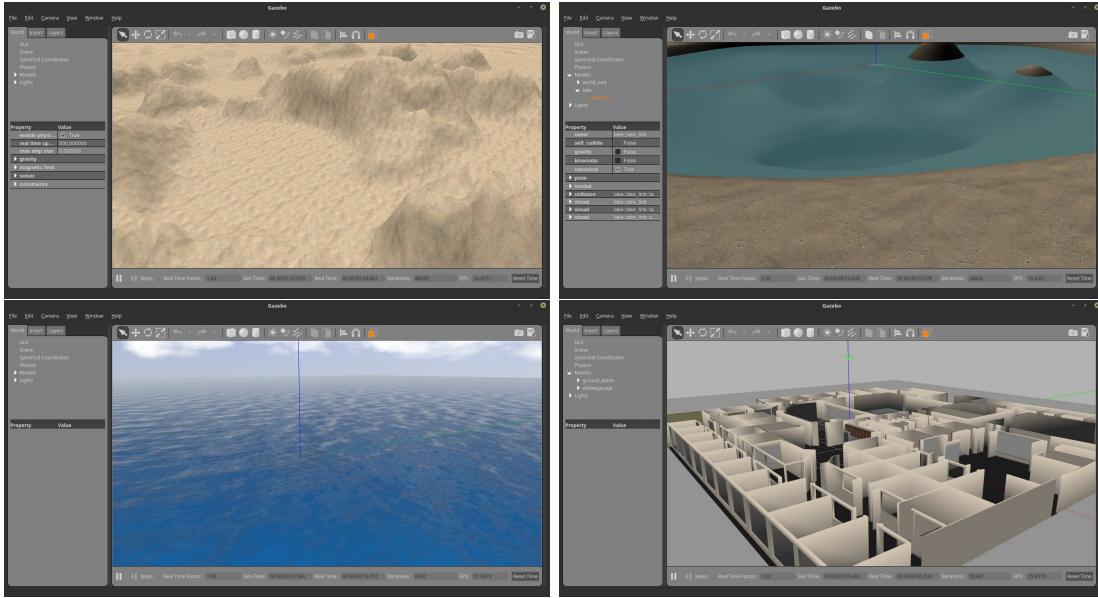


A Figura 7 ilustra com um exemplo, o funcionamento da estrutura de tópicos e nodos do ROS. O nodo *Sensor Sonar* é responsável por aquirir os dados da coleta e publicar esses dados no tópico *Scan*. Já o nodo *Aplicação* recebe os dados do tópico *Scan* e com base nesses dados realiza ações, como por exemplo, evitar uma colisão do robô com objetos no ambiente.

2.3 Simulador Gazebo

O Gazebo é um robusto *software* de código aberto para simulação de ambientes virtuais 3D para robôs. O simulador possui uma excelente ferramenta de simulação física, gráficos de alta qualidade, interface gráfica, ferramentas de criação de cenários realísticos e robôs, entre outras qualidades. Além disso, o *software* é integrado com o ROS, colaborando para o desenvolvimento de algoritmos, componentes, dispositivos e resoluções de problemas na área da robótica. A interface gráfica é acessível e intuitiva para o usuário, permitindo manipular o ambiente e seus elementos de forma fácil, como ilustra a Figura 8. Porém, a simulação em ambientes submersos ainda não está próximo do real, pois a ferramenta ainda não é capaz de adicionar uma das principais características contida no meio subaquático e fundamental para o desenvolvimento da área da visão subaquática, a turbidez na água.

Figura 8 – Interface gráfica do simulador Gazebo



Contudo, o desenvolvimento em código aberto facilita a criação de pacotes com plugins que auxiliam na simulação, como os pacotes *UUV Simulator* e o Simulador Netuno, desenvolvido por, respectivamente, (MANHÃES et al., 2016) e (LONGARAY, 2017). Ambos trabalhos contribuíram para a expansão da área da robótica subaquática.

2.4 GPU Sonar Simulator

O *GPU Sonar Simulator* é um simulador baseado no método proposto por (CER-QUEIRA et al., 2016), que utiliza *shaders* para processar imagens de sonar MSIS e FLS com baixo custo computacional. Os *shaders* são um conjunto de instruções que definem o comportamento da superfície dos objetos em um processamento gráfico. Eles são executados na GPU do computador, que é o processador dedicado especialmente para a renderização de gráficos em tempo real. Seu funcionamento é determinado pelo *pixel pipeline* ou *pipeline* de renderização 3D, que é a unidade encarregada pela transferência de dados referentes aos pixels que formam uma imagem. Portanto, o *pipeline* de renderização 3D é exclusivamente processado na GPU, resultando em uma redução na utilização de recursos da CPU e acelerando o processamento gráfico do computador.

O simulador foi inicialmente criado para a plataforma ROCK⁴ (do inglês *Robot Construction Kit*), porém através do trabalho de (LONGARAY, 2017) foi realizada a adaptação para o ROS. No desenvolvimento foi utilizado a linguagem C++ e a ferramenta OSG⁵ (do inglês *Open Scene Graph*). Quando um ambiente é renderizado no OSG, o *shader* computa

⁴O ROCK é um *framework* para o desenvolvimento de sistemas robóticos, semelhante ao ROS. Saiba mais em: <<https://www.rock-robotics.org/>>

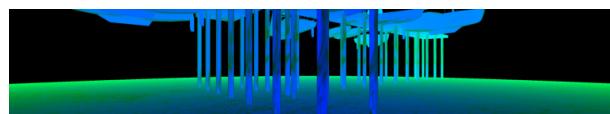
⁵O OSG é um conjunto de ferramentas gráficas 3D de alta performance de código livre e desenvolvida na linguagem C/C++. Saiba mais em: <<http://www.openscenegraph.org/>>

dados em um processamento a partir de uma câmera inserida na cena, onde tais dados são divididos em três categorias:

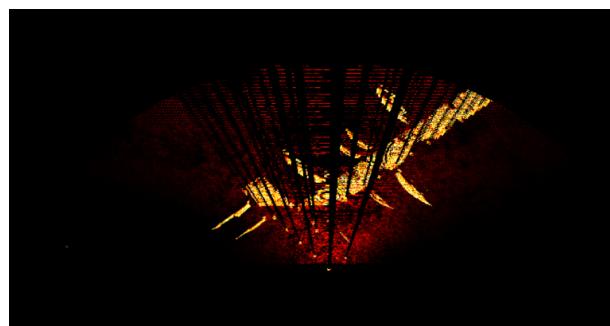
- **Intensidade:** simula a energia acústica refletida do objeto baseado no ângulo da superfície de contato;
- **Profundidade:** retrata a distância euclidiana entre o ponto central da câmera e o ponto da superfície de contato do obstáculo;
- **Distorção angular:** representa o ângulo entre a coluna central e as colunas nas extremidades da câmera.

A intensidade é calculada de acordo com os canais azul e verde, que representam, respectivamente, a intensidade e a distância, como mostra a Figura 9. Quanto mais claro é o azul em um determinado ponto, maior é a intensidade da energia acústica refletida, ao passo que quanto mais claro for o verde, maior é a distância ou profundidade do sensor até o ponto.

Figura 9 – Imagens correspondentes geradas pelo *GPU Sonar Simulator*



(a) Imagem gerada pelo *shader*



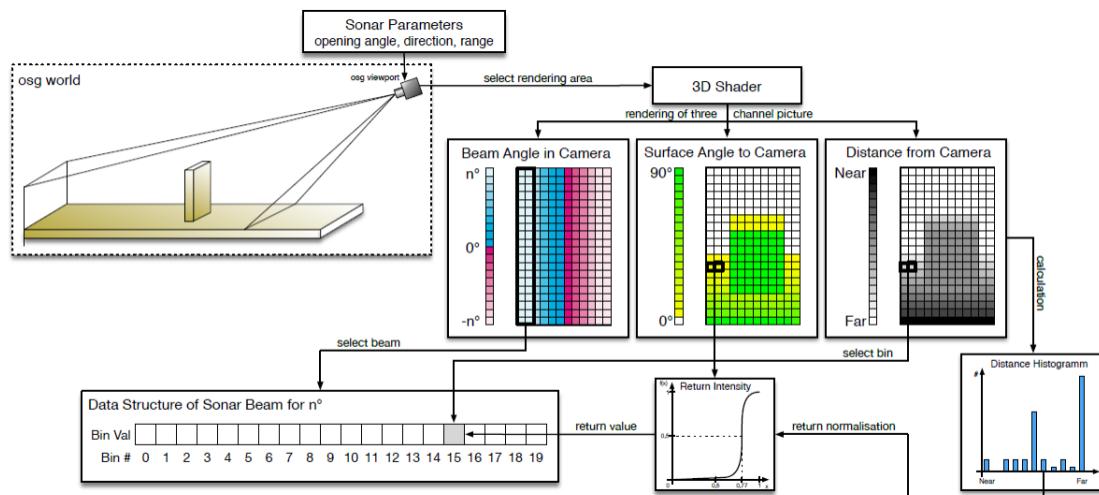
(b) Imagem acústica resultante.

Fonte: Adaptado de ([LONGARAY, 2017](#)).

O simulador do sonar possui alguns parâmetros de simulação que são definidos pelo usuário, tais como:

- **Quantidade de *beams*:** Define o número de colunas que a imagem será dividida. Se esse parâmetro for igual a 1, então, o simulador executará o sonar MSIS, se for maior que 1, então executará o FLS;
- **Quantidade de *bins*:** Quantidade de valores presentes em um *beam*;
- ***Beam width*:** Representa o ângulo de abertura horizontal da observação;
- ***Beam height*:** Representa o ângulo de abertura vertical da observação;
- ***Range*:** Define a distância máxima a ser observado (o alcance do sinal do sonar).

Figura 10 – Descrição gráfica do método utilizado no simulador do sonar



Fonte: ([CERQUEIRA et al., 2016](#)).

Esses parâmetros afetam no processamento do simulador e resolução da imagem. Quanto maior o número de *bins*, melhor será a qualidade da imagem acústica. A Figura 10 ilustra o método utilizado no simulador e descrito nessa seção.

2.5 Nuvem de Pontos

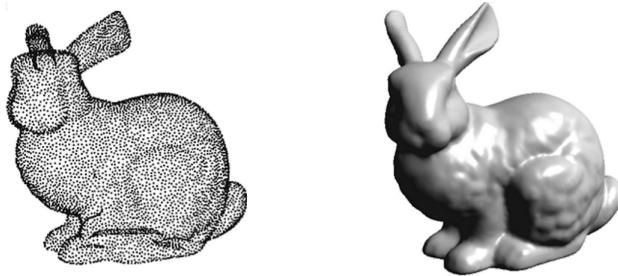
Uma nuvem de pontos é um conjunto de pontos expresso em um mesmo sistema de coordenadas, que possuem diversas finalidades, como criar modelos em 3D de animações, renderizações, objetos manufaturados, entre outros.. Em um sistema de coordenadas tridimensional, estes pontos são definidos por coordenadas X, Y e Z e comumente têm o objetivo de representar a superfície externa de um objeto.

2.6 Malha

Uma malha, também conhecida como malha de polígonos, malha de triângulos ou superfície, é um modelo que descreve um objeto tridimensional nos campos da computação gráfica. A malha é composta por um conjunto de pontos ou uma nuvem de pontos e uma coleção de faces, onde cada face é constituída por um conjunto de vértices. As faces geralmente são constituídas de triângulos ou quadriláteros, uma vez que estas formas simplificam o processo de renderização.

A Figura 11 ilustra a diferença entre uma nuvem de pontos e uma malha tridimensional.

Figura 11 – Diferença entre nuvem de pontos (esquerda) e uma malha de polígonos (direita).

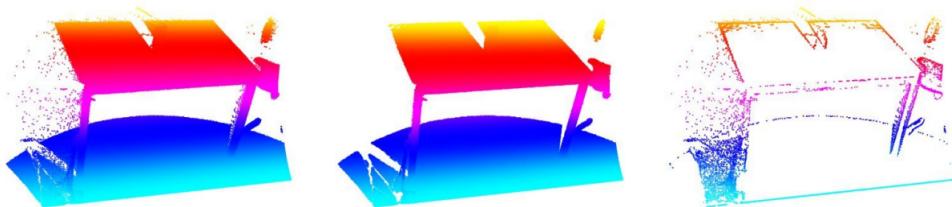


Um modelo tridimensional pode ser descrito de forma explícita ou implícita. A forma explícita é representada por um conjunto de vértices e por um conjunto de faces, que se relacionam para formar um objeto em 3D. Por sua vez, na forma implícita o conjunto de pontos já está organizado em forma de matriz, gerando automaticamente o conjunto de faces. Além disso, por ser organizado em forma de matriz, o modelo implícita torna a malha em uma espécie de imagem em 3D, possibilitando a aplicação de transformações utilizadas em processamento de imagens digitais, tais como os filtros digitais (na Seção 3.3 é abordado alguns filtros).

2.7 Point Cloud Library

A PCL (Biblioteca de Nuvem de Pontos, do inglês *Point Cloud Library*) é um *framework* para o processamento de imagens e nuvens de pontos em 2D/3D desenvolvido para a linguagem C++. O *framework* é um projeto de código aberto em larga escala criado por (RUSU; COUSINS, 2011). A PCL contém diversos algoritmos no estado da arte, incluindo filtros, estimativa de features, reconstrução de superfícies, segmentação, entre outros. Em paralelo, (STRAW, 2012) adaptou parte da PCL para a linguagem python.

Figura 12 – Exemplo de utilização da PCL em dados adquiridos por um sensor a laser.



Fonte: (RUSU; COUSINS, 2011).

A Figura 12 ilustra um exemplo da aplicação da PCL em uma nuvem de pontos. Na imagem a esquerda mostra uma nuvem de pontos adquirida a partir de um sensor a laser, a imagem do meio mostra a nuvem resultante após passar por um filtro⁶, por último, os pontos

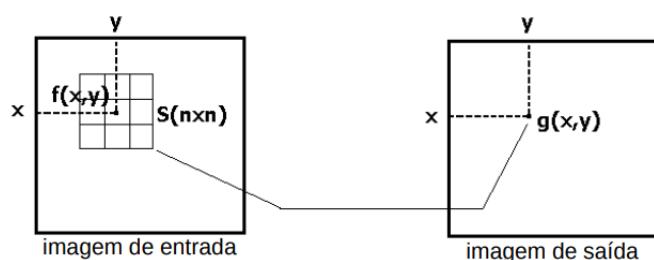
⁶O filtro utilizado se chama *StatisticalOutlierRemoval* (filtro estatístico de remoção de *outliers*) descrito na Seção 3.3.5. A documentação do filtro se encontra em: <http://pointclouds.org/documentation/tutorials/statistical_outlier.php>.

descartados são mostrados na figura na direita.

2.8 Filtros Espaciais

Filtros espaciais são um conjunto de técnicas de processamento de imagens que operam os valores de uma vizinhança para modificar uma imagem digital (atuando diretamente nos *pixels* da imagem). A vizinhança e o *pixel* que estão sendo operados, estão organizados em uma matriz, chamada de *kernel* ou janela. Quanto maior o tamanho da janela, maior será a intensidade do filtro. Para fazer a modificação, é necessário utilizar uma função que combine valores de um certo número de *pixels* da imagem de entrada com valores do *kernel*, para gerar o valor resultante na imagem de saída (Figura 13). Existem diversos filtros e diversos tipos de *kernels*, entre eles estão o filtro gaussiano e o filtro de mediana.

Figura 13 – Aplicação de um filtro em uma imagem de entrada e seu resultado na imagem de saída.



2.8.1 Filtro Gaussiano

O Filtro Gaussiano é um operador de convolução 2D (varredura de imagem por uma máscara), que é usado para suavizar bordas em imagens e remover ruídos, como mostra a Figura 14.

Figura 14 – Aplicação do Filtro Gaussiano. Imagem original à esquerda, e imagem filtrada à direita.



2.8.2 Filtro de Mediana

Filtro da mediana é uma transformação utilizada para suavizar ruído do tipo impulsivo em sinais e imagens digitais.

Figura 15 – Aplicação do Filtro de Mediana. Imagem com ruído à esquerda, e imagem filtrada à direita.



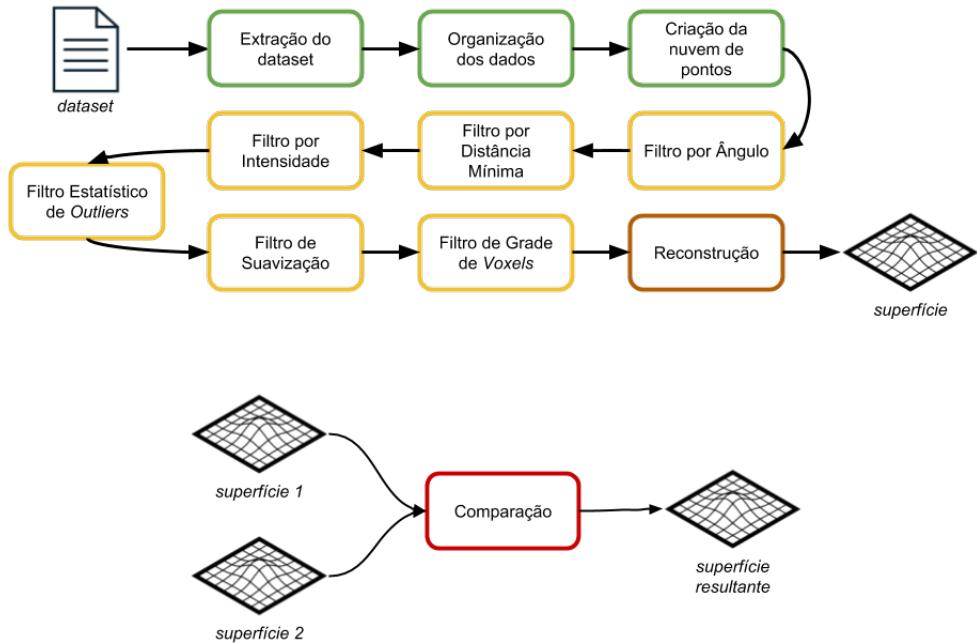
3 METODOLOGIA

Para uma melhor organização, a efetivação do trabalho é composta por duas partes, onde a primeira é a coleta de dados e, a segunda, o desenvolvimento da aplicação. Na segunda parte da execução do trabalho, será o desenvolvimento do algoritmo de reconstrução, que consistirá em quatro etapas, no qual, para cada etapa é desenvolvido uma rotina. Primeiro os dados serão preparados, em seguida serão refinados e filtrados, por fim, renderizados. O produto final do processamento, será uma superfície tridimensional. Cada parte está segmentada em etapas, no qual cada etapa é um processo a ser executado. Por conseguinte, a organização da metodologia apresenta os itens:

- Coleta de dados
 - Dados em simulação
 - Dados *in loco*
- Desenvolvimento
 - Pré-processamento
 - Filtragem
 - Reconstrução
 - Comparação

A Figura 16 mostra as etapas do desenvolvimento da aplicação. Primeiramente, os dados são inseridos através de um *dataset*, em verde são representados as etapas do pré-processamento, em amarelo são representados os processos de filtragem, o processo de reconstrução é representado em marrom, e por fim, a superfície é gerada. A etapa de comparação requer duas superfícies na entrada do processo, para gerar a superfície resultante.

Figura 16 – Etapas do processo de desenvolvimento da aplicação.



3.1 Coleta de Dados

A coleta de dados será realizada de duas formas: em simulação virtual do ambiente e *in loco*. A coleta em simulação virtual facilita a obtenção de dados, por não ser necessário possuir um sensor ou robô e por dispensar planejamento logístico que uma saída de campo requer. Contudo, é importante fazer a coleta *in loco*, pois, as informações obtidas em simulação não serão próximo do real quanto às coletados em campo.

A obtenção de dados por simulação virtual contará com o auxílio de *softwares* e ferramentas. O *software* Gazebo (seção 2.3) será o responsável por simular o ambiente virtual e executará dentro do ROS (seção 2.2) junto com outros pacotes, tais como o modelo de robô ROV (RexROV) e o simulador do sonar MSIS (seção 2.4). O RexROV é um modelo de robô desenvolvido pelos autores (MANHÃES et al., 2016) e está incluso no pacote *UUV Simulator*.

A aquisição de dados pela coleta em campo, será realizado no local ainda a ser definido e será utilizado o ROV, modelo LBV300-5¹ (Figura 17), do grupo do NAUTEC do C3. Acoplado ao robô, será utilizado o sensor MSIS modelo Tritech Micron Sonar² que fará a captura dos dados.

¹Informações completas do modelo do ROV em: <<http://www.teledynmarine.com/lbv300-5>>

²Informações completas do modelo do sensor em: <<http://www.tritech.co.uk/media/products/small-rov-mechanical-sector-scanning-sonar-tritech-micron.pdf?id=e3070c7b>>

Figura 17 – ROV utilizado no trabalho.

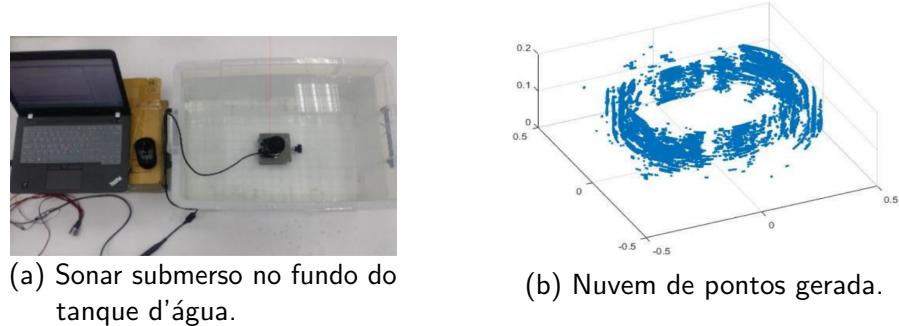


Para analisar a diferença entre os dados de ambas coletas, será realizada comparações entre os resultados obtidos em cada procedimento. Os materiais coletados, formarão um *dataset* que servirá como entrada no algoritmo de reconstrução.

3.2 Pré-processamento

A etapa de pré-processamento serve para manipular os dados do *dataset* e dispor em uma nuvem de pontos. No final da fase de pré-processamento, o produto esperado é uma nuvem de pontos da superfície que foi escaneada pelo sonar. Esse resultado servirá como entrada para o próximo processo.

Figura 18 – Nuvem de pontos gerada a partir de dados coletados por um sonar MSIS.



Fonte: ([DONG; CHOU; YAO, 2017](#)).

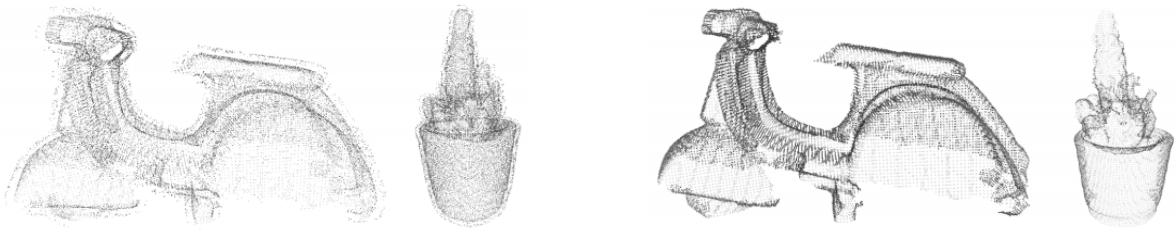
Na Figura 18a é possível observar um sonar MSIS mergulhado em um recipiente com água. O sensor faz o reconhecimento das paredes do recipiente e gera uma nuvem de pontos (Figura 18b).

3.3 Filtragem

O processo de filtragem ou *denoising* é composto por um conjunto de filtros em forma de funções que tem como objetivo remover ou atenuar informações desnecessárias da

nuvem de pontos, como ruídos e *outliers*³. O resultado da etapa de filtragem é uma nuvem de pontos mais limpa e legível, como ilustrado na Figura 19. Desse modo, com um conjunto de pontos sem ruídos, torna mais simples o processo para ligar os pontos e formar um modelo tridimensional.

Figura 19 – Remoção de ruído em uma nuvem de pontos.



(a) Nuvem de pontos com ruído.

(b) Nuvem de pontos após remoção do ruído.

Fonte: Adaptado de ([ARVANITIS et al., 2017](#))

Para realizar essa etapa, os filtros foram separados em dois grupos. O primeiro grupo é composto por filtros simples, porém removem maior parte dos pontos desnecessários. O segundo conjunto é formado por filtros mais complexos onde muitos já estão implementados na PCL (Seção 2.7). Os filtros simples são representados pelo filtro por ângulo, filtro por intensidade e filtro por distância mínima, enquanto que os filtros complexos, são retratados pelo Filtro de Grade de Voxels⁴, Filtro de Suavização e o Filtro Estatístico de Remoção de *Outliers*.

3.3.1 Filtro por Ângulo

O filtro por ângulo, remove qualquer ponto que está fora da área cônica definida pelo ângulo horizontal θ . Para esclarecer, na nuvem de pontos da Figura 20a são mostrados em cinza e em preto, os pontos descartados e mantidos pelo filtro, respectivamente. O ângulo de abertura no sensor geralmente é configurável no próprio hardware, porém o critério do descarte fica por conta da aplicação, pois em um *dataset*, os mesmos pontos que são inúteis em um trabalho podem ser úteis em outro. Por padrão, foi utilizado 120 graus para o ângulo θ .

3.3.2 Filtro por Distância Mínima

O filtro por distância mínima faz a remoção de pontos que estão próximos ao sensor, onde a presença de ruído costuma ser elevada. Da mesma forma como abordado no filtro anterior, os pontos em cinzas são descartados enquanto que os pretos são mantidos, como

³*Outliers* é um termo da área da estatística utilizado para referenciar elementos atípicos ou observações que estão muito afastadas do restante da série.

⁴Um voxel é um *pixel* que contém volume em um espaço tridimensional. Fazendo uma analogia, o voxel seria um *pixel* em 3D.

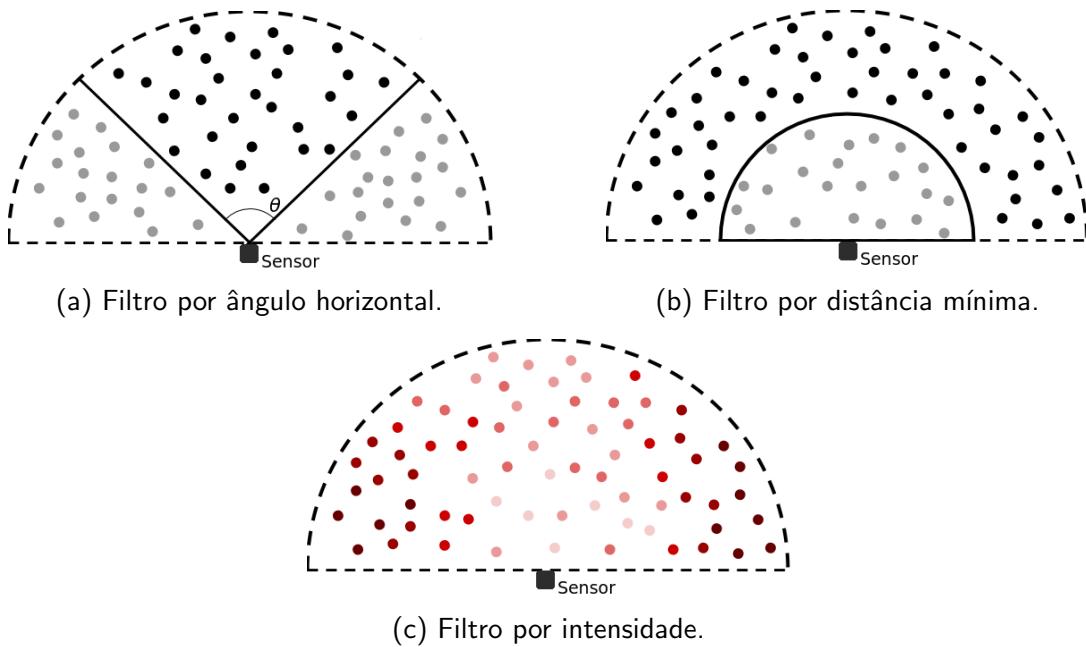
mostra a Figura 20b. Por padrão, o valor adotado para o raio de exclusão é de 5 metros, ou seja, serão descartados todos os pontos que estiverem até 5 metros de distância do sensor.

3.3.3 Filtro por Intensidade

Este processo remove os pontos que possuem baixo valor de intensidade. Os motivos para o sinal de intensidade ser baixo em um determinado ponto, pode ser:

- O ponto estar distante do sensor. Nesse caso, o sinal da onda estará fraco ao entrar em contato com o ponto, por conta da perda de energia para o meio ao longo da distância percorrida;
- O ângulo de incidência da onda com o ponto ser obtuso. Quando maior o ângulo de incidência entre a onda e o ponto, mais fraco será o sinal de retorno;
- A composição do objeto detectado pela onda. Quanto mais denso for o objeto, maior será o sinal de retorno.

Figura 20 – Filtros simples de remoção de *outliers*.



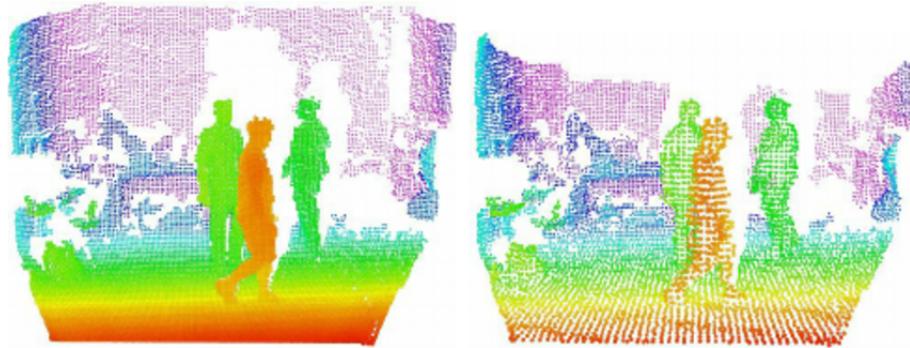
O exemplo ilustrado na Figura 20c, mostra uma nuvem onde os pontos possuem diversos valores de intensidade. Quanto mais claro for o ponto, maior será o seu valor de intensidade do sinal de retorno. Portanto, o filtro descarta os pontos mais escuros na imagem. Por padrão, é descartado todos os pontos que possuírem o valor de intensidade igual ou menor que 5% da diferença entre o valor máximo e o valor mínimo de intensidade da nuvem de pontos.

3.3.4 Filtro de Grade de Voxels

O Filtro de Grade de Voxels (do inglês *Voxel Grid Filter*) tem a finalidade de diminuir a resolução/densidade e ordenar os pontos em uma nuvem. A Figura 21 exibe uma nuvem de

pontos de entrada (esquerda) e o resultado gerado após aplicar o filtro (direita).

Figura 21 – Exemplo da aplicação do Filtro de Grade de *Voxels*.



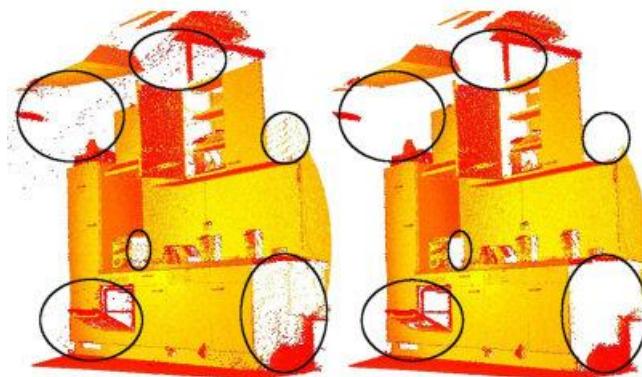
Fonte: ([MUNARO; BASSO; MENEGATTI, 2012](#)).

Para diminuir a quantidade de pontos e ordená-los, o algoritmo cria uma grade no conjunto de dados, onde os pontos são deslocados para o centroide da célula (*voxel*) da grade em que estão contidos (o tamanho do *voxel* é definido pelo usuário). A diminuição da quantidade e ordenação dos pontos facilita o processo de triangularização (etapa posterior responsável por transformar a nuvem de pontos em uma superfície).

3.3.5 Filtro Estatístico de Remoção de *Outliers*

O SORF (Filtro Estatístico de Remoção de *Outliers*, do inglês *Statistical Outlier Removal Filter*) é um filtro da biblioteca PCL que é responsável por remover conjuntos de pontos que possuem uma densidade menor do que o restante dos conjuntos de pontos dentro da nuvem. A Figura 22 exemplifica a aplicação do filtro. Na imagem à esquerda mostra a nuvem original que contém regiões em destaque, onde a densidade de pontos é menor que no restante da nuvem, enquanto que na imagem à direita mostra a remoção desses conjuntos de pontos.

Figura 22 – Exemplo do uso do SORF.



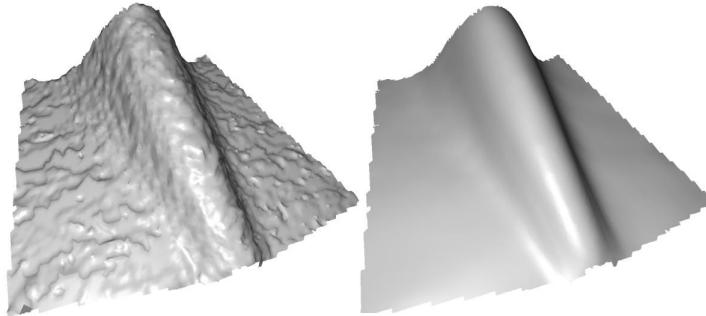
Fonte: Adaptado de ([RUSU; COUSINS, 2011](#)).

O filtro percorre a nuvem calculando, para cada ponto, a distância média entre seus pontos vizinhos (o número de vizinhos é definido pelo usuário). Assumindo o resultado como uma distribuição Gaussiana com uma média e um desvio padrão, todos os pontos cuja distância média estão fora de um intervalo definido pela média das distâncias globais e desvio padrão podem ser considerados *outliers* e, portanto, removidos do conjunto de dados.

3.3.6 Filtro de Suavização

O Filtro de Suavização é um filtro que está presente na biblioteca PCL e é responsável por suavizar e reconstruir a superfície de pontos na nuvem. Para realizar o procedimento, o filtro utiliza o MLS (do inglês *Moving Least Squares*), um método de reamostragem, desenvolvido por ([LEVIN, 1998](#)), capaz de recriar partes ausentes ou suavizar superfícies por meio da interpolação polinomial entre pontos. Para exemplificar, a Figura 23 exibe uma superfície sem tratamento (imagem à esquerda) e a mesma superfície após a aplicação do filtro (imagem à direita).

Figura 23 – Exemplo do uso do Filtro de Suavização.



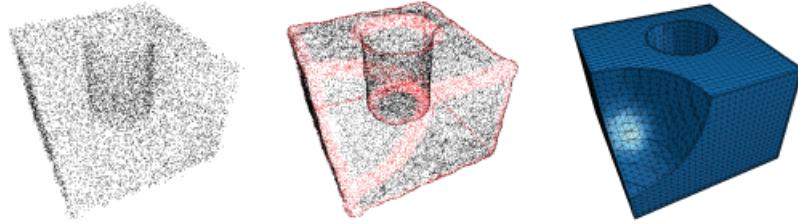
Fonte: Adaptado de ([RIDEL et al., 2015](#)).

No trabalho, o filtro é utilizado com o intuito de suavizar e corrigir boa parte do ruído e pequenos erros de medição de distância gerado nos dados, provenientes da movimentação do robô enquanto realiza a coleta.

3.4 Reconstrução

A reconstrução do modelo tridimensional é gerada a partir da nuvem de pontos, onde é obtido a forma final da superfície. Como exemplo, a Figura 24 mostra a reconstrução de um objeto passo a passo.

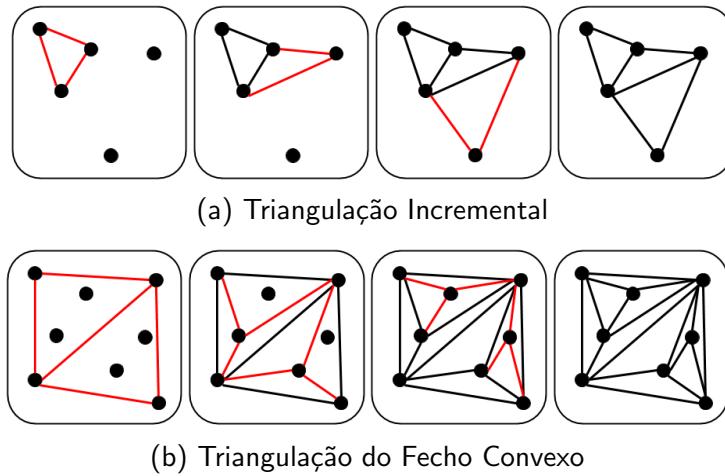
Figura 24 – Reconstrução de um objeto em 3D a partir de uma nuvem de pontos.



Fonte: Adaptado de ([JENKE et al., 2006](#)).

Para dar forma à superfície é necessário uma lógica para ligar os pontos, para tal lógica chamamos de triangulação. No trabalho, a triangulação é feita através da construção de triângulos sem fazer sobreposições, como ilustrado nos exemplos da Figura 25. Apesar do nome triangulação remeter à triângulos, ela pode ser constituída por qualquer espaço em simplexos ou polígonos. Os simplexos são extensões de triângulos em outras dimensões, tais como segmentos de reta, tetraedros, e etc. Existem diversas formas para fazer a triangulação em um conjunto de pontos, entre elas estão a triangulação incremental e a triangulação do fecho convexo.

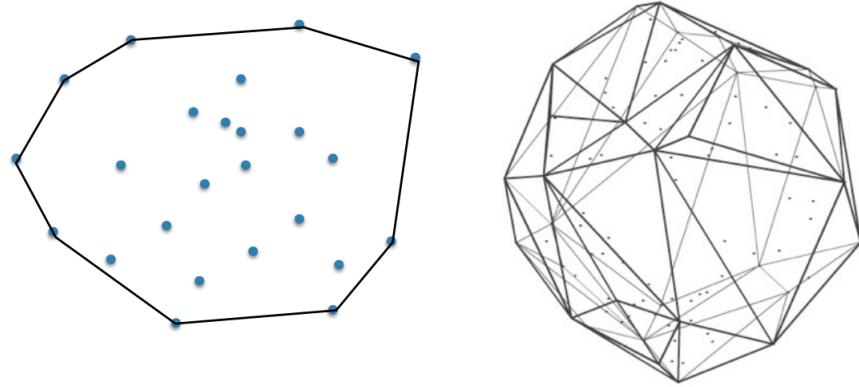
Figura 25 – Exemplos de modelos de triangulação.



Na Figura 25a mostra a triangulação incremental, onde é realizada a construção de triângulos a partir da seleção aleatória do ponto inicial, seguindo a ligação pelo ponto mais próximo. Na Figura 25b mostra a triangulação do fecho convexo, na qual se realiza o fecho convexo do conjunto de pontos seguindo pela triangulação. Após esse primeiro passo, para cada ponto no interior dos triângulos formados, se divide em mais três triângulos.

Fecho convexo (também conhecido como invólucro convexo ou envoltório convexo) são estruturas que englobam o conjunto de pontos, através da intersecção com seus pontos marginais. No plano, o fecho convexo é um polígono, enquanto que no espaço é representado por uma superfície (Figura 26).

Figura 26 – Fecho convexo representado no plano (esquerda) e no espaço (direita).

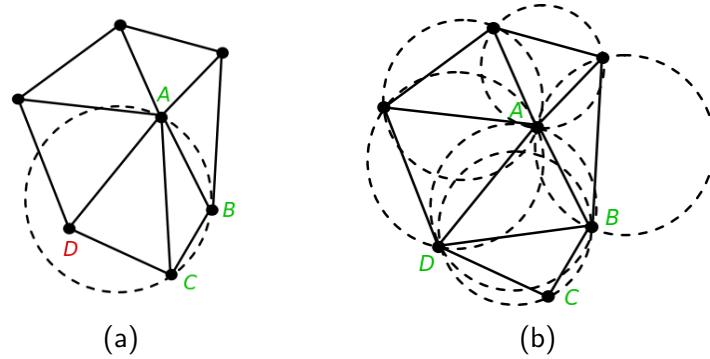


Uma triangulação nem sempre é adequada para um certo problema, pois os triângulos precisam ser parecidos com o triângulo equilátero (triângulos que não possuem ângulos internos agudos). Essas triangulações especiais usam critérios para definir triângulos, entre os modelos conhecidos existe a Triangulação de Delaunay.

3.4.1 Triangulação de Delaunay

A triangulação de Delaunay, proposto por (DELAUNAY et al., 1934), é o método responsável por fornecer uma triangulação com o maior ângulo mínimo, ou seja, os melhores triângulos de forma única, em um determinado conjunto de pontos. O método pode ser formulado e estudado no espaço euclidiano de dimensão n , mas para fins explicativos será definida e exemplificada apenas no plano.

Figura 27 – (a) Triangulação que não satisfaz o critério de Delaunay; (b) Triangulação de Delaunay.

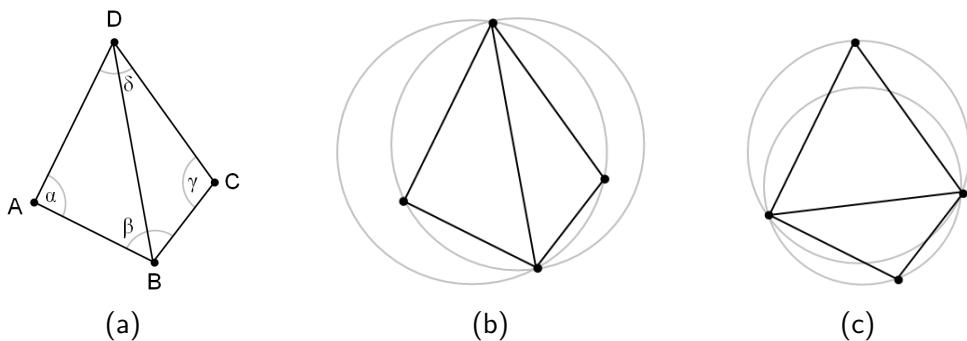


Fonte: Adaptado de (PITERI; JUNIOR, 2007).

Para a criação dos triângulos, se pode utilizar qualquer metodologia de triangulação (inclusive as que foram apresentadas na Figura 25). Após a criação, há uma verificação, que deve ser realizada em cada triângulo, definida pela regra do circuncírculo, que determina se o triângulo é ou não de Delaunay. A regra descreve que para cada triângulo da malha, é traçado

um círculo que passa pelos três vértices do triângulo, se esse círculo conter apenas os 3 pontos do triângulo, então ele satisfaz a regra. Se haver mais de 3 pontos dentro, então é aplicada a troca (*flip*) entre arestas contidas no círculo. Para explicar o termo *flip*, observe a Figura 27a, onde o triângulo ABC não satisfaz a regra do circuncírculo, pois há outro vértice contido (D). Para resolver esse problema se utiliza o *flip* entre os triângulos ABC e ACD, que faz a troca da aresta AC pela aresta DB. Refazendo a regra do circuncírculo nos triângulos, é possível observar (Figura 27b) que ambos são triângulos de Delaunay.

Figura 28 – Regra do circuncírculo que define se um triângulo é ou não de Delaunay.



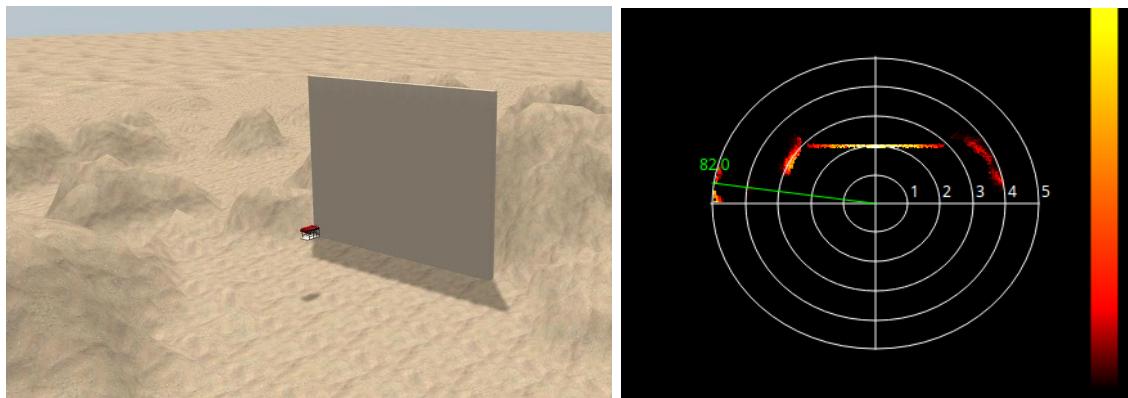
Na Figura 28 há outro exemplo de *flip*. A Figura 28b mostra ambos triângulos que não obedecem a regra do circuncírculo. Realizando o *flip* entre as arestas BD e AC, a regra se torna válida (Figura 28c).

3.5 Comparação

A comparação é a última etapa do processo de desenvolvimento. Essa parte é possível fazer a comparação entre dois modelos e determinar o quanto se perdeu ou ganhou em relação ao volume. Na fase de comparação será possível modelar coletas do mesmo local em diferentes períodos e fazer a comparação entre os resultados obtidos.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Figura 29 – Imagem acústica do sonar gerada em um ambiente de simulação.



(a) Simulação do cenário no Gazebo.

(b) Imagem acústica correspondente.

5 CONCLUSÃO

5.1 Trabalhos Futuros

5.2 Considerações Finais

Referências

- ARVANITIS, G. et al. Real-time removing of outliers and noise in 3d point clouds applied in robotic applications. In: SPRINGER. **International Conference on Interactive Collaborative Robotics**. [S.I.], 2017. p. 11–19. Citado na página 19.
- CERQUEIRA, R. et al. Custom shader and 3d rendering for computationally efficient sonar simulation. **SIBIGRAPI 2016**, 2016. Citado 2 vezes nas páginas 10 e 12.
- CHRIST, R. D.; SR, R. L. W. **The ROV manual: a user guide for remotely operated vehicles**. [S.I.]: Butterworth-Heinemann, 2013. Citado na página 5.
- COELHO, R. P.; HAVENS, K. Crise nas águas. **Educação, ciência e governança, juntas, evitando conflitos gerados por escassez e perda de qualidade das águas**, Editora Recóleo, Belo Horizonte – MG, p. 103–109, 2015. Citado na página 1.
- DELAUNAY, B. et al. Sur la sphere vide. **Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk**, v. 7, n. 793-800, p. 1–2, 1934. Citado na página 24.
- DONG, M.; CHOU, W.; YAO, G. Underwater 3d reconstruction based on geometric transformation of sonar and depth information. In: IOP PUBLISHING. **IOP Conference Series: Materials Science and Engineering**. [S.I.], 2017. v. 261, n. 1, p. 012014. Citado na página 18.
- HALLIDAY JEARL WALKER, R. R. D. **Fundamentals of Physics**. 8^a. ed. [S.I.]: LTC, 2008. v. 8^a. Citado na página 5.
- HANSEN, J. et al. Global temperature change. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 103, n. 39, p. 14288–14293, 2006. Citado na página 1.
- JENKE, P. et al. Bayesian point cloud reconstruction. In: WILEY ONLINE LIBRARY. **Computer Graphics Forum**. [S.I.], 2006. v. 25, n. 3, p. 379–388. Citado na página 23.
- KERLE, N.; JANSSEN, L. L.; HUURNEMAN, G. C. Principles of remote sensing. **ITC, Educational textbook series**, v. 2, p. 250, 2004. Citado na página 2.
- LEVIN, D. The approximation power of moving least-squares. **Math. Comput.**, American Mathematical Society, Boston, MA, USA, v. 67, n. 224, p. 1517–1531, out. 1998. ISSN 0025-5718. Disponível em: <<http://dx.doi.org/10.1090/S0025-5718-98-00974-0>>. Citado na página 22.
- LONGARAY, L. M. Desenvolvimento de datasets simulados em ambientes subaquáticos para uso em aplicações de deep learning. 2017. Citado 2 vezes nas páginas 10 e 11.
- MANHÃES, M. M. M. et al. UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In: **OCEANS 2016 MTS/IEEE Monterey**. [S.I.]: IEEE, 2016. Citado 2 vezes nas páginas 10 e 17.
- MUNARO, M.; BASSO, F.; MENEGATTI, E. **Tracking people within groups with RGB-D data**. 2012. 2101-2107 p. Citado na página 21.

- PACHAURI, R. K. et al. **Climate change 2014: synthesis report. Contribution of Working Groups I, II and III to the fifth assessment report of the Intergovernmental Panel on Climate Change.** [S.I.]: IPCC, 2014. Citado na página 1.
- PITERI, M. A.; JUNIOR, M. M. Triangulação de delaunay e o princípio de inserção randomizado. **II Simpósio Brasileiro de Geomática-V Colóquio Brasileiro de Ciências Geodésicas, Presidente Prudente-SP**, p. 9, 2007. Citado na página 24.
- RIBAS, D.; RIDAO, P.; NEIRA, J. **Underwater SLAM for structured environments using an imaging sonar.** [S.I.]: Springer, 2010. v. 65. Citado 2 vezes nas páginas 6 e 7.
- RIBAS, D. et al. Slam using an imaging sonar for partially structured underwater environments. In: IEEE. **Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on.** [S.I.], 2006. p. 5040–5045. Citado na página 7.
- RIDEL, B. et al. Parabolic-cylindrical moving least squares surfaces. **Computers and Graphics**, Elsevier, p. 7, 2015. Disponível em: <<https://hal.inria.fr/hal-01169572>>. Citado na página 22.
- RIETBROEK, R. et al. Revisiting the contemporary sea-level budget on global and regional scales. **Proceedings of the National Academy of Sciences**, National Academy Sciences, v. 113, n. 6, p. 1504–1509, 2016. Citado na página 1.
- RUSU, R. B.; COUSINS, S. 3D is here: Point Cloud Library (PCL). In: **IEEE International Conference on Robotics and Automation (ICRA)**. Shanghai, China: [s.n.], 2011. Citado 2 vezes nas páginas 13 e 21.
- STRAW, A. **python-pcl.** [S.I.]: GitHub, 2012. <<https://github.com/strawlab/python-pcl>>. Citado na página 13.