



**Guilherme Ferreira Gusmão**

## **Desenvolvimento e Validação de Sensor LiDAR Virtual**

### **Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Metrologia (Área de concentração: Metrologia para Qualidade e Inovação) da PUC-Rio.

Orientador: Prof. Carlos Roberto Hall Barbosa  
Coorientador: Prof. Alberto Barbosa Raposo

Rio de Janeiro  
Março de 2020



**Guilherme Ferreira Gusmão**

## **Desenvolvimento e Validação de Sensor LiDAR Virtual**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Metrologia (Área de concentração: Metrologia para Qualidade e Inovação) da PUC-Rio. Aprovada pela Comissão Examinadora abaixo.

**Prof. Carlos Roberto Hall Barbosa.**

Orientador

Programa de Pós-Graduação em Metrologia — PUC-Rio

**Prof. Alberto Barbosa Raposo.**

Coorientador

Departamento de Informática — PUC-Rio

**Prof. Eduardo Costa da Silva.**

Departamento de Engenharia Elétrica — PUC-Rio

**Prof. Guilherme Penello Temporão.**

Departamento de Engenharia Elétrica — PUC-Rio

**Prof. Waldemar Celes (Filho).**

Instituto Tecgraf — PUC-Rio

Rio de Janeiro, 11 de março de 2020

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Guilherme Ferreira Gusmão**

Graduado em Engenharia Elétrica com Ênfase em Eletrônica pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), em 2017. Atualmente mestrando do Programa de Pós-Graduação em Metrologia (PósMQI) pela PUC-Rio.

#### Ficha Catalográfica

Gusmão, Guilherme Ferreira

Desenvolvimento e validação de sensor LiDAR virtual / Guilherme Ferreira Gusmão ; orientador: Carlos Roberto Hall Barbosa ; coorientador: Alberto Barbosa Raposo. – 2020.

104 f. : il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Centro Técnico Científico, Programa de Pós-Graduação em Metrologia, 2020.

Inclui bibliografia

1. Metrologia – Teses. 2. Metrologia para Qualidade e Inovação – Teses. 3. Sensor virtual. 4. LiDAR. 5. Nuvem de pontos sintética. 6. Sensoriamento remoto. 7. Rastreamento de raios. I. Barbosa, Carlos Roberto Hall. II. Raposo, Alberto Barbosa. III. Pontifícia Universidade Católica do Rio de Janeiro. Centro Técnico Científico. Programa de Pós-Graduação em Metrologia. IV. Título.

CDD: 389.1

Em memória dos meus avôs, Edson de Souza Leal e José Cícero Ferreira, que  
forjaram grande parte de quem sou.  
Deixaram de ser homens para se tornarem lendas.

## Agradecimentos

A Deus por me reerguer todas as vezes que caí. Sem Ele, nada seria.

Ao meu avô, Edson de Souza Leal, por me ensinar a ser amável com todos que cruzarem meu caminho. Em mostrar que uma família unida, atravessa obstáculos colossais. O mundo é menos gentil sem seu carinho.

Ao meu avô, José Cícero Ferreira, por me ensinar a me manter firme, independente da adversidade. Em acreditar que tudo é possível, se trabalharmos por nossos objetivos. O mundo é menos alegre sem seu sorriso.

Ao meu orientador Carlos Roberto Hall Barbosa por todos esses anos de ensinamentos e conselhos, além de ouvir pacientemente meus inúmeros devaneios.

Ao meu coorientador Alberto Barbosa Raposo, por acreditar no meu trabalho e me dar uma oportunidade sem igual.

Ao Renato Cherullo, por toda dedicação e motivação dada durante este trabalho. Esta dissertação também pertence a você.

A equipe Tecgraf por partilharem suas experiências comigo. Em especial: Alejandro Díaz, pelo direcionamento e conversas filosóficas. César Sierra, pelas orientações na dissertação. Daniel Radetic, pelas contribuições estatísticas. Eduardo Albuquerque, pelo bom humor. Felipe Carvalho, pela oportunidade e apoio. José Hugo, pelos conhecimentos em CUDA. Leonardo Santiago, por ajudar na construção do módulo Cenário Virtual. Thiago Motta, por todo o suporte dentro e fora do trabalho.

A professora Karla Tereza Figueiredo Leite, por contribuir com o dispositivo LiDAR utilizado nesta dissertação.

Ao Engenheiro Rogério Regazzi, pela gentil cessão dos dados de escaneamento aéreo apresentado no Capítulo 5.

Aos membros da comissão examinadora por suas contribuições a este trabalho.

À professora Ana Maria Beltran Pavani por dar asas as minhas ideias e conselhos nas tormentas.

Aos professores, funcionários e alunos do Programa de Pós-graduação em Metrologia por enriquecerem minha vida.

As minhas avós, Lanuza e Leny, por serem as pessoas mais fortes que eu conheço. A felicidade de vocês, me motiva a sempre seguir em frente.

A minha mãe, Eliane Ferreira Gusmão, pela guerreira que é, perseverante e altruísta, com um coração capaz de carregar o mundo e mais um pouco.

Ao meu pai, Janio César Souza Gusmão, por se esforçar todos os dias da minha vida para me fazer sorrir.

A minha irmã, Rafaela Ferreira Gusmão, por cuidar de mim e me motivar a sonhar. Sempre uma nova aventura ao seu lado.

A minha namorada, Thábata Rayane Guimarães, por ser minha estrela vespertina no meio de tanta escuridão. Eu sempre vou esperar por você, não importa quantas léguas precisarei atravessar.

Ao meu tio Dielson e minha tia Deise, por me acolherem e vibrarem com minhas vitórias.

A toda minha família pelo suporte ao longo desses anos turbulentos.

Aos meus amigos, Bernardo Antônio Correa, José Guilherme Cucco, Nicolás Conley, Rodolfo Raposo e William Gomes. Meus irmãos, cada um de vocês é uma joia rara. Um tesouro sem igual. Obrigado por estarem ao meu lado.

À CAPES e à PUC-Rio, pelos auxílios concedidos.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

## Resumo

Gusmão, Guilherme Ferreira; Hall Barbosa, Carlos Roberto; Barbosa Raposo, Alberto. **Desenvolvimento e Validação de Sensor LiDAR Virtual**. Rio de Janeiro, 2020. 104 p. Dissertação de Mestrado – Programa de Pós-Graduação em Metrologia. Área de concentração: Metrologia para Qualidade e Inovação, Pontifícia Universidade Católica do Rio de Janeiro.

As tecnologias de imageamento em três dimensões (3D) vêm tendo seu uso cada vez mais disseminado no meio acadêmico e no setor industrial, especialmente na forma de nuvens de pontos, uma representação matemática da geometria e superfície de um objeto ou área. No entanto, a obtenção desses dados pode ainda ser cara e demorada, reduzindo a eficiência de muitos procedimentos que são dependentes de um grande conjunto de nuvens de pontos, como a geração de *datasets* para treinamento de aprendizagem de máquina, cálculo de dossel florestal e inspeção submarina. Uma solução atualmente em voga é a criação de simuladores computacionais de sistemas de imageamento, realizando o escaneamento virtual de um cenário feito a partir de arquivos de objetos 3D. Este trabalho apresenta o desenvolvimento de um simulador de sistema LiDAR (*light detection and ranging*) baseado em algoritmos de rastreamento de raio com paralelismo (*GPU raytracing*), com o sensor virtual modelado por parâmetros metrológicos e calibrado por meio de comparação com um sensor real, juntamente com um gerador flexível de cenários virtuais. A combinação destas ferramentas no simulador resultou em uma geração robusta de nuvens de pontos sintéticas em cenários diversos, possibilitando a criação de datasets para uso em testes de conceitos, combinação de dados reais e virtuais, entre outras aplicações.

## Palavras-chave

Metrologia – teses; metrologia para qualidade e inovação – teses; sensor virtual; LiDAR; nuvem de pontos sintética; sensoriamento remoto; rastreamento de raio.

## Abstract

Gusmão, Guilherme Ferreira; Hall Barbosa, Carlos Roberto (Advisor); Barbosa Raposo, Alberto (Co-advisor). **Development and Validation of a LiDAR Virtual Sensor**. Rio de Janeiro, 2020. 104 p. Dissertação de Mestrado – Programa de Pós-Graduação em Metrologia. Área de concentração: Metrologia para Qualidade e Inovação, Pontifícia Universidade Católica do Rio de Janeiro.

Three dimensional (3D) imaging technologies have been increasingly used in academia and in the industrial sector, especially in the form of point clouds, a mathematical representation of the geometry and surface of an object or area. However, obtaining this data can still be expensive and time consuming, reducing the efficiency of many procedures dependent on a large set of point clouds, such as the generation of datasets for machine learning training, forest canopy calculation and subsea survey. A trending solution is the development of computer simulators for imaging systems, performing the virtual scanning of a scenario made from 3D object files. At the end of this process, synthetic point clouds are obtained. This work presents the development of a LiDAR system simulator (light detection and ranging) based on parallel ray tracing algorithms (GPU raytracing), with its virtual sensor modeled by metrological parameters. A way of calibrating the sensor is displayed, by comparing it with the measurements of a real LiDAR sensor, in addition to surveying error models to increase the realism of the virtual scan. A flexible scenario creator was also implemented to facilitate interaction with the user. The combination of these tools in the simulator resulted in a robust generation of synthetic point clouds in different scenarios, enabling the creation of datasets for use in concept tests, combining real and virtual data, among other applications.

## Keywords

Metrology – thesis; metrology for quality and innovation - thesis; virtual sensor; LiDAR; synthetic point cloud; remote sensing; ray tracing.



# Sumário

1. Introdução	20
1.1 Contextualização	20
1.2 Motivação	21
1.3 Objetivos: geral e específicos	23
1.4 Estrutura da dissertação	24
2. Técnicas de imageamento 3D	26
2.1. Métodos e técnicas de imageamento 3D para geração de nuvem de pontos	26
2.1.1. Métodos Ópticos Passivos	27
2.1.2. Métodos Ópticos Ativos	30
2.2. LiDAR	34
2.2.1. Componentes e Características	35
2.2.2. Princípio de Funcionamento	37
2.2.3. Nuvem de Pontos	37
2.2.4. Aplicações	38
3. Fundamentos de escaneamento virtual e ferramentas de software	40
3.1. Escaneamento Virtual	40
3.2. Estado da Técnica	46
3.3. Ferramentas para Desenvolvimento do Simulador	48
3.3.1. Rastreamento de Raios e API Nvidia OptiX	48
3.3.2. Unity 3D	50
4. Desenvolvimento e Validação do Simulador de Sistema LiDAR	52
4.1. Simulador	52
4.1.1. Cenário Virtual	54
4.1.2. Sensor Virtual	59
4.1.3. Gerador de Nuvem de Pontos Sintética	63
4.2. Validação	66

5. Aplicações em Cenários Típicos	73
5.1. Especificações das Máquinas para Simulação:	73
5.2. Inspeção em Campo de Petróleo Submarino	73
5.3. Segmentação Semântica de Nuvem de Pontos para Deep Learning	77
5.4. Simulação para Escaneamento a Laser Aéreo (ALS)	81
6. Discussão, conclusões e trabalhos futuros	86
6.1. Discussão e Conclusões	86
6.2. Trabalhos Futuros	87
Referências bibliográficas	91
Anexo A – Artigo CBM 2019	99

## Lista de figuras

Figura 1 – Diagrama de Classificação das técnicas de imageamento 3D, com as técnicas de interesse para a pesquisa circuladas em verde. 27

Figura 2 – À esquerda, quatro fotos do *dataset* de uma estátua de querubim fornecida pela 3DFLOW. À direita, a nuvem de pontos densa gerada e uma ampliação do rosto, para melhor evidenciar a característica pontual da nuvem [27]. 28

Figura 3 – Princípio de funcionamento da visão estéreo binocular [28]. 29

Figura 4 – Princípio de medição da técnica de forma a partir da distância focal (SfF) [29]. 29

Figura 5 – Princípio de aquisição de imagem da técnica de forma a partir do movimento da câmera (SfM) [31]. 30

Figura 6 – O interferômetro de Michelson. O feixe laser inicial é dividido em dois feixes e depois recombinado. Este processo altera a fase dos feixes que, ao serem superpostos, interferem de forma construtiva ou destrutiva [32]. 31

Figura 7 – Princípio básico de luz estruturada [33]. 32

Figura 8 – Método de Triangulação. PSD é um sensor sensível de posição [34]. 32

Figura 9 – Método de deslocamento de fase.  $(V_i)_r$  é o sinal de referência,  $(V_i)_f$  é o sinal de medição e  $V_e$  são os pulsos contados. A relação do gráfico linear inferior é entre distância e o número de pulsos contados em  $V_e$  [35]. 33

Figura 10 – Método de Tempo de Voo. Um feixe luminoso é disparado e o dispositivo sensor obtém a distância até um alvo usando o tempo de ida e retorno do feixe a sua origem [36]. 34

Figura 11 – a) O sensor LiDAR HDL-64E da Velodyne [37], e b) o sensor LiDAR SL3 da 3D at Depth [38]. 35

Figura 12 – Nuvem de pontos com informação adicional de cor (Abadia de Sint-Baafs, Gent, Bélgica) [39]. 38

Figura 13 – a) Uso do Velodyne Alpha Puck para obter informações do ambiente ao redor do carro autônomo [37]. b) Observação de nuvens de pontos da cobertura florestal da Amazônia para avaliar o impacto da seca provocada pelo El Niño [40]. c) Nuvem de pontos da Acropolis de Copan

- em Honduras [32]. d) Nuvem de pontos de uma inspeção submarina de um poço submarino pela 3D at Depth [38]. 39
- Figura 14 – Nuvem de pontos de LiDAR de Washington, DC [42]. 41
- Figura 15 – Escaneamento da aeronave submersa TBD-1 Devastator da época da segunda guerra mundial [43]. 41
- Figura 16 – a) Um cenário virtual para simulação de imageamento 3D. b) A nuvem de pontos sintética gerada pelo sensor virtual a partir de uma posição de escaneamento [17]. 42
- Figura 17 – No topo, um carro de F1 da Ferrari [46]. Abaixo, o modelo 3D do carro (disponível nos formatos FBX, OBJ ou 3DS) [47]. 44
- Figura 18 – Comparação visual entre uma nuvem de pontos sintética (a) e uma nuvem de pontos real (b) de um alvo e sua contraparte virtual [1]. 46
- Figura 19 – Visualização da geometria dos raios disparados pelo sensor real e virtual [2]. 46
- Figura 20 – Princípio de funcionamento do algoritmo de raytracing [36]. 49
- Figura 21 – Execução de um programa em CUDA C. O host code chama os kernels, funções de execução em paralelo com device code, para serem executados. Este processo continua até o fim da execução do programa [51]. 50
- Figura 22 – Diferentes aplicações feitas com a plataforma Unity3D. a) Uma aplicação em realidade virtual (VR) para treinamento na montagem de veículos para Volkswagen. b) Criação de jogos fotorrealistas. c) Desenvolvimento de animações em 2D e 3D, como o Giant Bear da F.D. Films|Taqqut Productions. d) Visualização de BIM (Building Information Model) em 3D de plantas baixas usando realidade aumentada (AR) [52]. 51
- Figura 23 – Fluxograma de operação: paralelogramos são arquivos de dados ( ), retângulos são módulos ( ) e losangos são aplicações ( ). 54
- Figura 24 – Cenário simples construído com o editor do módulo Cenário Virtual. A cruzeta no topo indica a direção dos eixos. 56
- Figura 25 – a) Parâmetro posição. b) Parâmetro rotação. 57
- Figura 26 – a) Parâmetro escala; b) Parâmetro número de identificação (Id). 58
- Figura 27 – Posicionamento do sensor virtual no cenário. À direita, a previsão das imagens observadas pelos sensores. 58

Figura 28 – a) Técnica de tempo de voo usado em sensor LiDAR. b) Fundamento do algoritmo de raytracing [36].	59
Figura 29 – a) Plano cartesiano. b) Plano esférico [2].	60
Figura 30 – a) Pilotis do prédio Leme, PUC-Rio. b) Piloti escaneado no experimento.	62
Figura 31 – Curva de ajuste polinomial de grau 2, para obtenção da equação de erro.	62
Figura 32 – Nuvem de pontos sintética gerada para cada posição do sensor virtual definido pelo arquivo de caminho do sensor.	64
Figura 33 – Diferentes visualizações de nuvem de pontos. a) visualização de pontos. b) Visualização da ação dos vetores normais. c) Visualização de cores. d) Visualização de identificadores com vetores normais em uso para facilitar visualização.	65
Figura 34 – Uma série de escaneamentos virtuais são realizados gerando um grande volume de nuvens de pontos de uma única face. Em seguida, esse dataset é unificado em uma nuvem de pontos sintética combinada.	66
Figura 35 – Hokuyo URG-04LX-UG01 [56]	67
Figura 36 – Mapa de distância a 700 mm: Pontos de intercessão do LiDAR ('o') e pontos de intercessão do sensor virtual ('x')	69
Figura 37 – Mapa de distância a 1500 mm: Pontos de intercessão do LiDAR ('o') e pontos de intercessão do sensor virtual ('x').	69
Figura 38 – Diferença entre dados reais e sintéticos. Linha contínua, erro na distância do ângulo central a 700 mm. Linha tracejada, erro na distância do ângulo central a 1500 mm.	70
Figura 39 – Mapa de distância a 700 mm: Pontos de intercessão do LiDAR ('o') e pontos de intercessão do sensor virtual de modelo híbrido ('x').	71
Figura 40 – Mapa de distância a 1500 mm: Pontos de intercessão do LiDAR ('o') e pontos de intercessão do sensor virtual de modelo híbrido ('x').	71
Figura 41 – Diferença entre dados reais e sintéticos gerados com o modelo híbrido. Linha contínua, erro na distância do ângulo central à 700 mm. Linha tracejada, erro na distância do ângulo central à 1500 mm	72
Figura 42 – Exemplo de imagens obtidas de equipamentos submarinos em uso [58,59].	74

Figura 43 – Nuvem de pontos de equipamentos submarinos em um campo de petróleo offshore [38].	74
Figura 44 – (a) Árvore de Natal Molhada [60]. (b) Modelo 3D de uma ANM.	75
Figura 45 – Cenário Virtual de um campo de petróleo em diferentes ângulos e iluminação.	76
Figura 46 – Nuvem de pontos do Manifold.	76
Figura 47 – Nuvens de pontos das ANMs.	77
Figura 48 – Nuvem de pontos do campo de petróleo virtual com informações de classificação.	77
Figura 49 – Segmentação semântica da classe “carro” em uma imagem [61]. Todos os objetos identificados como pertencente a esta classe estão em destaque.	78
Figura 50 – Processo de segmentação semântica de nuvens de pontos feita pelo sistema PointNet [53]. A entrada é constituída por nuvens de pontos sem classificação e a saída são nuvens de pontos com segmentação. As diferentes cores indicam diferentes classes de objetos.	79
Figura 51 – Cenário virtual de uma sala de estar. Os quadros exibem quatro posições de um total de dezoito para o sensor virtual. Todos os modelos 3D foram obtidos de [62].	80
Figura 52 – Nuvem de pontos sintética gerada com segmentação semântica vistas em diferentes ângulos no visualizador Cloud Compare [63]. Classes por cores: Vermelho – Estante, Amarelo – Sofá, Verde escuro – Mesa, Verde claro – Cadeira, Azul – Monitor/TV, Cinza – Parede e Laranja – Chão.	80
Figura 53 – Princípio básico de ALS. GPS e IMU ajudam a georreferenciar os dados [64].	81
Figura 54 – Modelo 3D de um mapeamento aéreo com drone.	82
Figura 55 – Zona de interesse para escaneamento virtual.	83
Figura 56 – Vista aérea da nuvem de pontos sintética exibindo informação de cores e topografia.	84
Figura 57 – Vista frontal da nuvem de pontos sintética exibindo informação de cores e topografia.	84
Figura 58 – Propriedades Ópticas. (a) Reflexão [68]. (b) Espalhamento [69]. (c) Absorção [67].	89

## Lista de siglas e abreviaturas

2D	2-Dimensional
3D	3-Dimensional
AD	Analog-to-Digital
ALS	Airborne Laser Scanning
ANM	Árvore de Natal Molhada (equipamento submarino)
API	Application Programming Interface
AR	Augmented Reality
BIM	Building Information Model
BMW	Bayerische Motoren Werke AG
c	Velocidade da Luz
CAD	Computer Aided Design
CARLA	Car Learning to Act
CCD	Charge-Coupled Device
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
GPS	Global Positioning System
GPU	Graphics Processing Unit
GTA-V	Grand Theft Auto V
IA	Inteligência Artificial
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging
NASA	National Aeronautics and Space Administration (USA)
PMT	Photomultiplier Tube
PósMQI	Programa de Pós-graduação em Metrologia
PUC-Rio	Pontifícia Universidade Católica do Rio de Janeiro
RMSE	Root Mean Square Error
SfF	Shape from Focus
SfM	Shape from Motion

SPD	Single Photon Detector
t	Tempo
TD	Time-to-Digital
ToF	Time of Flight
TUM	Technische Universität München (Technical University of Munich)
VR	Virtual Reality
VTD	Virtual Test Drive
$\lambda$	Comprimento de onda



*Estas coisas vos tenho dito para  
que tenhais paz em mim. No mundo, passais por aflições, mas tende bom ânimo;  
eu venci o mundo.*  
João 16:33

# 1

## Introdução

Os avanços das tecnologias de escaneamento e processamento de imagens vêm permitindo, por meio de dispositivos sensores e câmeras especializadas, coletar dados massivos de um objeto ou região alvo e processá-los no computador, para gerar modelos e mapas com ótimo nível de resolução. No entanto, o processo pode ser difícil, demorado e de elevado custo, inviabilizando ou atrasando procedimentos que necessitem de uma quantidade enorme dessas informações, como, por exemplo, cálculo de dossel florestal e geração de conjunto de dados (*dataset*) para aprendizagem profunda (*deep learning*). Uma das soluções em voga é o escaneamento virtual, no qual modelos 3D, parciais ou totais, do objeto ou área de interesse são usados em simulações computacionais para obter dados verossimilhantes aos reais [1–6].

Este capítulo inicial tem como objetivo contextualizar as motivações por trás do desenvolvimento de simuladores para escaneamento virtual, assim como exibir os objetivos e estrutura desta dissertação.

### 1.1

#### Contextualização

O ser humano é uma criatura altamente visual, que sempre buscou registrar o mundo e até o seu próprio âmag, por meio de desenhos e imagens. Artistas e estudiosos ao longo da história criaram mapas, pinturas, esculturas, o design de peças, écorchers anatômicos, entre outros. Obras como estas permitiram o estudo, bem como a obtenção de informações relevantes para diversas áreas da sociedade.

A própria ideia de imageamento para cálculo de distância possui anos de existência. No artigo de François Blais, o autor comenta que o princípio de medição por triangulação foi utilizado pelos gregos para navegação e astronomia [7]. Com os avanços da optoeletrônica e o surgimento de microcomputadores, entre os anos

de 1970 e 1980, foi possível o surgimento de sensores com custo competitivo para aplicação industrial [7].

Atualmente, a tecnologia de escaneamento e processamento de imagens é capaz, por meio do uso de sensores, câmeras especializadas, independente do uso de veículos operados tanto remotamente como automatizados, de coletar dados de um objeto ou região alvo e processá-los no computador para gerar modelos e mapas com um ótimo nível de precisão, com uma resolução capaz de chegar à ordem de milímetros [8–11].

Estes mapas são, normalmente, conjuntos de pontos em um mesmo sistema de coordenadas, passíveis de obtenção de informações geo-espaciais/métricas. Devido a aparência da disposição dos dados, este foi denominado “nuvem de pontos” (*point cloud*). Possuem aplicações nas mais diversas áreas, a saber: geografia, geologia, oceanografia, engenharia florestal, geomorfologia, automobilística, militar etc.

Dentre os sistemas de imageamento 3D, o dispositivo LiDAR (*Light Detection and Ranging*) vem recebendo cada vez mais atenção, tanto do setor industrial quanto do meio acadêmico. Sua precisão e velocidade de aquisição de dados, somadas à capacidade de embarcá-lo em veículos não-tripulados ou remotos (caso dos drones), vêm permitindo uma gama de aplicações interessantes como batimetria e inspeção submarina [12–14].

## 1.2 Motivação

Os dispositivos de imageamento 3D ainda possuem alto custo, além de serem expostos, com frequência, às intempéries do ambiente ou riscos de dano. A necessidade de cuidado no manuseio, em conjunto com as grandes extensões da área de interesse, torna o processo demorado, podendo demandar dias para um escaneamento completo [2,5,6,15].

Outros procedimentos podem trazer problemas de segurança para os envolvidos nos experimentos. Por exemplo, o treinamento de *deep learning* para carros autônomos necessita ser capaz de reconhecer obstáculos, principalmente pessoas e outros veículos. Realizar esses experimentos na prática seria algo muito arriscado [2,16].

O escaneamento virtual vem como solução para estes problemas. Ele consiste na simulação de um sensor virtual que interage com um cenário virtual 3D, resultando em uma nuvem de pontos artificial.

O cenário simulado pode ser feito com uso de modelos CAD ou modelos feitos a partir de nuvens de pontos reais. Como, em um simulador, o ambiente e situações podem ser controlados, a aquisição de *datasets* se torna segura e tão rápida quanto o processamento dos computadores permitir [2,6]. Consequentemente, o custo de operação é reduzido.

Além disso, esses simuladores podem ser úteis na educação. Como mencionado no artigo de Bharat Lohani e R K Mishra, o simulador permitiria a geração de dados em diversas situações diferentes dentro de um laboratório, permitindo melhor entendimento das tecnologias de imageamento pelos estudantes [15].

A literatura indica que o sensor virtual baseado no dispositivo LiDAR tem sua modelagem usualmente feita com algoritmos de rastreamento de raios (*ray tracing*) [2 – 6,14,15,17,18]. Com a recente inovação nas tecnologias de computação gráfica, o uso de algoritmos de rastreamento de raios processados em paralelo em GPU cresce a cada dia, devido a sua rapidez de execução.

Como exemplo, Timo Hanke et al. produziram um simulador baseado em LiDAR, com uso da API da NVIDIA de *ray tracing*, chamada Optix [19], para uso no simulador de direção VTD da Vires, produzindo *datasets* para treinamento de carros autônomos [2]. Um método similar foi desenvolvido por Wang F et al., onde nuvens de pontos sintéticas com informações de identificação dos objetos detectados foram geradas no simulador de direção autônoma CARLA e depois experimentos foram feitos mostrando sua efetividade no uso de treinamento do método de *deep learning* [20].

Além da área automobilística, setores de meio ambiente, robótica, mapeamento e reconhecimento parecem ter grande interesse pelo assunto [3–6]. Todavia, a literatura científica indica a necessidade de mais realismo nessas simulações, aumentando a qualidade dos dados gerados [1,2,6]. Para garantir a qualidade, os ruídos, erros e distorções de informação do escaneamento real precisam ser considerados também no mundo virtual. Estes ruídos vêm sendo configurados de acordo com a área de foco. Uma maneira possível seria considerar a influência do ambiente sobre o sensor virtual, como partículas suspensas ou condições climáticas

adversas, assim como os erros de medição do equipamento devido às suas características de construção.

A lacuna que a presente pesquisa de mestrado busca preencher, dando uma contribuição ao referido tema, é o desenvolvimento de um simulador modular, mais genérico na sua aplicação, e a criação de modelos de erros robustos para o sensor virtual.

### 1.3 **Objetivos: geral e específicos**

A pesquisa de mestrado em Metrologia aqui apresentada tem como objetivo geral desenvolver e validar um sensor virtual baseado em LiDAR capaz de simular as imagens geradas por um LiDAR real, levando em consideração as influências das características físicas do sensor, assim como os erros percebidos nas medições sobre o escaneamento.

Para alcançar este objetivo geral, definem-se os seguintes objetivos específicos:

- Caracterizar o estado da arte sobre modelos de sensores virtuais baseados em sensores de imageamento 3D;
- Caracterizar os principais métodos e técnicas de imageamento 3D aplicadas em engenharia para geração de nuvens de pontos, em particular no setor de óleo & gás;
- Caracterizar a tecnologia de escaneamento laser LiDAR, descrevendo o instrumento de medição, seus princípios físicos, os dados de saída (nuvem de pontos) e as principais grandezas que afetam a qualidade da imagem;
- Investigar as ferramentas, técnicas e plataformas de software disponíveis para o desenvolvimento de um sensor virtual baseado em LiDAR;
- Desenvolver e implementar o sensor virtual na plataforma computacional selecionada, modelando o ruído ambiente introduzido em imagens LiDAR e incluindo uma parametrização metrológica que permita simular sensores de dispositivos LiDAR comerciais reais;
- Realizar ensaios de bancada com dispositivos LiDAR reais já disponíveis na PUC-Rio;
- Validar o sensor virtual LiDAR com base nos ensaios de bancada;

- Aplicar o sensor virtual LiDAR já validado à simulação de cenários reais e aplicações típicas; e
- Explicitar os diferenciais do modelo proposto em relação aos modelos que vêm sendo adotados no mundo para simulação em ambiente virtual.

## 1.4 Estrutura da dissertação

A presente dissertação está estruturada em seis capítulos, que abordam temas relacionados ao escaneamento virtual com uso de sensor virtual baseado em LiDAR, fornecendo ao leitor uma abordagem que identifica e contextualiza lacunas da literatura científica especializada, contribuindo para a melhoria da obtenção da nuvem de pontos artificial por meio de dados adquiridos no decorrer desta pesquisa.

Este Capítulo 1 refere-se à introdução da dissertação, na qual busca-se estabelecer uma relação do leitor com o tema proposto, contextualizando os equipamentos de imageamento 3D, escaneamento virtual e sua utilização nas mais diversas áreas. Mostra a evolução da tecnologia ao longo dos anos e identifica as técnicas em voga para escaneamento virtual.

O Capítulo 2 descreve os principais métodos e técnicas de imageamento 3D, para a geração de nuvem de pontos, com destaque para o dispositivo LiDAR, apresentando seu princípio de funcionamento e levantando suas características metrológicas.

O Capítulo 3 justifica e conceitua a técnica de escaneamento virtual. Na sequência, investigam-se as ferramentas, técnicas e plataformas de software disponíveis para o desenvolvimento do simulador baseado em LiDAR, concluindo com a caracterização da plataforma de desenvolvimento escolhida.

O Capítulo 4 apresenta e descreve o simulador desenvolvido e suas ferramentas, assim como a validação por comparação com um sensor LiDAR real.

O Capítulo 5 mostra o uso do simulador em algumas aplicações típicas descritas na literatura.

Finalmente, o Capítulo 6 apresenta as considerações finais e a sugestão de trabalhos futuros.

No Anexo A apresenta-se um artigo científico apresentado no 10º Congresso Brasileiro de Metrologia, realizado em Florianópolis em 2019.

## 2

### Técnicas de imageamento 3D

Dispositivos de imageamento 3D têm como um dos seus principais usos a geração de modelos tridimensionais de um alvo de interesse com alto nível de resolução, sendo uma dessas representações a nuvem de pontos [21]. Este capítulo visa caracterizar os métodos e técnicas ópticas evidenciadas no estado da arte como propícias para geração deste tipo de dados. Além disso, a última seção trata especificamente do sensor LiDAR, cujas características e funcionamento são fundamentais para este trabalho.

#### 2.1.

#### **Métodos e técnicas de imageamento 3D para geração de nuvem de pontos**

Como descrito no Capítulo 1 desta dissertação, nuvens de pontos são conjuntos de pontos em um mesmo sistema de coordenadas passíveis de obtenção de informações geo-espaciais/métricas obtidos do processo de imageamento 3D de um determinado objeto ou área de interesse. Para tal, são utilizados dispositivos sensores ou câmeras que capturam as propriedades geométricas, como forma e curvatura, e/ou fotométricas, como cor e reflexividade do material, destes alvos. A saída desses equipamentos fornece os dados brutos necessários para que algoritmos computacionais obtenham as informações métricas e espaciais, para uma geração coerente das nuvens de pontos [2,5,6,21,22].

A literatura indica um grande número de técnicas de imageamento 3D [12,21–26], dividindo-as em categorias, como exibido na Figura 1. Para os fins deste trabalho, o interesse são as técnicas de imageamento 3D baseadas em princípios ópticos para geração de nuvem de pontos, destacadas em verde na Figura 1. A subseção seguinte trata das técnicas reflexivas ópticas mais proeminentes na literatura, que estão divididas entre métodos passivos e ativos [26].



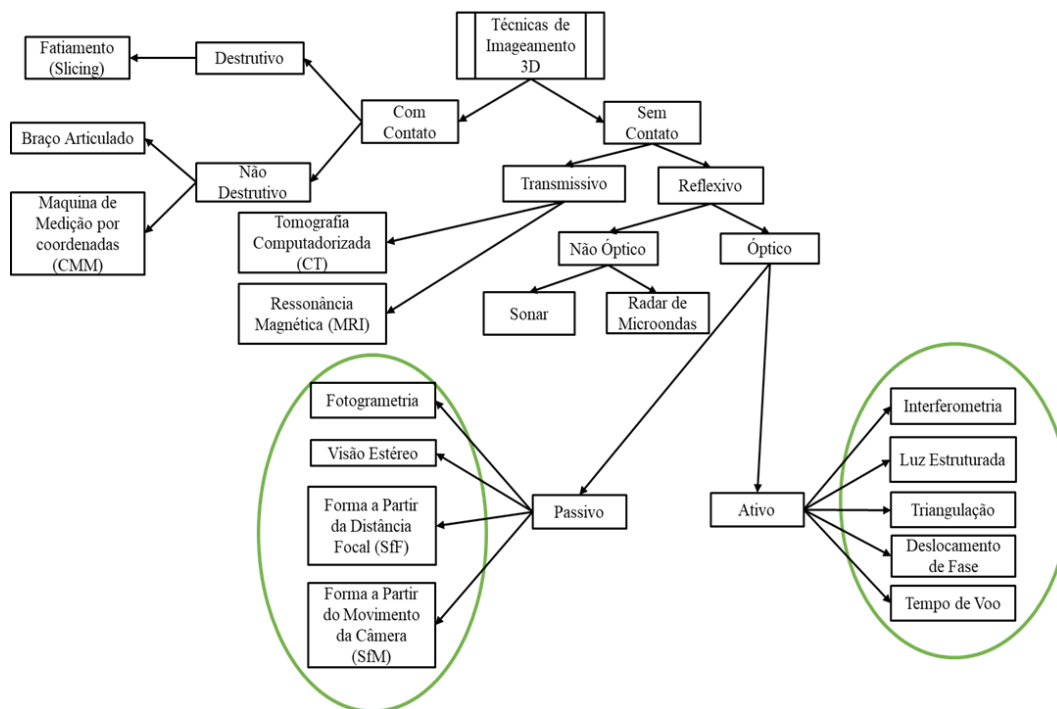


Figura 1 – Diagrama de Classificação das técnicas de imageamento 3D, com as técnicas de interesse para a pesquisa circuladas em verde.

### 2.1.1. Métodos Ópticos Passivos

São classificados como métodos passivos de imageamento 3D aqueles que dependem apenas da iluminação do ambiente e das características ópticas dos objetos para obter os dados geométricos e espaciais dos alvos de interesse [23–26].

Destacam-se:

#### A) Fotogrametria:

A fotogrametria utiliza fotos ou vídeos feitos com câmeras fotográficas especializadas, ou mesmo simples, configuradas de forma a obter a melhor iluminação e nitidez dos alvos de interesse. Estes *datasets* são processados por algoritmos computacionais que estimam as coordenadas em um sistema métrico [22].

O típico *pipeline* (segmentação de instruções), apresentado em [22], segue a seguinte ordem: calibração do sensor/câmera, orientação da imagem, medição da superfície, extração das características e geração de ortofoto. Na etapa de “medição da superfície” já existe informação suficiente para construção de uma nuvem de pontos. Na Figura 2, encontram-se algumas fotos do *dataset* fornecido pela 3DFLOW, para uso

no seu software de fotogrametria 3DF Zephyr e a nuvem de pontos densa produzida [27].



Figura 2 – À esquerda, quatro fotos do *dataset* de uma estátua de querubim fornecida pela 3DFLOW. À direita, a nuvem de pontos densa gerada e uma ampliação do rosto, para melhor evidenciar a característica pontual da nuvem [27].

## B) Visão Estéreo:

A técnica consiste no uso de duas ou mais câmeras para obtenção de informações geométricas e da superfície de um alvo comum [23-26]. A reconstrução do modelo tridimensional, de acordo com [26], segue os seguintes passos: aquisição das imagens, modelagem da câmera, extração das características da superfície dos alvos, análise para correspondência dos pontos e triangulação das imagens projetadas pelas câmeras. A Figura 3 exhibe o princípio de funcionamento da visão estereo.

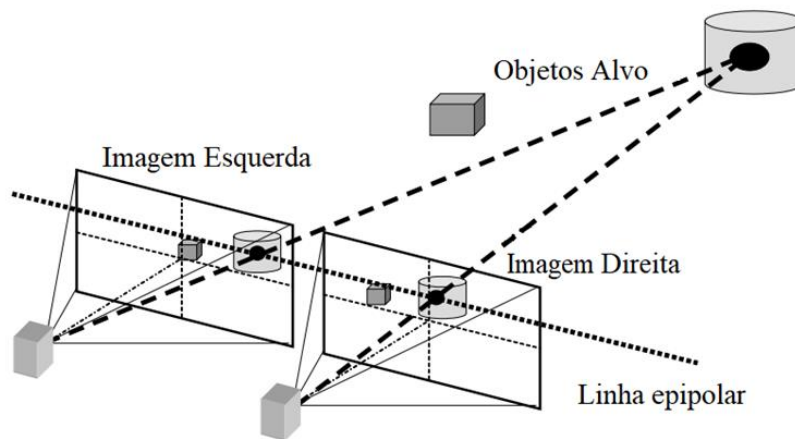


Figura 3 – Princípio de funcionamento da visão estéreo binocular [28].

As principais vantagens são a simplicidade e baixo custo da técnica. No entanto, existe uma dificuldade para identificação de pontos comuns nas imagens pares [26].

### C) Shape from Focus (SfF):

Esta técnica utiliza-se do fenômeno de profundidade de campo de visão para realizar medições de distância por meio das propriedades focais das lentes de uma câmera. A imagem alvo é desfocada proporcionalmente à distância entre os pontos dessa imagem e ao plano do objeto em foco [26]. O princípio de medição é apresentado na Figura 4.

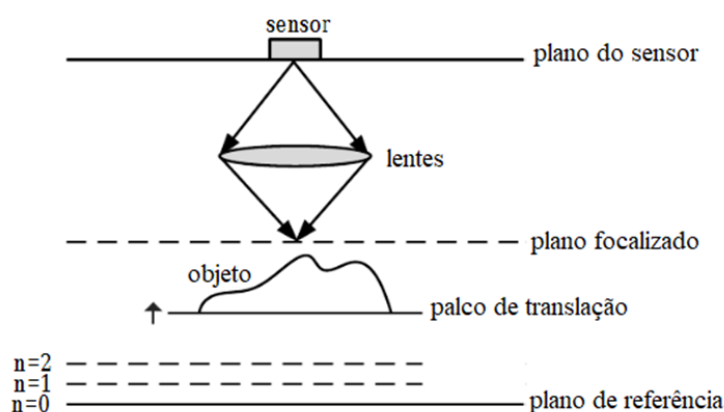


Figura 4 – Princípio de medição da técnica de forma a partir da distância focal (SfF) [29].

#### D) Shape from Motion (SfM):

Forma a partir do movimento da câmera é uma técnica fundamentada na estimação do movimento da câmera com o uso de imagens sobrepostas para realizar a reconstrução 3D do alvo de interesse. A reconstrução é feita de forma automática usando detalhes comuns nas imagens capturadas, como mostrado na Figura 5 [30,31]. No entanto, devido às características do processo, existe uma falta de informações sobre escala e orientação, que estão presentes em outras técnicas, como a fotogrametria [31].

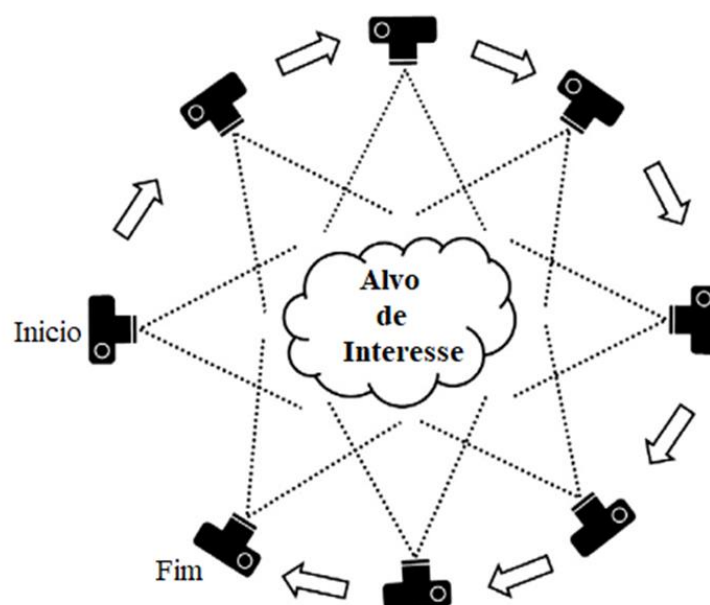


Figura 5 – Princípio de aquisição de imagem da técnica de forma a partir do movimento da câmera (SfM) [31].

#### 2.1.2. Métodos Ópticos Ativos

São classificados como métodos ativos as técnicas de imageamento 3D que utilizam uma fonte luminosa, por exemplo um laser, como forma de obter informações dos objetos de interesse [23–26].

##### A) Interferometria:

A técnica consiste na superposição ou interferência de um sinal padrão de referência com a luz refletida de um padrão luminoso

projetado sobre uma superfície. A geometria da superfície é obtida da demodulação do sinal de referência [26]. A Figura 6 exibe o interferômetro de Michelson, baseado na interferência de dois raios luminosos [32].

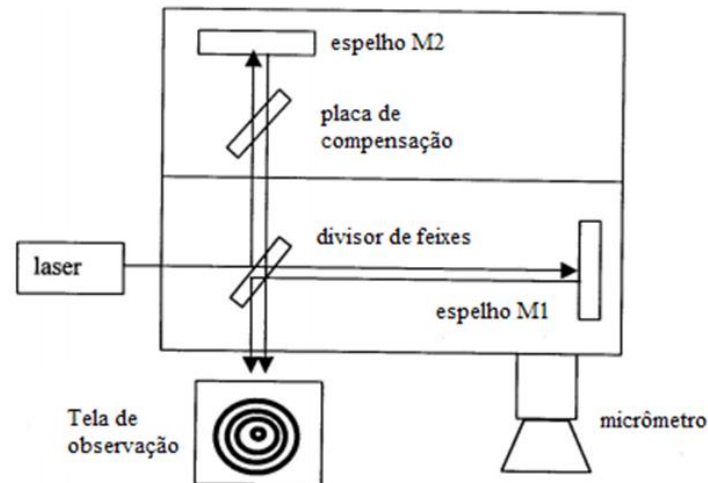


Figura 6 – O interferômetro de Michelson. O feixe laser inicial é dividido em dois feixes e depois recombinado. Este processo altera a fase dos feixes que, ao serem superpostos, interferem de forma construtiva ou destrutiva [32].

### B) Luz Estruturada:

No caso da técnica de luz estruturada, a geometria e características da superfície escaneada são capturadas por meio do uso de padrões luminosos e câmeras. O padrão luminoso é distorcido ao ser projetado sobre o objeto de interesse. São essas variações no padrão luminoso sobre a superfície que são analisadas nas imagens capturadas para a reconstrução 3D [26,33]. O princípio é mostrado na Figura 7.

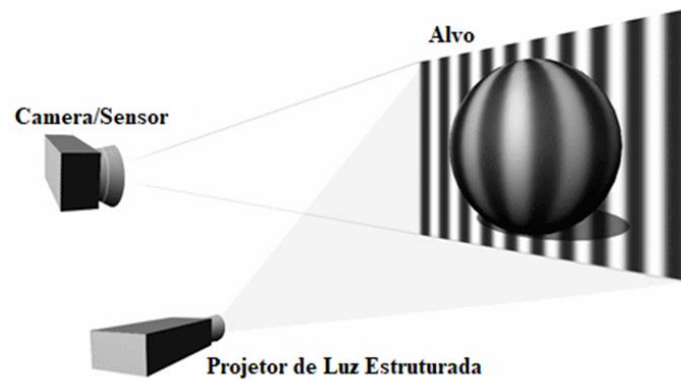


Figura 7 – Princípio básico de luz estruturada [33].

### C) Triangulação:

A triangulação é método de comparação entre medidas na qual, dados dois ou mais dispositivos sensores/câmeras observando um alvo comum, pode-se obter a posição do mesmo pela interseção dos raios da projeção da imagem. Estes raios formam um triângulo entre o espaço amostral e os dispositivos de escaneamento que são a base do método, como visto na Figura 8 [12,23].

Devido à grande paralaxe existente no método, a qualidade do resultado é tanto melhor quanto mais próximos os dispositivos sensores estiverem do alvo e quanto mais distantes os dispositivos sensores estiverem afastados entre si [12].

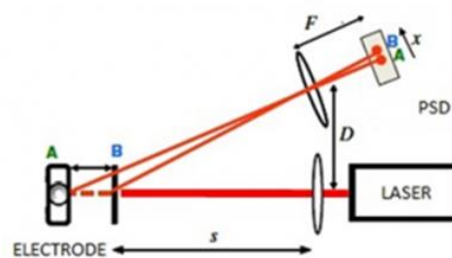


Figura 8 – Método de Triangulação. PSD é um sensor sensível de posição [34].

### D) Deslocamento de Fases:

Neste método, a distância do feixe luminoso é calculada a partir da diferença de fase de dois sinais, um de medição e outro de referência

[35]. Uma das formas de demodulação frequentemente usadas nesse método é a contagem de tempo, onde o intervalo de tempo entre os sinais é contado e a distância calculada em relação aos comprimentos dos pulsos, como visto na Figura 9.

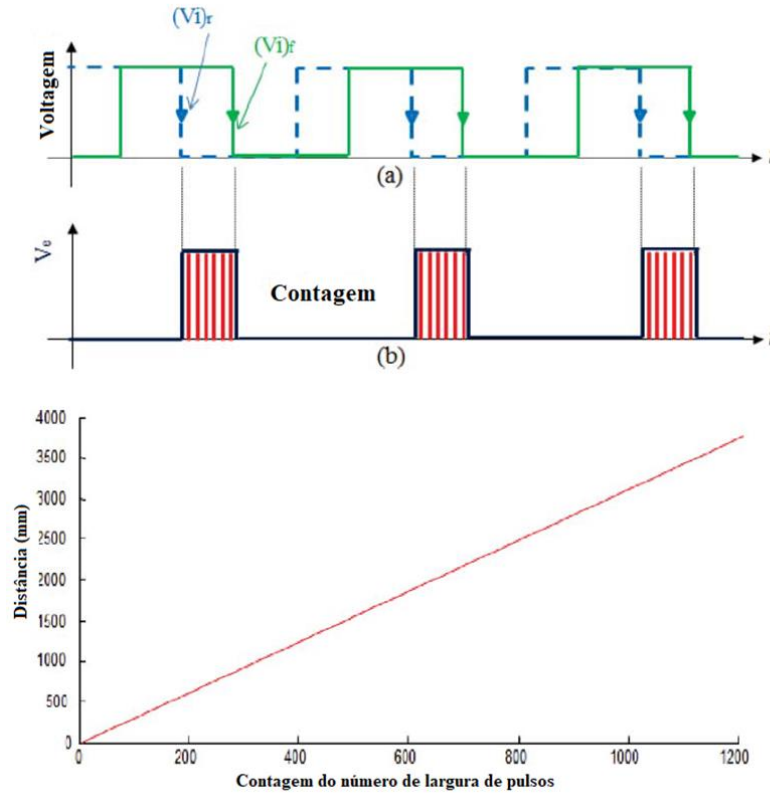


Figura 9 – Método de deslocamento de fase.  $(Vi)_r$  é o sinal de referência,  $(Vi)_f$  é o sinal de medição e  $Ve$  são os pulsos contados. A relação do gráfico linear inferior é entre distância e o número de pulsos contados em  $Ve$  [35].

### E) Tempo de Voo:

O tempo de voo de um sinal é definido como o tempo total de ida e retorno para o dispositivo de origem. Dada a velocidade do sinal no meio de propagação pode-se calcular a distância percorrida até a colisão do feixe com alguma superfície, como mostrado na Figura 10 [2,3,28]. Como a velocidade da luz é conhecida e praticamente constante, a distância pode ser calculada por [7–10]

$$d = \frac{ct}{2}, \quad (1)$$

onde

$c$  é a velocidade da luz ( $299792458 \frac{m}{s}$ )

$t$  é o tempo de voo (*time of flight* – ToF)

Quando o alvo está muito próximo do dispositivo sensor ( $d < 1$  m), a baixa precisão na medição do tempo pode provocar um erro relativo muito grande [7–10].

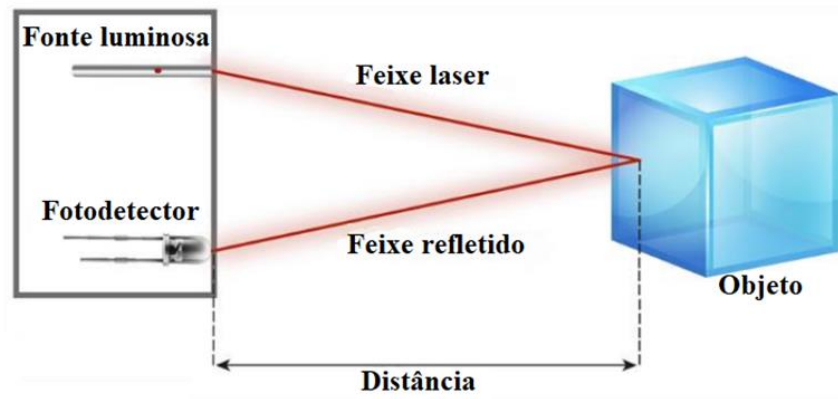


Figura 10 – Método de Tempo de Voo. Um feixe luminoso é disparado e o dispositivo sensor obtém a distância até um alvo usando o tempo de ida e retorno do feixe a sua origem [36].

## 2.2. LiDAR

Nesta dissertação de mestrado, a tecnologia de imageamento 3D de interesse foi o sistema LiDAR, um sistema de imageamento 3D capaz de obter medições de distâncias, que após o processamento dos dados por um computador, são organizadas em um conjunto de dados pontuais, com ou sem georreferências, chamado de nuvem de pontos. A Figura 11 apresenta alguns sensores de sistema LiDAR.



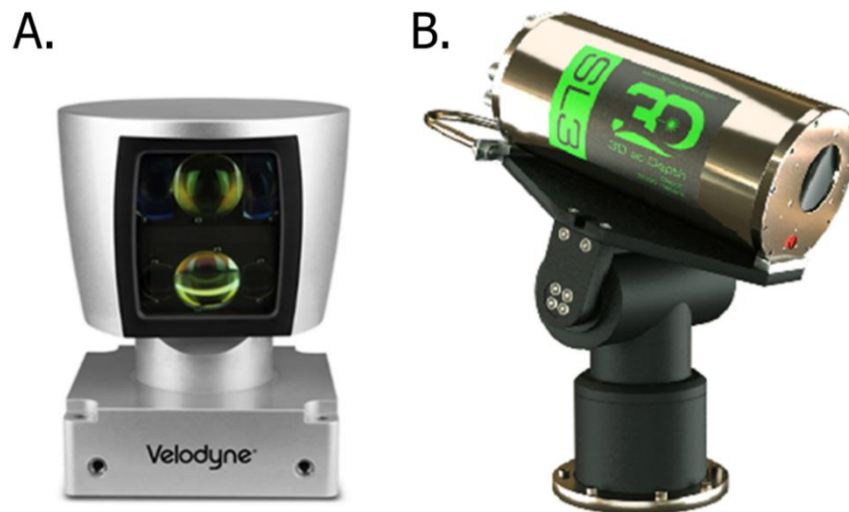


Figura 11 – a) O sensor LiDAR HDL-64E da Velodyne [37], e b) o sensor LiDAR SL3 da 3D at Depth [38].

A escolha do sistema LiDAR para este trabalho será melhor evidenciada ao final desta seção, onde são dados exemplos de aplicação deste dispositivo, e no Capítulo 3, onde encontram-se informações sobre o software e técnicas utilizadas para geração de nuvens de pontos sintéticas, por meio de simulações computacionais.

### 2.2.1. Componentes e Características

O sistema LiDAR básico, normalmente, é composto pelos seguintes componentes: unidade receptora GPS, fonte luminosa do tipo laser, fotorreceptor e computador [7–11].

A unidade GPS (Global Positioning System) registra a localização exata no plano cartesiano, empregando um marco zero de referência para maior exatidão. Muitas aplicações usam, em conjunto com o GPS, uma unidade de medição inercial (IMU) para obter georreferências nos pontos [7–11].

A fonte laser produz os pulsos empregados no escaneamento do alvo, sendo assim o LiDAR é um transdutor ativo [7–11]. Usualmente, este laser é refletido por um espelho em movimento. Na maioria das aplicações o laser possui sua faixa espectral próxima ao infravermelho, porém para batimetria e escaneamento submarino são usados lasers na faixa espectral do azul ao verde. Na literatura, o

laser verde com comprimento de onda 532 nm é o mais habitual em ambientes submarinos [12].

O fotorreceptor pode ser um fotomultiplicador (PMT), detectores de fótons (SPD), fotodiodos e até câmeras. O sistema pode ser construído com um conversor A/D (conversor analógico para digital) ou conversor T/D (conversor de tempo para digital) de alta velocidade [11,12].

Necessita-se de um computador confiável e com processadores de alto desempenho para processar e armazenar a quantidade massiva de dados adquiridos de forma coerente [7–11].

Quanto ao sensor LiDAR, estudando a literatura [1–12], pode-se ressaltar que suas principais características metrológicas de interesse são:

- I) Fundo de Escala: Faixa de valores onde o erro do instrumento de medição se mantém dentro dos limites estabelecidos;
- II) Distância de medição (Resolução): A menor diferença entre indicações que pode ser percebida pelo sistema;
- III) Campo de visão: A região angular capaz de ser observada/varrida pelo sensor;
- IV) Resolução da movimentação e rotação: O passo com o qual os sensores são movimentados e rotacionados;
- V) Método de medição: O LiDAR pode ser uma combinação de ToF com outros métodos de imageamento 3D, como deslocamento de fase;
- VI) Precisão do ângulo: O quanto o ângulo da fonte laser desvia da posição correta durante sua movimentação;
- VII) Divergência do raio: O quanto o ângulo de saída do disparo varia de acordo com o ambiente;
- VIII) Número máximo de pontos por varredura: Quantidade limite de pontos registrados;
- IX) Temperatura de operação: Faixa de temperatura de funcionamento ótimo do equipamento; e
- X) Classe do Laser: Para medidas de segurança quanto ao funcionamento do laser.

### **2.2.2. Princípio de Funcionamento**

O sistema LiDAR é baseado em ToF [7–11], ou seja, o sensor dispara um feixe de laser e mede o tempo de ida e retorno para o fotodetector. Como mencionado na seção 2.1.2, utilizando esse tempo de voo na equação (1), pode-se encontrar a distância entre o sensor e a superfície detectada. Os dados de saída brutos, juntamente com os ângulos vertical e horizontal do disparo do laser, servem de entrada para os algoritmos geométricos que irão calcular o ponto espacial 3D equivalente àquela distância e ângulos, assim como outras informações relevantes. Ao final, com todos os pontos calculados e posicionados de forma coerente no conjunto, tem-se a nuvem de pontos [21,22].

### **2.2.3. Nuvem de Pontos**

A saída de um sistema LiDAR é uma nuvem de pontos. Este conjunto de dados, que pode ter uma ordem massiva de pontos ( $10^6$ , tipicamente), é uma representação matemática em três dimensões da superfície escaneada, criando um mapa preciso do alvo.

Cada ponto na nuvem é posicionado de acordo com seu vetor de posição 3D (coordenadas  $x$ ,  $y$ ,  $z$ ). Além dessa informação, os pontos podem ter outras características vinculadas como vetor normal à superfície detectada, georreferências, elevação, marcações de classificação, entre outras [1–12,21,22,39]. Como exemplo, apresenta-se a nuvem de pontos feita com o sistema Leica ScanStation 2 na Figura 12 [39], onde a nuvem de pontos possui informações de cor, devido ao emprego de uma câmera especializada no sistema, já que o sensor LiDAR sozinho não consegue obter informações de cores.



Figura 12 – Nuvem de pontos com informação adicional de cor (Abadia de Sint-Baafs, Gent, Bélgica) [39].

#### 2.2.4. Aplicações

A tecnologia vem sendo empregada nas mais diversas áreas do conhecimento. Carros autônomos realizam mapeamento em tempo real dos seus arredores com o uso de sensores LiDAR, acoplados à carroceria, em conjunto com redes de *machine learning*, reduzindo os riscos de colisão, como exemplificado na Figura 13a [1,2,16,17,20,37]. Extensões de florestas podem ter seu dossel florestal calculado ou analisado por meio de nuvens de pontos geradas por sistemas LiDAR embarcados em drones ou aviões, como na Figura 13b [3–5,40]. De forma similar, o emprego da tecnologia em arqueologia vem ajudando a identificar ruínas de cidades há muito corroídas pelo tempo, como na Figura 13c [21,22,39,41]. Operações submarinas já utilizam o LIDAR para inspeções de poços de petróleo, naufrágios etc., como na Figura 13d [12–14,38]. Esta última aplicação, inclusive, é uma das motivações desta dissertação de mestrado, visando aumentar o número de *datasets* de nuvens de pontos em meio submarino, já que a aquisição destes em campo possui alto nível de complexidade e custo.

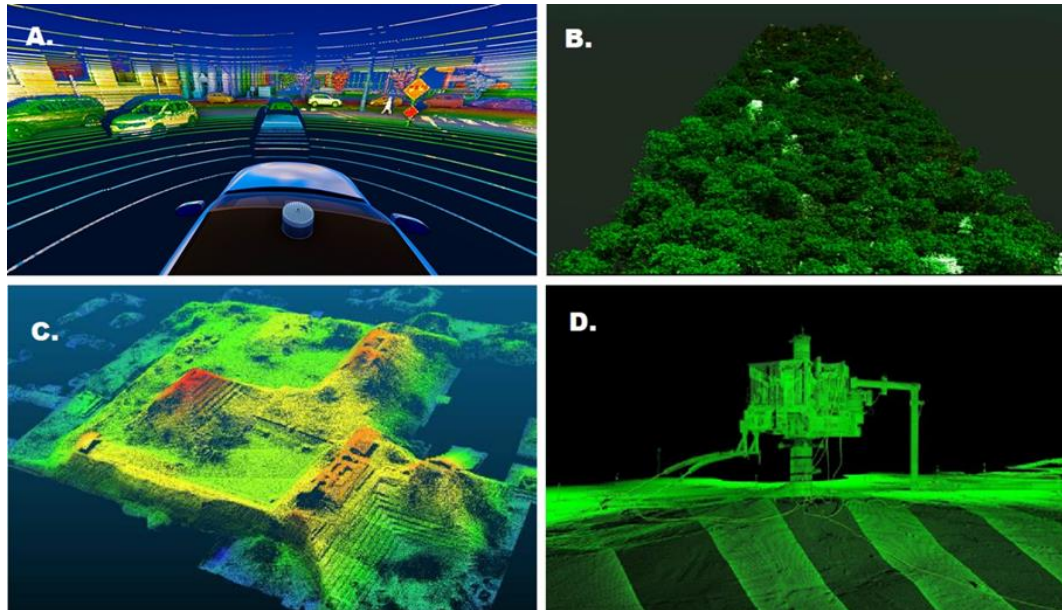


Figura 13 – a) Uso do Velodyne Alpha Puck para obter informações do ambiente ao redor do carro autônomo [37]. b) Observação de nuvens de pontos da cobertura florestal da Amazônia para avaliar o impacto da seca provocada pelo El Niño [40]. c) Nuvem de pontos da Acropolis de Copan em Honduras [32]. d) Nuvem de pontos de uma inspeção submarina de um poço submarino pela 3D at Depth [38].

### 3

## Fundamentos de escaneamento virtual e ferramentas de software

Escaneamento virtual é uma técnica que visa a geração de nuvens de pontos sintéticas por meio de simulações computacionais, normalmente, com o propósito de fornecer *datasets* em situações nas quais a obtenção de dados em campo possui um alto custo, risco e tempo de uso. Este capítulo começa com a contextualização, justificativa e conceituação da técnica, seguido de um breve estudo do estado da arte. O capítulo é encerrado com a definição do algoritmo de rastreamento de raios, da API Nvidia OptiX e Unity 3D, escolhidos para a implementação do simulador desenvolvido neste trabalho.

### 3.1.

#### Escaneamento Virtual

Como apresentado na seção 2.2, as tecnologias de imageamento 3D para geração de nuvens de pontos, em especial o LiDAR, têm um uso cada vez mais disseminado em diferentes áreas do conhecimento. No entanto, existem alguns problemas associados ao uso dessas tecnologias.

A aquisição dos dados brutos é diretamente dependente da aplicação e tamanho do alvo de interesse. Como exemplo da magnitude que um projeto destes pode ter, apresenta-se a Figura 14, onde exhibe-se a nuvem de pontos aérea de Washington, DC, EUA [42].

Em algumas aplicações, existem situações que são impossíveis de serem testadas devido a restrições de segurança e preservação do bem-estar humano, como no caso da aquisição de dados para treinamento de carros autônomos em situações de atropelamento ou colisão [2–6,15,17,20]. Também existe o problema da aquisição em ambientes de difícil acesso ou de grande risco de dano ao equipamento. Há problemas recorrentes nos escaneamentos em meio submarino, como o mapeamento de naufrágios visto na Figura 15, e na arqueologia, como o



mapeamento de ruínas em regiões isoladas [6,12,21,22,41]. Ademais, o hardware associando a essas tecnologias ainda é caro. Com isso, o processo pode tornar-se custoso, arriscado e demorado [2–6,15,17,20].

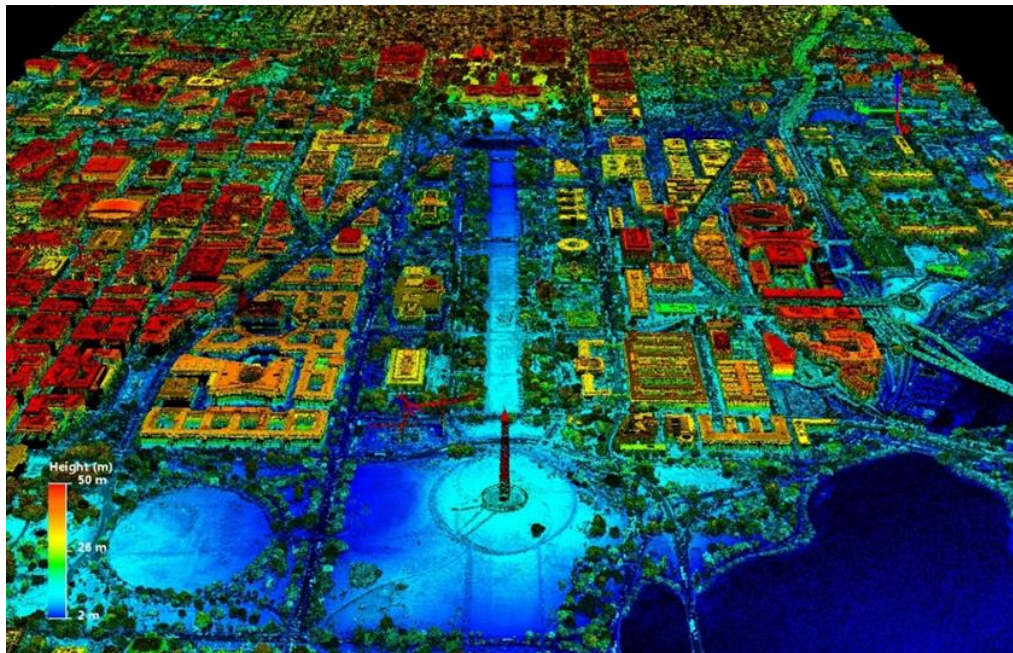


Figura 14 – Nuvem de pontos de LiDAR de Washington, DC [42].

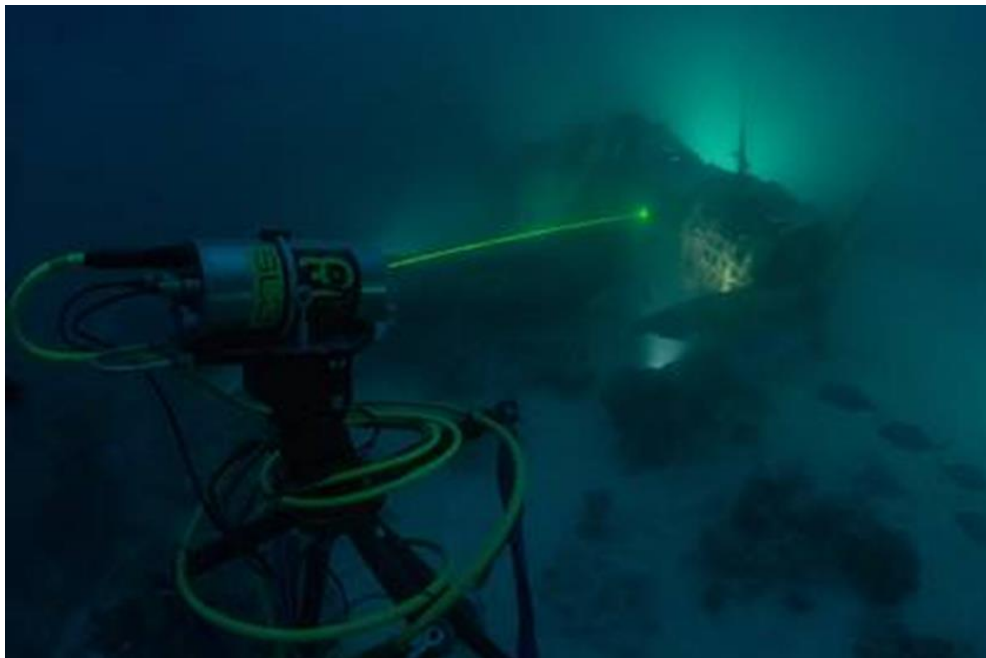


Figura 15 – Escaneamento da aeronave submersa TBD-1 Devastator da época da segunda guerra mundial [43].

Estes aspectos prejudicam aplicações dependentes de um número massivo de *datasets* para trabalho, como é o caso dos métodos de aprendizagem de máquina (*machine learning*), aprendizagem profunda (*deep learning*) e testes de novas técnicas de mapeamento para cálculo de dossel florestal [2–6,15,17,20,40]. Outro problema é a classificação dos dados das nuvens de pontos obtidas de escaneamentos com dispositivos de imageamento 3D, devido à quantidade gigantesca de pontos presentes nos *datasets* [2–6,15,17,20].

Uma solução em voga para superar essas dificuldades, principalmente nos setores automobilístico, de robótica e ambiental, é o **escaneamento virtual** [2–6,15,17,20]. Esta técnica consiste em simulações computacionais do escaneamento de um cenário virtual por dispositivos de imageamento 3D, usualmente sistemas LiDAR, replicando as características do alvo de interesse real, resultando em nuvens de pontos sintéticas ao final do processo. A Figura 16 exemplifica este conceito.

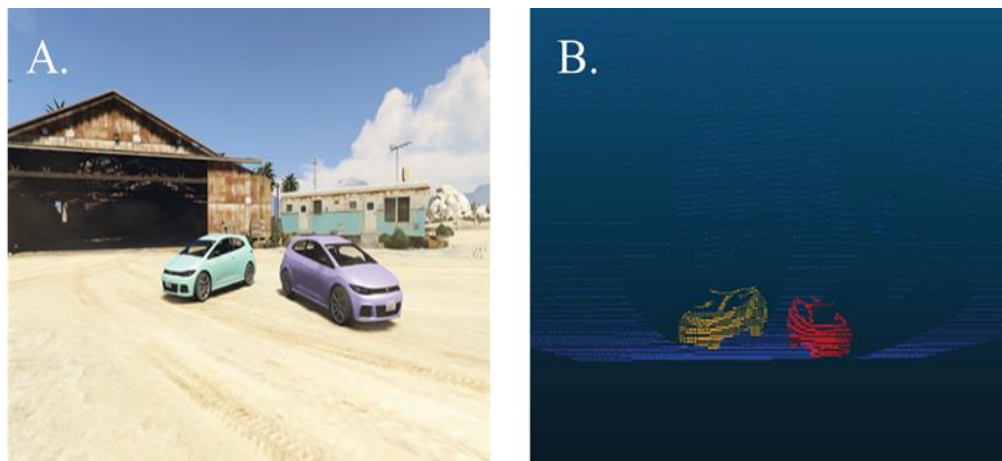


Figura 16 – a) Um cenário virtual para simulação de imageamento 3D. b) A nuvem de pontos sintética gerada pelo sensor virtual a partir de uma posição de escaneamento [17].

Como o simulador não é submetido às limitações existentes na aquisição de dados reais, é possível reproduzir um grande número de cenários e experimentos, até mesmo aqueles onde poderia haver riscos para o equipamento ou pessoal. O tempo de geração das nuvens de pontos também é muito menor do que no caso real, já que o tempo de processamento é limitado apenas pelo poder computacional disponível [2–6,15,17,20]. Estas limitações da aquisição em campo podem ser



observadas no artigo [44], onde foram necessários cinco dias para aquisição dos dados brutos com um carro com sistema de sensores e somente durante o dia.

Além deste método produzir dados artificiais, ele também permite o estudo das tecnologias de imageamento 3D por alunos e pesquisadores em um ambiente controlado. Sendo assim, supondo uma boa modelagem dos dispositivos sensores, o simulador traz experiência prévia para o usuário antes de realizar aplicações reais, diminuindo erros no manuseio e aumentando a velocidade de aprendizagem de uso desses equipamentos [5,6,15].

A maioria desses simuladores é desenvolvida pela seguinte diretriz: Desenvolve-se ou busca-se um construtor de cenários virtuais, modela o sensor virtual que irá realizar a simulação e finalmente valida-se o modelo em comparação com dados reais [2,6,17,20].

#### **A) Cenário Virtual**

O cenário virtual é uma representação computacional de um alvo real de interesse, seja ele um ambiente ou um único objeto. Esses cenários são construídos com o uso de arquivos de modelos 3D, podendo ser do tipo CAD, objetos usados em jogos ou até mesmo modelos com malhas construídas a partir de nuvens de pontos [2–6,15,17,20]. Estes arquivos contêm informações de um objeto em três dimensões, incluindo coordenadas de posição, textura, vetor normais, entre outros. Os formatos típicos são PLY, OBJ, STL, DXF, FBX, 3DS [45].

Na Figura 16, por exemplo, o cenário foi construído usando recursos do jogo Grand Theft Auto V (GTA-V), enquanto em [2] foi utilizado o simulador automobilístico VTD da Vires.

A geração das nuvens de pontos sintéticas será tão mais robusta quanto mais detalhados forem os modelos 3D utilizados no cenário [2–6,15,17,20]. A Figura 17 mostra um carro de Fórmula 1 real e seu modelo 3D para comparação.

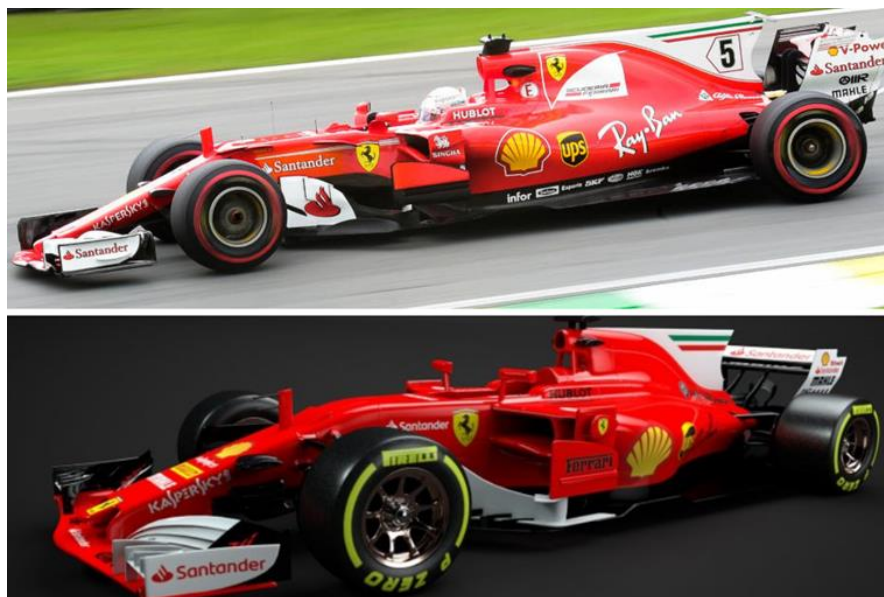


Figura 17 – No topo, um carro de F1 da Ferrari [46]. Abaixo, o modelo 3D do carro (disponível nos formatos FBX, OBJ ou 3DS) [47].

## B) Sensor Virtual

O principal componente em um sistema de imageamento 3D é o seu dispositivo sensor, já que este irá ditar quais métodos de aquisição de dados podem ser utilizados. A contrapartida digital desses dispositivos na simulação computacional é o sensor virtual, uma representação matemática baseada no comportamento de um sensor e nos princípios de aquisição dos dados da técnica de interesse. Esta definição é válida para qualquer modelagem baseada em sensores reais [2,15,48,49].

No caso dos métodos de imageamento 3D, para geração de nuvens de pontos, isto se traduz em como emular a aquisição de dados brutos por um sensor real, seja dos métodos de imageamento ópticos 3D passivos ou ativos, na forma de imagens ou distâncias [2–6,15,17,20]. A qualidade das informações obtidas afeta diretamente a eficiência do escaneamento virtual, pois os dados brutos reais também são processados por algoritmos computacionais para a geração das nuvens de pontos [2–6,15,17,20].

Estes tipos de sensores virtuais encontram-se em duas categorias, como explicado em [2]:

I – Modelo baseado em medições: a modelagem do comportamento do sensor virtual é baseada no processo de medição feito pelo sensor real.

Logo, existe a preocupação de replicar os princípios físicos utilizados na captura das informações do alvo de interesse, assim como o posicionamento e orientação do sensor real. Exemplificando com o LiDAR, uma modelagem baseada em medição tentaria replicar o princípio de tempo de voo no simulador; ou

II – Modelo baseado em erros: a modelagem do comportamento do sensor virtual é baseada nos erros estatísticos percebidos nas medições de um sensor real, levando em consideração erros sistemáticos e aleatórios. Este caso de modelagem também pode ser feito de forma probabilística [7,18]. Para um sensor virtual baseado em LiDAR existiria a preocupação de modelar os desvios nas medições provocadas pelas condições e características ópticas do ambiente, assim como os erros provenientes do manuseio e construção do dispositivo.

Modelos da categoria I são os mais utilizados na literatura, especialmente em robótica e setor automobilístico, enquanto a categoria II começou a ser mais explorada recentemente [2,5,6]. Em ambos os tipos de modelagens, há a necessidade de reproduzir da forma mais fiel possível as características metrológicas do dispositivo de imageamento 3D. Usando o sensor LiDAR mais uma vez como exemplo, a sua contraparte digital deve ter as suas características e limitações de operação, como ângulo máximo de escaneamento e alcance máximo do laser [2–6,16,17,20].

### C) Validação do Modelo

A validação do modelo do sensor virtual é a forma de garantir a robustez das nuvens de pontos sintéticas geradas na simulação. Na literatura, esta etapa é vislumbrada de duas formas [1,2,5,6,16,17,20].

I – Por meio da análise do erro entre uma nuvem de pontos sintética e uma nuvem de pontos real, ambas geradas a partir de um alvo e de sua contraparte digital, com parâmetros de escaneamento similares [1,5,6,16,20], como ilustrado na Figura 18.

II – Por meio da análise do sinal primário gerado pelo sensor virtual, usando o sensor real como referência, com ambas as medições feitas de um alvo e sua contraparte digital com parâmetros de escaneamento similares

[2,17,18], como ilustrado na Figura 19. Esta opção funciona como um método de calibração para o sensor virtual, supondo que o sensor real já esteja devidamente calibrado [17].

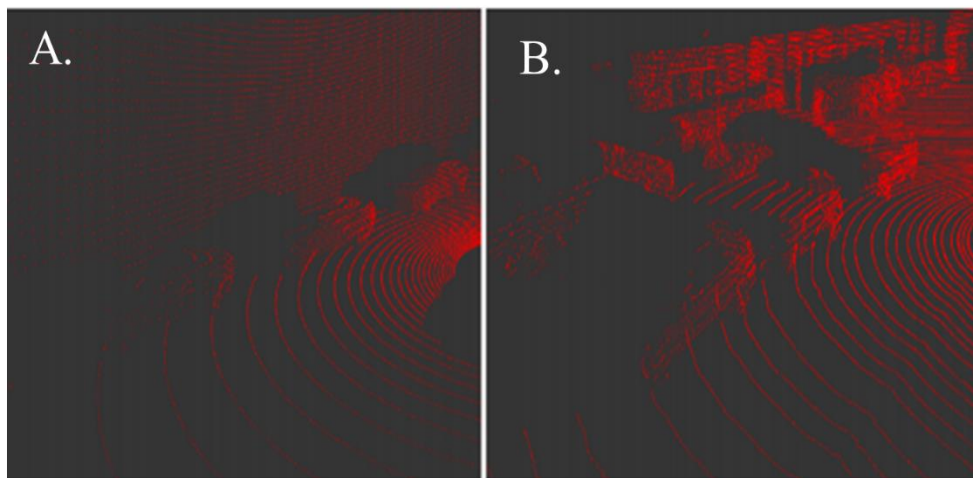


Figura 18 – Comparação visual entre uma nuvem de pontos sintética (a) e uma nuvem de pontos real (b) de um alvo e sua contraparte virtual [1].

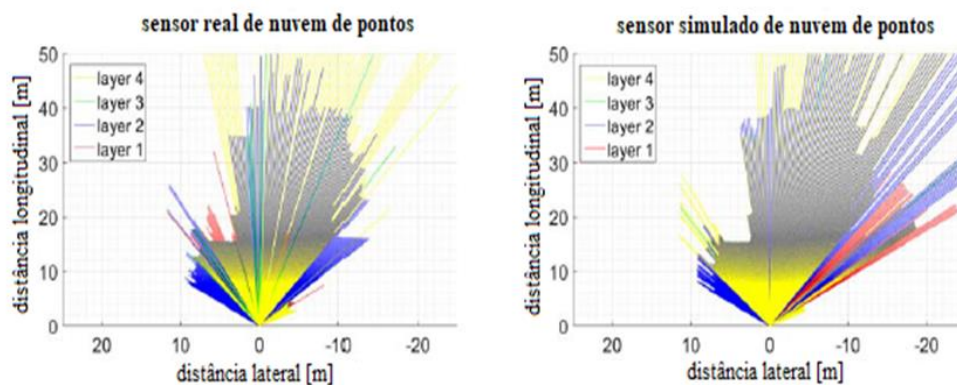


Figura 19 – Visualização da geometria dos raios disparados pelo sensor real e virtual [2].

### 3.2. Estado da Técnica

A literatura mostrou que, apesar da ideia e concepção do escaneamento virtual existir desde 2004, foi somente com os avanços obtidos a partir de 2016 que sua implementação se tornou viável com o aprimoramento do poder computacional, principalmente na parte gráfica [1–6,14–20]. Sendo assim, muitos artigos com

maior impacto são recentes, muitos tendo sido publicados ao longo da pesquisa e desenvolvimento desta dissertação.

Em 2004, Constantini e Susstrunk publicaram um artigo onde apresentam um sensor virtual de imageamento baseado em modelos de erros, tanto ópticos quanto eletrônicos, com o intuito de estudar e analisar as características de erros nos mais diversos cenários, assim podendo desenvolver algoritmos de processamento de imagem mais eficientes [48].

Em 2015, Majek e Bedkowski propuseram o uso de algoritmos de rastreamento de raios (*raytracing*) com paralelismo na GPU para simular o comportamento de sensores de distância, já que eles “geram uma quantidade de dados massivos por segundo” [18]. O simulador foi desenvolvido usando o OptiX, a engine de *raytracing* da Nvidia [19].

Em 2017, Hanke et al., em uma colaboração entre a BMW e faculdades alemãs, apresentaram um *pipeline* para o desenvolvimento e validação de simuladores para escaneamento virtual baseados em sistema LiDAR, para a geração de nuvens de pontos sintéticas usadas em *datasets* para treinamento de *deep learning* na área de carros autônomos. A API OptiX foi utilizada em conjunto com o simulador de carros VTD da Vires para criação dos cenários [2].

Em 2018, Xiangyu et al. desenvolveram um simulador baseado em LiDAR usando uma API de *ray casting* em conjunto com recursos do vídeo game GTA V, com o intuito de gerar nuvens de pontos sintéticas com algum nível de classificação dos objetos contidos nelas. Neste artigo, também mostram um método de calibração para o sensor virtual [17].

Neste mesmo ano, Öhman apresentou em sua dissertação de mestrado um simulador baseado em LiDAR, usando *raytracing*, para testar métodos de localização de árvores e observar a influência de certos parâmetros de interesse nestes métodos [5].

Também em 2018, Tallavajhula apresentou em sua tese de doutorado um simulador para geração de nuvens de pontos sintéticas para treinamento de robôs autônomos, assim como uma revisão da literatura, evidenciando a necessidade de simuladores cada vez mais realistas para trabalhar em conjunto com os dados gerados em campo [6].

Em 2019, Wang et al. publicaram um artigo mostrando um simulador de sistema LiDAR baseado em *raycasting*, adaptado no simulador de carros autônomos CARLA, para geração automática de nuvens de pontos sintéticas com informações de classificação já contidas em seus dados. Eles concluem o artigo evidenciando que o uso de dados sintéticos em conjunto com dados reais aumenta a eficiência dos processos de *deep learning/machine learning* [20].

### 3.3.

#### Ferramentas para Desenvolvimento do Simulador

O levantamento do estado arte salientou que, para o desenvolvimento de simuladores com a finalidade de escaneamento virtual baseado em sistema LiDAR, o algoritmo de *raytracing* é o método mais utilizado para modelar o sensor virtual. Ademais, nota-se que é necessário o emprego de *engines* ou plataformas que permitam a construção de cenários virtuais com a utilização de modelos 3D.

Dessa forma, para o desenvolvimento do simulador desta dissertação, foi escolhida a aplicação de uma *engine* de *raytracing* em conjunto com uma *engine* de jogos, descritos a seguir.

#### 3.3.1.

##### Rastreamento de Raios e API Nvidia OptiX

O rastreamento de raios (*raytracing*) é um algoritmo recursivo de computação gráfica para renderização de imagens. O princípio básico de funcionamento do algoritmo é traçar, para cada pixel na tela, o caminho de um raio partindo de centro de projeção (olho/câmera), passando pelo pixel, e prosseguindo até que haja a colisão deste raio com algum objeto do cenário virtual, conforme ilustrado na Figura 20. Dada esta colisão, podemos retornar a cor e a distância do objeto atingido pelo raio, ou explorar a natureza recursiva do algoritmo, gerando novos raios para calcular as múltiplas reflexões do raio original, havendo, para isso, necessidade de informações sobre as propriedades do objeto e modelos de iluminação e sombras [50].

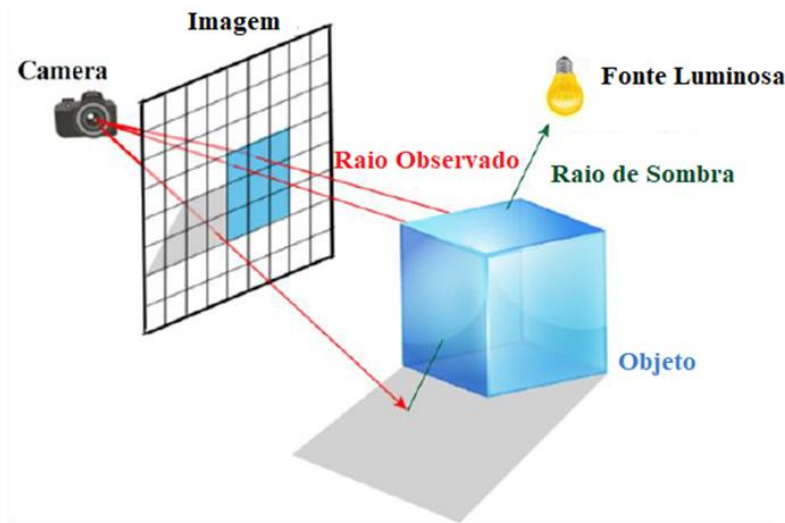


Figura 20 – Princípio de funcionamento do algoritmo de raytracing [36].

Para o desenvolvimento do simulador foi escolhida a *engine* de *raytracing* OptiX da Nvidia [19]. A API permite o uso de um enorme conjunto de ferramentas e algoritmos de *raytracing* com alto grau de paralelismo, ou seja, o processamento de cada raio disparado pode ser executado de forma independente, reduzindo o tempo total dos cálculos computacionais [19,51].

A linguagem de programação adotada pela API é o CUDA (*Computer Unified Device Architecture*), que tem como foco a programação em placas de vídeos da Nvidia. Ela possui um *wrapper* em C++, função de sub-rotinas que chamam outras rotinas em CUDA, a fim de reduzir a complexidade de programação para o usuário. A estrutura do CUDA C se divide em dois tipos: *host code* (código processado na CPU) e *device code* (código processado na GPU) [51]. A Figura 21 exemplifica o funcionamento de um programa em CUDA.

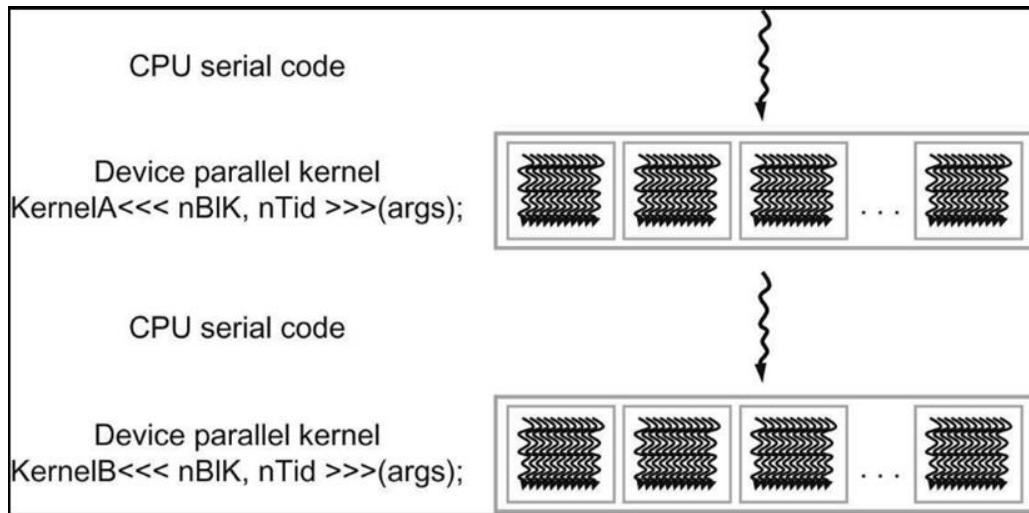


Figura 21 – Execução de um programa em CUDA C. O host code chama os kernels, funções de execução em paralelo com device code, para serem executados. Este processo continua até o fim da execução do programa [51].

### 3.3.2. Unity 3D

Para facilitar a construção dos cenários virtuais foi selecionada a plataforma Unity3D da Unity Technologies [52]. Graças a sua popularidade, esta *engine* de jogos, lançada em 2005, possui um grande acervo de ferramentas para trabalhar com recursos 3D, que trazem flexibilidade para o desenvolvimento de diferentes aplicações, como exibido na Figura 22.

A linguagem utilizada é o C#, possuindo integração com a plataforma de programação Visual Studio da Microsoft. A plataforma é gratuita para desenvolvedores desde que as aplicações produzidas tenham lucro inferior a US\$ 100.000,00 [51].



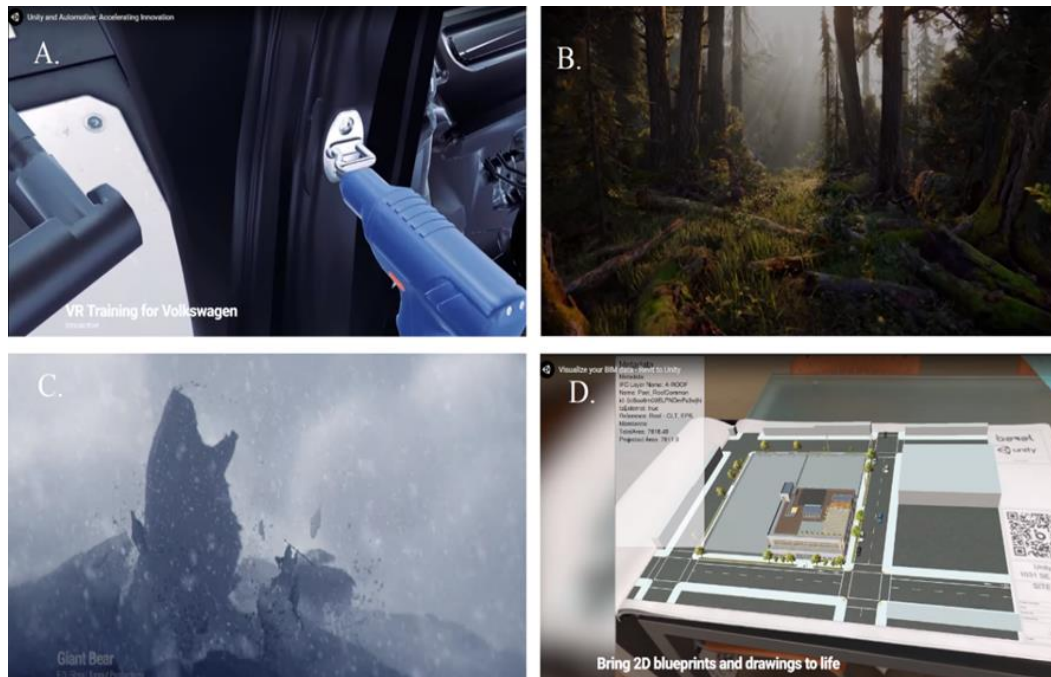


Figura 22 – Diferentes aplicações feitas com a plataforma Unity3D. a) Uma aplicação em realidade virtual (VR) para treinamento na montagem de veículos para Volkswagen. b) Criação de jogos fotorrealistas. c) Desenvolvimento de animações em 2D e 3D, como o Giant Bear da F.D. Films|Taqqut Productions. d) Visualização de BIM (Building Information Model) em 3D de plantas baixas usando realidade aumentada (AR) [52].

## 4

# Desenvolvimento e Validação do Simulador de Sistema LiDAR

O estado da arte em escaneamento virtual, apresentado no Capítulo 3, mostrou a eficiência da técnica na geração de nuvens de pontos sintéticas, principalmente para complementação de aplicações que demandam quantidades massivas de dados. No entanto, como os simuladores apresentados são focados em resolver problemas específicos de suas áreas de interesse, existe uma lacuna para o desenvolvimento de um simulador mais genérico. Este capítulo apresenta o simulador desenvolvido ao longo desta pesquisa de mestrado em Metrologia, explicando cada módulo necessário para geração de uma nuvem de pontos sintética. O capítulo termina com a validação do sensor virtual com base em ensaios experimentais com um sensor LiDAR real, que funciona como um processo de calibração para o simulador.

### 4.1. Simulador

A técnica de escaneamento virtual vem sendo aplicada com sucesso para geração de dados sintéticos para os mais diversos fins, principalmente nas áreas de robótica, automobilística e ambiental. No entanto, como cada simulador tem seu sensor virtual modelado para uma aplicação específica, faz-se necessário que novos simuladores sejam estruturados desde os fundamentos, toda vez que se deseja tratar de outro problema. Ademais, a literatura mostra ser imperativo o constante aperfeiçoamento do realismo dos escaneamentos virtuais, para reduzir a divergência entre os dados sintéticos e reais [2,5,6,15,17]. Esta dissertação de mestrado visa preencher estas lacunas com o desenvolvimento de um simulador modular baseado em um sensor virtual híbrido, ou seja, usando um modelo baseado em medições e erros de um sensor real. As influências do ambiente sobre o escaneamento serão consideradas em uma etapa futura de desenvolvimento do simulador, conforme descrito no Capítulo 6.

O simulador desta dissertação nasceu da necessidade de encontrar uma maneira mais eficiente de gerar nuvens de pontos no meio submarino, principalmente no setor de óleo e gás, já que o ambiente hostil, somado ao alto custo dos equipamentos, torna inviável a aquisição de uma quantidade adequada de dados para aplicações como criação de *datasets* para treinamento de *machine learning* [12,6]. Os dados sintéticos gerados permitiriam provar conceitos de estudos, planejamento de aquisição de dados em campo e aumento do acervo de dados (*data augmentation*) [53].

Como apresentado nos capítulos anteriores, o levantamento do estado da arte indicou um grande uso dos sistemas de imageamento 3D do tipo LiDAR para aquisições de dados submarinos, assim como para as outras áreas mencionadas no início da seção. Sendo assim, este dispositivo foi escolhido como alicerce da modelagem do sensor virtual desta pesquisa.

O simulador, apelidado de optScan, foi desenvolvido com uma abordagem modular, usando a plataforma Unity [52] e a *engine* de *raytracing* NVIDIA OptiX [19]. Ele pode ser dividido em três módulos: **Cenário Virtual**, **Sensor Virtual** e **Gerador de Nuvem de Pontos Sintética**. Esta última possui duas aplicações internas (simulação do escaneamento e fusão das nuvens de pontos). Todo o processo de simulação até a geração dos dados artificiais segue o fluxograma da Figura 23.

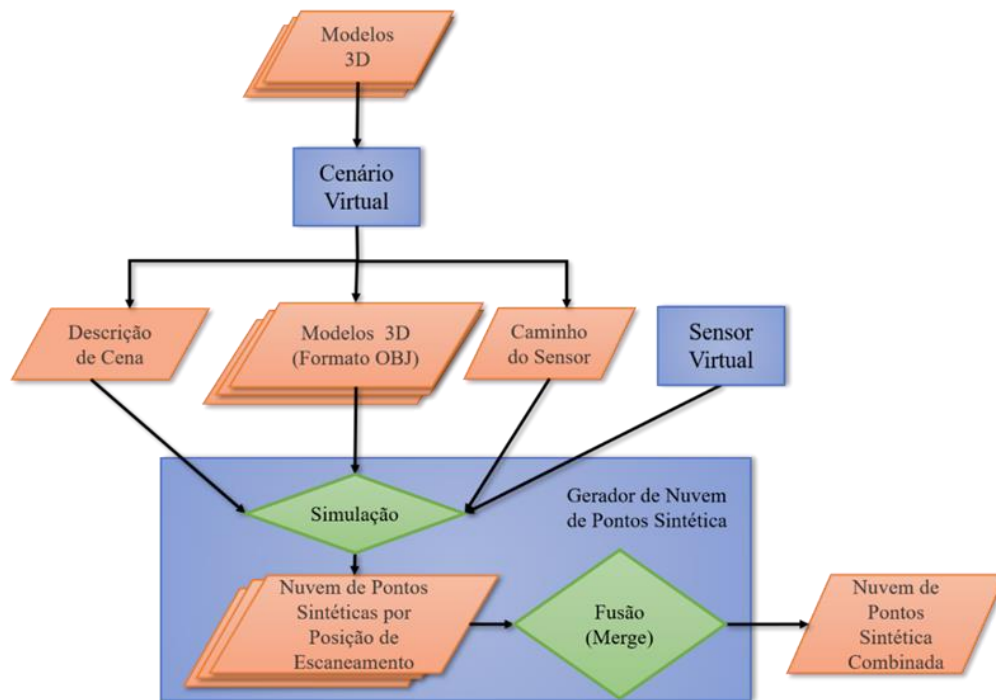


Figura 23 – Fluxograma de operação: paralelogramos são arquivos de dados (▭), retângulos são módulos (▭) e losangos são aplicações (◊).

A característica modular permite trabalhar com o simulador na forma de blocos autônomos, facilitando a substituição ou implementação dos componentes. Exemplificando, o módulo **Cenário Virtual** poderia ser substituído por outro software, bastando apenas adaptar as saídas esperadas pelo módulo **Gerador de Nuvem de Pontos Sintética**. A seguir descreve-se em detalhes os 3 módulos indicados na Figura 23.

#### 4.1.1. Cenário Virtual

Este módulo do optScan permite que o usuário crie a representação computacional do seu alvo de interesse por meio de um editor de cenários virtuais construídos a partir de arquivos de objeto 3D. O desenvolvimento deste editor foi feito com a plataforma Unity, que oferece inúmeras ferramentas para trabalhar tanto com objetos 2D quanto 3D.

O módulo funciona da seguinte maneira:

- 1) O usuário carrega no editor arquivos de objeto 3D que representem o alvo de interesse.

- 2) Após armazenados no projeto, estes objetos podem ser utilizados no cenário virtual, como exemplificado na Figura 24.
- 3) Os objetos possuem quatro características importantes, todas elas definidas pelo usuário durante a construção do cenário:
  - a. **Posição:** indica o posicionamento do objeto no cenário, em relação ao ponto de origem do mundo criado, como ilustrado na Figura 25.
  - b. **Rotação:** indica o quanto o objeto está rotacionado em relação a sua posição angular original, como ilustrado na Figura 25.
  - c. **Escala:** indica o quanto o objeto está ampliado ou reduzido de seu tamanho original, como ilustrado na Figura 26.
  - d. **Id:** é um número de identificação único para cada classe de objeto, criado no momento que o mesmo é posicionado no cenário. Por exemplo, na Figura 26, há 3 objetos distintos, cada um com um número de identificação diferente: a árvore de natal molhada ANM<sup>1</sup>, o caranguejo e o solo arenoso.
- 4) Uma vez finalizada a construção do cenário, o usuário define quais as posições desejadas para escaneamento, preferencialmente seguindo técnicas de posicionamento para aquisição de imagens com imageamento óptico 3D. Isto é feito com uso da entidade “câmera” da plataforma Unity, representada por uma câmera de filme na Figura 27. A posição e direção de cada câmera indica a região que deve ser escaneada. A Figura 27 também mostra a ferramenta *câmera preview*, que exibe a imagem observada dentro do campo de visão do sensor ao ser acionado. Esta ferramenta funciona como um primeiro teste para as eventuais nuvens de pontos geradas durante a simulação, pois elas devem respeitar estas previsões feitas no editor.
- 5) Ao final desta seleção, ao acionar o gerenciador do projeto criado, inicia-se o processo de importação dos dados, criando três tipos de arquivos:
  - a. **Descrição de Cena:** Este arquivo de texto contém os parâmetros definidos pelo usuário de cada objeto contido no cenário virtual,

---

<sup>1</sup> Equipamento típico em campos de petróleo submarino instalado no poço de petróleo [53]

além de alguns comandos *default* para a execução da simulação no módulo **Gerador de Nuvem de Pontos Sintética**.

- b. **Modelos 3D no formato OBJ:** Como o módulo **Gerador de Nuvem de Pontos Sintética** foi desenvolvido com a plataforma Optix, somente arquivos de objetos 3D no formato PLY ou OBJ podem ser utilizados. Para não reduzir o acervo de arquivos permitidos para uso, o editor foi programado com um conversor de objetos 3D para formato OBJ.
- c. **Caminho do Sensor:** Este arquivo de texto contém a posição e orientação de escaneamento do sensor em cada ponto definido pelo usuário. Os parâmetros armazenados supõem que o sensor é mantido estático em cada posição.

Estes arquivos de saída, somados ao módulo **Sensor Virtual**, servem de entrada para a simulação contida no módulo **Gerador de Nuvem de Pontos Sintética**.

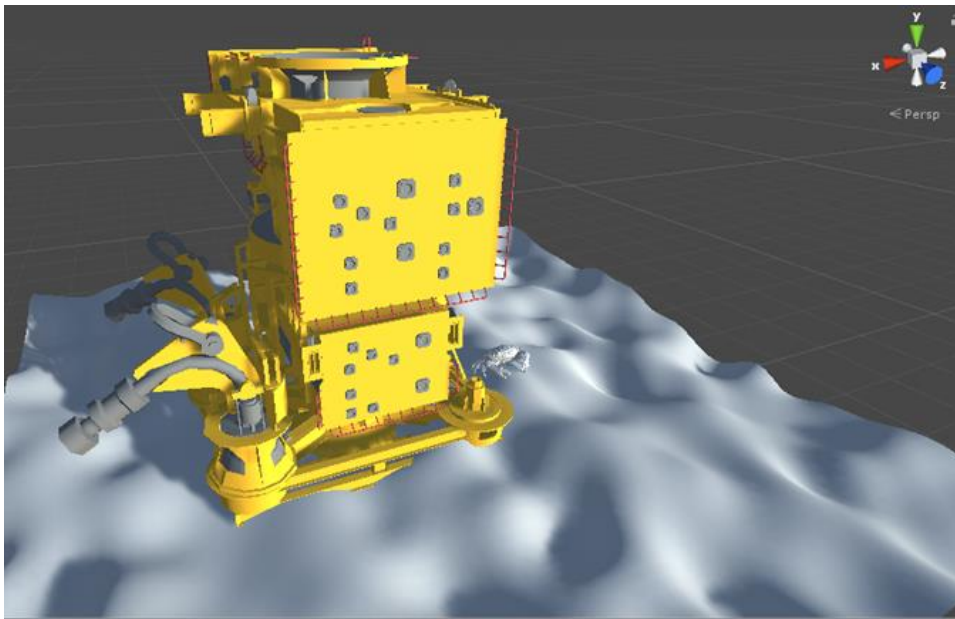


Figura 24 – Cenário simples construído com o editor do módulo Cenário Virtual. A cruzeta no topo indica a direção dos eixos.

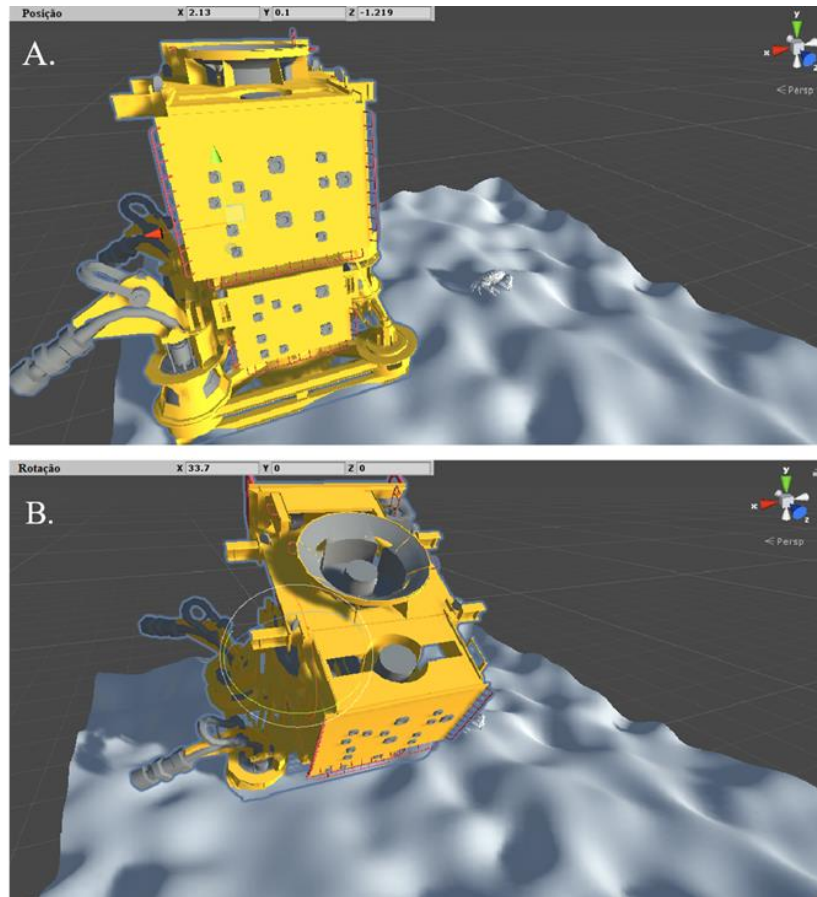


Figura 25 – a) Parâmetro posição. b) Parâmetro rotação.



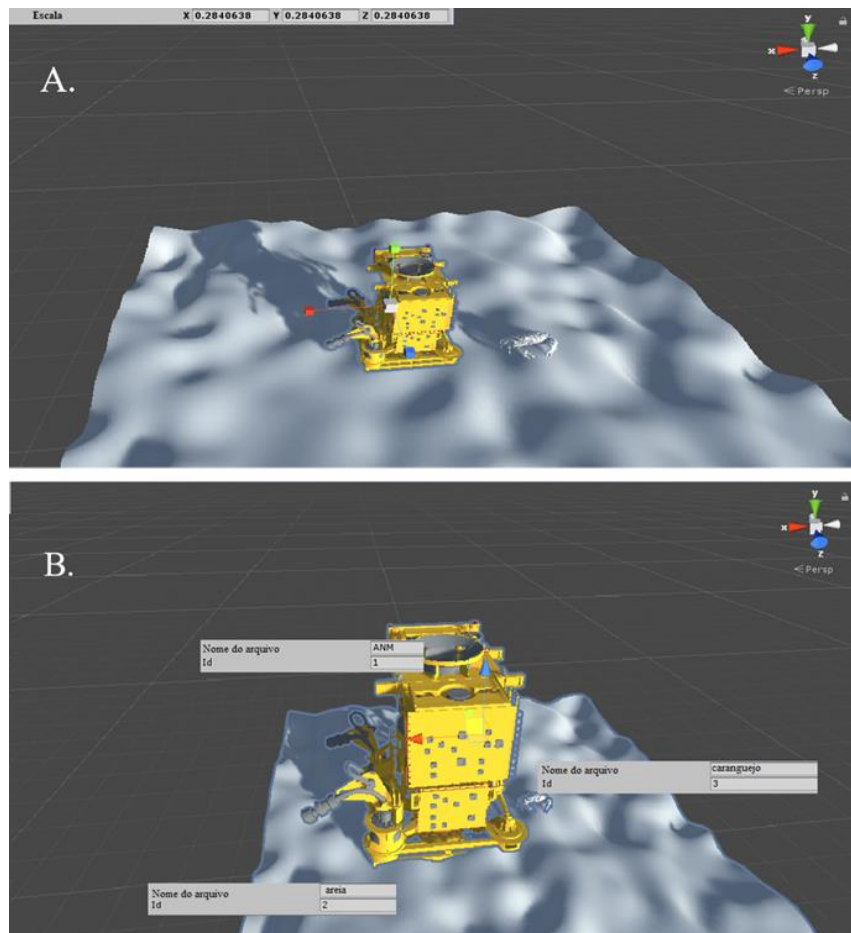


Figura 26 – a) Parâmetro escala; b) Parâmetro número de identificação (Id).

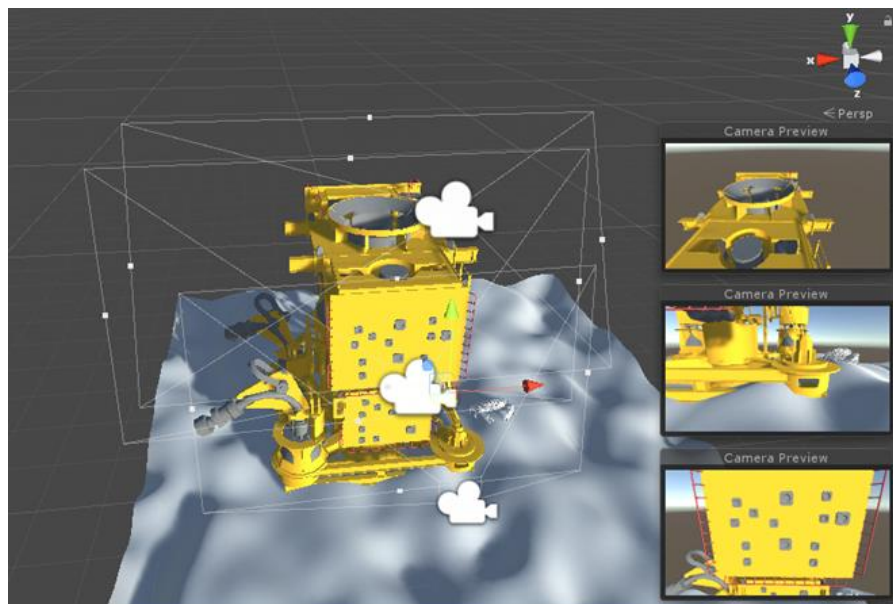


Figura 27 – Posicionamento do sensor virtual no cenário. À direita, a previsão das imagens observadas pelos sensores.



#### 4.1.2. Sensor Virtual

Como explicado no Capítulo 3, a parte mais importante de um escaneamento virtual é o sensor virtual, a contraparte digital de um sensor em um sistema de imageamento 3D. O módulo **Sensor Virtual** deste simulador foi modelado a partir de um dispositivo LiDAR, com uma abordagem aqui nomeada de híbrida, pois une o modelo baseado em medições virtuais com um modelo baseado em erros (estimados a partir de ensaios experimentais com um sensor LiDAR real) para aumentar o realismo da aquisição de dados brutos para geração da nuvem de pontos.

Para realizar a modelagem do sensor virtual foi necessário o levantamento de suas características metrológicas e de seus princípios de funcionamento, para ter uma maior clareza sobre o comportamento do LiDAR. Estes dados foram apresentados no Capítulo 2 e alguns deles serão reiterados ao longo dessa subseção.

Como um marco inicial, a pesquisa de mestrado focou em como replicar a técnica de tempo de voo empregada pelo sensor LiDAR em uma simulação computacional. Assim como na literatura, concluiu-se que algoritmos de rastreamento de raios (*raytracing*) são uma excelente forma de reproduzir a técnica, devido à similaridade entre seus fundamentos, como exibido na Figura 28.

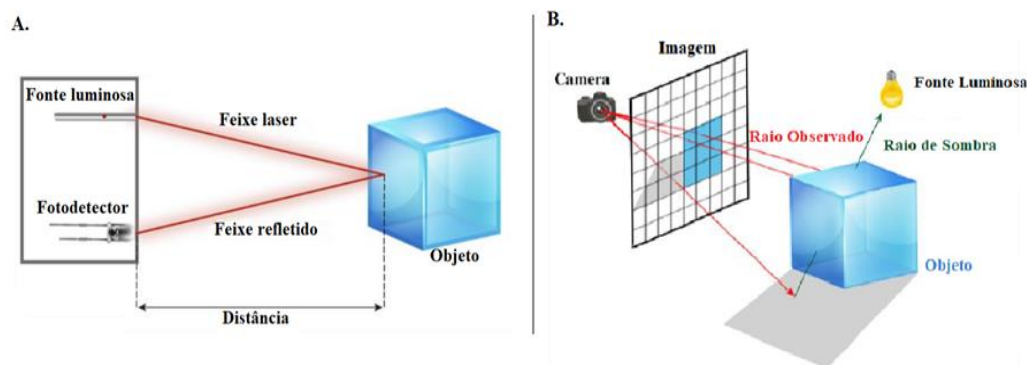


Figura 28 – a) Técnica de tempo de voo usado em sensor LiDAR. b) Fundamento do algoritmo de raytracing [36].

Essas informações levaram à escolha da *engine* NVIDIA OptiX, que possui uma gama de ferramentas e algoritmos de *raytracing* prontas para uso, já incluindo paralelismo na GPU.

Para os propósitos deste projeto, foi necessário adaptar algumas funções e características do algoritmo de rastreamento de raios para melhor modelar o sensor virtual nas condições desejadas. As principais adaptações foram:

- 1) **Plano Esférico:** Sensores LiDAR usualmente escaneiam uma área dentro de um arco de circunferência, com um ângulo de passo definido pelo fabricante. Como o *raytracing* trabalha em um plano cartesiano, foi necessária a conversão para um plano esférico [2]. Isto traz distorções cônicas típicas do escaneamento com LiDAR [2]. A Figura 29 exibe os dois planos para *raytracing*.
- 2) **Ângulo de disparo:** Em consequência da alteração para o plano esférico, a quantidade de raios disparados depende do tamanho da tela de simulação e do passo angular definido pelo usuário. Sendo assim, os raios agora respeitam um ângulo máximo azimutal (horizontal) e um ângulo máximo zenital (vertical). Isto emula os ângulos de operação para escaneamento de um LiDAR.
- 3) **Alcance máximo de detecção:** A distância máxima do raio disparado é definida pelo usuário de acordo com a distância de operação do LiDAR. Isto faz com que objetos fora deste limite não sejam registrados pelo sensor virtual, tal como ocorre com um sensor real.

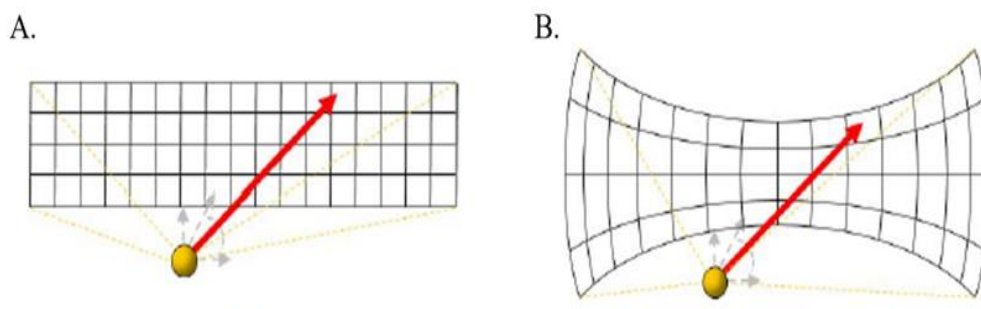


Figura 29 – a) Plano cartesiano. b) Plano esférico [2].

Estas alterações fazem parte do modelo baseado em medições, apresentado no artigo [36], pois replicam os princípios físicos do dispositivo sensor.

Já para o modelo baseado em erros foi realizada uma abordagem empírica, por meio do seguinte experimento:

- 1) Foi escolhido um sensor LiDAR para realizar escaneamentos de um objeto de geometria simples, neste caso um dos pilotis da PUC-Rio, como o mostrado na Figura 30. O formato cilíndrico, em conjunto com o material com baixa reflexividade, facilita isolar os erros provenientes do dispositivo daqueles advindos do ambiente. As características deste LiDAR encontram-se na seção 4.2.
- 2) O sensor foi posicionado em frente ao piloti de forma que seu feixe central atingisse o meio da estrutura, a uma altura fixa de 1240 mm (tamanho do suporte até o chão).
- 3) O processo foi realizado 16 vezes para distancias indo de 500 mm até 2000 mm do feixe central ao meio do piloti, de forma a cobrir a maior parte da faixa de detecção do LiDAR usado (60 mm – 4000 mm). Uma trena laser Bosch, com incerteza de medição de 1,5 mm, foi utilizada como referência para garantir o posicionamento correto do sensor.
- 4) Para cada uma dessas distâncias, foram feitos 500 escaneamentos, obtendo por volta de 60 pontos a cada varredura, obtendo-se assim um total de 480000 pontos.
- 5) Esses dados foram tratados de modo a remover pontos com distância nula. Em seguida, foram obtidas as propriedades estatísticas desse conjunto.
- 6) Uma equação do erro em função da distância foi obtida por meio do ajuste de curva polinomial de grau 2 dos dados de média e seu desvio padrão calculados no passo anterior.

$$\begin{aligned}
 &erro(distancia) \\
 &= -5,139.10^6.distancia^2 \\
 &+ 9,92.10^{-4}.distancia \\
 &+ 15,26 [mm]
 \end{aligned}
 \tag{2}$$

O erro médio quadrático (RMSE) foi de 3,689. A curva obtida é apresentada na Figura 31.

- 7) Além dessa equação de erro, um ruído gaussiano ( $N(0,3)$ ) foi adicionado para incluir erros aleatórios nas medições. Esta distribuição foi feita em cima de números aleatórios gerados na placa de vídeo paralelamente.
- 8) Finalmente, este polinômio de erro pode ser implementado no algoritmo de *raytracing* de modo a perturbar cada raio disparado pelo módulo **Sensor Virtual**.

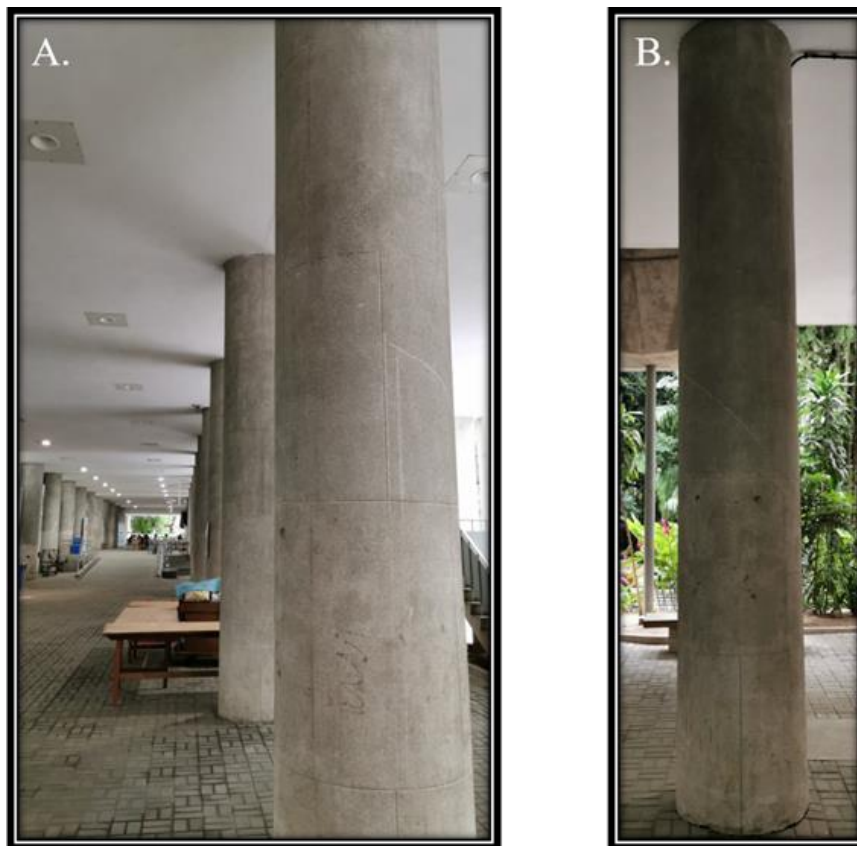


Figura 30 – a) Pilotis do prédio Leme, PUC-Rio. b) Piloti escaneado no experimento.

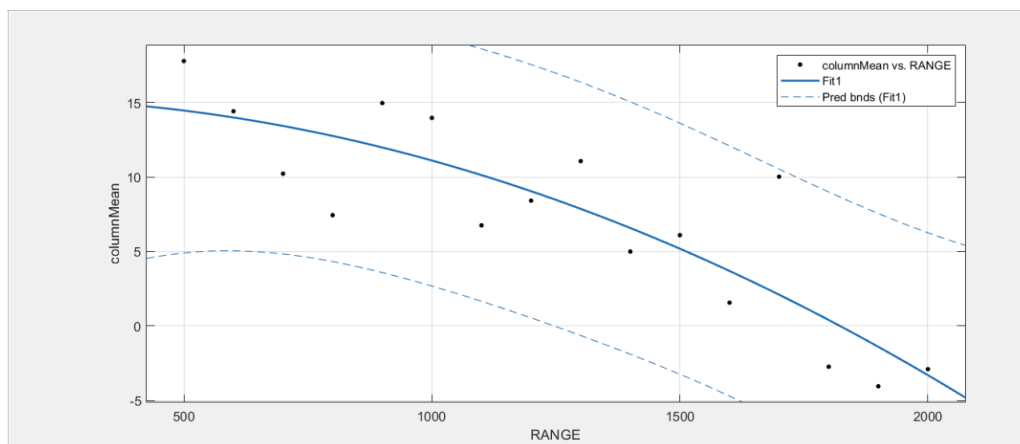


Figura 31 – Curva de ajuste polinomial de grau 2, para obtenção da equação de erro.

A combinação desses dois modelos de sensores virtuais permite uma geração robusta das nuvens de pontos pelo próximo módulo, pois produz dados brutos mais coerentes com os de um escaneamento real, o que ficará mais evidente na seção 4.2.

#### 4.1.3. Gerador de Nuvem de Pontos Sintética

O último módulo do simulador é o responsável pela geração da nuvem de pontos sintética. A execução de todo o seu processo é feita por duas aplicações:

- 1) **Simulação:** Esta aplicação recria o cenário virtual dentro da *engine* OptiX por meio dos arquivos criados no módulo **Cenário Virtual**. Em seguida, o escaneamento virtual é executado pelo módulo **Sensor Virtual**, que obtém informações do cenário quando um raio disparado por ele colide com um objeto. A distância de intercessão obtida dessas colisões são os dados brutos desse processo de imageamento 3D. Dada estas distancias, pode-se calcular a posição 3D da colisão, criando assim o ponto relativo a superfície colidida pelo raio. Este vetor  $\hat{P}$  com coordenadas (x,y,z) é calculado da seguinte forma:

$$\hat{P}(d) = \hat{O} + \hat{D}.d \quad (3)$$

Onde:

$\hat{O}$  é o vetor da posição de origem do raio disparado.

$\hat{D}$  é o vetor de direção do raio disparado, que é calculado a partir dos vetores de orientação do sensor virtual (Up, Right, Forward) e dos vetores de ângulo de disparo.

$d$  é a distância de intercessão encontrada da colisão do raio com um objeto no Cenário Virtual.

Estes dados são então organizados na forma de nuvens de pontos para cada posição definida no módulo **Cenário Virtual**, como exibido na Figura 32.

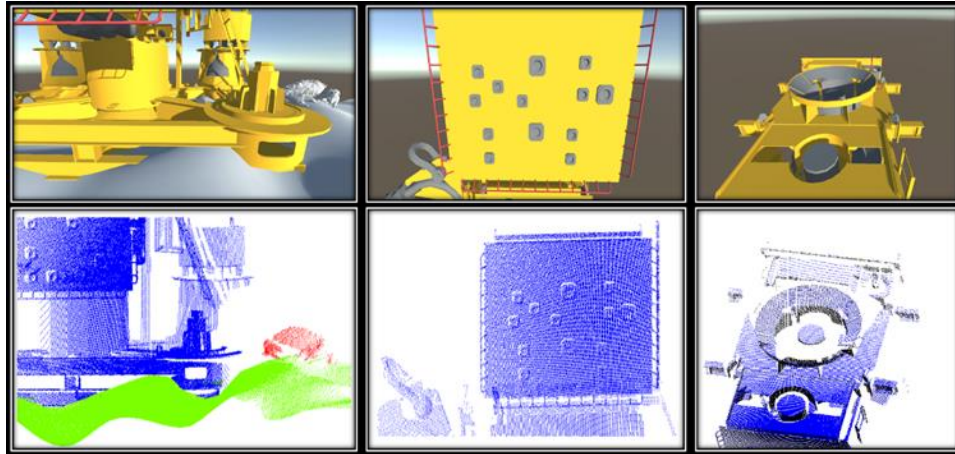


Figura 32 – Nuvem de pontos sintética gerada para cada posição do sensor virtual definido pelo arquivo de caminho do sensor.

Cada ponto dessas nuvens possui as seguintes características:

- a. **Coordenadas 3D:** Definem a posição do ponto em relação ao cenário virtual construído  $(x,y,z)$ . É a informação básica de uma nuvem de pontos. A visualização dos pontos é apresentada na Figura 33a.
- b. **Vetor Normal:** São vetores perpendiculares à superfície dos objetos detectados. Usados para definições de iluminação do material e úteis em várias rotinas de *deep learning* e segmentação de dados [53]. A visualização da ação dos vetores é apresentada na Figura 33b.
- c. **Cor:** Definida na forma de vetores RGB. Útil para verificar a correspondência da nuvem de pontos com seu modelo 3D. Alguns sistemas de LiDAR já possuem câmeras no espectro visível para obter a informação de cor, pois o sensor sozinho não é capaz de adquirir esses dados [39]. A visualização de cores é apresentada na Figura 33c.
- d. **Número de Identificação:** Este número é um identificador único, definido pelo usuário na criação do cenário, como mencionado na subseção 4.1.1. Esta informação facilita a visualização de diferentes objetos dentro de uma mesma nuvem, o que também facilita processos de catalogação e segmentação de dados. A

visualização pelo número de identificação é apresentada na Figura 33d.

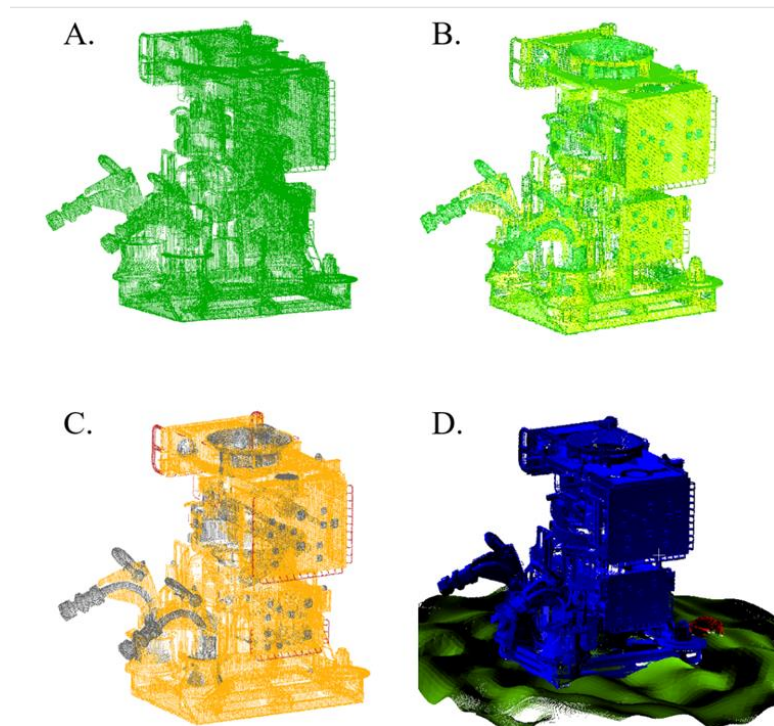


Figura 33 – Diferentes visualizações de nuvem de pontos. a) visualização de pontos. b) Visualização da ação dos vetores normais. c) Visualização de cores. d) Visualização de identificadores com vetores normais em uso para facilitar visualização.

- 2) **Fusão (Merge):** Dado um número suficiente de posições para escaneamento, é possível capturar o alvo de interesse na sua totalidade. Todavia, essas nuvens de pontos sintéticas de uma única face precisam ser unificadas em um único modelo 3D, para que seus dados possam ser mais bem utilizados na maioria dos procedimentos que trabalham com nuvens de pontos. Sendo assim, foi desenvolvido, com uso da biblioteca de algoritmos geométricos Geogram [55], uma aplicação de fusão desses dados.

A aplicação tem como entrada as nuvens de pontos selecionadas pelo usuário dentre as que foram geradas na simulação. Os dados são então combinados em pares e inicia-se um processo de comparação entre os pontos existente nessas nuvens de pontos provisórias. O objetivo desta comparação é definir a existência de pontos redundantes, ou seja, pontos



na nuvem que a posição 3D coincidam ou estão dentro de um limiar de proximidade definido pelo usuário. Este limiar irá variar dentro da faixa de valores de (0,1). A eficiência desse método dependerá dos parâmetros metrológicos configurados no módulo **Sensor Virtual**. Para as nuvens de pontos geradas no Capítulo 4 e 5, foi utilizado o valor de 0,1 mm. A seguir, os pontos redundantes são agrupados e eliminados da nuvem de pontos provisória. Esta filtragem chama-se decimação por espaço [55].

O processo de combinação, comparação e decimação irão se repetir até que todo o dataset tenha sido transformado em uma única nuvem de pontos. Ao final, tem-se uma nuvem de pontos combinada com redução de pontos, podendo ser salva nos formatos XYZ e PLY. A Figura 34 ilustra o processo do módulo **Gerador de Nuvem de Pontos Sintética**.

Caso seja necessário, a aplicação também permite normalizar nuvem de pontos de forma que suas fronteiras estejam contidas dentro de um cubo de aresta de 2 unidades [36,53]. Esta normalização é importante para os dados de treinamento de alguns métodos de *deep learning*.

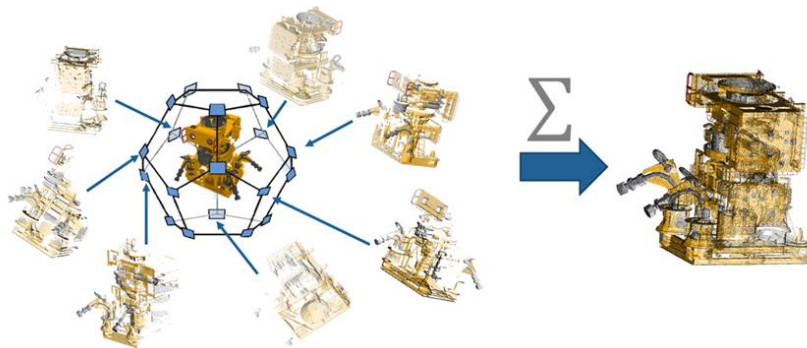


Figura 34 – Uma série de escaneamentos virtuais são realizados gerando um grande volume de nuvens de pontos de uma única face. Em seguida, esse dataset é unificado em uma nuvem de pontos sintética combinada.

## 4.2. Validação

Apesar do simulador gerar nuvens de pontos sintéticas seguindo parâmetros metrológicos definidos pelo usuário, foi necessário fazer uma comparação com um



sensor real, para garantir que o comportamento do sensor virtual fosse de fato similar ao de um sensor LiDAR.

Como mencionado no Capítulo 3, existem duas maneiras principais de validar a robustez da geração de nuvem de pontos dos sensores virtuais: comparação do sinal primário do sensor virtual com o sensor real como referência; ou pelo erro da diferença entre nuvens reais e sintéticas. Nesta dissertação foi escolhido o primeiro método.

O processo inicia-se com a análise dos parâmetros metrológicos do sensor de referência escolhido, no caso deste experimento o modelo URG-04LX-UG01 da Hokuyo, visto na Figura 35. Ele foi o mesmo utilizado no desenvolvimento do modelo baseado em erros do módulo Sensor Virtual. Este dispositivo possui as seguintes características:

- 1) Laser infravermelho com comprimento de onda  $\lambda = 785 \text{ nm}$ ;
- 2) Área de escaneamento azimutal:  $240^\circ$  (resolução angular  $\sim 0,3516^\circ$ ) | Sem área de escaneamento zenital.
- 3) Duas margens de alcance operacional:
  - a. 60 mm – 1000 mm, com incerteza de medição<sup>2</sup> de 30 mm.
  - b. 1000 mm – 4000, mm com incerteza de medição de 3 % da distância detectada.



Figura 35 – Hokuyo URG-04LX-UG01 [56]

O experimento de validação foi dividido em duas etapas, uma para cada modelo pertencente ao módulo Sensor Virtual. A primeira etapa desse experimento foi previamente descrita no artigo [36], que se encontra no Anexo A. Para trazer

<sup>2</sup> Fator de abrangência não disponibilizado

maior clareza nos resultados dessa dissertação, o experimento presente no Anexo A foi replicado com o mesmo modelo utilizado na etapa 2 (um piloti da PUC-Rio) nesta seção.

Para a validação do modelo, baseado em medições, os seguintes procedimentos foram tomados:

- 1) Os parâmetros metrológicos do LiDAR foram configurados no módulo Sensor Virtual.
- 2) Construiu-se um cenário virtual com um cilindro modelado matematicamente com o diâmetro do piloti.
- 3) Tanto no experimento virtual quanto no real, os respectivos sensores foram posicionados de forma que o ângulo central (ângulo zero) coincidissem com o centro da pilastra a uma mesma altura.
- 4) Uma série de escaneamentos foram realizados, do piloti e de sua contraparte digital. No entanto, o simulador foi alterado de forma a obter somente as distâncias das interseções ao invés da nuvem de pontos, já que estes são os dados brutos de um sensor LiDAR. Além disso, a equação de erro e o ruído gaussiano foram desativados da simulação.

Estes procedimentos resultaram em mapas de distância para ambos os sensores. Para facilitar o estudo foram selecionadas duas distâncias centrais, cada uma pertencente a uma faixa de alcance de operação do LiDAR. A Figura 36 e a Figura 37 mostram os mapas de distância a 700 mm e a 1500 mm, respectivamente. Os pontos de intercessão do LiDAR são marcados com ‘o’, enquanto os do sensor virtual são marcados com ‘x’.

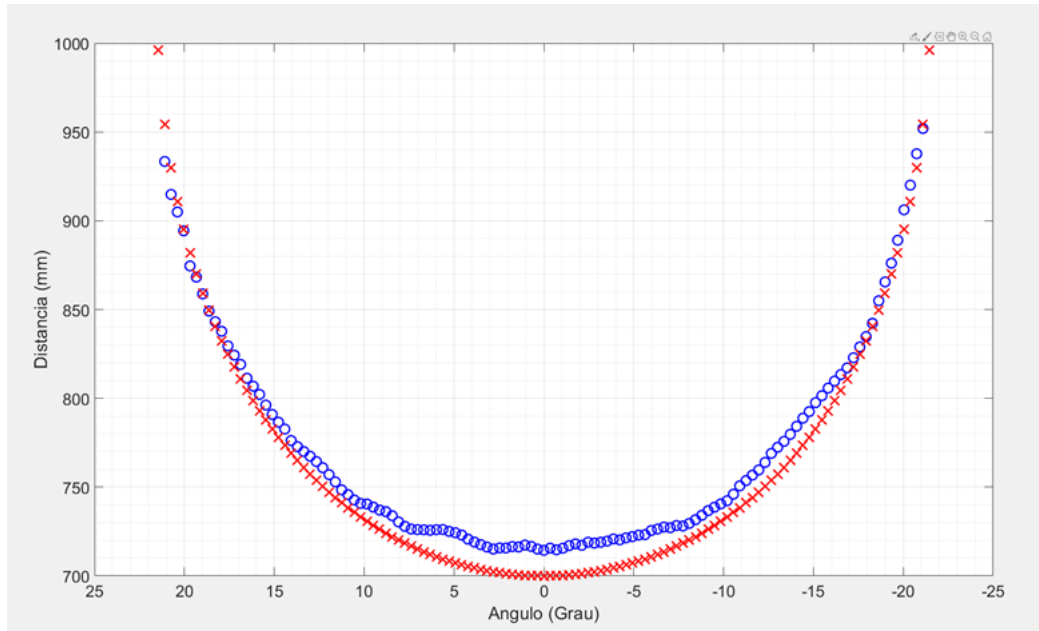


Figura 36 – Mapa de distância a 700 mm: Pontos de intercessão do LiDAR ('o') e pontos de intercessão do sensor virtual ('x')

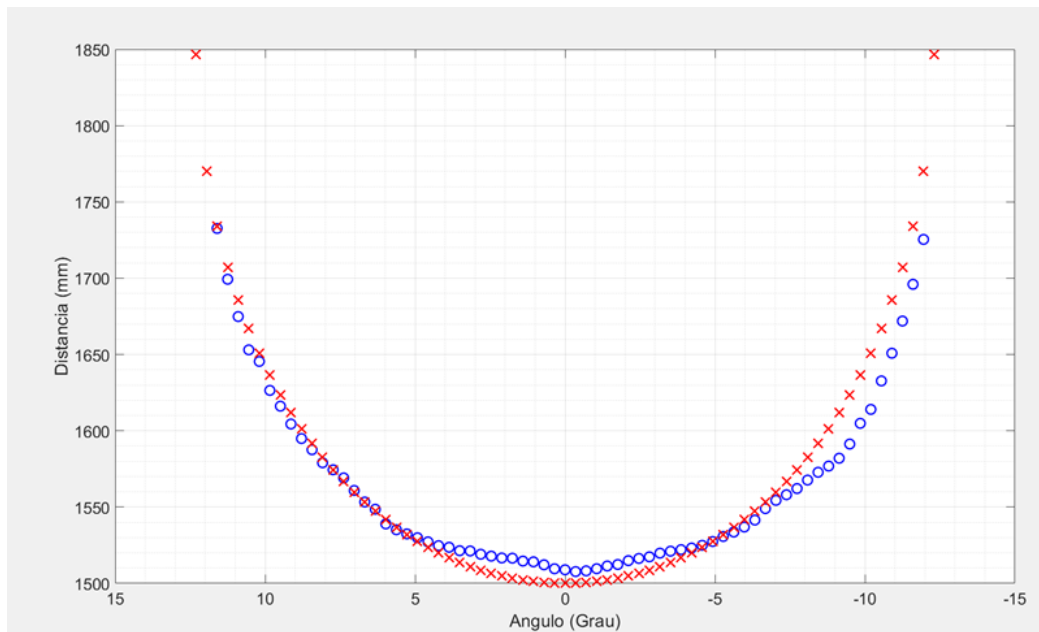


Figura 37 – Mapa de distância a 1500 mm: Pontos de intercessão do LiDAR ('o') e pontos de intercessão do sensor virtual ('x').

Uma vez obtidos estes mapas, foi encontrado o erro quadrático médio para cada distância de interesse usando a diferença entre os dados virtuais e os dados de referência, exibidos na Figura 38. Para a posição do ângulo central a 700 mm, o RMSE encontrado foi 11,9396 mm, enquanto para a posição do ângulo central a

1500 m, o RMSE foi 21,834 mm. Estes cálculos só levaram em consideração os raios que detectaram o alvo de teste.

Estes valores são bons indícios da robustez do sensor virtual baseado em medições, pois o erro entre os dados reais e sintéticos está coerente com as incertezas de medição especificadas pelo fabricante. Isto é esperado, já que a simulação não possuía ruídos ou perturbações e o ambiente do experimento no mundo real, foi controlado para haver o menor impacto possível nas medições.

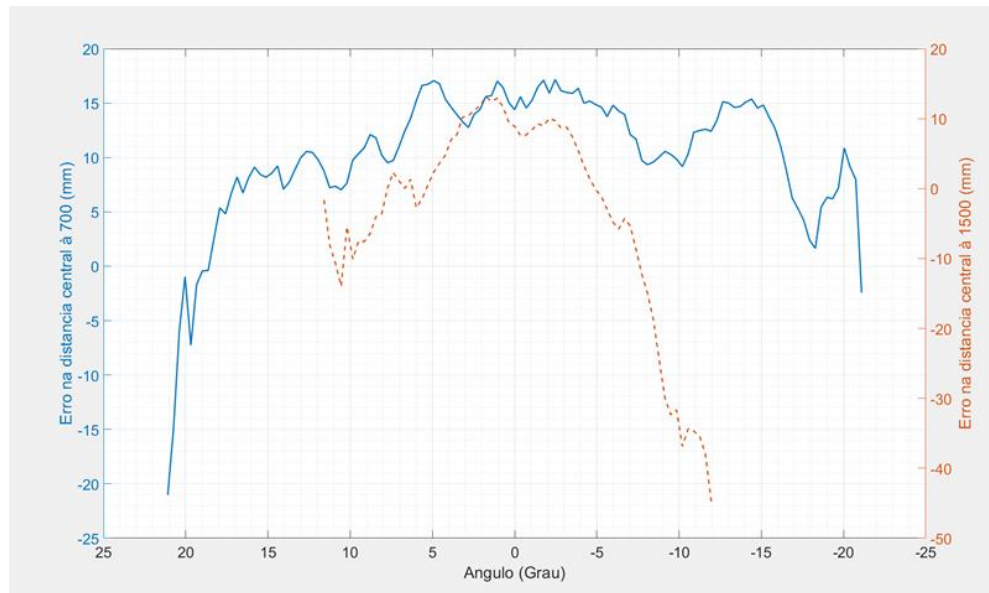


Figura 38 – Diferença entre dados reais e sintéticos. Linha contínua, erro na distância do ângulo central a 700 mm. Linha tracejada, erro na distância do ângulo central a 1500 mm.

Para a validação do modelo baseado em erros, os seguintes procedimentos foram tomados:

- 1) Os procedimentos de 1 a 3 da etapa 1 da validação foram repetidos.
- 2) Uma série de escaneamentos virtuais foram feitos, mas com a equação de erro e ruído gaussiano ativados durante o disparo dos raios.

Estes procedimentos resultaram em novos mapas de distância para o sensor virtual. Estes novos mapas foram comparados com os dados de referência, nas mesmas distâncias centrais da etapa 1. A Figura 39 e a Figura 40 mostram os mapas de distância a 700 mm e a 1500 mm, respectivamente. Mais uma vez, os pontos de intercessão do LiDAR são marcados com ‘o’, enquanto do sensor virtual são marcados com ‘x’.

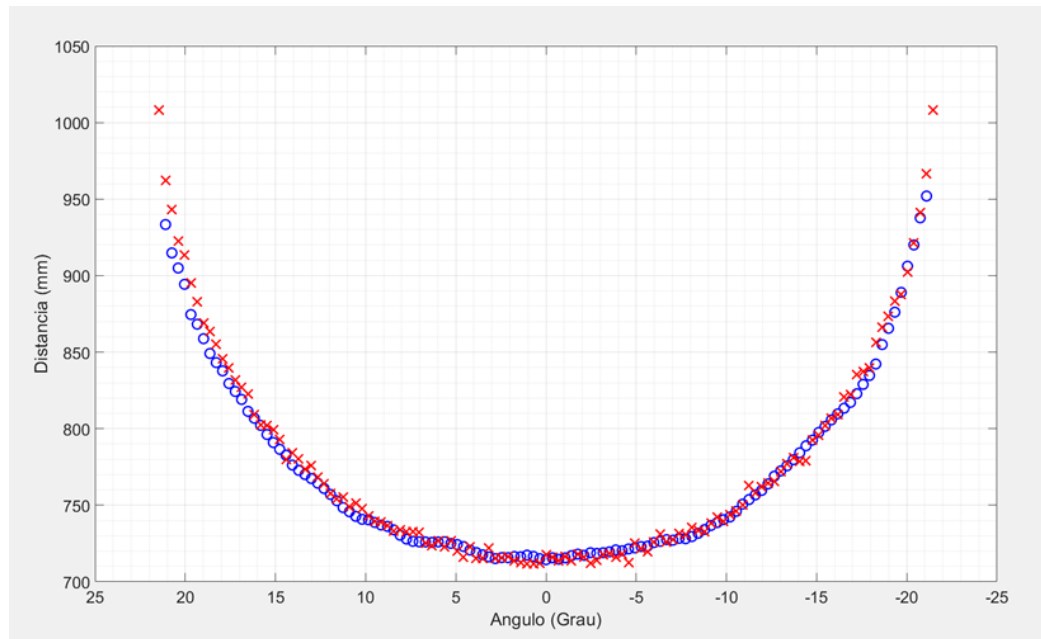


Figura 39 – Mapa de distância a 700 mm: Pontos de interseção do LiDAR ('o') e pontos de interseção do sensor virtual de modelo híbrido ('x').

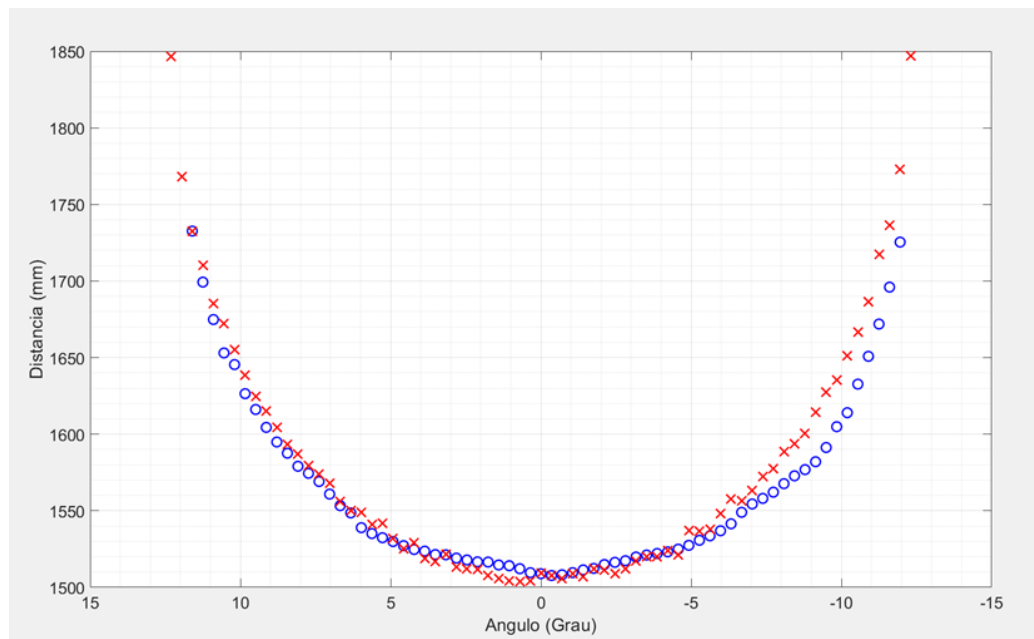


Figura 40 – Mapa de distância a 1500 mm: Pontos de interseção do LiDAR ('o') e pontos de interseção do sensor virtual de modelo híbrido ('x').

Obtidos estes mapas, repetiu-se o processo de cálculo dos RMSE pela diferença entre os dados de referência e os dados sintéticos, exibidos na Figura 41. Para posição do ângulo central a 700 mm, o RMSE encontrado foi 7,224 mm,

enquanto, para o ângulo central a 1500 m, o RMSE foi 14,1336 mm. Assim como na etapa 1, estes cálculos só levaram em consideração os raios que detectaram o alvo de teste.

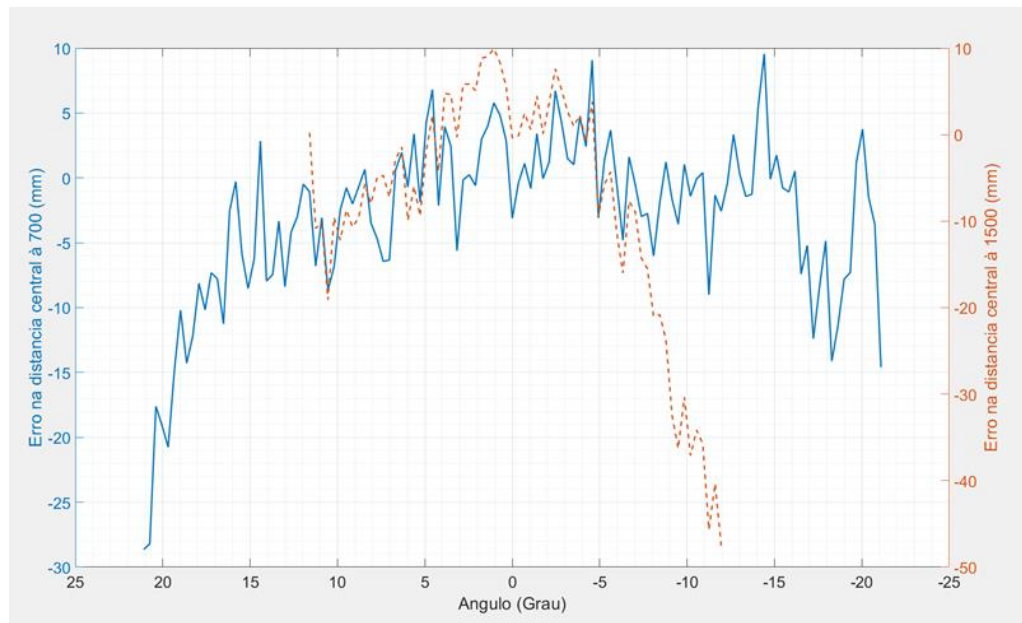


Figura 41 – Diferença entre dados reais e sintéticos gerados com o modelo híbrido. Linha contínua, erro na distância do ângulo central à 700 mm. Linha tracejada, erro na distância do ângulo central à 1500 mm

Essa validação do modelo de erros comprova os resultados e suposições levantadas na etapa 1. Observa-se que ao reativar as fontes de erros, o resultado da simulação praticamente se iguala ao resultado do escaneamento com LiDAR. Isto comprova a robustez de todo o simulador, principalmente do módulo Sensor Virtual, que mostrou ter um comportamento similar a um sensor LiDAR real. Ademais, todo esse processo funciona como um processo de calibração do sensor virtual por comparação com o sensor de referência, supondo que o mesmo tenha sido previamente calibrado corretamente.

## 5

### Aplicações em Cenários Típicos

Os dados na forma de nuvem de pontos têm tido um crescente interesse pelas mais diversas áreas. Estes datasets apresentam um mapa tridimensional de um alvo de interesse e, devido à alta resolução dos dispositivos sensores durante a aquisição de dados, as nuvens de pontos trazem características importantes do alvo, como danos em estruturas, por exemplo [1–6,7–10]. Este Capítulo apresenta o emprego do simulador que foi desenvolvido nesta dissertação e descrito nos Capítulos anteriores em algumas aplicações em cenários típicos indicados na literatura.

#### 5.1.

##### Especificações das Máquinas para Simulação:

Duas máquinas foram utilizadas nas simulações. Suas especificações seguem abaixo.

Máquina A:

- Processador: Intel Core i7 - 7500U.
- Memória RAM: 16 GB.
- Placa de vídeo: Nvidia GeForce 940 MX.

Máquina B:

- Processador: Intel Core i7 - 8700K.
- Memória RAM: 32 GB.
- Placa de vídeo: Nvidia GeForce GTX 1070 Ti 6GB.

#### 5.2.

##### Inspeção em Campo de Petróleo Submarino

A indústria de Óleo & Gás vem buscando soluções mais eficazes para a inspeção submarina feita em campos de petróleo **offshore**<sup>3</sup>, devido às condições hostis do ambiente submarino, que afetam a qualidade das imagens obtidas [12,13,57], como mostra a Figura 42.

---

<sup>3</sup> No meio marítimo



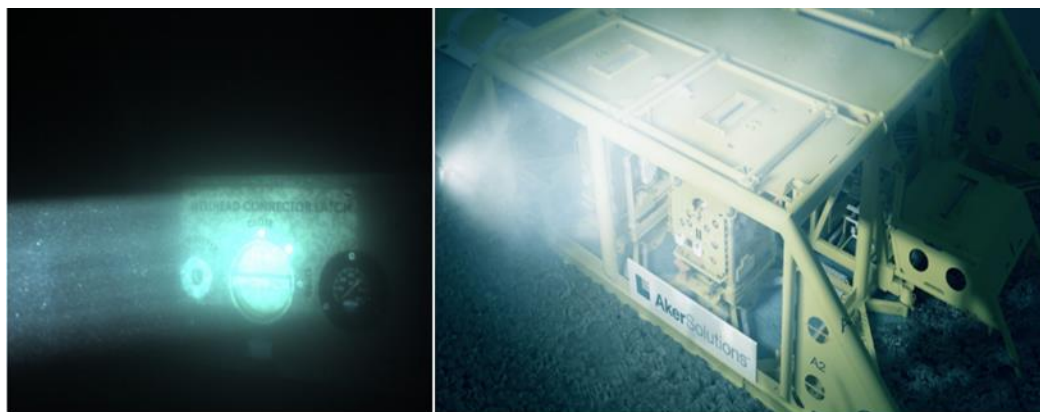


Figura 42 – Exemplo de imagens obtidas de equipamentos submarinos em uso [58,59].

Uma dessas soluções envolve o uso de tecnologias de imageamento 3D, em especial dispositivos LiDAR, para a geração de nuvens de pontos bem detalhadas, como exibido na Figura 43.

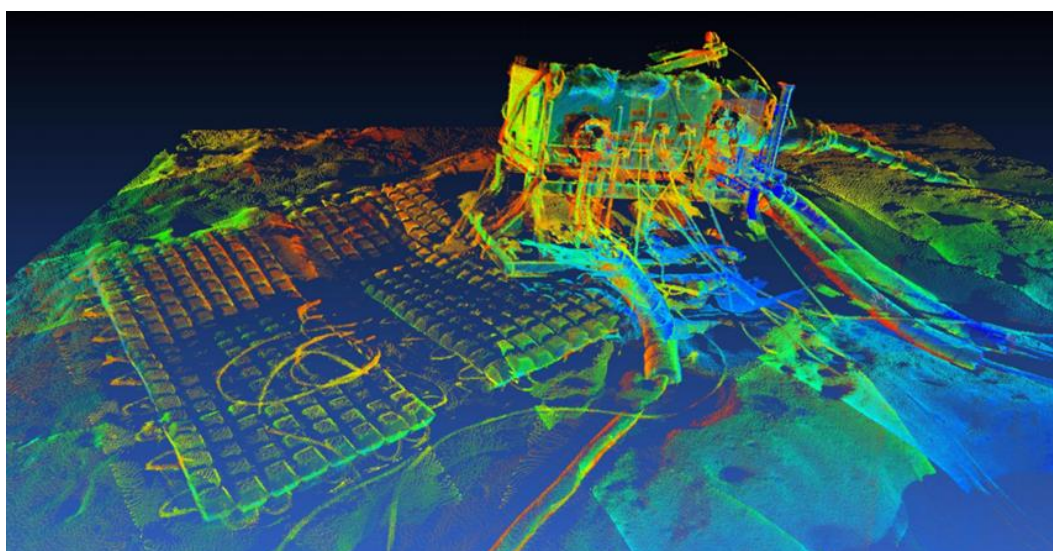


Figura 43 – Nuvem de pontos de equipamentos submarinos em um campo de petróleo offshore [38].

Com este tipo de dados é possível fazer comparações da evolução das condições do equipamento no campo de petróleo, assim como das condições do ambiente. Operações planejadas com estas informações reduzem significativamente custos e riscos [12,38,57–59].

No entanto, como mencionado, o fundo do oceano é um ambiente extremamente hostil, prejudicando a aquisição massiva de dados,



consequentemente inviabilizando a aplicação de técnicas de treinamento de IA que viriam a facilitar a análise das estruturas contidas nestas nuvens de pontos [53].

Para aumentar o acervo de datasets pode-se usar um simulador de escaneamento virtual, como o desenvolvido nesta dissertação. Primeiro, deve-se obter bons modelos de engenharia, o mais fidedignos possível dos alvos de interesse. Por exemplo, na Figura 44 tem-se uma Árvore de Natal Molhada (ANM), equipamento típico de um campo de petróleo submarino, bem como seu modelo digital equivalente.

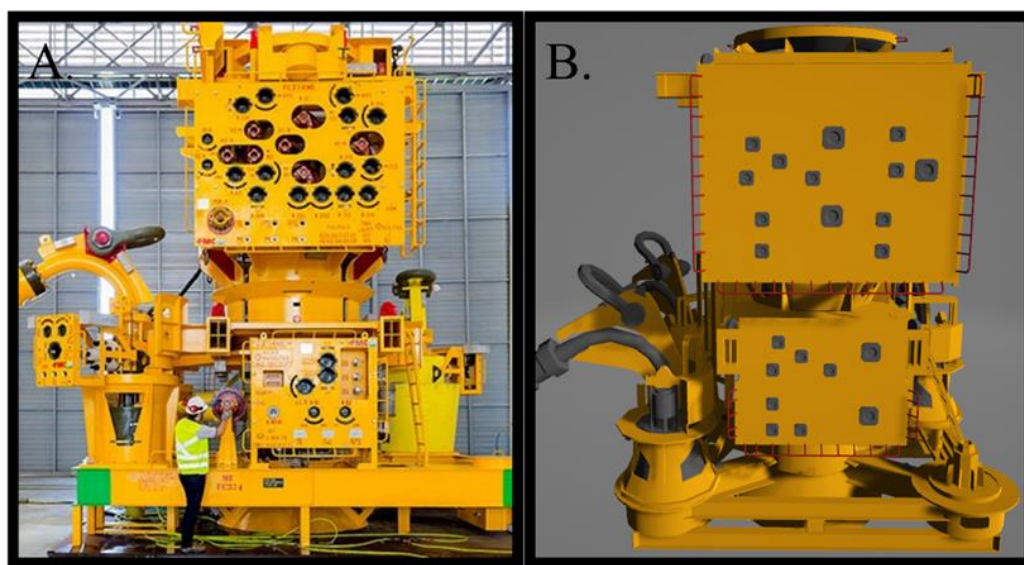


Figura 44 – (a) Árvore de Natal Molhada [60]. (b) Modelo 3D de uma ANM.

Com o uso do módulo Cenário Virtual, apresentado no Capítulo 4, foi construído um cenário simples para representação de uma pequena fração de um campo de petróleo offshore. Os modelos 3D usados na Figura 45 são: dois modelos de ANM; um manifold<sup>4</sup>; um caranguejo (desproporcionalmente aumentado para facilitar a visualização aqui); e o solo arenoso. Dentro deste cenário os sensores virtuais foram posicionados de forma a escanear os alvos em sua totalidade.

<sup>4</sup> Equipamento usado para simplificar o sistema submarino do campo de petróleo, reduzindo o número de dutos [53].

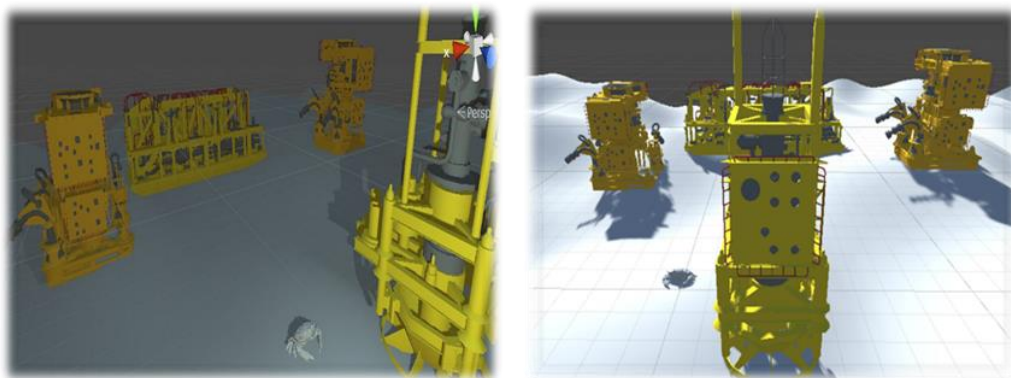


Figura 45 – Cenário Virtual de um campo de petróleo em diferentes ângulos e iluminação.

A execução dos próximos dois módulos do simulador resultou em uma série de nuvens de pontos equivalentes dos modelos 3D dos equipamentos submarinos. A Figura 46 mostra a nuvem de pontos do manifold, enquanto a Figura 47 mostra as nuvens de pontos de dois modelos de ANM.

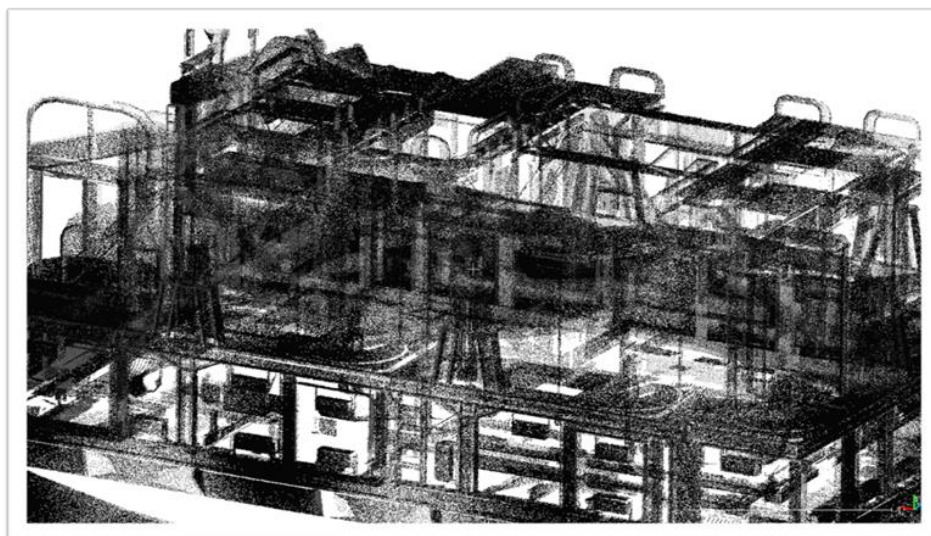


Figura 46 – Nuvem de pontos do Manifold.

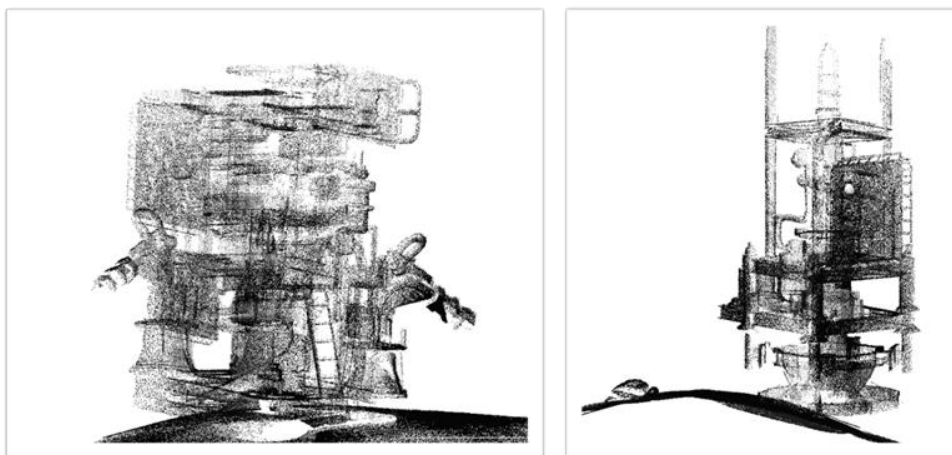


Figura 47 – Nuvens de pontos das ANMs.

Finalmente, a Figura 48 exibe a nuvem de pontos unificada de todo o alvo de interesse, com suas informações de classificação em evidência. O principal objetivo da geração desses dados sintéticos é a otimização da aquisição de dados em campo, pois a IA treinada com estes dados sintéticos indicará o tamanho necessário do dataset para treinamento. Sendo assim, a aquisição de dados pode ser organizada com base no total de dados necessários e não no tempo de medição [2,20,53].



Figura 48 – Nuvem de pontos do campo de petróleo virtual com informações de classificação.

### 5.3. Segmentação Semântica de Nuvem de Pontos para Deep Learning

A técnica de segmentação semântica é o processo de classificação de cada pixel de uma imagem em classes específicas [53,61]. Por exemplo, na Figura 49,

houve a segmentação de todos os objetos na imagem que podem ser classificados como pertinentes a classe “carro”.



Figura 49 – Segmentação semântica da classe “carro” em uma imagem [61]. Todos os objetos identificados como pertencente a esta classe estão em destaque.

Muitas tecnologias dependentes de visão computacional e de sensoriamento, como carros autônomos e sistemas robóticos, dependem deste tipo de resultado, já que as aplicações precisam ser capazes de interpretar o ambiente onde estão operando, normalmente com o uso de deep learning [2,6,17,20,53,60].

A segmentação semântica também pode ser aplicada no contexto de nuvens de pontos, só que, ao invés de se classificarem os pixels, são os pontos do dataset que recebem atribuições de classes [20,53]. A Figura 50 exibe uma nuvem de pontos com segmentação.



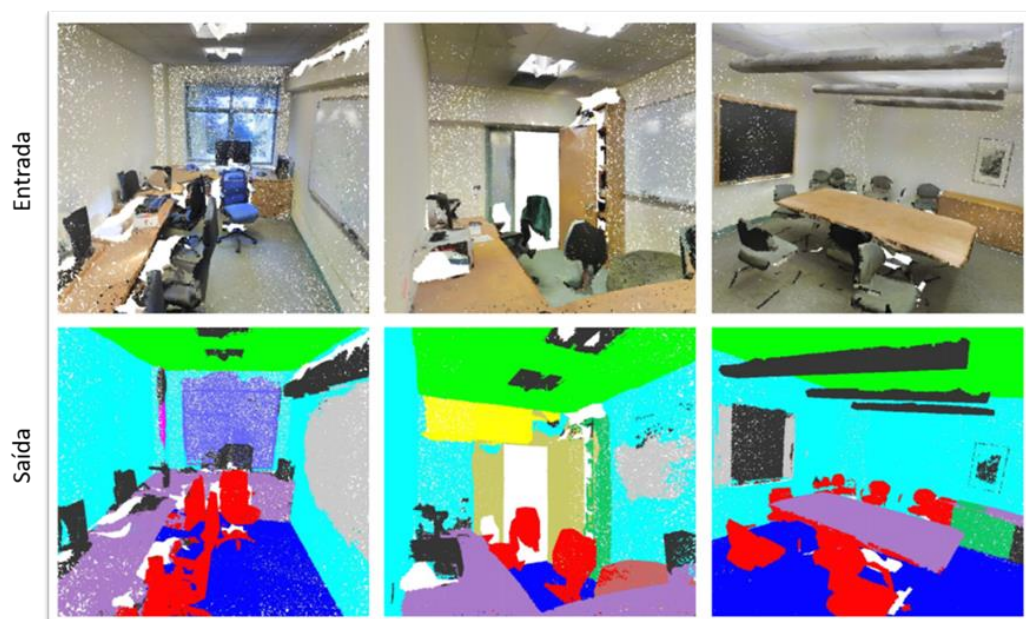


Figura 50 – Processo de segmentação semântica de nuvens de pontos feita pelo sistema PointNet [53]. A entrada é constituída por nuvens de pontos sem classificação e a saída são nuvens de pontos com segmentação. As diferentes cores indicam diferentes classes de objetos.

No entanto, devido às características geométricas das nuvens de pontos, como a invariância na permutação dos pontos do conjunto e a não ordenação dos elementos, o processo de classificação pode ser bastante complicado e demorado [53].

O simulador desenvolvido nesta dissertação gera as nuvens de pontos com níveis de classificação atribuídos a cada ponto. Este processo de anotação semântica é realizado indiretamente pelo usuário, passando as informações de classe no módulo Cenário Virtual, as quais são anotadas nos pontos detectados durante o escaneamento virtual.

Como exemplificação deste processo, um cenário de uma sala com vários objetos foi construído e foram definidas várias posições para o sensor virtual, como visto na Figura 51. Os objetos foram separados nas seguintes sete classes: Chão; Parede; Monitor/Televisão; Cadeira; Mesa; Sofá; e Estante.

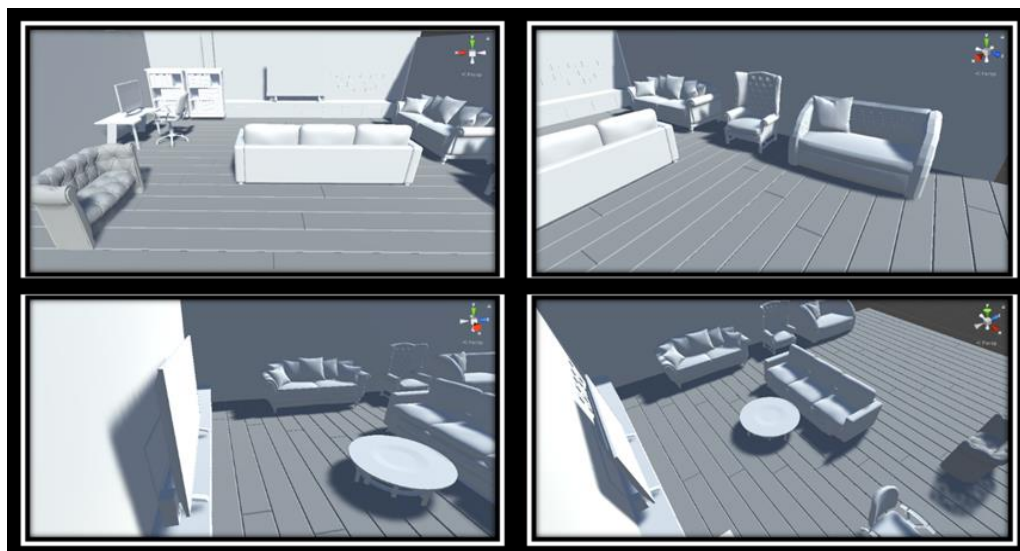


Figura 51 – Cenário virtual de uma sala de estar. Os quadros exibem quatro posições de um total de dezoito para o sensor virtual. Todos os modelos 3D foram obtidos de [62].

O processo completo da simulação resultou em uma nuvem de pontos completa do cenário alvo, com informações de segmentação semântica, exibida na Figura 52.

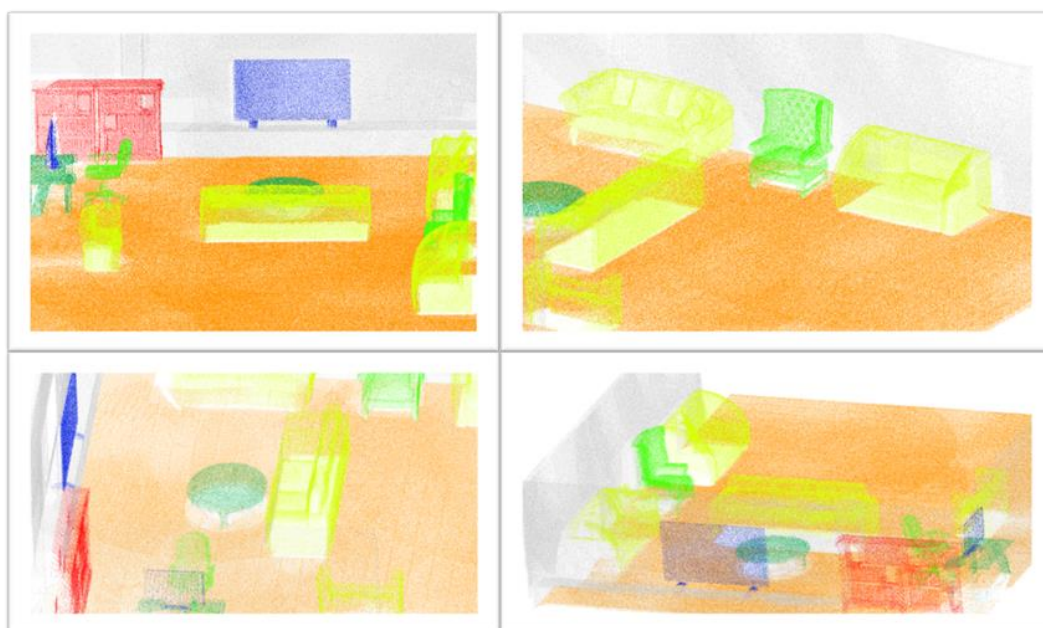


Figura 52 – Nuvem de pontos sintética gerada com segmentação semântica vistas em diferentes ângulos no visualizador Cloud Compare [63]. Classes por cores: Vermelho – Estante, Amarelo – Sofá, Verde escuro – Mesa, Verde claro – Cadeira, Azul – Monitor/TV, Cinza – Parede e Laranja – Chão.

Estes dados sintéticos gerados em conjunto com dados reais auxiliam no processo de treinamento de redes de deep learning para segmentação, aumentando a robustez do processo [2,20].

#### 5.4. Simulação para Escaneamento a Laser Aéreo (ALS)

Um dos principais usos dos dispositivos LiDAR é o mapeamento aéreo de grandes extensões de interesse, com o intuito de obter informações topográficas, por exemplo de florestas e de zonas urbanas [3–5,15]. Isto é feito pelo método de escaneamento a laser aéreo (Airborne Laser Scanning), onde um ou mais sensores LiDAR acoplados a drones ou aeronaves sobrevoam a região alvo, obtendo informação dos feixes de laser refletidos de volta [3–5,64], como ilustrado na Figura 53.

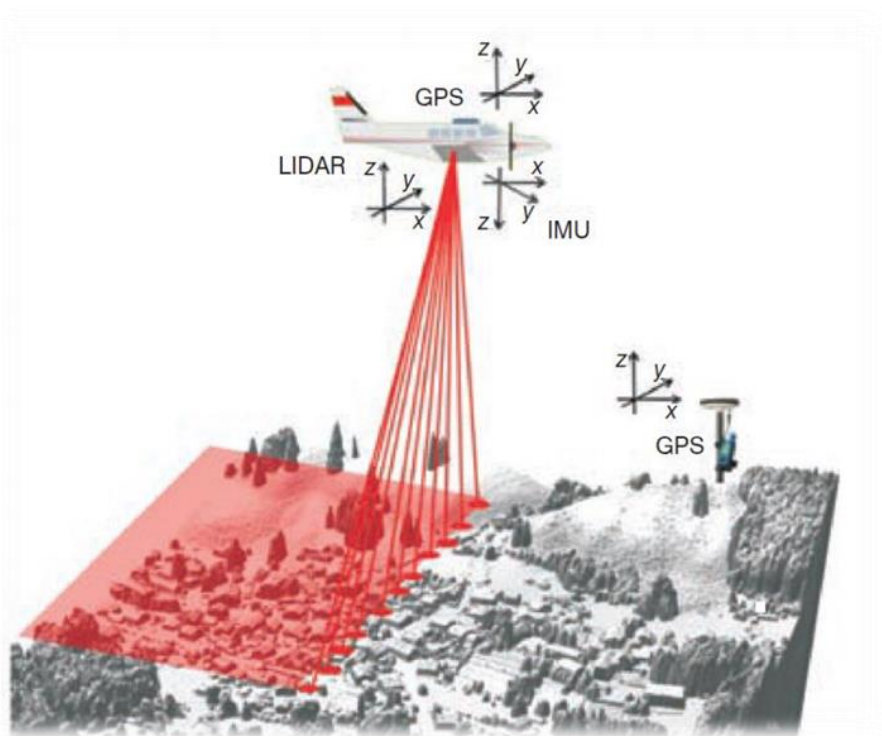


Figura 53 – Princípio básico de ALS. GPS e IMU ajudam a georreferenciar os dados [64].

No entanto, quando existe a necessidade de criar novas soluções para interpretação dos dados medidos pelo LiDAR, aquisições de dados em campo devem ser realizadas novamente. Devido aos problemas inerentes ao ALS, como risco ao equipamento e tamanho da região de interesse, o teste de novos métodos



em desenvolvimento é prejudicado. Esta necessidade promoveu a criação de simuladores de ALS baseados em LiDAR, capazes de gerar dados sintéticos para estudos dentro da literatura, principalmente na área de engenharia florestal, visando o cálculo de dossel florestal [3–5]. Estas aplicações normalmente usam dados reais para criação de modelos 3D mais robustos para os testes [3–5].

Um escaneamento virtual do modelo 3D<sup>5</sup>, apresentados na Figura 54, de uma área mapeada por um drone acoplado com dispositivos de imageamento 3D na região do Colégio Naval em Angra dos Reis, foi realizado para geração de dados topográficos.



Figura 54 – Modelo 3D de um mapeamento aéreo com drone.

Foi então definida uma série de 164 posições para escaneamento de uma sub-região do modelo com o sensor virtual, conforme exibido na Figura 55. Ao fim

---

<sup>5</sup> Dados cedidos pelo engenheiro Rogério Regazzi



da simulação, uma nuvem sintética unificada do alvo de interesse foi gerada, exibida em duas vistas nas Figuras 56 e 57.



Figura 55 – Zona de interesse para escaneamento virtual.



Figura 56 – Vista aérea da nuvem de pontos sintética exibindo informação de cores e topografia.



Figura 57 – Vista frontal da nuvem de pontos sintética exibindo informação de cores e topografia.

O simulador, com sua abordagem genérica, já é capaz de gerar nuvens de pontos sintéticas topográficas, o que já permite seu uso nestes estudos. Com as

devidas alterações no módulo Sensor Virtual, como a influência da velocidade de voo sobre o disparo dos feixes laser do sensor, pode-se adaptar o seu uso para ALS.

## 6

### **Discussão, conclusões e trabalhos futuros**

Nesta dissertação de mestrado em Metrologia foi apresentado um simulador modular com sensor de modelo híbrido baseado em LiDAR, capaz de gerar nuvens de pontos sintéticas. Este Capítulo final traz as discussões e conclusões advindas dos resultados obtidos na pesquisa, assim como sugestões de trabalhos futuros para aperfeiçoamento do simulador desenvolvido.

#### **6.1.**

##### **Discussão e Conclusões**

As tecnologias de imageamento 3D têm um uso cada vez mais disseminado, tanto no meio acadêmico quanto no setor industrial, principalmente com o crescente poder computacional, que permite o uso desses dispositivos até em forma portátil [2,6,9,12,15]. Apesar disso, a demanda massiva por dados de imageamento tornou necessário encontrar novas formas de aquisição para complementar estes dados, em especial em setores que trabalham com ambientes hostis ou muito extensos, como por exemplo o setor de óleo & gás no meio submarino e na arqueologia [12,22].

Como apresentado no Capítulo 3, o escaneamento virtual permite suprir esta demanda por dados, reduzindo custo e tempo de produção. No entanto, o foco nos setores automobilístico e robótica trouxe soluções muito específicas para seus problemas, obrigando o desenvolvimento de novos simuladores sempre que uma nova necessidade é percebida.

A pesquisa desenvolvida aqui buscou preencher esta lacuna, propondo uma abordagem modular para o simulador, dando a seus componentes um certo grau de independência. Além disso, o simulador desenvolvido permite a geração de nuvens de pontos sintéticas com dados de classificação e gerados de forma automática, limitado apenas pelo poder computacional do equipamento onde ocorre a simulação.

O Capítulo 4 mostrou que algoritmos de rastreamento de raios são extremamente eficazes para simulação da técnica de tempo de voo, princípio fundamental dos dispositivos do tipo LiDAR. Também apresentou-se uma fusão entre os dois tipos de modelagem típicas dos sensores virtuais para escaneamento virtual, criando assim um modelo híbrido que traz mais robustez na geração da nuvem de pontos sintética, pois leva em consideração tanto o comportamento físico dos dispositivos LiDAR, quanto seus erros documentados. Ademais, o processo de validação descrito funciona como um método de calibração do dispositivo digital em referência a um LiDAR real.

Já no Capítulo 5 foi exibido o emprego da técnica em cenários variados, mostrando a versatilidade da criação de cenários com o simulador. Esta facilidade de produção de cenários virtuais, juntamente com a característica modular do simulador, permite uma fácil adaptação para atender a novas necessidades do usuário.

Desta maneira, o simulador desenvolvido nesta dissertação de Metrologia mostrou-se eficaz e robusto na geração automática de datasets de nuvens de pontos com identificadores de classificação com um sensor virtual calibrado por um sensor de sistema LiDAR.

Espera-se que, com mais algumas melhorias, o simulador seja capaz de incluir a influência do ambiente sobre o escaneamento, permitindo a simulação de meios hostis, como no fundo do mar ou com condições climáticas desfavoráveis (neblina, neve, chuva). Estas possíveis melhorias encontram-se descritas na próxima seção.

## **6.2. Trabalhos Futuros**

Apesar da robustez da geração de nuvem de pontos sintética pelo simulador, existem algumas maneiras de aperfeiçoá-lo. Algumas abordagens podem ser feitas inicialmente.

### **1) Generalização do modelo de erro:**

A construção atual deste modelo foi realizada com base no sensor disponível para pesquisa. Idealmente, o processo de criação da equação

de erro deveria ser repetido para vários outros modelos de LiDAR, na tentativa de obter uma generalização do modelo.

Outras fontes de erros podem ser observadas na construção de um dispositivo LiDAR, pois, dependendo do uso, a própria eletrônica e temperatura afetam o processo de detecção da luz de retorno. Também existem problemas inerentes à tecnologia, como situações onde o raio retorna na direção da sua fonte e não para o detector (backscattering) [10,12].

## 2) Sensor Virtual baseado em CCD:

Outra possibilidade de melhoria, seria o acréscimo de um sensor virtual baseado em um dispositivo de carga acoplada (CCD). Estes dispositivos são capazes de obter imagens de alta resolução, normalmente em aplicações médicas, metrológicas ou visão computacional [65]. Sendo assim, o sensor virtual baseado em CCD, passaria a ser responsável pela obtenção de imagens, que processadas junto com os dados gerados da simulação do sensor virtual baseado em LiDAR, trariam informações de cores do cenário virtual obtidas de uma maneira mais fidedigna com escaneamentos reais, já que o sensor LiDAR não é capaz de obter informações de cores.

## 3) Modelo de comportamento da luz:

Como o sensor do sistema LiDAR é baseado em uma fonte luminosa (laser), a trajetória dos feixes disparados será influenciada pelas propriedades ópticas dos objetos e do meio onde o sensor se encontra [66]. Este feixe pode sofrer reflexões múltiplas, como na Figura 58a, ou atravessar um meio com muitos particulados suspensos e sofrer um espalhamento, mudando drasticamente a trajetória do feixe, como na Figura 58b. O material também pode ser altamente absorvente, para todas ou uma faixa específica de comprimentos de onda, não gerando informação de retorno, já que toda a energia do feixe disparado é absorvida, como é o caso da máscara revestida de vantablack, o material mais escuro criado pelo ser humano [67], na Figura 58c.

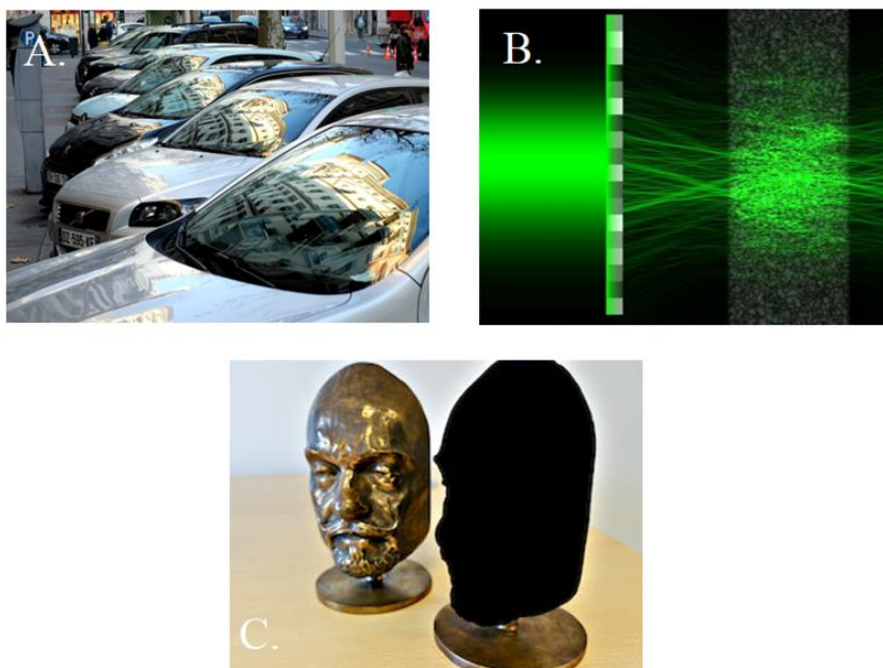


Figura 58 – Propriedades Ópticas. (a) Reflexão [68]. (b) Espalhamento [69]. (c) Absorção [67].

Estas propriedades podem influenciar a saída do sensor LiDAR, gerando uma falsa medição de um ponto, chamada de artefato, ou ausência de informação, chamada de buraco, na nuvem de pontos.

Logo, a modelagem desses fenômenos no simulador traria uma nova gama de resultados, além de aumentar a robustez dos dados gerados.

O estudo da literatura indicou duas abordagens para esta modelagem [7,70–72]:

- a. Teorias físicas de espalhamento e absorção da luz: Como na maioria dos escaneamentos com dispositivo LiDAR, as partículas responsáveis por absorção e espalhamento são muito maiores que o comprimento da luz, a solução de Mie para a equação de Maxwell pode ser usada. Devido a sua complexidade, a teoria não será apresentada aqui em detalhes. As simulações que usam a teoria de Mie tentam reproduzir fielmente a propagação dos fótons no meio de interesse. Apesar de ser um resultado poderoso, o custo computacional é muito alto [2].

Outra possibilidade é a criação de texturas para os modelos 3D com informações de reflexão e refração provenientes da lei de Snell [2,6].

- b. Modelos probabilísticos do método de Monte Carlo (MC): Com o uso de amostras aleatórias, o método estatístico de Monte Carlo pode calcular resultados tanto determinísticos quanto não determinísticos [72]. Para o problema proposto neste trabalho, o método pode ser usado para calcular a propagação dos fótons, dadas amostras aleatórias da posição inicial desses fótons dentro de um espaço de tempo definido previamente. Este tipo de solução pode ser mais rápido do que calcular a teoria de Mie, especialmente se feita uma abordagem de paralelismo na placa de vídeo (GPU) [72].



## Referências bibliográficas

- [1] ALLDÉN, T.; CHEMANDER, M.; DAVAR, S.; JANSSON, J.; LAURENIUS, R.; TIBOM, P. Virtual generation of Lidar data for autonomous vehicles. 2017. 64 p. Trabalho de conclusão de curso (Bachelor thesis in Data and Information technology) - Department of Computer Science and Engineering. University of Gothenburg, Chalmers University of Technology, Gothenburg, Suécia, 2017.
- [2] HANKE, T.; SCHAERMANN, A.; GEIGER, M.; WEILER, K.; HIRSENKORN, N.; RAUCH, A.; SCHNEIDER, S-A; BIEBL, E. **Generation and validation of virtual point cloud data for automated driving systems.** In **Proceedings...** IEEE 20th International Conference on Intelligent Transportation Systems, 2017
- [3] BENSON, M.; PIERCE, L.; SARABANDI, K. **Model-based estimation of large area forest canopy height and biomass using radar and optical remote sensing with limited lidar data.** In **Proceedings...** IEEE International Geoscience and Remote Sensing Symposium (IGARSS) in Fort Worth, Texas, 2017.
- [4] BREMER, M; WICHMANN, V; RUTZINGER, M. Calibration and validation of a detailed architectural canopy model reconstruction for the simulation of synthetic hemispherical images and airborne LiDAR data. **Remote Sensing, MDPI**, v.9, n.3, p. 220, 2017.
- [5] ÖHMAN, N. **Simulation of LiDAR data for forestry applications.** 2018. 29 p. Dissertação (Mestrado) - Department of Physics. Umeå University, Umeå, 2018.
- [6] TALLAVAJHULA, A. **Lidar simulation for robotic application development: modeling and evaluation.** 2018. 107 p. Dissertação (Doutorado) - The Robotics Institute. Carnegie Mellon University. Pittsburgh, 2018.

- [7] BLAIS, F. Review of 20 years of range sensor development. **Journal of Electronic Imaging**, v.13, n.1, p.231-240, 2004.
- [8] ADAMS, M D. Coaxial range measurement – current trends for mobile robotic applications. **IEEE Sensors Journal**, v.2, n.1, p. 2-13, 2002.
- [9] CHOW, J. C. K. **Multi-Sensor Integration for Indoor 3D Reconstruction**. 2014. 239 p. Dissertação (Doutorado) - Department of Geomatics Engineering. University of Calgary, Alberta, 2014.
- [10] GOKTURK, S. B.; YALCIN, H.; BAMLI, C. **A time-of-flight depth sensor – system description, issues and solutions**. In **Proceedings...** Conference on Computer Vision and Pattern Recognition Workshop in Wanshigton, 2004.
- [11] **LIDAR Pulsed Time of Flight Reference Design**. Texas Instruments Design, 2016 (revisado 2018).
- [12] OLIVER-CODINA, G; MASSOT-CAMPOS, M. Optical sensors and methods for underwater 3D reconstruction. **Sensors, MDPI**, v.15, n.12, p. 31525-31557, 2015.
- [13] ZIHAO Cui; ZHAOSHUO Tian; YANCHAO Zhang; ZONGJIE Bi; GANG Yang. **Research on the underwater target imaging based on the streak tube laser lidar**. In **Proceedings...** Young Scientists Forum in Shanghai, China, 2017.
- [14] FANLIN, Yang; DIANPENG, Su; YUE, Ma; CHENGKAI, Feng; ANXIU, Yang; MINGWEI, Wang; Refraction correction of airborne LiDAR bathymetry based on sea surface profile and ray tracing. **IEEE Transactions on Geoscience and Remote Sensing**, v. 55, n.11, p. 6141-6149, 2017.
- [15] LOHANI, B; MISHRA, R K. **Generating LIDAR data in laboratory: LIDAR simulator**. In **Proceedings...** ISPRS Workshop on Laser Scanning and SilviLaser in Espoo, Finlandia, 2007.
- [16] MICKS, A. E.; MYERS, S. V.; BANVAIT, H.; KADETOTAD, S. **Virtual sensor data generation for bollard receiver detection**. United States Patent No. US 2018/0011953 A1. Dearborn, USA: Ford Global Technologies, 2018.

- [17] XIANGYU Yue; BICHEN Wu; SESHIA, S; KEUTZER, K; SANGIOVANNI-VICENTELLI, A; **A LiDAR point cloud generator: from a virtual world to autonomous driving**. In **Proceedings...** the 2018 ACM on International Conference on Multimedia Retrieval, 2018.
- [18] MAJEK, K; BEDKOWSKI, J. **Range Sensors Simulation Using GPU Ray Tracing**. In **Proceedings...** the 9th International Conference on Computer Recognition Systems CORES in Cham, 2015.
- [19] PARKER, S G.; BIGLER, J; DIETRICH, A; FRIEDRICH, H; HOBEROCK, J; LUEBKE, D; MCALLISTER, D; MCGUIRE, M; MORLEY, K; ROBISON, A; STICH, M. OptiX: a general purpose ray tracing engine. **ACM Transactions on Graphics**, v.29, n.4, p.66, 2010.
- [20] WANG F; ZHUANG Y; GU H; HU H; Automatic Generation of Synthetic LiDAR Point Clouds for 3-D Data Analysis. **IEEE Transactions on Instrumentation and Measurement**, v. 68, n. 7, p. 2671-2673, 2019.
- [21] BÖHLER, W; 2005. **Comparison of 3D laser scanning and other 3D measurement techniques**. Recording, Modeling and Visualization of Cultural Heritage. In **Proceedings...** the International Workshop, Centro Stefano Franscini, Monte Verita, Ascona, Switzerland, p.89-99, 2006.
- [22] REMONDINO, F; Heritage Recording and 3D Modeling with Photogrammetry and 3D Scanning. **Remote Sensing**, v. 3, n. 6, p. 1104-1138, 2011.
- [23] SÁ, A M; DE MEDEIROS FILHO, E S; CARVALHO, P C P; VELHO, L. Coded Structured Light for 3D-Photography: An Overview. **RITA**, v. 9, n. 2, p. 203-219, 2002.
- [24] PAGE, D; KOSCHAN, A; VOISIN, S; ALI, N; ABIDI, M. 3D CAD model generation of mechanical parts using coded-pattern projection and laser triangulation systems. **Assembly Automation**, v. 25, n. 3, p. 230-238, 2005.
- [25] KUMAR, A; JAIN, P.K; PATHAK, P.M. Industrial application of point cloud/STL data for reverse engineering. **DAAAM International Scientific Book**, p.445-462, 2012.

- [26] SANSONI, G; TREBESCHI, M; DOCCHIO, F. State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation. **Sensors**, v. 9, n. 1, p. 568-601, 2009.
- [27] **3DFLOW**. 3DF Zephyr tutorials. Disponível em: <https://www.3dflow.net/technology/documents/3df-zephyr-tutorials/>. Acesso em: 9 dez. 2019.
- [28] HARYAMA, M; YOKOYAMA, N; KAMEYAMA, M. Design of a trinocular-stereo-vision VLSI processor based on optimal scheduling. **IEICE transactions on electronics**, v. 91, n. 4, p. 479-486, 2008.
- [29] KIM, H.J; MAHMOOD, M; CHOI, T.S. An efficient neural network for shape from focus with weight passing method. **Applied Sciences**, v. 8, n. 9, p. 1648, 2018.
- [30] KO, J; HO, Y.S. 3d point cloud generation using structure from motion with multiple view images. In **Proceedings...** The Korean Institute of Smart Media Fall Conference, p. 91-92, 2016.
- [31] WESTOBY, M.J; BRASINGTON, J; GLASSER, N.F; HAMBREY, M.J; REYNOLDS, J.M., 2012. 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. **Geomorphology**, v. 179, p. 300-314, 2012.
- [32] **Physics 251**. Experimental Atomic Physics. Disponível em: [https://ixnovi.people.wm.edu/phys251/web\\_manuals/interferometry.pdf](https://ixnovi.people.wm.edu/phys251/web_manuals/interferometry.pdf). Acesso em: 14 jan. 2020.
- [33] **Robotics Tomorrow**. What is Structured Light Imaging? Disponível em: <https://www.roboticstomorrow.com/article/2018/04/what-is-structured-light-imaging/11821>. Acesso em: 14 jan. 2020.
- [34] **Julight**. Disponível em: <http://www.julight.it/laser-triangulation/>. Acesso em: 10 dez. 2019.
- [35] YOON, H; SONG, H; Park, K. A phase-shift laser scanner based on a time-counting method for high linearity performance. **Review of Scientific Instruments**, v.82, n.7, 2011.

- [36] GUSMÃO, G. F.; HALL BARBOSA, C. R.; RAPOSO, A. B.; OLIVEIRA, R. C. A LiDAR system simulator using parallel raytracing and validated by comparison with a real sensor. In **Proceedings... X Congresso Brasileiro de Metrologia**, Florianopolis. Anais do X Congresso Brasileiro de Metrologia, 2019.
- [37] **Velodyne Lidar**. Disponível em: <https://velodynelidar.com/index.html>. Acesso em: 11 dez. 2019.
- [38] **3D at Depth**. Disponível em: <https://www.3datdepth.com/>. Acesso em: 11 dez. 2019.
- [39] NUTTENS, T; DE MAEYER, P; DE WULF, A; GOOSSENS, R; STAL, C. 2011. **Comparison of 3D accuracy of terrestrial laser scanning and digital photogrammetry: an archaeological case study**. In **Proceedings... Remote sensing and geoinformation not only for scientific cooperation: 31st EARSeL symposium**, p. 66-74, 2011.
- [40] **NASA**. 3D View of Amazon Forests Captures Effects of El Niño Drought. Disponível em: <https://www.nasa.gov/feature/goddard/2018/3d-view-of-amazon-forests-captures-effects-of-el-ni-o-drought>. Acesso em 12 dez. 2019.
- [41] VON SCHWERIN, J; RICHARDS-RISSETTO, H; REMONDINO, F; SPERA, M; AUER, M; BILLEN, N; LOOS, L; STELSON, L; REINDEL, M. Airborne LiDAR acquisition, post-processing and accuracy-checking for a 3D WebGIS of Copan, Honduras. **Journal of Archaeological Science: Reports**, v. 5, p. 85-104, 2016.
- [42] **Jason Stoker**. Public Domain. Disponível em: <https://www.usgs.gov/media/images/lidar-point-cloud-washington-dc-0n>. Acesso em: 17 dez. 2019.
- [43] **Marine Technology News**. Subsea Wreck Brought to Life by Lasers. Disponível em: <https://www.marinetechologynews.com/news/subsea-wreck-brought-lasers-587916>. Acesso em: 16 jan. 2020.
- [44] GEIGER, A; LENZ, P; STILLER, C; URTASUN, R. Vision meets robotics: The kitti dataset. **The International Journal of Robotics Research**, v. 32, n. 11, p. 1231-1237, 2013.

- [45] **The Graphics File Formats Page.** Disponível em: <http://www.martinreddy.net/gfx/>. Acesso em: 16 jan. 2020.
- [46] **Gazeta Esportiva.** Fernando Dantas/Gazeta Press. Disponível em: <https://www.gazetaesportiva.com/motor/site-revela-rompimento-da-ferrari-com-patrocinador-de-100-milhoes/>. Acesso em: 16 jan. 2020.
- [47] **Free3D.** Ferrari Formula 1 3d Model Submitted by dill\_afroze. Disponível em: <https://free3d.com/3d-model/ferrari-formula-1-72527.html>. Acesso em: 16 jan. 2020.
- [48] CONSTANTINI, R; SUSSTRUNK, S; **Virtual sensor design.** In **Proceedings...** Electronic Imaging in San Jose, California, United States, 2004.
- [49] HILL, D J; YONG L; MARINI, L; KOOPER, R; Rodriguez, A; FUTRELLE, J; MINSKER, B; MYERS, J; MCLAREN, T. A virtual sensor system for user-generated, real-time environmental data products. **Environmental Modelling & Software**, v.26, n.12, p. 1710-1724, 2011.
- [50] GATTASS, M. **Notas de aula sobre computação gráfica interativa.** Rio de Janeiro: Departamento de Informática, PUC-Rio, 2013.
- [51] KIRK, David B.; HWU, Wen-mei W. **Programming Massively Parallel Processors: 3.** Ed. Cambridge: Morgan Kaufmann, 2017.
- [52] **Unity3D.** Disponível: <https://unity.com/>. Acesso em: 28 dez. 2019.
- [53] QI, C R; SU, H; MO K; GUIBAS, L. J. **Pointnet: Deep learning on point sets for 3d classification and segmentation.** In **Proceedings...** IEEE Conference on Computer Vision and Pattern Recognition, p. 652-660, 2017.
- [54] **Petrobras.** Conheça curiosidades sobre equipamentos de nossos sistemas submarinos. Disponível em: <https://petrobras.com.br/fatos-e-dados/conheca-curiosidades-sobre-equipamentos-de-nossos-sistemas-submarinos.htm>. Acesso em: 6 fev. 2020.
- [55] **ALICE project-team.** Geogram. Disponível em: <http://alice.loria.fr/index.php/home.html>. Acesso em: 23 jun. 2019.
- [56] **Hokuyo.** Rangefinder URG-04LX-UG01. Disponível em: <https://www.hokuyo-aut.jp/search/single.php?serial=166>. Acesso em: 23 jun. 2019.

- [57] SHEHHI, S. A; SAYED, T. A; HAMMADI, H. A; MUKHMARI, H. A; DHANHANI, K. A; KALLINKIL, S. Photo Realistic 3D Cloud of Points of Subsea Structures Utilizing ROV. In **Proceedings...** Abu Dhabi International Petroleum Conference. Society of Petroleum Engineers, 2016.
- [58] DETRY, R.; KOCH, J; PAILEVANIAN, T; GARRET, M; LEVINE, D; YAHNKER, C; GILDNER, M; 2018, May. Turbid-water subsea infrastructure 3D reconstruction with assisted stereo. In **Proceedings...** 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO), p. 1-6., 2018.
- [59] **Aker Solutions**. Disponível em: <https://www.akersolutions.com/what-we-do/products-and-services/subsea-structures/>. Acesso em: 02 fev. 2020.
- [60] **TN Petróleo**. Árvore de Natal Submarina da FMC Technologies Brasil recebe Certificado de Conteúdo Local. Disponível em: <https://tnpetroleo.com.br/noticia/arvore-de-natal-submarina-da-fmc-technologies-brasil-recebe-certificado-de-conteudo-local/>. Acesso em: 6 fev. 2020.
- [61] **Heartbeat**. A 2019 Guide to Semantic Segmentation. Disponível em: <https://heartbeat.fritz.ai/a-2019-guide-to-semantic-segmentation-ca8242f5a7fc>. Acesso em: 6 fev. 2020.
- [62] **Free3D**. Disponível em: <https://free3d.com/>. Acesso em: 6 fev. 2020.
- [63] **Cloud Compare**. 3D point cloud and mesh processing software. Disponível em: <https://www.danielgm.net/cc/>. Acesso em: 6 fev. 2020.
- [64] JOKELA, J. **CityGML building model production from airborne laser scanning**. 2016. 64 p. Dissertação (Mestrado) – School of Engineering. Aalto University. Espoo, 2016
- [65] **TC281 CCD Image Sensor Data Sheet (Rev. D)**. Texas Instruments, 2003.
- [66] HULST, Hendrik C. van de; **Light Scattering by Small Particles**. New York: Wiley, 1957.
- [67] **Surrey nanosystems**. Disponível: <https://www.surreynanosystems.com/about/vantablack>. Acesso em 24 out. 2019.

- [68] **Weekend Reflections.** Disponível em: <https://weekendreflection.blogspot.com/2017/05/multiple-reflections-in-lyon-weekend.html>. Acesso em: 24 out. 2019.
- [69] **Phys.org.** Disponível em: <https://phys.org/news/2019-03-laser-opaque-media.html>. Acesso em: 24 out. 2019.
- [70] CARLES, G; HARVEY, A R. **Monte-Carlo simulation of imaging systems with turbid media through optical ray-trace.** In **Proceedings...** Imaging and Applied Optics Congress, 2016.
- [71] GOVAERTS, Y M; VERSTRAETE, M M. Raytran: a Monte Carlo ray-tracing model to compute light scattering in three-dimensional heterogeneous media. **IEEE Transactions on Geoscience and Remote Sensing**, v. 36, n. 2, p. 493-505, 1998.
- [72] QIANQIAN, Fang; Mesh-based Monte Carlo method using fast ray-tracing in Plücker coordinates. **Biomed Opt Express**, v.1, n.1, p. 165-175, 2010.



## Anexo A – Artigo CBM 2019

### A LiDAR System Simulator Using Parallel Raytracing and Validated by Comparison with a Real Sensor

G. F. Gusmão<sup>12</sup>, C. R. H. Barbosa<sup>1\*</sup>, A. B. Raposo<sup>2</sup>, R. C. de Oliveira<sup>2</sup>

<sup>1</sup>Postgraduate Program in Metrology, PUC-Rio, R. Marquês de São Vicente, 225, Gávea, Brazil; <sup>2</sup>Tecgraf/PUC-Rio, R. Marquês de São Vicente, 225, Gávea, Brazil.

\*Email: <hall@puc-rio.br>

**Abstract.** The advances in 3D imaging and Computer Vision are allowing for a massive acquisition of data, especially in the form of point clouds. However, scanning with these sensors can be costly and time consuming, reducing the efficiency of certain procedures, like deep learning dataset generation and forest canopy calculation. One trending solution is the creation of 3D imaging sensor simulators. This paper presents a simulator for light detection and ranging (LiDAR) systems with a parallel raytracing approach and a flexible scene creator. Finally, the simulator sensor is validated with a real LiDAR data.

#### 1. INTRODUCTION

There is a rising interest on the use of 3D imaging sensors by both academy and industry, especially now that computing power allows processing of massive datasets, even in mobile form factors. These type of sensor systems permit the gathering of geospatial information from a desired target, usually resulting in a dense point cloud [1-3].

However, the process of scanning is still time consuming and expensive, depending on the size of the target, with hours of field work required and hazards. This hinders applications that are data intensive, like training of deep learning models [4-7].

One solution seeing a lot of development is the creation of 3D imaging sensors simulators, especially for light detection and ranging (LiDAR) systems. These simulators are able to, given 3D models in a virtual scene, generate a synthetic point cloud. As the virtual world doesn't have the same restrictions of the physical world, a user could generate a great volume of datasets in days, in contrast to weeks in field, with reduced costs and risks [4,5,7].

The automobile industry concentrates the most prominent researchers of this solution, focusing on deep learning models for autonomous cars [7-10]. One work [7] shows the benefits of using a ray tracing approach to simulate the LiDAR sensor behavior in a simulator for generating deep learning dataset for autonomous vehicles. Recently [10], a similar approach used a car simulator called Carla to generate synthetic point clouds with "semantic labels" of the detected obstacles.

As most of these researches focus on autonomous vehicles, there is a lack of more general simulators in the literature. In this paper, it is presented a LiDAR system based simulator with a parallel ray tracing approach and a scene builder.

## 2. SIMULATOR

The simulator was developed with NVIDIA OptiX ray tracing engine, described as “a programmable system designed for NVIDIA GPUs and other highly parallel architectures” [11]. The engine enables a varied set of ray tracing algorithms and applications.

The simulator can be divided in two parts, *the scene* and *the virtual sensor*. The used workflow is shown in Fig. 1.

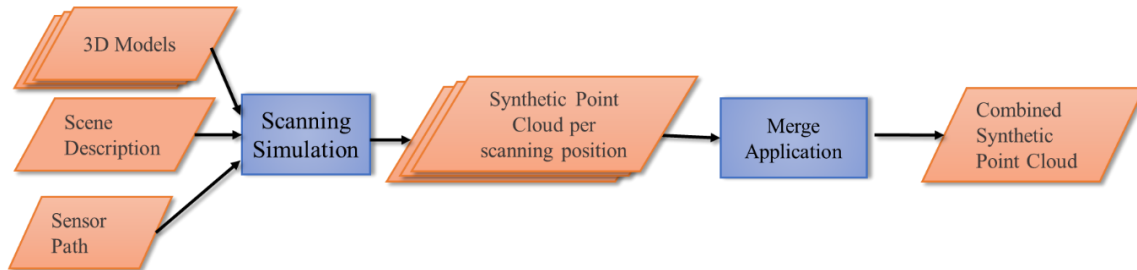


Figure 1: System workflow – parallelograms are data files and squares are applications.

### 2.1. Virtual Sensor

The scene is constructed loading a scene description file, 3D models files of desired targets and a sensor path file into the application.

The scene description contains information about scale, rotation, position, and identification number of the 3D models in the virtual world. The identification number is a unique number assigned for each object in the virtual world. For example, if a forest area is created: the terrain, the trees and rocks could each be marked with an individual identification number. This information can be passed for each point in the output point clouds, as seen in Fig. 2. Many procedures benefit from this type of data structure, especially segmentation [10,15]

The 3D model is a file that contains a 3D object including coordinates, texture maps, among other information. The used format is OBJ [17].

The sensor path is a file with coordinates of where to position the sensor for each scanning shot. The file contains 3D position and orientation.

This flexible virtual environment permits creating point clouds from simple objects, outdoors, indoors or even hazardous environment, like subsea survey.

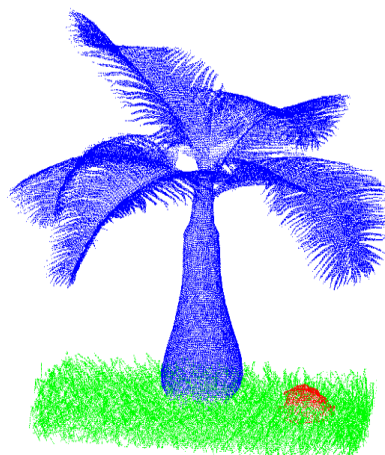


Figure 2: The synthetic point cloud of the forest area. The points are colored based on their objects' identification number by a point cloud viewer.

## 2.2. Virtual Sensor

A virtual sensor is a mathematical representation, an algorithm, which emulates a real sensor's behaviour [12]. For the simulator described in this paper, it was used the definition “*sensor measurement model*”, described in [7] as “models based on the measurement process and generating low level data from the built scene”.

### 2.2.1. LiDAR:

LiDAR system is a ranging measurement device that generates, after the data gathered is processed by a computer, massive sets of geospatial points called point clouds. This system components are, usually, a GPS unit, an inertial measurement unit and an active sensor based on a laser and a photodetector [1,2,3,13].

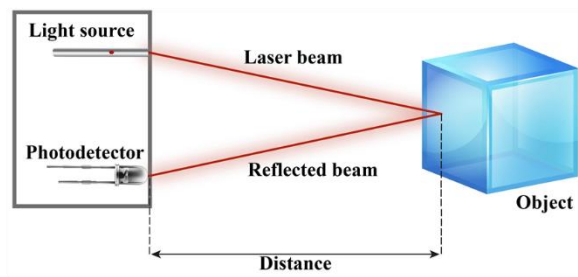
The LiDAR sensor's working principle is based on time of flight, i.e., the time a laser beam shot from the sensor takes to return to the photodetector, as shown in Fig. 3a. This time is used to calculate the distance of the beam travelling path as

$$distance = \frac{time\ of\ flight * light\ speed}{2} . \quad (1)$$

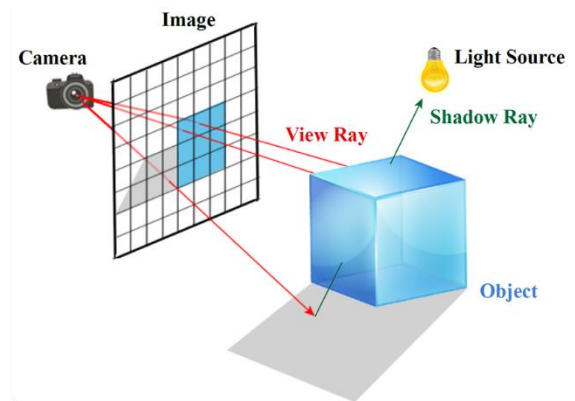
The sensor has an operational scanning angle and distance range. The sensor shoots successive rays, separated by the step angle, within the operational range. Distance range is the distance interval where the LiDAR manufacturer guarantees the measurements will be within the expected measurement uncertainty.

### 2.2.2. Sensor Model:

The literature shows that the ray tracing algorithm is a good starting point to simulate the LiDAR sensor behavior [4-10]. The idea behind the algorithm [11] is tracing the ray's path from an eye (or camera) to a pixel in the screen, returning the color and distance information of the object hit by the ray, as shown in Fig 3b.



(a)



(b)

Figure 3 : (a) Time of flight principle of a LiDAR. (b) Ray tracing.

For the purposes of this work, the rays shot in the OptiX engine were modeled considering the LiDAR characteristics highlighted in the previous subsection. The ray tracing now follows an angle range, shooting rays with a defined step angle and within a distance range.

This yielded a realistic generation of synthetic point clouds, as the simulator virtual sensor is modeled by real LiDAR specifications.

### 2.2.3. Synthetic point cloud generation:

The point cloud data is obtained from the collision of each ray with an object in the virtual world. When this occurs, the program calculates the intersection distance, position, normal vector and color of the surface hit point, plus the identification number of the object.

*3D coordinates* are the position of the point in the world. It's the basic information of a point cloud. However, these coordinates are not geospatial, only corresponding to the scene built in the simulation.

*Normal vectors* are vectors perpendicular to the surface of an object. Like the *identification numbers*, previously defined in section 2.1, they are very useful for many approaches in training deep learning models and segmentation routines [15].

*Color*, obtained in the form of RGB vectors, helps verifying if the point cloud corresponds to its target 3D model, although LiDAR systems usually don't have this kind of information.

*Intersection distance* is the raw information from a LiDAR sensor. Combined with rays' directions, the LiDAR system calculates the 3D coordinates and builds the point cloud [3,7,10].

As described in section 2.1, these synthetic point clouds are taken from a series of shots following the instructions from the sensor path file. The application generates as many point clouds as there are scanning positions.

Nevertheless, some procedures [4,5,15], need a point cloud from the entire or a big part of target model. Using Geogram [14], a programming library of geometric algorithms, an application was developed to merge the single viewpoint clouds in a combined point cloud. It removes redundant points (points which 3D coordinates coincide or are too close to one another) based on a defined proximity threshold. The combined point cloud can also be normalized to be limited by a cube with an edge 2 units. The combined point cloud, as shown in Fig. 4, can then be saved as a XYZ or PLY file.

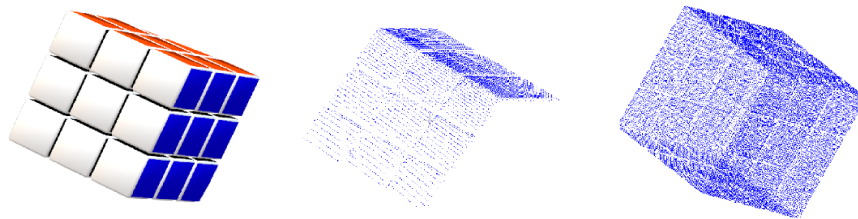


Figure 4: On the left, a 3D model of a Rubik Cube. On the middle, the single view point cloud. On the right, the combined point cloud.

## 3. VALIDATION

The measured intersection distance of the virtual sensor model was compared to the intersection distance measured by a LiDAR sensor from a target object to verify if the virtual sensor behaviour was coherent with the LiDAR specifications. The LiDAR sensor URG-04LX-UG01 from Hokuyo was used for acquisition of data. It has an infrared laser (wavelength 785 nm) with a horizontal scan area of 240° (angle resolution approx. 0.3516°) and no vertical scan area. It has two operational distance ranges: from 60 mm to 1000 mm with measurement uncertainty of 30 mm; and from 1000 mm to 4000 mm with measurement uncertainty of 3% of detected distance.

For the target object a soda can was chosen, shown in Fig. 5a, as it has a good reflective material and simple geometry. The middle of the can was positioned at 650 mm and 1300 mm from the LiDAR sensor in front of an infrared non-reflective wall, so no background information was gathered. For the simulator, the scene was based on the real target area and the virtual sensor configured with the Hokuyo LiDAR sensor specifications. The 3D model used was a can with its body covered by a cylinder mathematically generated with the same diameter as the soda can, as shown in Fig 5b. Both virtual sensor and LiDAR sensor were positioned to hit only the middle height of their respective cans.

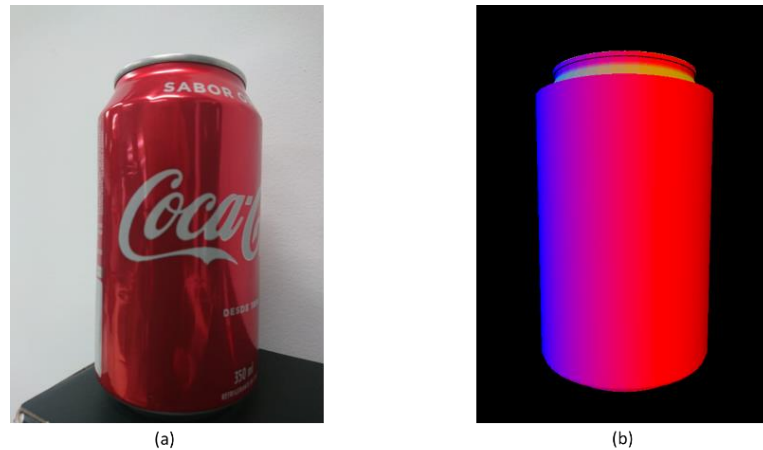


Figure 5: Validation Experiment: (a) 350 ml soda can; (b) 3D model of a 350 ml soda can covered with a cylinder to smooth the surface.

These scanning experiments resulted in four range maps plotted with the measured intersection distances and shooting angle, two from the LiDAR and two from the virtual sensor. Fig. 6 shows the range maps at distance 1300 mm and Fig. 7 shows the range maps at distance 650 mm. LiDAR measured distances are marked as 'o' and virtual sensor measured distance are marked as 'x'.

To calculate the root mean square error (RMSE), the difference between the virtual and real sensors data was used, shown in Fig. 8. For the can positioned at 650 mm, the RMSE was 33.36 mm. For the can positioned at 1300 mm, the RMSE was calculated only for successful rays returned to the LiDAR, as the sensor missed information from two rays. The RMSE at position 1300 mm was 7.16 mm.

These are good indications of the virtual sensor behaviour, as the simulator scene has no noise and background information from the environment was reduced for the real target, so the error between them is expected to be within the measurement uncertainty of the Hokuyo' LiDAR. In fact, both RMSEs calculated are within the uncertainty range.

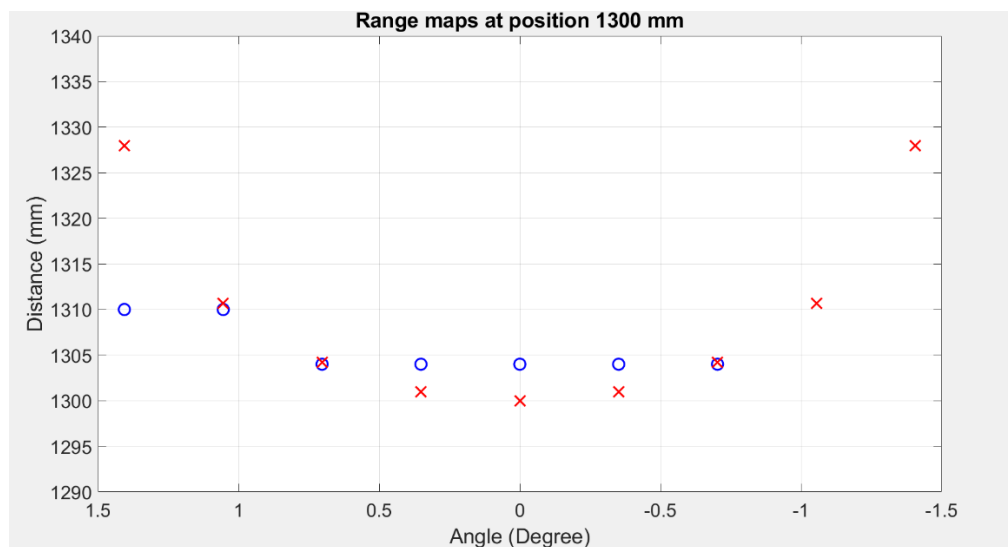


Figure 6: Range maps from the Hokuyo URG-04LX-UG01 LiDAR ('o') and the virtual sensor ('x') at position 1300 mm (distance vs ray shooting angle).

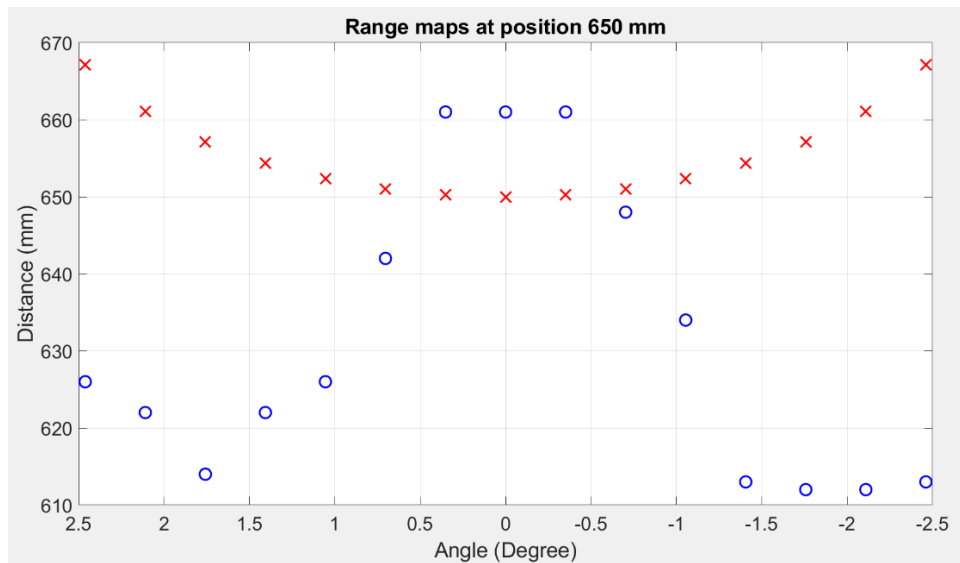


Figure 7: Range maps from the Hokuyo URG-04LX-UG01 LiDAR ('o') and the virtual sensor ('x') at position 650 mm (distance vs ray shooting angle).



Figure 8: Difference between real and virtual range maps at position 650 mm (dashed line) and at position 1300 mm (solid line)

The results from the validation experiments verifies that the virtual sensor follows the scanning behaviour of the LiDAR sensor it was modelled after, as the rays hit the surface of the can at the same ray shooting angle and the difference between measurements are within the uncertainty of the real sensor. However, even though all the affirmations above are true, some interesting phenomena appear in Fig. 6 and Fig. 7.

In Fig. 6, the LiDAR didn't receive information from two rays, meaning that the rays that hit the surface of the can were reflected away from the photodetector. Also, one of the successful rays hit was detected nearer than it truly is. Both issues show that the system has more difficult measuring the distance close to the curvature of the can. This can be related with known LiDAR problems, like backscattering and diffuse reflection, as they can cause the system to register a false data position, called artifacts, or no information at all, called holes [1-3,13].



In Fig. 7, the LiDAR sensor measurements shows an intensification of the problems observed in Fig. 6, as the distances measured from the can's curvature were even smaller than the real distances. This is expected from time of flight devices when measuring distances below 1000 mm [3,13], as the light returns too fast for the device photodetector to register the information correctly.

#### 4. CONCLUSION AND FUTURE WORKS

This manuscript presented a LiDAR system based simulator able to consistently generate synthetic point clouds, as the virtual sensor is modeled by real LiDAR specifications, but it could be improved initially in two different aspects:

- Adding inherent documented sensor measurements errors, like those shown in the Hokuyo LiDAR sensor, during the virtual sensor scanning procedures [4,5,7,10]. An error distribution model is already being developed for future uses of this simulator;
- Adding environmental noises, like scattering particles, and target media or object material characteristics, like refractive and reflexive index [4,5,7-9]. This would allow for creating more accurate hazardous environment scenarios for the simulator. A Monte Carlo Mie Scattering approach is being researched to develop filters that disturbs the rays travelling path like in the mentioned issues.

#### Acknowledgements

The authors thank for the financial support provided by the Brazilian funding agencies CNPq, CAPES, FINEP and FAPERJ.

#### References

- [1] Blais F 2004 "Review of 20 years of range sensor development" *Journal of Electronic Imaging*, v.13, n.1, p.231-240.
- [2] Chow J C K 2014 "Multi-Sensor Integration for Indoor 3D Reconstruction". Alberta, 239 p. PhD Dissertation - Department of Geomatics Engineering. University of Calgary.
- [3] Oliver-Codina G, Massot-Campos M 2015 "Optical sensors and methods for underwater 3D reconstruction. Sensors" v.15, n.12, p. 31525-31557.
- [4] Tallavajhula A 2018 "Lidar simulation for robotic application development: modeling and evaluation" Pittsburgh, 107 p. PhD Dissertation - The Robotics Institute. Carnegie Mellon University.
- [5] Öhman N 2018 "Simulation of LiDAR data for forestry applications. Umeå, 29 p. MSc Thesis - Department of Physics. Umeå University.
- [6] Majek K, Bedkowski J "Range Sensors Simulation Using GPU Ray Tracing", *Proc. the 9<sup>th</sup> International Conference on Computer Recognition Systems CORES 2015*, Cham.
- [7] Hanke T; Schaermann A, Geiger M, Weiler K, Hirsenkorn N, Rauch A, Schneider S-A, Biebl E, "Generation and validation of virtual point cloud data for automated driving systems", *Proc. IEEE 20th International Conference on Intelligent Transportation Systems 2017*.
- [8] Hanke T, Hirsenkorn N, Rauch A, Dehlink B, Rasshofer R, Biebl E, "Generic Architecture for Simulation of ADAS Sensors", *16th International Radar Symposium (IRS) p. 125-130 2015*, Dresden.
- [9] Hirsenkorn N, Kolsi H, Selmi M, Schaermann A, Hanke T, Rauch A, Rasshofer R, Biebl E, "Learning Sensor Models for Virtual Test and Development" *In Workshop Fahrerassistenz und automatisiertes Fahren 2017*.
- [10] Wang F, Zhuang Y, Gu H and Hu H 2019, "Automatic Generation of Synthetic LiDAR Point Clouds for 3-D Data Analysis" *IEEE Transactions on Instrumentation and Measurement*. v. 68, n. 7, pp. 2671-2673
- [11] Parker S G, Bigler J, Dietrich A, Friedrich H, Hoberock J, Luebke D, ... and Stich, M. 2010, "OptiX: a general purpose ray tracing engine. *In Acm transactions on graphics* v. 29, n. 4, p. 66.

- [12] Constantini R, Sussstrunk S, “Virtual sensor design” *Proc. Electronic Imaging 2004*, San Jose, California, United States.
- [13] Gokturk S B, Yalcin H, Bamli C, “A time-of-flight depth sensor – system description, issues and solutions”, *Proc. Conference on Computer Vision and Pattern Recognition Workshop 2004*, Wanshigton.
- [14] ALICE project-team, [Online]. Available: <http://alice.loria.fr/index.php/home.html> [Accessed 23 06 2019]
- [15] Qi C R, Su H, Mo K and Guibas L J, “Pointnet: Deep learning on point sets for 3d classification and segmentation”. *Proc. IEEE Conference on Computer Vision and Pattern Recognition 2017* p. 652-660.
- [16] Salmon, J K, Moraes M A, Dror R O and Shaw D E, “Parallel random numbers: as easy as 1, 2, 3”, *Proc. International Conference for High Performance Computing, Networking, Storage and Analysis p.16 2011*, ACM.
- [17] The Graphics File Formats Page, [Online]. Available: <http://www.martinreddy.net/gfx/http://alice.loria.fr/index.php/home.html> [Accessed 23 06 2019].