

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Sistema de Triangulação Laser para Medição de Estruturas Submersas com Alta Definição

João Miguel Pesqueira Gaspar Pombo

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Professor Doutor Nuno Alexandre Cruz

Co-orientador: Doutor Andry Maykol Pinto

11 de Julho de 2016

Resumo

A investigação do meio aquático sempre foi uma área de grande interesse, tanto para instituições militares, como para instituições científicas. Atualmente, a utilização da tecnologia baseada em som já está bem estudada e os seus limites são conhecidos. Como tal, é necessário explorar novos métodos que permitam a extração de informação deste meio.

Posto isto, esta dissertação incide na análise, conceção e implementação de um sistema de aquisição 3D com base na projeção de linhas laser que, em conjunto com odometria, proporciona a criação de mapas de profundidade.

De modo a serem adquiridos dados com qualidade é essencial uma boa análise da arquitetura do sistema. Assim sendo, a primeira parte desta tese passa pelo estudo e validação conceptual do sistema de triangulação laser.

A segunda parte deste trabalho foca-se no uso de bibliotecas *Robot Operating System* (ROS) e *Point Cloud Library* (PCL) para desenvolver um algoritmo capaz de realizar a reconstrução de superfícies de estruturas submersas com alta definição. A implementação do algoritmo tem por base o uso de *octrees*, o que torna o sistema extremamente eficiente.

A viabilidade da implementação deste sistema é confirmada pelos mapas obtidos durante a fase de testes, realizados com recurso a dados simulados e dados experimentais. A partir da sua observação é possível perceber que o trabalho desenvolvido revela ter aplicabilidade em contexto real.

Abstract

The investigation of the aquatic environment has always been an area of great interest, both for military and scientific institutions. Currently, the use of sound-based technology is already well documented and their limits are known. Consequently, it is necessary to explore new methods that allow the extraction of information from this environment.

Having said that, this dissertation focuses on the analysis, design and implementation of a 3D acquisition system based on laser line projection, that in conjunction with odometry, provides the ability of creating depth maps.

In order to assure the quality of the acquired data, a good system architecture analysis is essential. Therefore, the first part of this thesis is focused on the study and conceptual validation of the laser triangulation system.

The second part of this thesis focuses on the use of Robot Operating System (ROS) and Point Cloud Library (PCL) libraries to develop an algorithm that is capable of performing the surface reconstruction of submerged structures with high definition. The implementation of the algorithm is based on the use of octrees, which makes the system extremely efficient.

The feasibility of implementing this system is confirmed by the maps obtained during the testing phase, carried out using simulated and experimental data. From their observation is possible to realize that this work reveals applicability in real context.

Agradecimentos

Em primeiro lugar, desejo agradecer ao Professor Doutor Nuno Alexandre Cruz e ao Doutor Andry Pinto pela disponibilidade, paciência e dedicação demonstrada desde que iniciei esta dissertação.

Aos meus pais, Jaime Pombo e Conceição Pombo, pelo apoio incondicional, por todos os sacrifícios que fizeram e por tudo o que me ensinaram.

Ao meu irmão, Pedro Pombo, por sempre me apoiarem nas minhas decisões e por acreditarem mais em mim do que eu.

Aos meus companheiros de curso, Daniel Barbosa, João Ventura e Manuel Silva, por toda a ajuda que me deram ao longo destes anos. Por todas as conversas e provérbios sem nexos, e por todos os momentos de insanidade mental, pânico e euforia. Vocês, sem dúvida, tornaram tudo isto muito mais fácil.

Ao Nuno Morais e à Catarina Mavilde, por me continuarem a levar para os "bons maus caminhos" e por terem estado presentes quando mais precisei. Obrigado por estarem sempre prontos para me desencaminhar.

Por fim, aos meus amigos de longa data, Andreia Silva, António Barbosa, Luís Sousa e André Rodrigues, por todos os momentos que partilhamos enquanto crescemos, e pela amizade e companheirismo que sempre esteve presente.

A todos eles, e tantas outras pessoas que de alguma forma me marcaram neste percurso,

Muito Obrigado.

João Pombo

*“In any moment of decision,
the best thing you can do is the right thing,
the next best thing is the wrong thing,
and the worst thing you can do is nothing.”*

Theodore Roosevelt

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Motivação	1
1.3	Objetivos	2
2	Revisão Bibliográfica	5
2.1	Introdução	5
2.1.1	Extração e processamento da informação do meio	6
2.2	Medição 3D em ambientes subaquáticos	7
2.2.1	Baseado em som	7
2.2.2	Baseado em imagem	10
3	Análise da arquitetura do sistema ótico de aquisição 3D	15
3.1	Introdução	15
3.2	Modelação do sistema de triangulação câmara-laser	16
3.2.1	Análise dos parâmetros α , Δx e z_{max} do sistema ótico	18
3.2.2	Análise da amplitude do alcance de deteção do sistema (Δz) em função de α e Δx	19
3.2.3	Análise de sensibilidade do sistema ao relevo	22
3.3	Impacto das limitações físicas espaciais no sistema	24
3.3.1	Influência de α e Δx em z_{max}	25
3.3.2	Impacto de α e Δx em Δz	28
3.3.3	Influência de α em $offset_x$	33
3.3.4	Conclusão	33
3.4	Impacto da quantização da câmara	34
3.4.1	Caso de estudo: influência da distância em z e do α no plano de imagem	35
3.5	Conclusão	37
4	Mapeamento	39
4.1	Introdução	39
4.2	Enquadramento do sistema de mapeamento desenvolvido	40
4.3	Algoritmo	43
4.3.1	Inicialização	43
4.3.2	ROS <i>Callbacks</i>	44
4.3.3	Fusão de Dados	45

5	Resultados Obtidos	51
5.1	Testes de validação com recurso a valores simulados	51
5.1.1	Simulação de rampas	52
5.1.2	Simulação de ondas	59
5.1.3	Simulação de um tubo	65
5.1.4	Eficiência dos métodos de concatenação	70
5.1.5	Conclusão	71
5.2	Testes de validação realizados em laboratório	72
5.2.1	Reconstrução Discretizada	73
5.2.2	Reconstrução em <i>free run</i>	81
5.3	Conclusão	89
6	Conclusões e Trabalho Futuro	91
6.1	Conclusão	91
6.2	Trabalho Futuro	92
	Referências	93

Lista de Figuras

2.1	Dados batimétricos do desfiladeiro Soquel	6
2.2	Veículo Aquático Autônomo equipado com SSS	9
2.3	Setup experimental de um sistema estereoscópico	11
2.4	C-Gauge	13
3.1	Plano do objeto	17
3.2	Vista superior do sistema de triangulação (referencial da câmara:O ; referencial do laser:L)	17
3.3	Análise da amplitude do alcance de detecção do sistema (Δz)	20
3.4	Vista superior da interceção do laser com dois planos	23
3.5	Vista frontal da interceção do laser com dois planos	23
3.6	Gráfico de α rad (radianos), Δx m (metros) e z_{max} m (metros)	25
3.7	Gráficos de α rad, Δx m e z_{max} m	25
3.8	Caso de estudo: impacto de α em z_{max}	26
3.9	Caso de estudo: impacto de Δx em z_{max}	27
3.10	Caso de estudo: impacto de α em z_{min}	29
3.11	Caso de estudo: impacto de Δx em z_{min}	30
3.12	Caso de estudo: impacto de α em Δz	31
3.13	Caso de estudo: impacto de Δx em Δz	32
3.14	Projeção de um ponto no plano de imagem	34
4.1	Interação entre nós de ROS	40
4.2	Diagrama de blocos do sistema	41
4.3	Extração da linha laser de uma <i>frame</i> - cores diferentes representam dimensões de altura diferentes	42
4.4	Algoritmo de mapeamento	43
4.5	Estrutura das mensagens ROS	44
4.6	Algoritmo das <i>callbacks</i>	45
4.7	Algoritmo da fusão de dados	46
4.8	Matriz da translação dos pontos	48
4.9	Matriz da rotação dos pontos em torno de x	48
4.10	Matriz da rotação dos pontos em torno de y	48
4.11	Matriz da rotação dos pontos em torno de z	48
4.12	Estrutura <i>Octree</i>	49
4.13	Subdivisão de <i>voxels</i> para acomodar pontos para mapear um coelho	50
5.1	Características do sistema usado nos testes	52
5.2	Simulação de uma rampa com 10^6 pontos	53
5.3	Simulação de uma rampa com 10^6 pontos (<i>top view</i>)	53

5.4	Simulação de uma rampa com 10^6 pontos (<i>side view</i>)	54
5.5	Gráfico da simulação da rampa com 10^6 pontos - tempo de ciclo vs número de pontos	54
5.6	Simulação de uma rampa com 5×10^6 pontos	56
5.7	Simulação de uma rampa com 5×10^6 pontos (<i>top view</i>)	56
5.8	Simulação de uma rampa com 5×10^6 pontos (<i>side view</i>)	57
5.9	Gráfico da simulação da rampa com 5×10^6 pontos - tempo de ciclo vs número de pontos	57
5.10	Gráfico da simulação da rampa com 5×10^6 pontos - tempo de ciclo vs número de pontos (<i>close view</i>)	58
5.11	Simulação de uma onda com 1.44×10^6 pontos	60
5.12	Simulação de uma onda com 1.44×10^6 pontos (<i>top view</i>)	60
5.13	Simulação de uma onda com 1.44×10^6 pontos (<i>side view</i>)	61
5.14	Gráfico da simulação da onda com 1.44×10^6 pontos - tempo de ciclo vs número de pontos	61
5.15	Simulação de uma onda com 2.88×10^6 pontos	62
5.16	Simulação de uma onda com 2.88×10^6 pontos (<i>top view</i>)	63
5.17	Simulação de uma onda com 2.88×10^6 pontos (<i>side view</i>)	63
5.18	Gráfico da simulação da onda com 2.88×10^6 pontos - tempo de ciclo vs número de pontos	64
5.19	Simulação de um tubo com 8×10^6 pontos (<i>top view</i>)	66
5.20	Simulação de um tubo com 8×10^6 pontos (<i>top view</i> com tubo destacado)	66
5.21	Simulação de um tubo com 8×10^6 pontos (<i>close up</i> com relevo destacado)	67
5.22	Simulação de um tubo com 8×10^6 pontos (<i>close up</i>)	67
5.23	Gráfico da simulação de um tubo com 8×10^6 pontos - tempo de ciclo vs número de pontos	68
5.24	Utilização do <i>CPU</i> e da memória <i>RAM</i> no início do mapeamento	69
5.25	Utilização do <i>CPU</i> e da memória <i>RAM</i> no fim do mapeamento	69
5.26	Mapa obtido através da concatenação	70
5.27	Mapa obtido através da octree	71
5.28	Plataforma usada para calibração do sistema câmara-laser	72
5.29	Fotografia do <i>setup</i> para mapeamento do tubo longo	74
5.30	<i>Top view</i> da reconstrução do tubo longo - com escala de comprimento em metros	75
5.31	<i>Top view</i> da reconstrução do tubo longo - com escala de largura em metros	76
5.32	Fotografia do <i>setup</i> para mapeamento do tubo curto	76
5.33	<i>Top view</i> da reconstrução do tubo curto - com escala de comprimento em metros	77
5.34	<i>Top view</i> da reconstrução do tubo curto - com escala de largura em metros	78
5.35	<i>Top view</i> da reconstrução do tubo curto - com escala de altura em metros	78
5.36	Fotografia do <i>setup</i> para mapeamento da caixa	79
5.37	<i>Top view</i> da reconstrução da caixa - com escala de comprimento em metros	80
5.38	<i>Top view</i> da reconstrução da caixa - com escala de largura em metros	80
5.39	<i>Front view</i> da reconstrução da caixa - com escala de altura em metros	81
5.40	Número de pontos mapeados ao longo do tempo - reconstrução do tubo longo	82
5.41	<i>Top view</i> da reconstrução em <i>free run</i> do tubo longo - com escala de comprimento	83
5.42	<i>Top view</i> da reconstrução em <i>free run</i> do tubo longo - com escala de largura em metros	83
5.43	<i>Top view</i> da reconstrução em <i>free run</i> do tubo longo - com escala de altura em metros	84

5.44	Gráfico do desempenho da reconstrução do tubo longo - tempo de ciclo vs número de pontos	84
5.45	Número de pontos mapeados ao longo do tempo - reconstrução da caixa	86
5.46	<i>Top view</i> da reconstrução em <i>free run</i> da caixa - com escala de comprimento em metros	86
5.47	<i>Top view</i> da reconstrução em <i>free run</i> da caixa - com escala de largura em metros	87
5.48	<i>Top view</i> da reconstrução em <i>free run</i> da caixa - com escala de altura em metros	87
5.49	<i>Side view</i> da reconstrução em <i>free run</i> da caixa - com escala de altura em metros	88
5.50	Gráfico do desempenho da reconstrução da caixa - tempo de ciclo vs número de pontos	88

Lista de Tabelas

3.1	Casos de estudo: impacto de α em z_{max}	27
3.2	Casos de estudo: impacto de Δx em z_{max}	28
3.3	Casos de estudo: impacto de α em Δz	31
3.4	Casos de estudo: impacto de Δx em Δz	32
3.5	Casos de estudo: impacto de α em $offset_x$	33
3.6	Casos de estudo A: impacto do ângulo $\alpha = 1.39rad$ e da distância em $z=0.5$ metros no plano de imagem	35
3.7	Casos de estudo B: impacto do ângulo $\alpha = 1.39rad$ e da distância em $z=3$ metros no plano de imagem	35
3.8	Casos de estudo A: resultados do impacto do ângulo $\alpha = 1.39rad$ e da distância em $z=0.5$ metros no plano de imagem	35
3.9	Casos de estudo B: resultados do impacto do ângulo $\alpha = 1.39rad$ e da distância em $z=3$ metros no plano de imagem	36
3.10	Casos de estudo C: impacto do ângulo $\alpha = 0.52rad$ e da distância em $z=0.5$ metros no plano de imagem	37
3.11	Casos de estudo C: resultados do impacto do ângulo $\alpha = 0.52rad$ e da distância em $z=0.5$ metros no plano de imagem	37
5.1	Pârametros de teste da rampa com 10^6 pontos	53
5.2	Resultados do teste da rampa com 10^6 pontos	55
5.3	Pârametros de teste da rampa com 5×10^6 pontos	55
5.4	Resultados do teste da rampa com 5×10^6 pontos	58
5.5	Pârametros de teste da onda com 1.44×10^6 pontos	59
5.6	Resultados do teste da onda com 1.44×10^6 pontos	62
5.7	Pârametros de teste da onda com 2.88×10^6 pontos	62
5.8	Resultados do teste da onda com 2.88×10^6 pontos	64
5.9	Pârametros de teste do tubo	65
5.10	Resultados do teste do tubo	68
5.11	Dimensões do tubo longo	75
5.12	Dimensões do tubo curto	77
5.13	Dimensões da caixa	79
5.14	Resultados do mapeamento do tubo longo	85
5.15	Resultados do mapeamento da caixa	89

Abreviaturas e Símbolos

AUV	<i>Autonomous Underwater Vehicle</i>
CCD	<i>Charge Coupled Device</i>
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
CPU	<i>Central processing unit</i>
CRAS	Centro de Robótica e Sistemas Autónomo
hz	<i>hertz</i>
ICCD	<i>Intensified Charge Coupled Device</i>
IMU	<i>Inertial measurement unit</i>
INESC TEC	Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência
LLS	<i>Laser Line Scan</i>
m	metro
MBES	<i>Multi Beam Echo Sounders</i>
mm	milímetro
pcd	<i>point cloud data</i>
PCL	<i>Point Cloud Library</i>
PLLS	<i>Pulsed Laser Line Scan</i>
px	<i>pixel</i>
rad	radiano
RAM	<i>Random Access Memory</i>
ROS	<i>Robot Operating System</i>
ROV	<i>Remotely Operated underwater Vehicle</i>
SBES	<i>Single Beam Echo Sounders</i>
SONAR	<i>Sound Navigation and Ranging</i>
SSS	<i>Side Scan Sonar</i>

Capítulo 1

Introdução

1.1 Enquadramento

Atualmente a investigação de métodos para aquisição de imagem tridimensional em ambiente aquático centra-se, em grande parte, no desenvolvimento e implementação de sistemas capazes de mapear estruturas a longo alcance (e.g. mapeamento do fundo do mar) [1].

Apesar disso, este tipo de soluções nem sempre satisfaz as necessidades no que toca ao mapeamento de estruturas submersas. Por conseguinte, é fundamental conceber tecnologia capaz de realizar aquisição de imagem a curta distância por forma a obter informação mais precisa (e.g. medições precisas de estruturas no fundo do mar).

Estes sistemas são normalmente instalados em veículos aquáticos autónomos e têm como finalidade realizar missões onde é essencial efetuar medições com elevada exatidão, que requerem, posteriormente, a reconstrução de imagem das estruturas submersas com alta definição.

Esta dissertação tem como foco principal a conceção e a demonstração de um protótipo de sistema de mapeamento de estruturas submersas, com base na projeção de linhas laser e de outra informação sensorial (e.g. posição), a fim de criar mapas de profundidade.

É importante notar que devido aos fenómenos de absorção e dispersão da luz no meio aquático o desempenho destes sistemas nem sempre é o mais eficiente [2], pelo que o objetivo do presente projeto não é criar um sistema que substitua métodos de aquisição a longa distância, que normalmente utilizam a propagação das ondas sonoras, mas sim desenvolver um que os complemente.

1.2 Motivação

O Centro de Robótica e Sistemas Autónomos (CRAS) do INESC TEC participa de forma ativa na investigação e desenvolvimento de soluções nos vários domínios da robótica, sendo que este projeto surge no âmbito da robótica aquática.

Em missões subaquáticas de inspeção, mapeamento ou monitorização, a distância e a informação relativa às estruturas a mapear (e.g. área, volume, forma) é, por norma, adquirida através de sonares devido à facilidade com que as ondas sonoras se propagam neste meio.

Porém, as medições apresentam níveis de incerteza que, por vezes, impossibilitam uma reconstrução precisa das estruturas submersas. Estes sistemas funcionam à base da emissão de ondas sonoras e receção dos ecos provenientes de estruturas presentes no meio. Como tal, estão limitados pela velocidade de propagação do som, que se revela demasiado baixa para aplicações que requerem uma elevada proximidade aos objetos alvo a mapear.

Assim sendo, é importante conceber e testar a viabilidade de uma solução robótica baseada em visão, que se destaca pelo uso da projeção de linhas laser em conjunto com uma câmara, em vez da utilização de um sistema baseado na propagação de ondas sonoras.

O sistema de aquisição de informação 3D deve ser capaz de realizar operações de mapeamento em ambiente aquático de uma forma eficaz e eficiente. Este, se especialmente dotado de fiabilidade ao funcionamento em ambientes subaquáticos e flexibilidade de atuação em diversas aplicações, representa um avanço tecnológico desta área de investigação.

Para além disso, a tecnologia hoje desenvolvida poderá ser aperfeiçoada para auxiliar a navegação de veículos aquáticos autónomos, por forma a não comprometer a sua integridade e a não perturbar ou danificar o ecossistema e estruturas nele presentes (e.g. redes de proteção de aquacultura *offshore*), quando realizada a curta distância.

1.3 Objetivos

O objetivo desta dissertação é desenvolver um sistema de mapeamento de estruturas submersas com base na projeção de linhas laser, que em conjunto com outra informação sensorial proporcione novas formas de criar mapas de profundidade.

Apesar deste ser o principal objetivo também se espera que o trabalho desenvolvido possibilite, com a otimização do sistema, a sua utilização noutras aplicações (e.g. auxílio na navegação de veículos aquáticos autónomos).

Posto isto, esta dissertação apresenta os seguintes objetivos:

- Estudo da disposição física dos lasers e câmaras, por forma a otimizar a qualidade de informação que é possível extrair, considerando para isso as limitações construtivas dos veículos. Parte deste projeto consiste na interação entre a projeção de linhas laser e a câmara que realiza a aquisição de imagem. Como tal, é preciso estudar a forma como estes elementos se relacionam de modo a encontrar uma zona de equilíbrio que não comprometa a qualidade da informação recebida.
- Desenvolver e implementar algoritmos para mapeamento tridimensional. Feita a extração de linhas laser, a partir das imagens adquiridas, é necessário criar os mapas de profundidade. Assim sendo, é preciso conceber algoritmos ou adaptar outros já existentes para mapear as estruturas.
- Implementar e testar o protótipo. O fim desta dissertação passa pela validação de todo o trabalho desenvolvido. Assim, a instalação e teste do sistema é essencial para que se possa analisar o seu comportamento.

Esta etapa permite não só inferir a viabilidade da integração deste tipo de tecnologia em veículos aquáticos autónomos, mas também, com base nos resultados obtidos, desenvolver novas aplicações para este sistema.

Capítulo 2

Revisão Bibliográfica

Atualmente, existem múltiplos métodos que permitem realizar o mapeamento de fundos subaquáticos ou estruturas submersas e inúmeras aplicações para estes sistemas, que vão desde a investigação científica à sua utilização em contexto militar [1, 3].

Apesar da importância que esta área de investigação tem na obtenção de conhecimento sobre o meio, as suas características adversas representam sérias dificuldades à obtenção de informação, quer através do uso de métodos de aquisição óticos, quer através de métodos de aquisição acústicos. Neste sentido é fundamental continuar a procurar novas soluções tecnológicas que permitam obter dados de boa qualidade, de forma eficiente e sem comprometer a integridade das pessoas e veículos que as utilizam.

Neste capítulo é feita uma revisão bibliográfica das tecnologias utilizadas no mapeamento de objetos submersos, leitos de rios ou fundos oceânicos. As secções que se seguem expõem algumas das técnicas mais utilizadas.

Nas secções 2.1 e 2.1.1 é feita uma introdução geral ao tema, expondo de forma sucinta quais os problemas associados a esta área e de que forma é realizada a exploração deste meio. Posteriormente, nas secções 2.2.1 e 2.2.2, são abordadas algumas das tecnologias existentes e usadas no mapeamento de estruturas submersas.

2.1 Introdução

As pequenas partículas presentes na água, que absorvem, refletem e alteram a direção quer da luz [4] quer do som no meio aquático, são responsáveis pela introdução de erros nos equipamentos de medição e, conseqüentemente, pela degradação da qualidade das imagens captadas [5]. Estes fatores, quando associados à falta de sensores de alta resolução com boa imunidade ao ruído, fazem com que o mapeamento a três dimensões no domínio aquático seja um grande desafio [6].

Além disso, métodos de medição como a batimetria (figura 2.1), que permitem o estudo da morfologia do meio aquático [7], são muitas vezes lentos no processo de criação de mapas com informação precisa e utilizam quantidades de informação elevadas para aplicações em que o estudo batimétrico não é o foco principal [8].

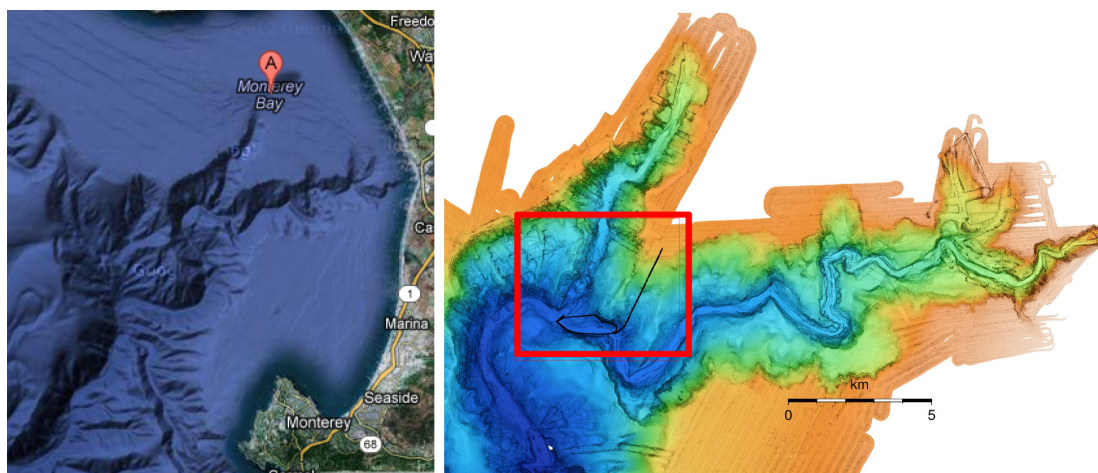


Figura 2.1: Dados batimétricos do desfiladeiro Soquel [9]

No entanto, o estudo deste ambiente é uma área que merece ser vastamente explorada, quer devido ao interesse que tem para as aplicações militares, que inicialmente contribuíram para o seu crescimento [3], quer para as áreas de investigação científica como por exemplo, o estudo da fauna [1] e biologia marinha [10] ou a descoberta, monitorização e preservação da herança cultural que se encontra submersa [11]. Consequentemente, é cada vez mais comum a automatização de processos de recolha de dados e a conceção de métodos e algoritmos adaptados às diversas aplicações existentes.

Por forma a explorar novas opções tecnológicas e dinamizar esta área de investigação, para que se possa mitigar ou até mesmo anular o impacto de alguns dos problemas inerentes ao meio, é imprescindível conhecer e analisar sistemas já existentes. Como tal, o presente capítulo reflete uma revisão bibliográfica essencial para o estudo, conceção e desenvolvimento do sistema de mapeamento de estruturas submersas proposto nesta dissertação.

2.1.1 Extração e processamento da informação do meio

A utilização de veículos aquáticos autónomos (AUV) ou de veículos operados remotamente (ROV) são ferramentas essenciais para a exploração do ambiente subaquático, pois reduzem a necessidade da intervenção humana (e.g. mergulhadores) [11] e permitem alcançar zonas onde as condições (e.g. pressão) são adversas ao corpo humano.

Atualmente, estes veículos são cada vez mais utilizados em ambientes complexos e desempenham diversas funções, entre as quais a inspeção em estruturas após desastres (e.g naufrágios), a monitorização regular de infraestruturas marinhas ou até mesmo missões de reconhecimento militar [6].

Segundo *Marcus Hammond e Stephen M Rock (2014)* [9], as missões de mapeamento realizadas por estes veículos apresentam várias etapas. Estas podem ser condensadas em duas fases, sendo que a primeira se foca na navegação e aquisição de informação do meio e a segunda na

análise e processamento dos dados adquiridos. Inicialmente, o veículo navega em volta da estrutura enquanto sistemas de aquisição, como o sonar ou sistemas baseados em visão artificial (e.g. câmaras), recolhem informação. Simultaneamente, os sistemas de navegação e outros sensores instalados guardam dados adicionais (e.g. odometria) que vão auxiliar a construção destes mapas. Posteriormente, toda a informação obtida é submetida a métodos que têm como objetivo extrair e fundir toda a informação sensorial para a produção dos mapas finais.

Os métodos associados à produção de mapas incluem técnicas ou etapas muitas vezes associadas ao processamento de imagem, tais como a filtragem, a segmentação, a reconstrução de superfícies, entre outras. Normalmente, os algoritmos onde se inserem estas técnicas são caracterizados pela sua complexidade computacional, resposta ao ruído, invariância perante diferentes condições (e.g. rotação, iluminação) [9], etc.

Como tal, a conceção, o desenvolvimento e a utilização de algoritmos robustos, capazes de realizar a extração das características essenciais e de lidar e corrigir erros, muitas vezes introduzidos pelo meio e por medições erradas por parte dos aparelhos de aquisição [3, 9], desempenha um papel fundamental na qualidade do produto final.

2.2 Medição 3D em ambientes subaquáticos

2.2.1 Baseado em som

Sound Navigation and Ranging, normalmente conhecido por sonar, é uma técnica profundamente estudada e utilizada em atividades no meio aquático. Atualmente existem duas formas de aplicar este conceito: o sonar ativo, que consiste no processo de emitir ondas acústicas e captar o eco refletido pelas estruturas atingidas, e o sonar passivo, que se baseia apenas na deteção de sons emitidos por objetos que se encontram no meio [12]. Em ambos os casos as imagens são posteriormente geradas sintetizando os sinais recebidos de cada onda acústica [13].

O conceito de sonar surgiu na segunda guerra mundial [12]. No que diz respeito à utilização militar o sonar ativo era usado para realizar comunicações, navegação e deteção de objetos enquanto que o sonar passivo era usado principalmente para vigilância [3]. Apesar disso, a sua aplicação transcende a utilização militar e hoje em dia é utilizado em diversas áreas tais como, a procura de destroços de navios ou aviões, o estudo da geologia marinha, o estudo da arqueologia marinha, o auxílio à pesca, entre outras [12].

A necessidade de desenvolver e melhorar sistemas capazes de aplicar esta técnica, motivada em parte pela procura das múltiplas indústrias onde o sonar é hoje em dia utilizado, fez com que ao longo dos anos surgissem múltiplas formas de aplicar este conceito.

É fundamental referir que as características do sonar (e.g. alcance, resolução) estão, em grande parte, dependentes da aplicação para o qual este se destina e, por conseguinte, os parâmetros de funcionamento (e.g. a frequência de ressonância) [12] são extremamente variáveis de sistema para sistema, pelo que existem diferentes classificações para diversas tecnologias que utilizam o sonar.

2.2.1.1 Single Beam Echo Sounders

O *Single Beam Echo Sounders* (SBES) está na gênese de muitos sistemas sonar [12]. Este sonar ativo baseia-se na transmissão de um impulso sinusoidal, sendo que a sua energia acústica é determinada tendo em consideração a frequência da onda portadora e a profundidade à qual o sistema vai ser usado [12]. A receção é normalmente limitada a um determinado período de tempo, por forma a evitar que o sinal enviado seja recebido diretamente do transmissor e para atenuar possíveis dispersões devido a partículas existentes no meio [12]. Apesar de se tratar de um conceito simples, e portanto fácil de aplicar, o SBES apresenta limitações relativamente à resolução e à área de terreno que pode mapear em cada ciclo, o que compromete a qualidade da informação adquirida e provoca um aumento significativo do tempo das missões [12].

2.2.1.2 Multi Beam Echo Sounders

O *Multi Beam Echo Sounders* (MBES) tem o mesmo princípio de funcionamento do SBES. Este sistema é capaz de adquirir medidas com boa resolução enquanto examina áreas com dimensões relativamente elevadas [14]. O que o distingue do SBES é o facto deste utilizar múltiplos feixes para realizar o mapeamento em vez de apenas um e por isso, ultrapassa algumas das limitações do SBES (e.g. área de cobertura reduzida). Atualmente o MBES pode estar equipado com tecnologia que permite o ajuste automático dos principais parâmetros do sonar (e.g. comprimento do impulso) ou filtros que otimizam a seleção da informação recebida [12]. Estas extensões tornam o sistema extremamente versátil pois é capaz de se ajustar às diferentes condições do meio e de filtrar os dados recolhidos sem ser necessária a intervenção humana [12].

Apesar disso, este sistema não é imune ao meio que o rodeia. O seu desempenho continua a estar dependente das condições do meio e das características da estrutura onde está instalado. Fatores como o barulho das hélices do veículo aquático ou o simples movimento do mesmo podem ser suficientes para comprometer a qualidade das imagens do sonar [12].

2.2.1.3 Side Scan Sonar

Os sistemas de varrimento lateral são constituídos por transdutores que emitem o sinal sonoro. Esta tecnologia é utilizada para obter informação sobre o solo oceânico ou objetos presentes em todo o meio aquático tendo em consideração o movimento do veículo [12].

No *Side Scan Sonar* (SSS) os transdutores, dispostos em linha e colocados nos lados das plataformas de mapeamento (figura 2.2), são capazes de enviar e receber sinais tanto no plano horizontal como no plano vertical [15], podendo existir múltiplas linhas do mesmo lado de modo a aumentar a qualidade da imagem [12]. Estes sistemas são usados para distinguir diferentes materiais no meio aquático e respetivas texturas, sendo empregues em diversas áreas que vão desde o estudo ambiental a aplicações militares (e.g. deteção de minas) [12]. Também podem ser utilizados para auxiliar a navegação de veículos autónomos aquáticos em ambientes não conhecidos [15].

Ainda que este sistema seja capaz de produzir imagens a duas dimensões com elevada resolução [12], a qualidade dos dados adquiridos está dependente do número de linhas de transdutores que o veículo pode transportar, pelo que o espaço disponível limita a qualidade das imagens do sonar.

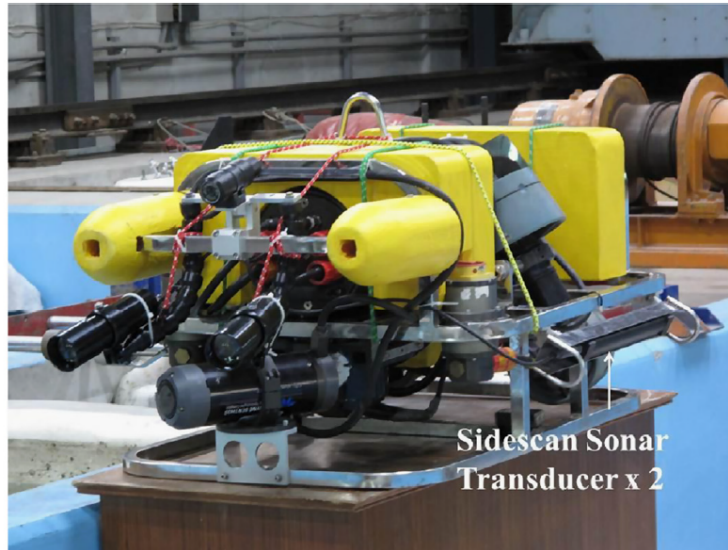


Figura 2.2: Veículo Aquático Autónomo equipado com SSS [15]

2.2.1.4 Conclusões

O sonar é uma técnica que tem sido estudada e constantemente melhorada ao longo de muito tempo. Este grande investimento na investigação da tecnologia acústica permitiu uma evolução de tal magnitude que, hoje em dia, os resultados obtidos com a utilização destes sistemas começam a ter uma qualidade muito próxima à dos sistemas óticos [14].

O facto do som se propagar mais facilmente do que a luz no meio aquático [16] é sem dúvida uma grande vantagem destes sistemas comparativamente a sistemas óticos (e.g. câmaras) [16, 17]. A possibilidade de combinar diferentes tecnologias que aplicam este conceito com o intuito de melhorar a capacidade de deteção de objetos no meio e a qualidade das imagens adquiridas [18] é também um ponto positivo que torna esta solução ainda mais atrativa.

No entanto, as imagens acústicas continuam a apresentar menos informação que as imagens óticas [13]. Outra grande desvantagem do sonar está relacionada com a velocidade de propagação do som. Esta característica por vezes limita severamente o desempenho destes sistemas onde a velocidade de deslocamento do veículo aquático pode ser o suficiente para perturbar a aquisição ao ponto da imagem ficar distorcida [13].

2.2.2 Baseado em imagem

2.2.2.1 Estereoscopia

No mundo que nos rodeia existem múltiplas pistas visuais que nos permitem criar a noção de profundidade [19]. De modo a que o cérebro humano perceção o ambiente que o envolve a três dimensões existem três princípios essenciais aos quais este recorre: conhecimento à priori do ambiente ou objeto, visão estereoscópica e paralaxe [19]. Na robótica, a estereoscopia é uma técnica onde as características presentes na visão humana são aplicadas à visão computacional por forma a criar a ilusão de profundidade. É fundamental perceber que cada pista visual fornecida pelo meio por si só não é suficiente para criar imagens detalhadas do mundo e por conseguinte é necessário utilizar vários indicadores para que a reconstrução do mapa ou modelo da estrutura seja o mais realista possível [19].

A estereoscopia é uma técnica estudada já há algum tempo e até aos dias de hoje o seu princípio base permanece presente em todos os sistemas que a aplicam. Fundamentalmente, esta técnica é caracterizada pela utilização de duas imagens do mesmo cenário obtidas de perspetivas diferentes, correspondentes à posição de duas câmaras, que são depois correlacionadas por forma a criar a ilusão de profundidade [19]. Atualmente esta correspondência entre imagens provenientes de diferentes ângulos é feita com recurso a ferramentas de *software* que também são responsáveis por executar outras funções tais como, a calibração dos equipamentos, a redução de ruído nas imagens e a correção de possíveis distorções provocadas pelo meio aquático [20].

Esta técnica tem diversas aplicações, sendo dada maior importância à sua utilização em missões onde a aquisição de imagem é feita a distâncias relativamente curtas devido às restrições que o meio aquático coloca à propagação da luz [21]. Dos múltiplos campos de interesse, onde a estereoscopia é utilizada, destacam-se a monitorização de locais de interesse arqueológico, o estudo geomorfológico da costa [20], a medição de objetos e a obtenção de distâncias [21].

As câmaras utilizadas são normalmente aparelhos de captura de vídeo (e.g. GoPro) [20] ou câmaras fotográficas digitais (e.g. Nikon D200s), com sensores *Charge Coupled Device* (CCD) ou *Complementary Metal Oxide Semiconductor* (CMOS), devidamente protegidas do meio aquático [1, 22], como se pode observar na figura 2.3. Devido à fraca luminosidade do ambiente subaquático estas câmaras são habitualmente emparelhadas com equipamentos de iluminação de modo a aumentar a visibilidade das zonas a mapear.

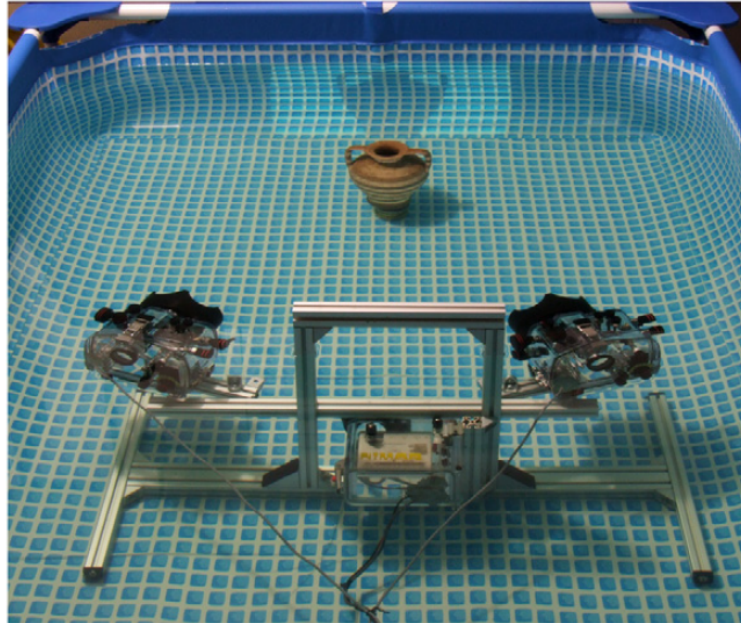


Figura 2.3: *Setup* experimental de um sistema estereoscópico [1]

Na estereoscopia, a utilização de duas câmaras nem sempre é suficiente para a reconstrução correta dos modelos a três dimensões. As ambiguidades provocadas por valores atípicos [23] e os artefatos presentes, numa ou em ambas as imagens, dificultam a obtenção de informação sobre a distância ou sobre as texturas dos objetos presentes no meio aquático. De modo a reduzir estas situações, pode ser benéfico utilizar mais do que duas câmaras para realizar a aquisição da imagem ou vídeo. Como tal, existem várias configurações possíveis para a utilização de múltiplas câmaras, sendo que estas se encontram, por norma, dispostas de modo a formarem estruturas triangulares ou estando paralelas entre si.

É ainda possível aumentar a complexidade destas arquiteturas agrupando as câmaras aos pares, criando múltiplos subsistemas estereoscópicos, ou utilizando as câmaras de forma independente, onde cada uma fornece informação essencial para a correlação correta de todos os dados adquiridos [23].

A adaptação de métodos estereoscópicos utilizados na robótica terrestre à robótica aquática podem, sem dúvida servir, para complementar sistemas que já realizam missões de reconhecimento neste meio.

A utilização de câmaras associadas a técnicas de iluminação não homogénea [24] ou de iluminação estruturada podem mitigar alguns dos problemas associados à propagação da luz no meio aquático, o que proporciona a aquisição de informação com melhor qualidade. Isto, aliado a métodos de processamento de imagem, faz com que esta tecnologia seja uma boa opção para a criação de modelos de objetos a três dimensões com uma boa resolução.

Porém, a estereoscopia, tal como todas as outras técnicas, tem as suas desvantagens das quais se destacam:

- O campo visual das câmaras presentes no sistema é um fator limitante [19], e portanto, objetos de dimensões superiores ao campo de visão tornam a tarefa de mapeamento mais complicada [22].
- A presença de erros de dimensão aparentemente pequena podem acumular-se ao longo de uma sequência de imagens sendo apenas detetados durante o processamento [11], o que pode comprometer a construção final dos modelos dos objetos.
- Os algoritmos de correlação entre imagens podem tornar-se computacionalmente pesados [20]. Para determinadas aplicações isto pode ser um entrave ao uso da estereoscopia, pois nem sempre é possível utilizar unidades de processamento capazes de suportar este tipo de carga computacional.
- O ajuste destes métodos a ambientes dinâmicos nem sempre é um processo simples [22]. Muitas das técnicas desenvolvidas para operar em ambientes estáticos raramente podem ser utilizadas em ambientes dinâmicos onde as condições do meio interferem na interação entre o equipamento de aquisição e o alvo a mapear [20].

2.2.2.2 Laser

A utilização de tecnologia laser no ambiente aquático pressupõe conhecimento básico do comportamento da luz neste meio e dos fenómenos que nele ocorrem.

Frequentemente, as imagens captadas debaixo de água têm má resolução e baixo contraste [25]. Isto acontece devido à turbulência do meio [26] e à atenuação da luz que é um processo caracterizado por fenómenos de absorção e de dispersão provocados não só pela água, mas também por partículas presentes nela [2].

Apesar disso, o uso do laser para mapeamento de estruturas submersas continua a ser uma solução interessante [25], pois permite examinar objetos num curto espaço de tempo e obter informação sobre eles (e.g. distância) com elevada precisão [27]. Por causa destas características esta tecnologia é usada em tarefas como a inspeção de tubos, o auxílio à navegação de veículos aquáticos autónomos [1], o estudo da biologia marinha, a deteção de objetos submersos [28], etc.

Estes sistemas são normalmente constituídos por uma fonte de luz (laser) e um sistema de aquisição de imagem (e.g. CCD, *Intensified Charge Coupled Device* (ICCD) ou CMOS) [26, 28]. Nas tecnologias que recorrem à projeção de linhas laser é comum haver alguma flexibilidade na definição dos parâmetros de *hardware* (e.g. potência do laser) [26], por isso é possível encontrar sistemas laser com princípios de funcionamento semelhantes mas com arquiteturas e outras características diferentes.

O *Laser Line Scan* (LLS) assemelha-se ao SBES, só que em vez de utilizar ondas acústicas este recorre à projeção de linhas laser. Tradicionalmente, esta técnica utiliza a transmissão contínua do feixe laser, com comprimentos de onda dentro da cor azul-verde, que é depois captada por sensores [29]. Esta estratégia é considerada como uma das melhores no que diz respeito ao alcance dentro das tecnologias de aquisição de imagens óticas em ambiente aquático.

Mesmo assim, o LLS pode ainda ser melhorado se a transmissão do feixe laser for feito de forma intermitente e se o tempo disponível para os sensores fazerem a aquisição for limitado [29].

O *Pulsed Laser Line Scan* (PLLS) é uma técnica ativa, usada na deteção e identificação de estruturas, capaz de produzir mapas de elevada qualidade. Esta mantém as vantagens do LLS só que a transmissão do laser é feita de forma intermitente [29]. Ao associar esta tecnologia a sistemas de aquisição com tempo de exposição limitado, a qualidade e o contraste das imagens melhoram [29], pois apenas as reflexões provenientes do alvo são captadas e a quantidade de ruído introduzido pelo meio é reduzido [27].

Por fim é apresentado um caso prático de um sistema baseado na projeção de linhas laser, já existente no mercado, desenvolvido pela OceanTools.

A OceanTools é uma empresa especializada no design e produção de soluções de engenharia no meio subaquático. Dos múltiplos projetos desenvolvidos existe um que se enquadra com o tema abordado nesta dissertação pela sua semelhança relativamente à tecnologia utilizada e às aplicações para o qual foi desenvolvido.

O *C-Gauge*, apresentado na figura 2.4, é um sistema de medição de elevada precisão, baseado na projeção de linhas laser, instalado num ROV de pequenas dimensões.

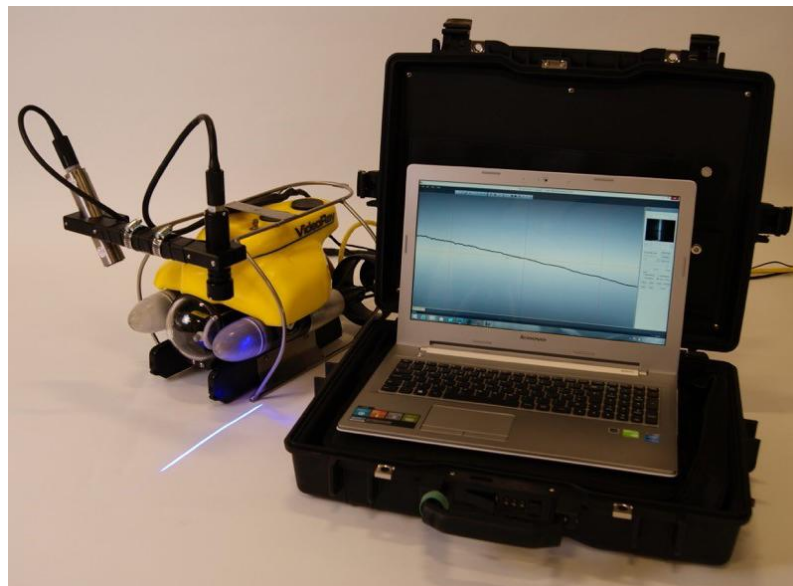


Figura 2.4: C-Gauge [30]

Este sistema permite realizar inspeções de tubos (e.g detecção de fendas) e de outras infraestruturas de forma rápida e precisa, fornecendo também informação sobre as suas dimensões e o seu estado de deterioração. Desta forma, as entidades responsáveis pela preservação destas estruturas podem planear e realizar operações de manutenção de forma informada com dados extremamente precisos, o que contribui para a preservação e extensão do tempo de vida destas estruturas ou componentes que as constituem.

2.2.2.3 Conclusões

A utilização de técnicas que recorrem à projeção de linhas laser permitem obter informação extremamente precisa, sendo possível construir modelos de estruturas submersas a três dimensões com elevada resolução independentemente da sua dimensão ou forma [20]. Devido ao facto de utilizarem a projeção de linhas laser, em vez de ondas acústicas, estes sistemas quando otimizados, permitem reduzir o tempo necessário à aquisição de imagem, assumindo que as condições do meio são favoráveis para a propagação da luz.

Não obstante, alguns sistemas laser ainda têm processos de instalação complexa [24], onde é difícil conciliar a disposição física dos equipamentos de aquisição com o espaço disponível, por forma a assegurar uma boa qualidade da informação obtida. Além disso, as adversidades do ambiente aquático impõem limites, por vezes severos, ao alcance deste tipo de sistemas tornando-os pouco versáteis, restringindo o seu uso para mapeamentos a curtas distâncias [29].

No entanto, casos como o *C-Gauge* demonstram que atualmente, os sistemas baseados na projeção de linhas laser, têm aplicação prática em operações no meio aquático e que o seu uso em determinadas situações pode ser vantajoso quando comparado com outras soluções mais antigas existentes. Este exemplo, é um bom incentivo à continuação da exploração destas tecnologias e demonstra a importância da realização de mais trabalhos de investigação e de implementação de projetos na área da robótica aquática.

Capítulo 3

Análise da arquitetura do sistema ótico de aquisição 3D

Este capítulo tem como finalidade apresentar os vários passos e os resultados obtidos através da análise da arquitetura do sistema ótico de aquisição 3D, constituído por uma câmara e um laser. O objetivo fundamental deste estudo é permitir identificar e obter a melhor configuração possível para o sistema ótico, considerando os limites físicos do veículo onde se pretende instalar o equipamento.

O capítulo 3 está organizado da seguinte forma. Na secção 3.1 é feita uma introdução, onde se abordam os conceitos de técnicas passivas e ativas. Na secção 3.2 é apresentado o esquema do sistema onde é possível observar a disposição dos vários elementos que o constituem e são também apresentados os cálculos efetuados. Nas secções 3.3 e 3.4 é apresentado o impacto das limitações físicas espaciais no sistema e o impacto da quantização da câmara, seguidos por uma breve conclusão.

3.1 Introdução

No capítulo 2 foram abordadas múltiplas técnicas para realizar o mapeamento de fundos aquáticos e estruturas submersas a três dimensões. Segundo *F. Bruno et al.* [1], estas podem ser separadas em dois grupos:

- Técnicas passivas;
- Técnicas ativas;

As técnicas passivas, por norma, utilizam a luz natural ou técnicas de iluminação artificial (i.e lâmpadas) de forma a iluminar o plano onde os objetos se encontram, sendo posteriormente recolhidas e sobrepostas múltiplas imagens obtidas a partir de diferentes ângulos e condições de iluminação [1]. Na secção 2.2.2.1 foi abordada uma técnica deste género.

Por sua vez as técnicas ativas utilizam laser onde são aplicados métodos baseados em princípios de triangulação. Resumidamente o feixe laser é projetado no objeto e o sensor de imagem,

colocado numa posição diferente, recebe a luz refletida. As coordenadas dos pontos são obtidas a partir da posição relativa dos dois dispositivos envolvidos no processo (i.e. laser e câmara) [1]. O trabalho realizado nesta dissertação enquadra-se neste grupo de técnicas.

O espaço disponível para a instalação destes sistemas, nos veículos aquáticos autónomos, por vezes limita o alcance e a sensibilidade. Face a este problema é essencial analisar a configuração do sistema para assegurar a qualidade da deteção 3D. Tendo estes fatores em consideração o sistema foi concebido de forma a cumprir os seguintes requisitos:

- Princípio de funcionamento robusto capaz de suportar as características do ambiente e adequado a operações de longa duração com reduzida intervenção humana.
- Eficiência energética como consequência da reduzida capacidade de processamento necessária ao seu funcionamento durante a recolha de dados.
- Capacidade de realizar operações de mapeamento 3D para tarefas de inspeção de objetos e fundos aquáticos a distâncias entre 0.5 e 2 metros - *close range navigation*.

3.2 Modelação do sistema de triangulação câmara-laser

Na sua forma mais simples o sistema é constituído por uma câmara e um laser e a sua análise tem como objetivos:

- Observar o impacto da variação do ângulo entre o referencial da câmara e o referencial do laser.
- Observar o impacto da variação da distância entre a câmara e o sistema de projeção da linha laser.

Na análise apresentada é considerado que o laser possui uma lente de linha, sendo que esta linha se encontra projetada no plano XZ . Como tal, o estudo que se segue incide sobre esse mesmo plano. Para além disso, por forma a simplificar a notação assume-se que o campo de visão é quadrado ($\Delta x = \Delta y$, figura 3.1) e que a câmara tem a mesma resolução horizontal e vertical .

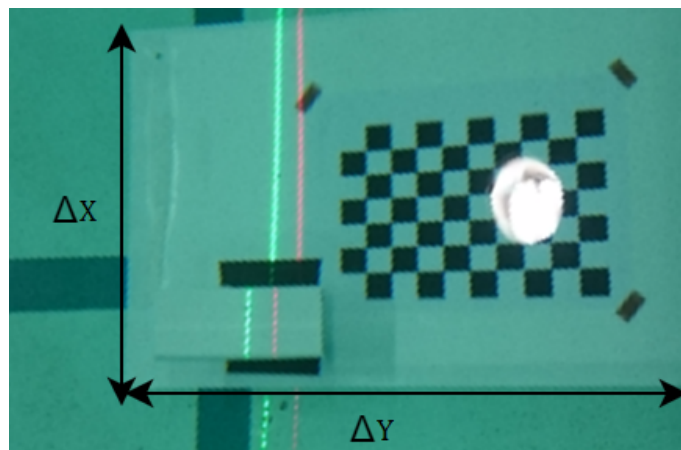


Figura 3.1: Plano do objeto

Na figura 3.2 é possível observar o esquema da arquitetura do sistema de triangulação.

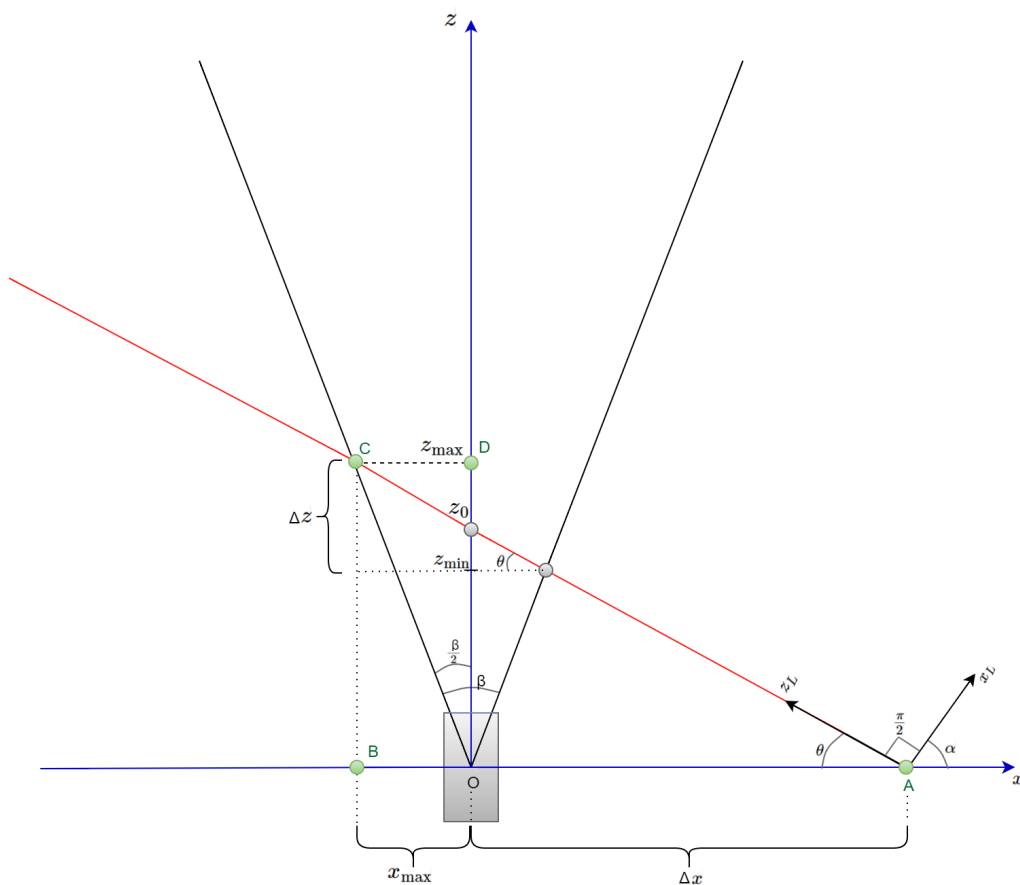


Figura 3.2: Vista superior do sistema de triangulação (referencial da câmara:O ; referencial do laser:L)

Na lista que se segue é apresentada uma descrição dos elementos fundamentais para a análise do sistema ótico de aquisição 3D que pode ser observado na figura 3.2:

z_{max} (metros) → alcance máximo do sistema. Normalmente caracterizado pela segunda interseção do feixe laser com o limite do campo de visão da câmara.

z_{min} (metros) → alcance mínimo do sistema. Normalmente caracterizado pela primeira interseção do feixe laser com o limite do campo de visão da câmara.

$(z_0, 0)$ → ponto onde o laser interceta o eixo ótico da câmara (z).

Δz (metros) → amplitude do alcance de deteção do sistema. Definido entre z_{min} e z_{max} .

β → ângulo de abertura da lente da câmara (em radianos). Neste caso foi considerado $\beta = 0.942rad$.

α → ângulo (em radianos) entre o referencial da câmara e o referencial do laser.

Δx (metros) → distância do sistema de projeção laser ao centro da câmara.

x_{max} (metros) → coordenada do eixo das abcissas correspondente a z_{max} .

3.2.1 Análise dos parâmetros α , Δx e z_{max} do sistema ótico

A parte inicial deste estudo centra-se na análise da interação entre os parâmetros α , Δx e z_{max} . O alcance máximo do sistema (z_{max}) é um parâmetro essencial para o sistema de aquisição, pois através deste é possível perceber a que distância a navegação do veículo, em torno das estruturas a mapear, deve ser feita.

No entanto, este parâmetro depende do ângulo α e do comprimento Δx e por isso, é necessário analisar qual o efeito que estes têm sobre z_{max} .

A equação que permite avaliar o comportamento do sistema nas condições mencionadas é construída a partir da análise do triângulo retângulo ABC.

A partir deste esquema (figura 3.2) é possível observar que:

- $\overline{AB} = x_{max} + \Delta x$;

- $\overline{BC} = z_{max}$;

Assim sendo a equação inicial é dada por:

$$\tan(\theta) = \frac{z_{max}}{x_{max} + \Delta x} \quad (3.1)$$

Para este caso de estudo o valor da inclinação do referencial do laser em relação ao referencial da câmara é dado por α e não por θ . Como tal, é necessário realizar o seguinte cálculo auxiliar:

$$\begin{aligned} \pi &= \theta + \alpha + \frac{\pi}{2} \Leftrightarrow \\ &\Leftrightarrow \theta = \frac{\pi}{2} - \alpha \end{aligned} \quad (3.2)$$

Da mesma forma x_{max} é obtido através do triângulo OCD onde

- $\overline{OD} = z_{max}$;
- $\overline{CD} = x_{max}$;

Posto isto, é possível inferir que:

$$\tan\left(\frac{\beta}{2}\right) = \frac{x_{max}}{z_{max}} \Leftrightarrow \quad (3.3)$$

$$\Leftrightarrow x_{max} = z_{max} \times \tan\left(\frac{\beta}{2}\right)$$

Substituindo θ e x_{max} na equação 3.1 pelos resultados obtidos em 3.2 e 3.3, respetivamente, obtém-se:

$$\tan\left(\frac{\pi}{2} - \alpha\right) = \frac{z_{max}}{z_{max} \times \tan\left(\frac{\beta}{2}\right) + \Delta x} \Leftrightarrow \quad (3.4)$$

$$\Leftrightarrow z_{max} = \frac{\Delta x}{\tan(\alpha) - \tan\left(\frac{\beta}{2}\right)}$$

A expressão 3.4 permite analisar a interação entre α , Δx e z_{max} . Assunto que será abordado posteriormente na secção 3.3.

3.2.2 Análise da amplitude do alcance de deteção do sistema (Δz) em função de α e Δx

A segunda parte desta análise passa por perceber a influência que a variação de α e Δx têm no valor da amplitude do alcance de deteção do sistema (Δz). Este pârmetro afeta a superfície de deteção, ou seja, a região onde é possível detetar a presença de objetos.

O modo como Δz varia em função de α e Δx é estudado através dos dados observados na figura 3.3.

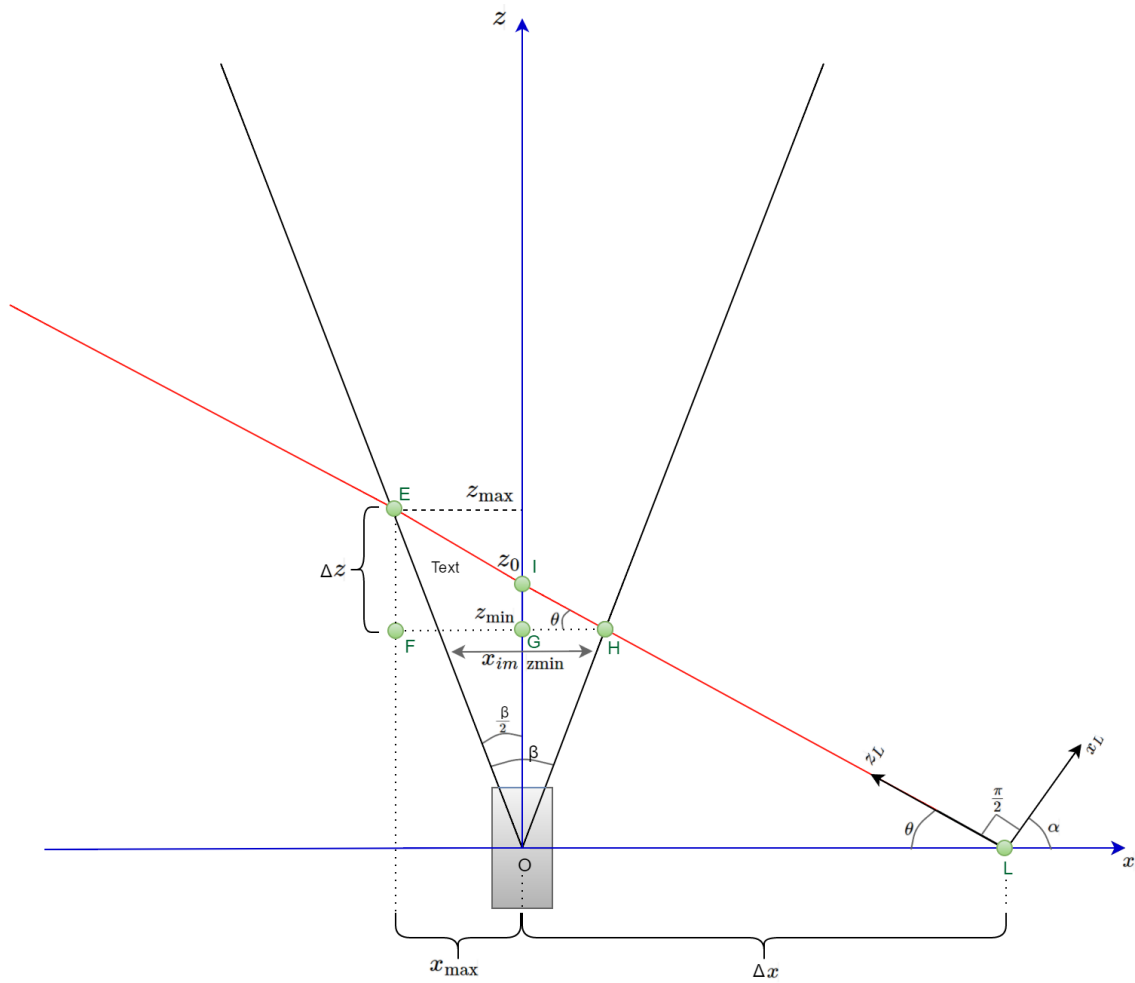


Figura 3.3: Análise da amplitude do alcance de detecção do sistema (Δz)

Inicialmente faz-se a análise da influência de α e Δx em Δz . Com o auxílio da figura 3.3 sabe-se que o triângulo HFE é constituído por:

- $\overline{EF} = \Delta z$;
- $\overline{HF} = x_{max} + \frac{x_{im;min}}{2}$;

Para simplificar a notação realiza-se a seguinte atribuição:

$$\frac{x_{im}z_{min}}{2} = x_{zmin} \quad (3.5)$$

Assim sendo, verifica-se que:

$$\tan\left(\frac{\pi}{2} - \alpha\right) = \frac{\Delta z}{x_{max} + x_{zmin}} \Leftrightarrow \quad (3.6)$$

$$\Leftrightarrow \Delta z = (x_{max} + x_{zmin}) \tan\left(\frac{\pi}{2} - \alpha\right)$$

Observando a equação 3.6, é possível perceber que Δz depende de x_{max} , que é obtido pela expressão 3.3, de α , que é uma variável do sistema, e de x_{zmin} , que pode ser calculado recorrendo à análise dos triângulos OGH e HGI, da figura 3.3, onde:

- $\overline{OG} = x_{zmin}$;
- $\overline{GH} = z_{min}$;
- $\overline{GI} = z_0 - z_{min}$;

Desta forma, pode-se deduzir que:

$$\tan\left(\frac{\beta}{2}\right) = \frac{x_{zmin}}{z_{min}} \Leftrightarrow \quad (3.7)$$

$$\Leftrightarrow z_{min} = \frac{x_{zmin}}{\tan\left(\frac{\beta}{2}\right)}$$

$$\tan\left(\frac{\pi}{2} - \alpha\right) = \frac{z_0 - z_{min}}{x_{zmin}} \Leftrightarrow \quad (3.8)$$

$$\Leftrightarrow x_{zmin} = \frac{z_0 - z_{min}}{\tan\left(\frac{\pi}{2} - \alpha\right)}$$

Com base na figura 3.3 é possível calcular, através da expressão 3.9, a distância z_0 à qual o feixe laser interceta o eixo ótico (z) da câmara. Isto deduz-se analisando o triângulo LOI:

- $\overline{LO} = \Delta x$;
- $\overline{OI} = z_0$;

$$\tan\left(\frac{\pi}{2} - \alpha\right) = \frac{z_0}{\Delta x} \Leftrightarrow \quad (3.9)$$

$$\Leftrightarrow z_0 = \Delta x \times \tan\left(\frac{\pi}{2} - \alpha\right)$$

Posto isto, pode-se continuar a resolução da expressão 3.8. Substituindo z_0 pelo resultado obtido na equação 3.9 e z_{min} pelo resultado da equação 3.7 obtém-se:

$$x_{zmin} = \frac{(\Delta x \times \tan(\frac{\pi}{2} - \alpha)) - (\frac{x_{zmin}}{\tan(\frac{\beta}{2})})}{\tan(\frac{\pi}{2} - \alpha)} \Leftrightarrow \quad (3.10)$$

$$\Leftrightarrow x_{zmin} = \frac{\Delta x \times \tan(\frac{\beta}{2})}{\tan(\alpha) \times \tan(\frac{\beta}{2})}$$

Finalmente, por forma a obter o valor Δz substituem-se os resultados obtidos a partir das expressões 3.3 e 3.10 em 3.6. Ou calcula-se a diferença $\Delta z = z_{max} - z_{min}$, onde z_{max} é dado por 3.4 e z_{min} por 3.7.

3.2.3 Análise de sensibilidade do sistema ao relevo

Por fim, falta entender como α afeta a sensibilidade do sistema ao relevo. Esta análise permite perceber de que forma o ângulo de incidência do laser, sobre um objeto, afeta a qualidade da informação obtida pelo sistema. Para isso assume-se o seguinte cenário:

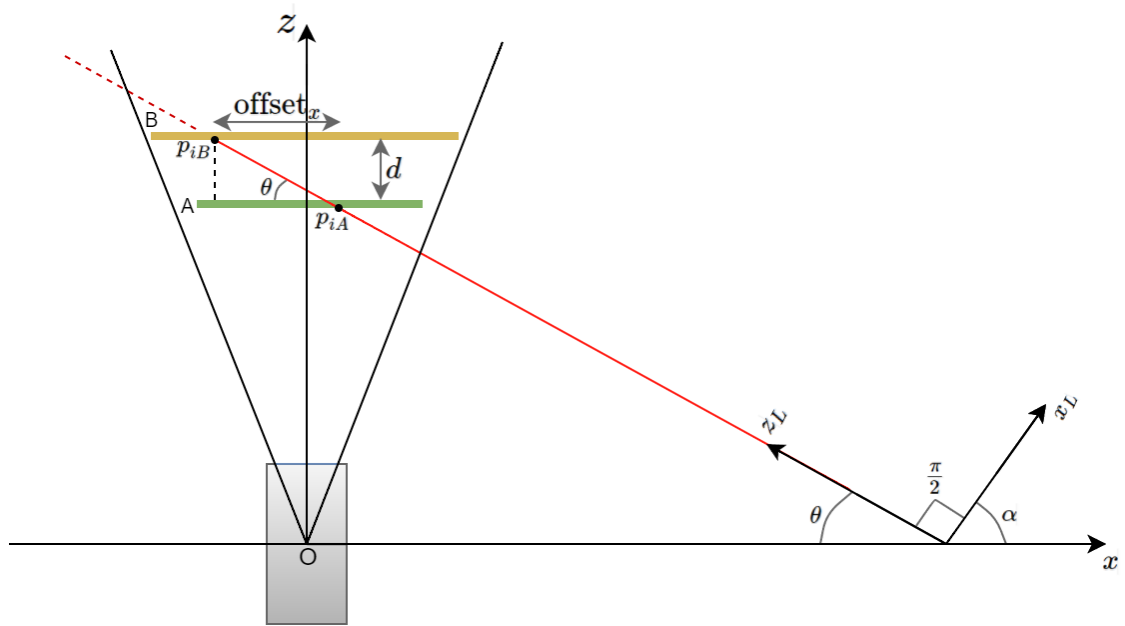


Figura 3.4: Vista superior da interceção do laser com dois planos

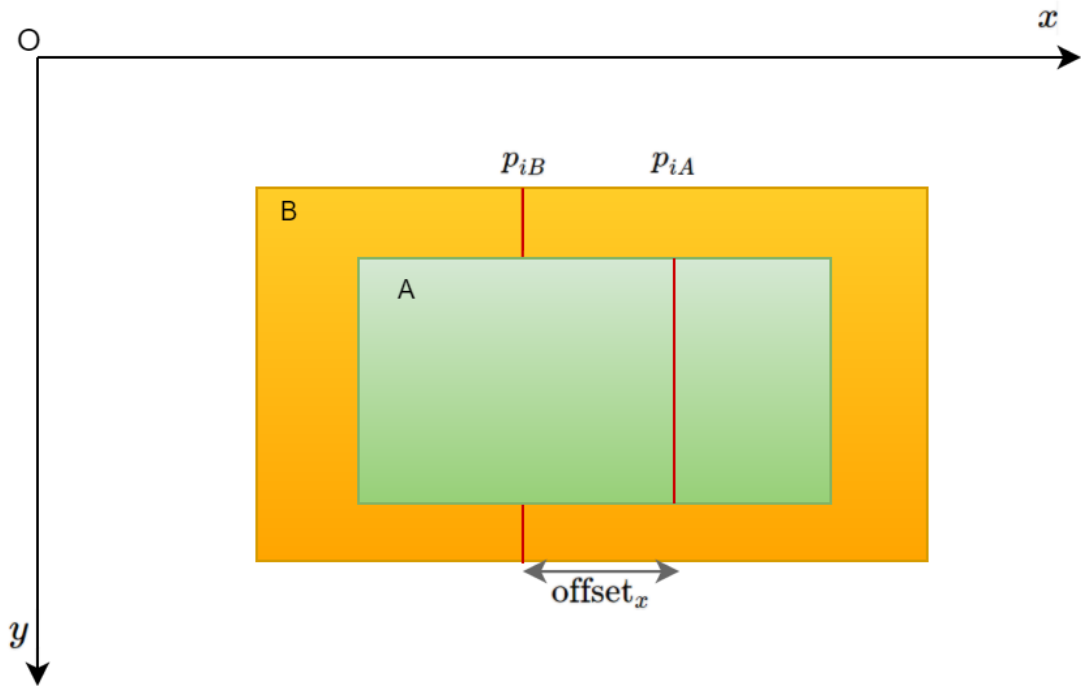


Figura 3.5: Vista frontal da interceção do laser com dois planos

Na figura 3.4 é possível observar a vista superior do plano A e do plano B que são interceptados pela linha laser com um ângulo de incidência $\theta (= \frac{\pi}{2} - \alpha)$ nos pontos P_{iA} e P_{iB} , respectivamente. Estes encontram-se separados por uma distância d (metros). Isto provoca um *offset* em x como se observa na figura 3.5.

É possível calcular este *offset* da seguinte forma:

$$\begin{aligned} \tan\left(\frac{\pi}{2} - \alpha\right) &= \frac{d}{offset_x} \Leftrightarrow \\ \Leftrightarrow offset_x &= \frac{d}{\tan\left(\frac{\pi}{2} - \alpha\right)} \end{aligned} \quad (3.11)$$

3.3 Impacto das limitações físicas espaciais no sistema

Após estarem deduzidas as expressões necessárias à análise do sistema, é agora possível obter informações sobre o seu comportamento. A partir da equação 3.4 obtém-se o gráfico representado nas figuras 3.6 e 3.7. Neste observa-se a relação entre α , Δx e z_{max} .

Uma análise rápida ao gráfico construído, permite observar que a variação do ângulo α aumenta ou diminui de forma exponencial a distância de alcance máxima z_{max} do sistema. Já a distância do sistema de projeção laser ao centro da câmara Δx demonstra influenciar z_{max} de forma linear. Também é possível observar que a partir de determinados valores, do ângulo α em combinação com Δx , deixa de haver grandes variações em termos de alcance do sistema. Assim sendo, no que diz respeito ao equilíbrio entre sensibilidade e alcance máximo, os parâmetros ótimos do sistema poderão estar situados na região central.

Estas relações serão estudadas em seguida com mais detalhe.

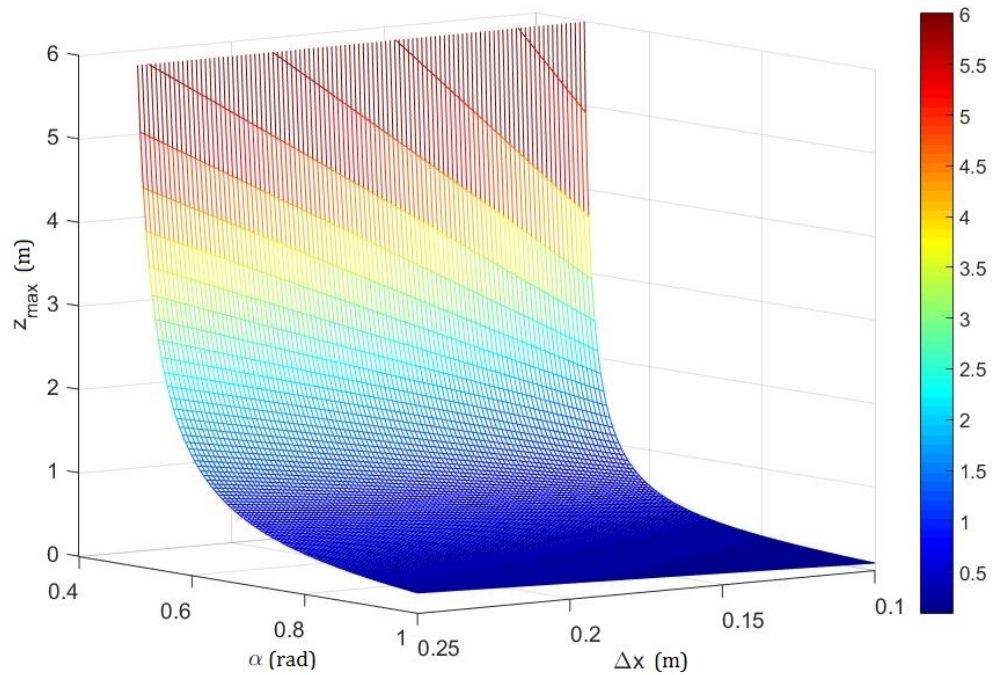


Figura 3.6: Gráfico de α rad (radianos), Δx m (metros) e z_{max} m (metros)

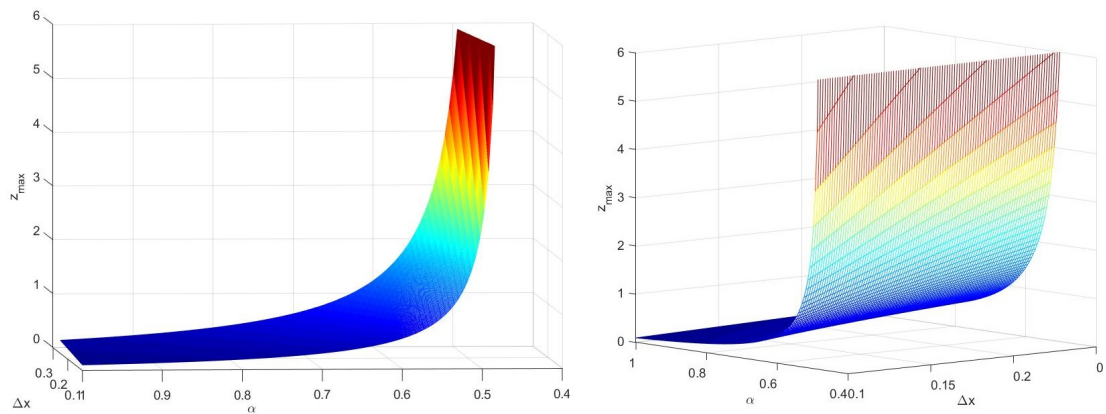


Figura 3.7: Gráficos de α rad, Δx m e z_{max} m

As subsecções que se seguem têm como objetivo avaliar o impacto que as diferentes variáveis estudadas na secção 3.2 têm no sistema. Para isso são analisadas algumas situações com valores arbitrários próximos dos limites máximo e mínimo para cada variável estudada nesta análise.

3.3.1 Influência de α e Δx em z_{max}

Esta subsecção tem como objetivo avaliar a influência de α e de Δx na amplitude do alcance de deteção do sistema. Esta análise é de extrema importância, pois permite extrair informação sobre

as relações entre α - z_{max} e Δx - z_{max} e, conseqüentemente auxiliar no dimensionamento do sistema.

No gráfico 3.8 são apresentadas 3 curvas, presentes no gráfico 3.6, que demonstram a relação entre α e z_{max} . Os seus valores são obtidos fixando a distância do laser ao centro da câmara, dada por Δx , na expressão 3.4, sendo que neste caso foram considerados:

- $\Delta x = 0.1$, que corresponde à distância mínima possível, perante os limites físicos espaciais do sistema.
- $\Delta x = 0.175$ metros, que corresponde a uma distância na região intermédia.
- $\Delta x = 0.25$ metros, que corresponde à distância máxima permitida para este caso.

A escolha destes pontos Δx permite isolar e analisar mais facilmente a influência de α em z_{max} nas zonas limite e intermédia.

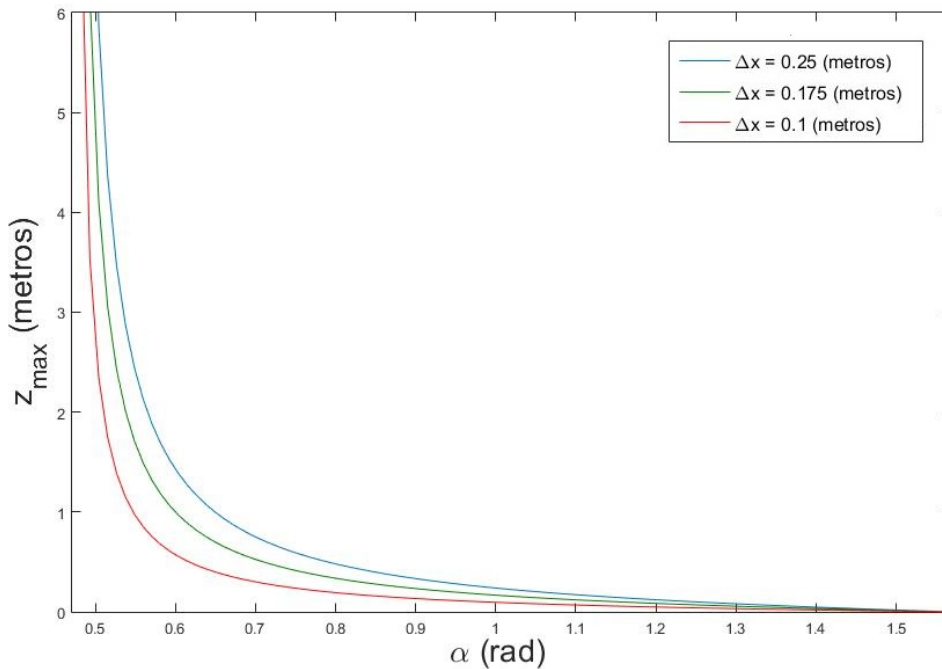


Figura 3.8: Caso de estudo: impacto de α em z_{max}

Uma primeira observação deste gráfico confirma a relação exponencial que existe entre α e z_{max} observada no gráfico 3.6. Para além disso, a partir de uma análise comparativa das curvas presentes no gráfico 3.8, é possível observar que à medida que o valor de Δx aumenta o comportamento exponencial da curva é atenuado, ou seja, variações em α não provocam alterações tão acentuadas em z_{max} .

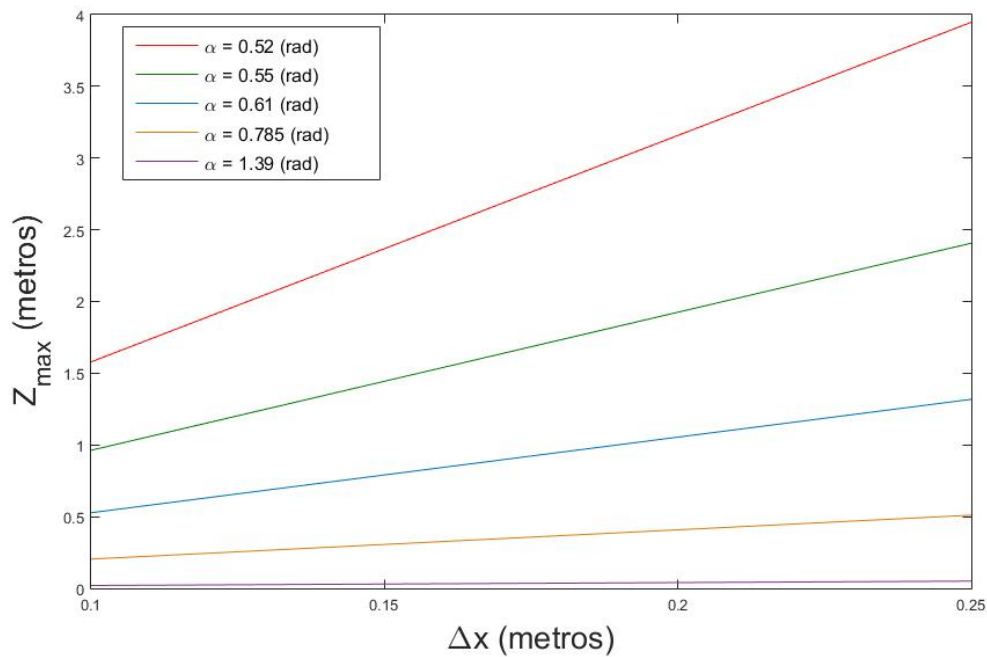
De modo a perceber, em termos quantitativos, a forma como a variação de α afeta o alcance z_{max} é apresentado na tabela 3.1 a análise de dois valores de α de uma das curvas. Neste caso a curva utilizada é a que corresponde a $\Delta x = 0.25$ metros.

Tabela 3.1: Casos de estudo: impacto de α em z_{max}

Caso	Variáveis		
	α (rad)	Δx (m)	z_{max} (m)
1	0.52	0.25	0.05
2	1.39	0.25	3.95

Considerando os valores α e Δx representados na tabela 3.1 e os valores de z_{max} obtidos através da equação 3.4, observa-se que a redução de α de $1.39rad$ para $0.52rad$ (caso 2 \rightarrow caso 1) provoca uma redução de 98.7% no alcance máximo do sistema.

A partir das retas presentes no gráfico da figura 3.9, extraídas da figura 3.4, observa-se que conforme a distância do laser ao centro da câmara aumenta, o alcance máximo do sistema aumenta. Além disso, verifica-se que o declive da reta é maior para valores de α mais pequenos. Como tal, o aumento em Δx apresenta um ganho superior, no alcance máximo, para esses ângulos comparativamente a valores de α maiores.

Figura 3.9: Caso de estudo: impacto de Δx em z_{max}

Tal como no caso analisado anteriormente na tabela 3.1, a análise que se segue foi feita em torno de uma das retas presentes no gráfico $\alpha = 0.785$ radianos. Esta análise quantitativa tem como objetivo perceber de que modo a alteração de Δx afeta o alcance z_{max} do sistema.

Tabela 3.2: Casos de estudo: impacto de Δx em z_{max}

Caso	Variáveis		
	α (rad)	Δx (m)	z_{max} (m)
1	0.785	0.1	0.20
2	0.785	0.25	0.51

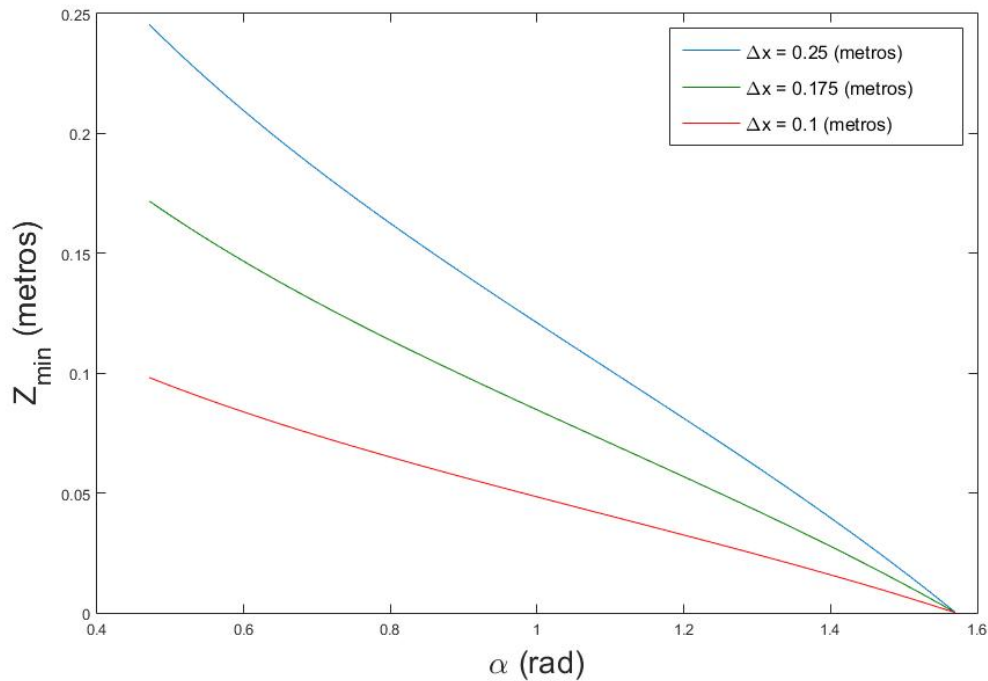
Com base na tabela 3.2 e nos resultados da equação 3.4, conclui-se que ao reduzir a distância Δx em $0.15m$ (caso 2 \rightarrow caso 1) o alcance máximo do sistema reduz em 60.8%.

3.3.2 Impacto de α e Δx em Δz

Nesta subsecção é analisada a influência de α e de Δx na amplitude do alcance de deteção do sistema. Este estudo é importante, na medida em que permite perceber de que forma Δz se comporta perante a alteração de α e Δx e consequentemente, avaliar quais os valores ideais para o sistema a implementar.

Inicialmente é feita uma breve análise, com base nos gráficos 3.10 e 3.11, sobre o modo como estes parâmetros afetam o valor do alcance mínimo do sistema (z_{min}). Apesar de não ser dada muita importância ao seu impacto no sistema, z_{min} é uma das variáveis que define o valor de Δz .

Do mesmo modo que se fez na secção 3.3.1, a partir das equações 3.7 e 3.10 foram isoladas 3 curvas, das várias existentes para diferentes combinações de α com Δx . Desta forma, a análise ao comportamento de z_{min} torna-se mais intuitiva. Optou-se novamente pelos valores de $\Delta x = [0.1$ metros, 0.175 metros, 0.25 metros].

Figura 3.10: Caso de estudo: impacto de α em z_{min}

Do gráfico 3.10 percebe-se que o valor do alcance mínimo diminui de forma quase linear à medida que o ângulo α aumenta. Para além disso, as curvas apresentam menor declive à medida que o laser se aproxima do centro da câmara. Pode-se então dizer que o sistema é menos sensível, relativamente a z_{min} , a variações de α para valores de Δx inferiores, ao contrário do que ocorre com z_{max} no gráfico 3.8.

Para além de ser afetado por α , o limite inferior da amplitude do alcance de deteção do sistema (z_{min}) também é influenciado por Δx . Como se pode observar pela figura 3.11.

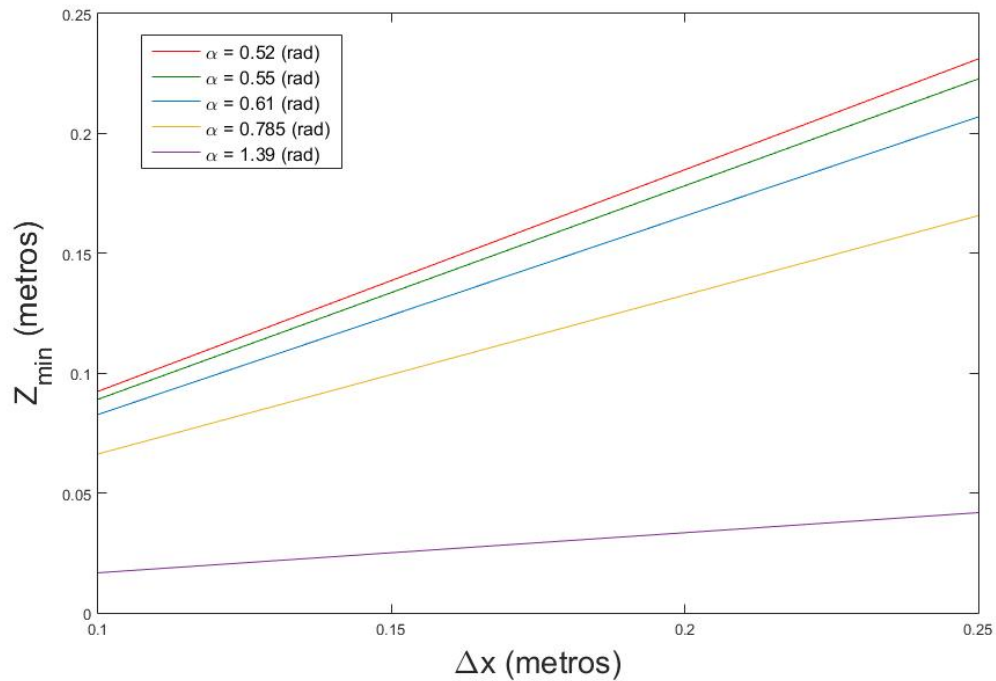


Figura 3.11: Caso de estudo: impacto de Δx em z_{min}

Através da figura (3.11) verifica-se que, tal como no caso do gráfico presente na figura 3.9, à medida que a distância do laser ao centro da câmara aumenta, o valor de z_{min} aumenta e que o declive da reta é maior para valores de α mais pequenos. Por conseguinte, existe maior ganho em z_{min} para valores de α mais pequenos conforme Δx aumenta.

Terminada a análise em z_{min} passa-se por fim à observação do modo como α e Δx afetam Δz . Sendo este dado pela diferença entre z_{max} e z_{min} .

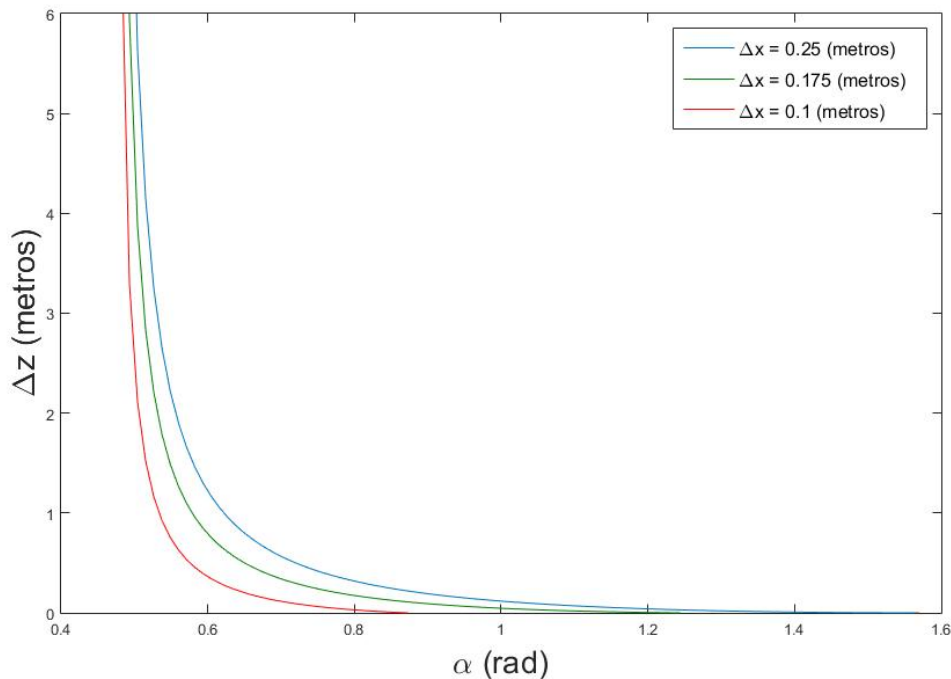


Figura 3.12: Caso de estudo: impacto de α em Δz

Tal como na situação analisada no gráfico 3.8, observa-se que a amplitude do alcance de deteção do sistema (Δz) é representado por curvas exponenciais. Novamente, os valores de Δx escolhidos são: $\Delta x = 0.1$ metros, $\Delta x = 0.175$ metros e $\Delta x = 0.25$ metros.

Através da análise destas funções conclui-se que à medida que Δx aumenta, o comportamento exponencial da curva é atenuado e, conseqüentemente, variações em α não provocam alterações tão acentuadas na amplitude Δz .

Na tabela 3.3 é apresentado um caso específico que permite realizar uma análise quantitativa da variação de Δz de acordo com a variação de α . Para isso escolheram-se 2 valores de α que fazem parte da curva (azul) onde Δx é igual a 0.25 metros.

Tabela 3.3: Casos de estudo: impacto de α em Δz

Caso	Variáveis					
	α (rad)	Δx (m)	z_0 (m)	z_{min} (m)	z_{max} (m)	Δz (m)
1	0.52	0.25	0.437	0.231	3.95	3.72
2	1.39	0.25	0.046	0.042	0.05	0.008

Como seria de esperar através da observação do gráfico 3.12, o aumento de α (caso 1 \rightarrow caso 2) resulta numa redução, neste caso de aproximadamente 99.8%, da amplitude Δz .

Por fim, é analisado o impacto de Δx em função de Δz . Na figura 3.13 são apresentadas 5 retas que representam a análise da amplitude do alcance de detecção para ângulos α fixos diferentes.

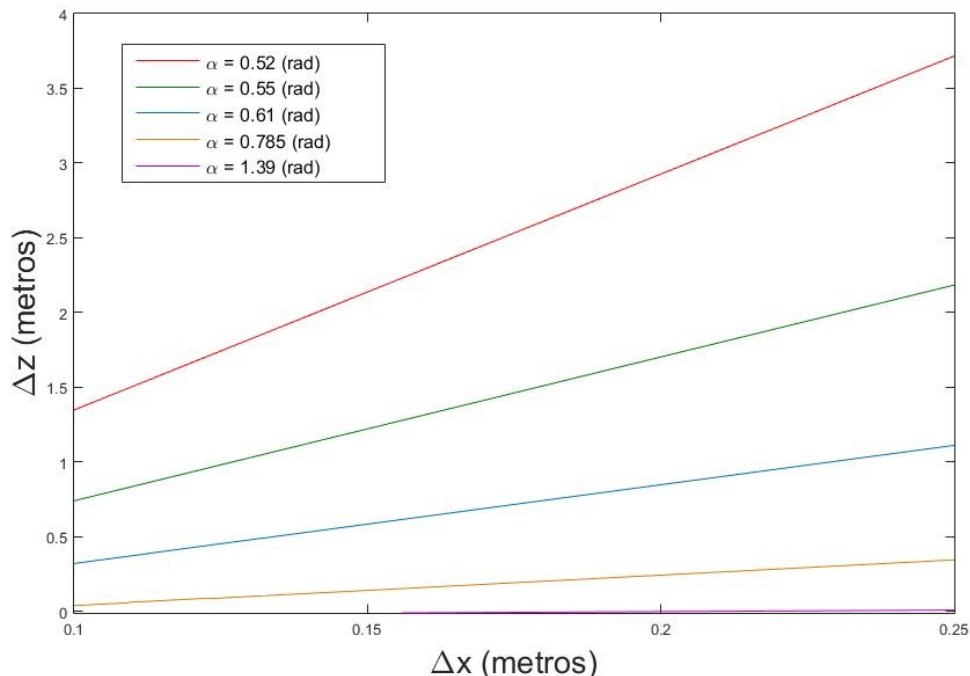


Figura 3.13: Caso de estudo: impacto de Δx em Δz

À semelhança de outras interpretações realizadas, para o mesmo valor de α , à medida que a distância do laser ao centro da câmara aumenta, o valor de Δz aumenta de forma linear. A magnitude dessa variação é dada pelo declive das retas observadas sendo que, para ângulos α maiores o ganho em termos de amplitude Δz revela ser mais pequeno.

O caso de estudo apresentado na tabela 3.4 tem como objetivo proporcionar uma análise quantitativa da relação entre a amplitude do alcance de detecção do sistema e a distância do laser ao centro da câmara. Para isso recorreu-se ao cálculo do valor de Δz , tendo como base a reta amarela da figura 3.13, onde α é igual a 0.785 radianos, e 2 pontos presentes nessa reta, onde Δx é igual a 0.1 metros e 0.25 metros.

Tabela 3.4: Casos de estudo: impacto de Δx em Δz

Caso	Variáveis					
	α (rad)	Δx (m)	z_0 (m)	z_{min} (m)	z_{max} (m)	Δz (m)
1	0.785	0.1	0.1	0.066	0.2	0.134
2	0.785	0.25	0.25	0.166	0.51	0.344

Baseado nos resultados Δz obtidos, é possível verificar que uma redução de 0.15 metros (caso

2 \rightarrow caso 1) causa uma perda de 61% na amplitude de detecção do sistema, assumindo que α é igual a 0.785 *rad*.

3.3.3 Influência de α em $offset_x$

A análise que se segue é meramente quantitativa e tem como finalidade observar como o ângulo α afeta a sensibilidade do sistema ao relevo e por conseguinte, a sua capacidade de extração de detalhe das estruturas submersas.

Consideram-se então 2 ângulos α diferentes, que se encontram na região limite dos parâmetros do sistema e que o desnível entre superfícies (d) é de 0.005 metros. O estudo deste caso é realizado com o auxílio das figuras 3.4 e 3.5, e dos resultados obtidos com a equação 3.11.

Tabela 3.5: Casos de estudo: impacto de α em $offset_x$

<i>Caso</i>	Variáveis		
	α (<i>rad</i>)	d (<i>m</i>)	$offset_x$ (<i>m</i>)
1	0.52	0.005	0.0029
2	1.39	0.005	0.0274

A partir dos valores da tabela 3.5 observa-se que o aumento do ângulo α leva a um aumento de $offset_x$ no plano do objeto em 0.0245 metros, ou seja, o *offset* observado é 9.4 vezes superior ao do caso 1. Como tal, quanto maior o ângulo α mais sensível se torna o sistema a variações de relevo.

3.3.4 Conclusão

A partir de todas as situações analisadas nas subsecções da secção 3.3 conclui-se que:

- Ao fixar o valor do ângulo α e variar Δx , os valores de z_{max} e Δz aumentam e diminuem de forma linear. Em termos genéricos, quanto mais afastado o laser estiver da câmara, maior será o alcance máximo e menor será o ângulo necessário para o obter. Como tal, a colocação do laser a 0.25 metros de distância minimiza o valor do ângulo α , maximizando a sensibilidade a alterações de relevo, quando comparado com a mesma instalação a distâncias menores (e.g $\Delta x = 0.1$ metros).
- Ao estabilizar a distância Δx e variar o ângulo α , os valores de z_{max} e Δz aumentam e diminuem de forma exponencial, sendo que a região ótima para este parâmetro, considerando missões de *close range navigation*, provavelmente se situa entre os 0.55 e os 0.9 radianos dependendo da distância do laser à origem da câmara. A partir destes valores de α o ganho em termos de alcance máximo não justifica as perdas de sensibilidade ao relevo e vice-versa.

3.4 Impacto da quantização da câmara

O modelo de uma câmara é uma função que mapeia o mundo tridimensional num plano bidimensional denominado plano de imagem. Dos vários modelos existentes, um dos fatores de diferenciação entre eles é a capacidade ou incapacidade destes capturarem o efeito de perspectiva, ou seja, se são capazes de diferenciar objetos próximos de objetos distantes através da representação do seu tamanho [31].

Um modelo simples capaz de capturar o efeito de perspectiva de uma imagem é o modelo *Pin-Hole* representado na figura 3.14. Este consiste apenas num ponto (centro da câmara) e num plano (plano de imagem). Para mapear um ponto do mundo no plano de imagem simplesmente move-se esse ponto em linha reta na direção do centro da câmara até que este intercepe o plano de imagem [31]. Tendo isto em consideração a posição do pixel é dada por:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

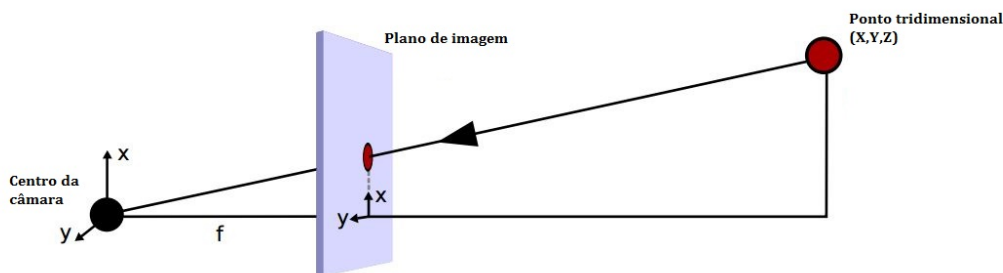


Figura 3.14: Projeção de um ponto no plano de imagem [31]

Onde f_x e f_y representam as distâncias focais e c_x e c_y as coordenadas do centro ótico em *pixels*. Assim sendo, quanto maior for o valor de z menor será a dimensão da projeção do objeto no plano da imagem e vice-versa.

3.4.1 Caso de estudo: influência da distância em z e do α no plano de imagem

O estudo realizado nesta secção tem como objetivo avaliar o impacto que a distância da câmara ao objeto e o ângulo α têm nas imagens obtidas e conseqüentemente, na qualidade da informação sobre relevo que pode ser extraída a partir da linha laser.

Para esta análise, a matriz dos parâmetros intrínsecos da câmara utilizada, *Mako G125C* com uma lente de 6mm, é dada por:

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.072e3 & 0 & 6.413e2 \\ 0 & 1.076e3 & 4.894e2 \\ 0 & 0 & 1 \end{bmatrix}$$

Inicialmente é realizada a análise do impacto que a distância do objeto em relação à câmara tem no plano da imagem. Para isso consideramos o $offset_x = 0.0274$ metros do caso 2 da tabela 3.5 abordado na secção 3.3.3.

Tabela 3.6: Casos de estudo A: impacto do ângulo $\alpha = 1.39rad$ e da distância em $z=0.5$ metros no plano de imagem

Variáveis			
Caso	$X(m)$	$Y(m)$	$Z(m)$
A_1	1.	1.	0.5
A_2	1.0274	1.	0.5

Tabela 3.7: Casos de estudo B: impacto do ângulo $\alpha = 1.39rad$ e da distância em $z=3$ metros no plano de imagem

Variáveis			
Caso	$X(m)$	$Y(m)$	$Z(m)$
B_1	1.	1.	3.
B_2	1.0274	1.	3.

Substituindo os valores X , Y e Z das tabelas 3.7 e 3.6 na matriz apresentada na secção 3.4 obtém-se os seguintes valores de x , y e w :

Tabela 3.8: Casos de estudo A: resultados do impacto do ângulo $\alpha = 1.39rad$ e da distância em $z=0.5$ metros no plano de imagem

Variáveis			
Caso	x	y	w
A_1	1.3927e3	1.3207e3	0.0005e3
A_2	1.4220e3	1.3207e3	0.0005e3

Tabela 3.9: Casos de estudo B: resultados do impacto do ângulo $\alpha = 1.39rad$ e da distância em $z=3$ metros no plano de imagem

Caso	Variáveis		
	x	y	w
B_1	2.9959e3	2.5442e3	0.0030e3
B_2	3.0253e3	2.5442e3	0.0030e3

O $offset_{x_A}$ e $offset_{x_B}$ calculado através de $x_{A_2} - x_{A_1}$ e $x_{B_2} - x_{B_1}$ respetivamente resulta em:

- $offset_{x_A} = 0.0293e3$
- $offset_{x_B} = 0.0294e3$

O que corresponde no plano de imagem a:

$$\frac{0.0293e3}{0.0005e3} = 58.6px \quad (3.12)$$

e

$$\frac{0.0294e3}{0.0030e3} = 9.4px \quad (3.13)$$

Através dos resultados de 3.12 e 3.13 conclui-se que quanto mais próximo o objeto se encontra da câmara, mais notório será o $offset_x$ no plano de imagem.

Relativamente à influência que o ângulo α tem no plano de imagem, considera-se uma nova situação C onde o $offset_x = 0.0029$ do caso 1 da tabela 3.5 abordado na secção 3.3.3.

Tabela 3.10: Casos de estudo C: impacto do ângulo $\alpha = 0.52rad$ e da distância em $z=0.5$ metros no plano de imagem

Variáveis			
Caso	$X(m)$	$Y(m)$	$Z(m)$
C_1	1.	1.	0.5
C_2	1.0029	1.	0.5

Tabela 3.11: Casos de estudo C: resultados do impacto do ângulo $\alpha = 0.52rad$ e da distância em $z=0.5$ metros no plano de imagem

Variáveis			
Caso	x	y	w
A_1	1.3927e3	1.3207e3	0.0005e3
A_2	1.3958e3	1.3207e3	0.0005e3

O $offset_{x_C}$ calculado através de $x_{C_2} - x_{C_1}$ resulta em:

- $offset_{x_C} = 0.00031e3$

O que corresponde no plano de imagem a:

$$\frac{0.00031e3}{0.0005e3} = 3.1px \quad (3.14)$$

Através dos resultados de 3.12 e 3.14 conclui-se que quanto maior o ângulo α , mais notório é o $offset_x$ no plano de imagem. Tal como ocorre no plano do objeto analisado na secção 3.2.

3.5 Conclusão

Apesar de existir uma região ótima para a escolha dos parâmetros deste sistema em operações de *close range navigation*, a sua seleção não deve ser permanente, a menos que fatores construtivos impeçam a sua alteração. Os objetivos da missão de mapeamento e as condições do meio onde esta decorre são alguns dos fatores necessários a ter em conta para que possa ser feita a escolha da melhor configuração.

As operações subaquáticas são feitas num meio extremamente dinâmico onde, por exemplo, a concentração de partículas pode limitar drasticamente o alcance máximo do sistema. Perante um cenário destes, escolher um Δx mais reduzido ou um α maior pode revelar-se benéfico. Da mesma forma, em operações onde a proximidade ao alvo é um fator impeditivo, escolher um Δx maior ou um α mais pequeno pode permitir a realização do mapeamento em condições de maior segurança.

Capítulo 4

Mapeamento

No final da secção 2.2.2.2, do capítulo 2, foi apresentado um sistema de medição baseado na projeção de linhas laser que apresenta bons resultados para aplicações associadas à inspeção subaquática, o *C-Gauge*.

No entanto, a exploração de novas técnicas e algoritmos, associados à necessidade de realizar operações de mapeamento para inspeção de estruturas submersas, promovem o crescimento da exploração subaquática a partir da utilização de sistemas robóticos, tornando-os cada vez mais robustos e versáteis.

Por esse motivo, neste capítulo são explicados os processos que compõem o algoritmo que, através da extração de linhas laser e em conjunto com odometria, possibilita a construção de mapas tridimensionais de estruturas submersas.

4.1 Introdução

O *software* apresentado tem como principal objetivo dotar o sistema da capacidade de criar superfícies de relevo a partir de informação (e.g *pointclouds*) enviada ao longo do tempo por outras entidades que o compõem. Assim sendo, este trabalho consiste na criação e integração de um módulo que implementa esta nova funcionalidade num sistema previamente concebido. De modo a que o seu desenvolvimento fosse possível, recorreu-se a algumas bibliotecas relacionadas com a área da robótica e do processamento de imagem.

Inicialmente, para que o processo de integração fosse o mais simples possível, recorreu-se a bibliotecas *Robot Operating System* (ROS). O principal motivo desta escolha prende-se com o facto de existirem processos de computação já desenvolvidos associados a este projeto, que partilham informação com o algoritmo desenvolvido nesta dissertação. Desta forma, o processo de comunicação entre os vários módulos que constituem o sistema é simplificado, na medida em que todos eles partilham a mesma *framework* [32]. Além disso, este conceito modular permite que no futuro sejam feitas melhorias e acréscimos ao trabalho hoje desenvolvido.

Mais especificamente, no que diz respeito à comunicação entre módulos, as bibliotecas ROS permitem criar nós que, sucintamente, são processos que com base num conjunto de dados de

entrada (*inputs*) executam operações de modo a produzir um ou múltiplos resultados (*outputs*), que podem depois ser difundidos recorrendo a tópicos [32]. Através da publicação e subscrição destes tópicos, como se observa na figura 4.1, os nós podem partilhar informação e utilizá-la caso esta seja útil para o desempenho das suas funções.

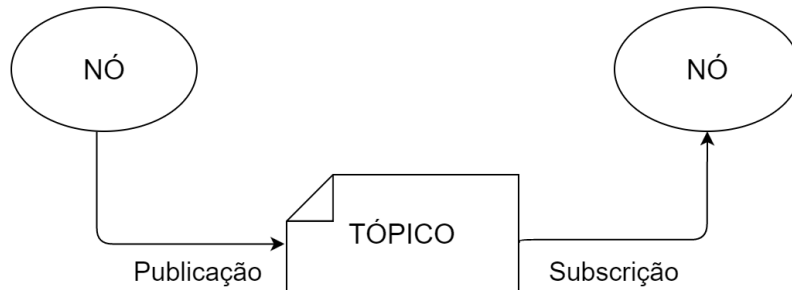


Figura 4.1: Interação entre nós de ROS

O exemplo ilustrado nesta figura (4.1) demonstra o caso mais simples onde a interação é de um para um, no entanto um nó pode publicar e subscrever a múltiplos tópicos.

A utilização do ROS foi complementada com o uso de uma biblioteca já bem integrada com esta *framework* [33], a *Point Cloud Library* (PCL). Esta permite a manipulação e processamento de conjuntos de pontos (*pointclouds*), representados num sistema de coordenadas (e.g cartesiano), que têm como objetivo caracterizar as superfícies externas dos objetos presentes no mundo. Além disso, a PCL contém estruturas de armazenamento extremamente eficientes (e.g *octrees*) que permitem guardar rapidamente todos os pontos das várias *pointclouds* que são publicadas.

4.2 Enquadramento do sistema de mapeamento desenvolvido

Como foi referido na secção 4.1, esta tese tem como objetivo desenvolver e integrar um nó ROS responsável pela criação de superfícies de relevo, usando *pointclouds* recebidas temporamente, num sistema já desenvolvido. Por conseguinte, antes de realizar a análise do algoritmo implementado é preciso compreender qual a relação entre os diferentes blocos que compõem o sistema.

No diagrama da figura 4.2 é apresentado, de forma simples, os processos necessários ao funcionamento do nó concebido.

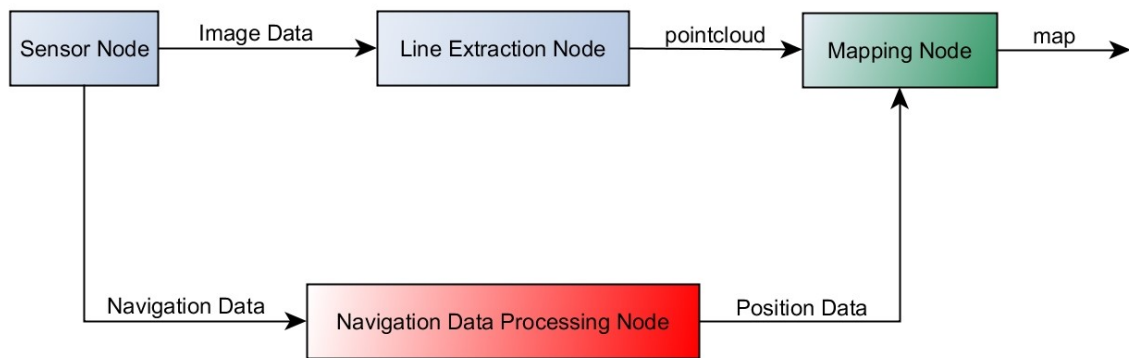


Figura 4.2: Diagrama de blocos do sistema

A partir da figura 4.2 observa-se que o sistema de mapeamento é composto por 4 blocos, sendo que os blocos a azul (*Sensor Node* e *Line Extraction Node*) correspondem a tecnologia que já se encontra implementada, o bloco a verde (*Mapping Node*) corresponde ao nó desenvolvido nesta tese e o bloco a vermelho (*Navigation Data Processing Node*) corresponde ao nó que, no final desta dissertação, fica a faltar conceber para que todo o sistema possa, no futuro, ser integrado num veículo aquático autónomo.

Mais detalhadamente, o *Sensor Node* representa os sistemas de aquisição de dados. Estes sistemas são responsáveis pelo envio de informação sobre a superfície das estruturas (*pointclouds*) e sobre os dados associados à navegação do veículo aquático (e.g velocidade). No âmbito desta tese foram considerados o sistema de ótico de aquisição 3D, estudado no capítulo 3, e um *Inertial measurement unit* (IMU). No entanto, no futuro poderão ser integrados mais sistemas de auxílio à navegação e de extração de informação sobre o meio, sem comprometer o funcionamento do sistema atualmente desenvolvido. A informação proveniente do *Sensor Node* é enviada para dois nós, o *Line Extraction Node* e o *Navigation Data Processing Node*, que realizam funções distintas.

O *Line Extraction Node* subscreve às mensagens que contêm as imagens captadas pelo sistema ótico e tem como principal função a extração dos dados sobre o relevo das estruturas a mapear. Sucintamente, através da análise da deformação da projeção do laser na imagem, este nó cria *pointclouds* que representam a porção da superfície medida. Na figura 4.3 é possível observar o resultado do processo de extração de uma linha laser.

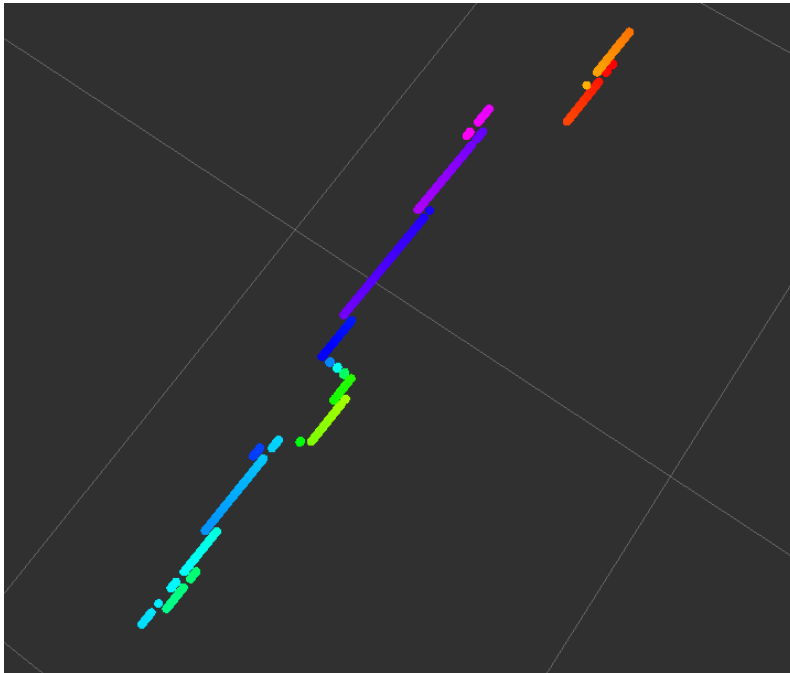


Figura 4.3: Extração da linha laser de uma *frame* - cores diferentes representam dimensões de altura diferentes

Por outro lado, o nó *Navigation Data Processing Node* subscreve à informação dos dispositivos de auxílio à navegação (e.g IMU), sendo que com base nesses dados estima a localização relativa (posição e orientação) do veículo durante o seu movimento em torno da estrutura a mapear. Desta forma, o nó responsável por criar mapas, tem acesso à informação necessária para realizar as transformações geométricas sobre a *pointcloud*, como será explicado mais à frente.

Por fim, o *Mapping Node* representa o nó responsável pela fusão da informação proveniente do *Line Extraction Node* (*pointclouds*) e do *Navigation Data Processing Node* (odometria). Com base nestes dados, o *Mapping Node* cria o modelo tridimensional da estrutura e disponibiliza-o para que possa ser, mais tarde, utilizado ou observado.

4.3 Algoritmo

De seguida é apresentado, de forma condensada, o fluxograma do algoritmo do *Mapping Node* destacado a verde na figura 4.2:

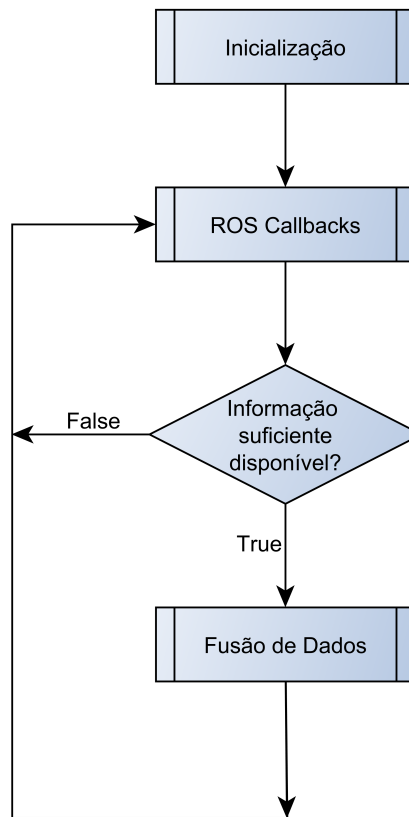


Figura 4.4: Algoritmo de mapeamento

Como se observa, na figura 4.4, o nó desenvolvido é composto por três etapas principais:

- Inicialização
- ROS *Callbacks*
- Fusão de Dados

4.3.1 Inicialização

A Inicialização é responsável por estabelecer as ligações, representadas pelas setas a preto na figura 4.2, que possibilitam a troca de informação entre os diferentes módulos do sistema. Quando a sua execução termina, o algoritmo passa a ser executado de forma cíclica. Nesta etapa, são também inicializadas todas as variáveis e estruturas utilizadas pelo algoritmo.

4.3.2 ROS Callbacks

As *callbacks* são funções definidas no código que são chamadas de forma automática, ou seja, o utilizador não tem controlo sobre o momento em que elas vão ocorrer. Estas são executadas, se subscritas, quando nova informação é disponibilizada pelo nó.

Na *framework* ROS, estas funções são normalmente controladas com recurso a *message handlers* que subscvem a tópicos de outros nós. Sempre que um tópico, ao qual o nó está subscrito, é publicado, o ROS invoca o *message handler* e passa a informação sobre essa mensagem para que a função de *callback* possa iniciar ao seu processamento [32].

As mensagens recebidas pelo nó desenvolvido apresentam estruturas semelhantes, demonstrada na figura 4.5, sendo que a principal diferença se encontra no tipo de informação que estas transportam.

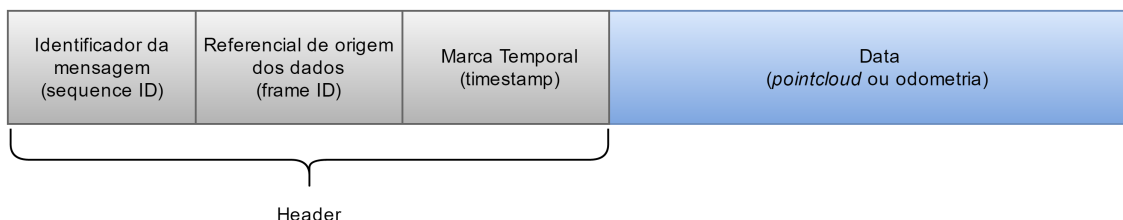


Figura 4.5: Estrutura das mensagens ROS

Como é possível observar na figura 4.5, as mensagens são compostas pelo *header* e pelo corpo da mensagem. O *header* é constituído pelo *sequence ID*, que é um identificador numérico, que é incrementado à medida que novas mensagens são criadas, pelo identificador do referencial onde foram obtidos os dados (*frame ID*) e pela marca temporal (*timestamp*), que corresponde à data (*Unix*) em que a mensagem foi gerada. Por fim, o corpo da mensagem contém a informação que vai ser manipulada pelo nó de mapeamento, neste caso, uma *pointcloud* ou valores de posição e orientação.

Estas mensagens são, muitas vezes, provenientes de nós diferentes e apresentam frequências de publicação distintas, e por isso, nem sempre é possível assegurar que a sua receção seja síncrona. Apesar de existirem filtros de sincronização nativos do ROS, no algoritmo desenvolvido foi implementada uma lógica de semáforos (*flags*) que permite coordenar a chamada do método responsável pela fusão de dados, assim que exista informação suficiente para a sua execução.

As *callbacks* implementadas, apesar de operarem de forma diferente sobre a informação que lhes é fornecida, são funções na sua génese semelhantes. Na figura 4.6 é apresentado, sucintamente, o algoritmo base das *callbacks* implementadas.

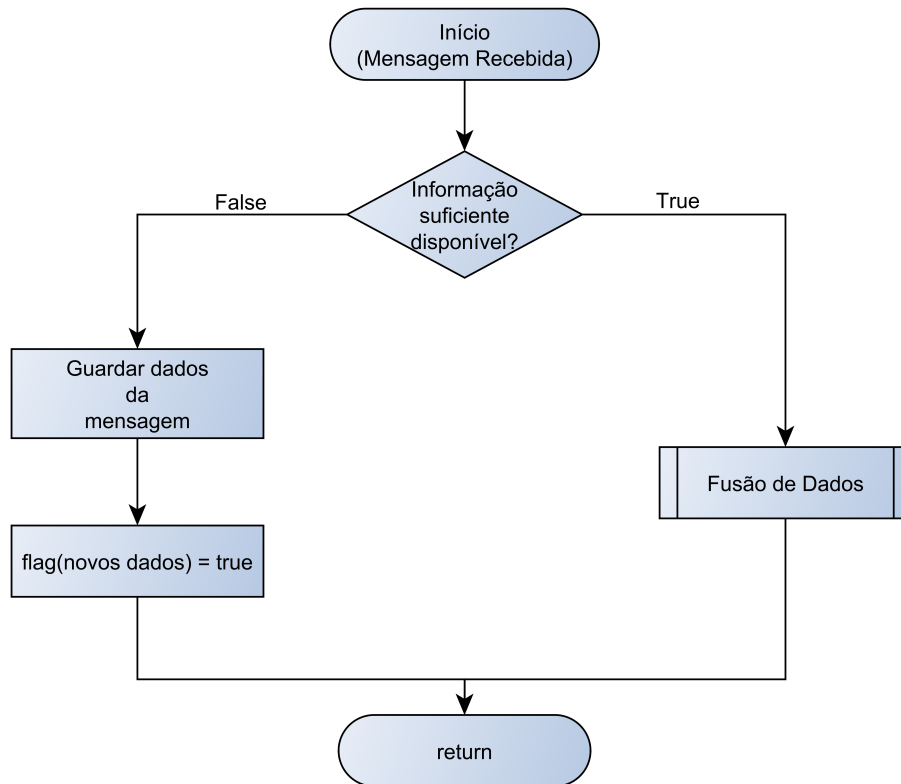


Figura 4.6: Algoritmo das *callbacks*

Resumidamente, sempre que é recebida uma mensagem *pointcloud* ou de odometria é verificado se já existe informação suficiente para realizar a fusão de dados. Caso se confirme essa condição, a função responsável pela fusão de dados é chamada. Na eventualidade de ainda não existir informação suficiente, os dados da mensagem são copiados para estruturas previamente criadas e uma *flag* é acionada para assinalar que existe informação sobre *pointclouds* ou sobre odometria disponível. Caso se receba nova informação antes da precedente ser utilizada, a mais antiga é descartada.

4.3.3 Fusão de Dados

Após se obter a informação necessária dos subscritores, a partir das *callbacks*, dá-se início ao processo de fusão dos dados provenientes da odometria e da *pointcloud*. Esta função, designada por "*pcl_join*", tem como parâmetros de entrada uma mensagem do tipo *PointCloud* e duas mensagens de odometria (*TwistedStamped*). O fluxograma, da figura 4.7 representa a sequência das operações realizadas nesta função.

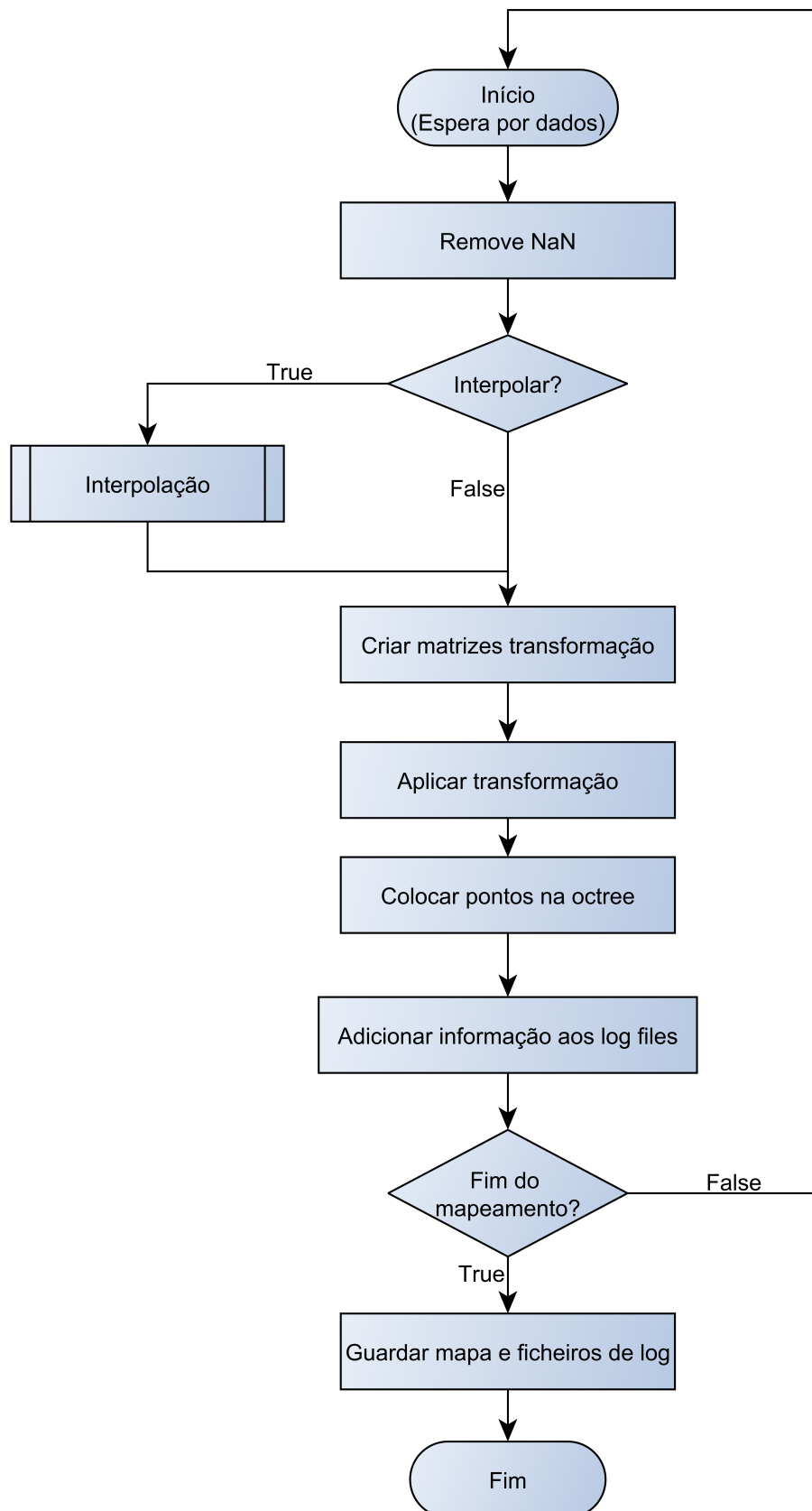


Figura 4.7: Algoritmo da fusão de dados

Inicialmente, a *pointcloud* recebida é filtrada, sendo os pontos de valor numérico não válido (*NaN*) removidos. Terminado este processo, é realizada a extração de informação dos *headers* das mensagens recebidas, especialmente das *timestamp*. Desta forma, é possível comparar a data de aquisição da *pointcloud* com a data de aquisição da odometria e, a partir da sua diferença perceber qual o intervalo temporal entre aquisições.

Caso esse intervalo ultrapasse o valor limite definido (atualmente 2 segundos), realiza-se uma interpolação dos valores da odometria para que se possa, mais tarde, colocar os pontos da *pointcloud* na região correta do referencial do mapa. Nas fórmulas usadas, para o cálculo dos novos valores de odometria, é considerado que a velocidade de deslocamento do veículo aquático é constante. As equações utilizadas para interpolar os valores de posição (4.1) e de orientação (4.2) são:

$$p_p = p_l + \frac{(p_a - p_l) \times (t_p - t_l)}{t_a - t_l} \quad (4.1)$$

p_p → novo valor de posição (x,y ou z) da *pointcloud*.

p_a → valor de posição (x,y ou z) fornecido pela última mensagem de odometria recebida.

p_l → valor de posição (x,y ou z) fornecido pela penúltima mensagem de odometria recebida.

t_p → valor da marca temporal da *pointcloud*.

t_a → valor da marca temporal da última mensagem de odometria recebida.

t_l → valor da marca temporal da penúltima mensagem de odometria recebida.

$$o_p = o_l + \frac{(o_a - o_l) \times (t_p - t_l)}{t_a - t_l} \quad (4.2)$$

o_p → novo valor de orientação (ϕ , θ ou ψ) da *pointcloud*.

o_a → valor de orientação (ϕ , θ ou ψ) fornecido pela última mensagem de odometria recebida.

o_l → valor de orientação (ϕ , θ ou ψ) fornecido pela penúltima mensagem de odometria recebida.

t_p → valor da marca temporal da *pointcloud*.

t_a → valor da marca temporal da última mensagem de odometria recebida.

t_l → valor da marca temporal da penúltima mensagem de odometria recebida.

É importante mencionar que o algoritmo é capaz de interpolar o valor da posição e orientação do veículo aquático autônomo, à data da aquisição da *pointcloud*, mas não se encontra preparado para fazer o processo inverso. Ou seja, extrapolar valores de posição ou preencher zonas sem informação sobre *pointclouds*. Contudo, esta funcionalidade poderá ser implementada no futuro.

Relativamente às mensagens com informação sobre as *pointclouds*, quando estas chegam ao nó de mapeamento, os valores das coordenadas dos seus pontos apenas representam as medidas da superfície, não existindo qualquer informação sobre a localização relativa deles. Como tal, é necessário realizar transformações geométricas tridimensionais para colocar esses pontos na zona

correta do referencial do mundo. Para isso, são criadas matrizes de translação e rotação, onde os seus parâmetros correspondem aos valores da odometria, sendo que estas transformações são aplicadas a toda a *pointcloud*.

$$\begin{bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Figura 4.8: Matriz da translação dos pontos[34]

$$\begin{bmatrix} x_{Rx} \\ y_{Rx} \\ z_{Rx} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) & 0 \\ 0 & \sin(a) & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Figura 4.9: Matriz da rotação dos pontos em torno de x [34]

$$\begin{bmatrix} x_{Ry} \\ y_{Ry} \\ z_{Ry} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(a) & 0 & \sin(a) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(a) & 0 & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Figura 4.10: Matriz da rotação dos pontos em torno de y [34]

$$\begin{bmatrix} x_{Rz} \\ y_{Rz} \\ z_{Rz} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(a) & -\sin(a) & 0 & 0 \\ \sin(a) & \cos(a) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Figura 4.11: Matriz da rotação dos pontos em torno de z [34]

Após todo o processamento da *pointcloud*, os pontos são adicionados à *octree* por forma a completar, com mais uma porção de informação, o mapa da estrutura submersa. Por fim, são

guardados em ficheiros de texto (.txt) os valores do tempo de ciclo do programa e a quantidade de pontos adicionados ao mapa.

Terminada a missão de mapeamento, e conseqüentemente a execução do programa, a informação presente na *octree* é guardada num ficheiro *point cloud data* (.pcd) para que possa ser posteriormente, analisada e é colocado num ficheiro de *log* (.txt) a data de início e de fim da operação e o número total de pontos do mapa.

4.3.3.1 Octree

Nesta dissertação, a receção de *pointclouds* é um processo que se repete ciclicamente e, por conseguinte, é necessária a utilização de uma estrutura onde estas possam ser agrupadas e guardadas para que no final se obtenha o mapa com todos os pontos recebidos durante as missões de mapeamento. Devido à sua eficiência na utilização dos recursos computacionais, nesta tese, a estrutura utilizada foi a *octree*.

As *octrees* são estruturas de dados em árvore, onde cada nó que a compõe tem oito filhos, ou então nenhum filho, capazes de organizar informação tridimensional proveniente de *pointclouds* [35].

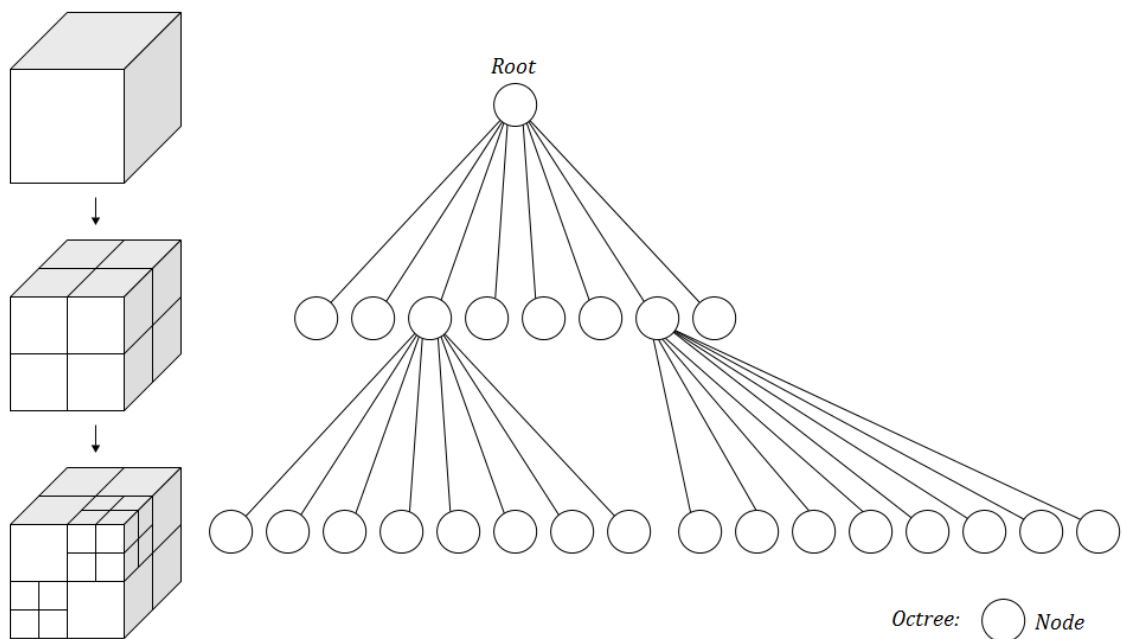


Figura 4.12: Estrutura *Octree* [36]

A raiz da árvore (*root*) é equivalente a uma caixa cúbica onde estão contidos todos os pontos que pertencem ao mapa. À medida que os nós vão sendo expandidos, a profundidade da árvore aumenta, e o espaço dentro dessa caixa vai-se subdividindo, recursivamente, em cubos (*voxels*) cada vez mais pequenos até um valor limite definido pelo parâmetro de resolução da *octree*. Este parâmetro descreve o comprimento do *voxel* que se encontra no último nível de profundidade da árvore. Por conseguinte, a profundidade é uma função da resolução escolhida [35].

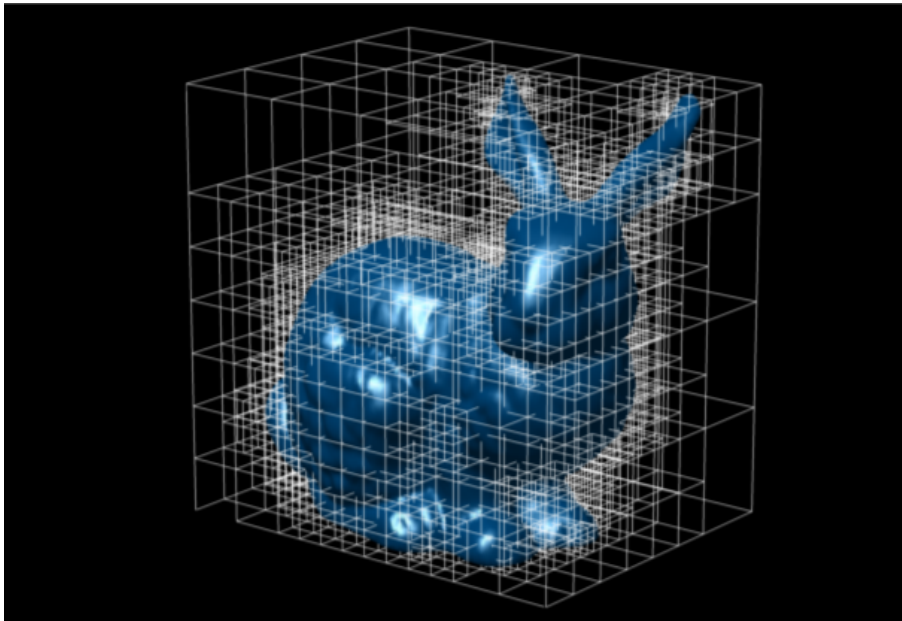


Figura 4.13: Subdivisão de *voxels* para acomodar pontos para mapear um coelho [37]

A importância de utilizar este tipo de estruturas, nesta dissertação, prende-se com o facto do sistema estar constantemente a receber nova informação para acrescentar ao mapa. Como o espaço disponível é definido pelo parâmetro de resolução, a alocação de memória é realizada à priori e não durante o processo de concatenação das várias *pointclouds*. Desta forma evitam-se processos de alocação e desalocação de memória que teriam impacto negativo na eficiência do algoritmo implementado.

Capítulo 5

Resultados Obtidos

Neste capítulo são apresentados os resultados obtidos durante os testes realizados ao sistema de mapeamento implementado. Na secção 5.1 são apresentados os resultados provenientes de testes realizados com dados simulados. Posteriormente, na secção 5.2, são demonstrados os resultados obtidos dos testes realizados no laboratório. Por fim, na secção 5.3 é feita uma breve conclusão ao capítulo.

5.1 Testes de validação com recurso a valores simulados

Para se proceder à análise do desempenho do algoritmo desenvolvido foi criado um nó ROS cuja função é construir e publicar periodicamente, a uma frequência de 25 *hz* (*hertz*), mensagens com *pointclouds* e mensagens com valores de posição (x,y,z) e de orientação (ϕ , θ ou ψ). Basicamente, a função deste nó é simular o envio da informação do *Line Extraction Node*, que funciona à frequência anteriormente mencionada, e do *Navigation Data Processing Node*.

Durante a execução deste programa, os parâmetros de posição e de orientação são iterativamente alterados, por forma a simular os movimentos de translação e rotação de um AUV, e as *pointclouds* são preenchidas de modo a simular alterações de relevo captadas pelo sistema de aquisição ótico.

Terminado o teste, o ficheiro do mapa, em formato *.pcd*, é visualizado e analisado com uma ferramenta apropriada, o *pcl_viewer*. Posteriormente, a informação dos ficheiros de *log*, onde são guardados o tempo de cada ciclo e o número de pontos do mapa, são tratados recorrendo ao Matlab.

O principal objetivo destes testes é avaliar o algoritmo responsável pela fusão dos dados provenientes dos tópicos aos quais este nó subscreve. No entanto, a sua performance é afetada pelo sistema onde este é executado e, como tal, é necessário ter essa componente em atenção. As experiências que são apresentadas nas subsecções que se seguem foram realizadas num sistema com as seguintes especificações:

Sistema	
Processador:	Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.20 GHz
Memória instalada (RAM):	8,00 GB

Figura 5.1: Características do sistema usado nos testes

5.1.1 Simulação de rampas

Inicialmente, foram realizados dois testes onde, em cada um deles, foi simulada uma superfície lisa com um declive de aproximadamente 21.8° . O principal objetivo destas simulações é observar o impacto da alteração do número de mensagens, com espaçamento entre *pointclouds* fixo, na capacidade de processamento do *CPU* e, conseqüentemente, no tempo de ciclo da execução do programa de fusão.

Para além disso, também é possível verificar se existe perda de mensagens ou falhas na reconstrução pois, sendo este um ambiente de teste controlado, é fácil perceber qual o resultado esperado à priori.

A legenda apresentada serve de auxílio à leitura das tabelas onde são apresentados os parâmetros das experiências:

- f - Frequência de envio das mensagens
- $n^\circ.M$ - Número de mensagens enviadas
- $n^\circ.P$ - Número de pontos por *pointcloud*
- $n^\circ.TP$ - Número total de pontos
- $e.p$ - Espaçamento entre pontos de 1 *pointcloud*
- $e.pcl$ - Espaçamento entre *pointclouds*
- $dist$ - Distância total mapeada
- $área$ - Área total mapeada

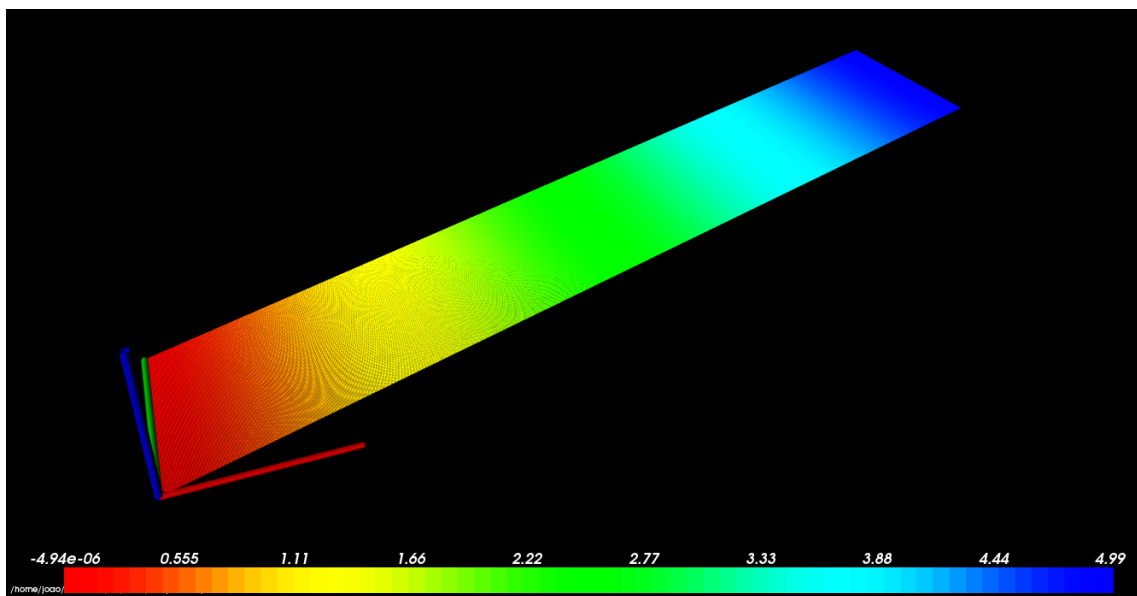
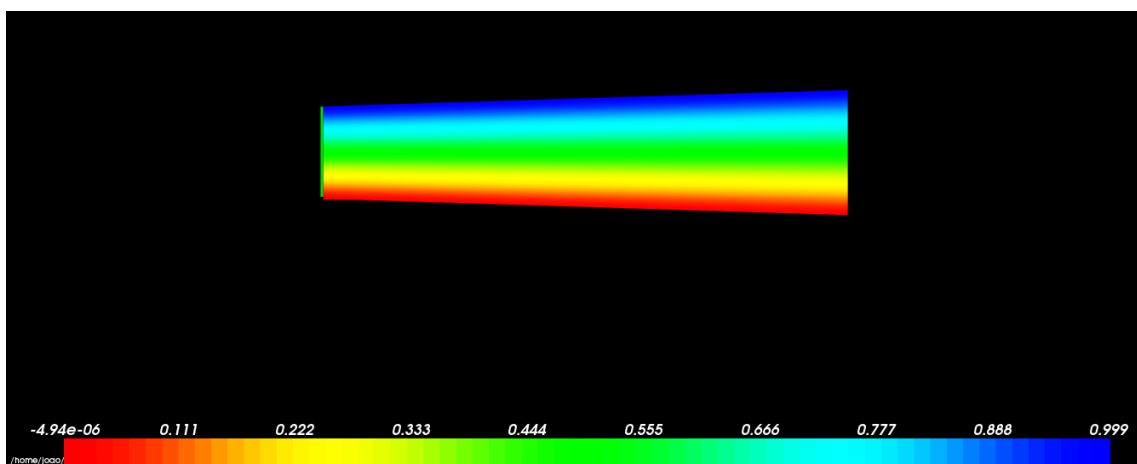
Na tabela 5.1 são apresentados os parâmetros do primeiro teste. Como mencionado anteriormente, foi considerada uma frequência de envio de mensagens de $25hz$. Ao todo foram enviadas 1000 mensagens onde foi considerado que o AUV se deslocou 0.005 metros entre aquisições de imagem sendo, no total, percorrida uma distância de 5 metros.

Também foi considerado que a câmara tem resolução vertical e horizontal igual e uma resolução de *1Megapixel* e, por conseguinte, realiza a extração de 1000 pontos, onde cada um representa 0.001 metros no mundo.

Tabela 5.1: Pârametros de teste da rampa com 10^6 pontos

<i>Mapa</i>	Pârametros							
	$f(hz)$	n°.M	n°.P	n°.TP	$e.p(m)$	$e.pcl(m)$	$dist(m)$	área(m^2)
Rampa A	25	1000	1000	10^6	0.001	0.005	5	5

Nas figuras que se seguem é possível observar o resultado da reconstrução. Na imagem 5.2 é apresentada a distância total percorrida (5 metros), na *top view* 5.3 observa-se a largura da superfície reconstruída (aproximadamente 1 metro) e na *side view* 5.4 é observada a altura da rampa (aproximadamente 2 metros).

Figura 5.2: Simulação de uma rampa com 10^6 pontosFigura 5.3: Simulação de uma rampa com 10^6 pontos (*top view*)

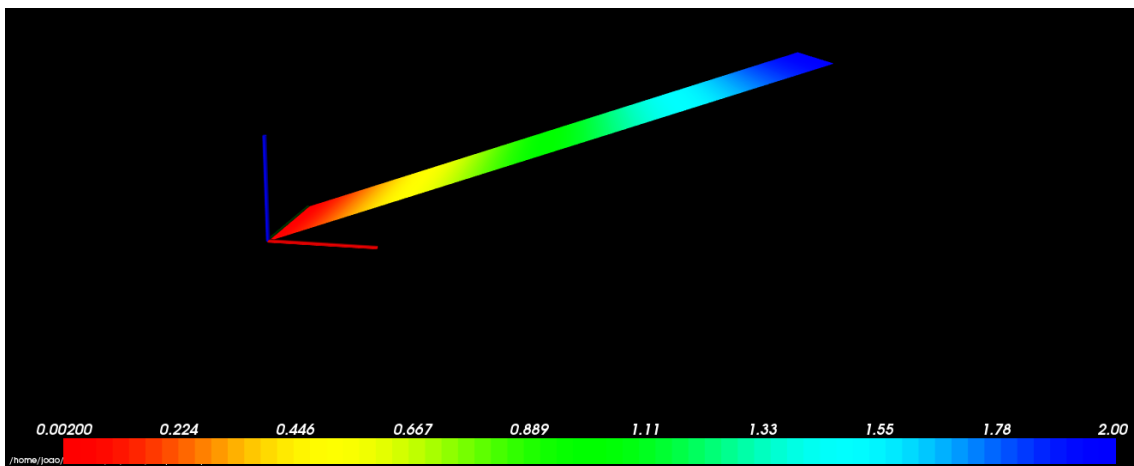


Figura 5.4: Simulação de uma rampa com 10^6 pontos (*side view*)

No gráfico 5.5 observam-se as variações dos tempos de ciclo, em milissegundos (ms), ao longo do processo de mapeamento. Através deste e da análise de tabela 5.2, é possível perceber se ocorreram perturbações na execução do algoritmo e, comparando com outros testes, compreender o motivo pelo qual estas acontecem.

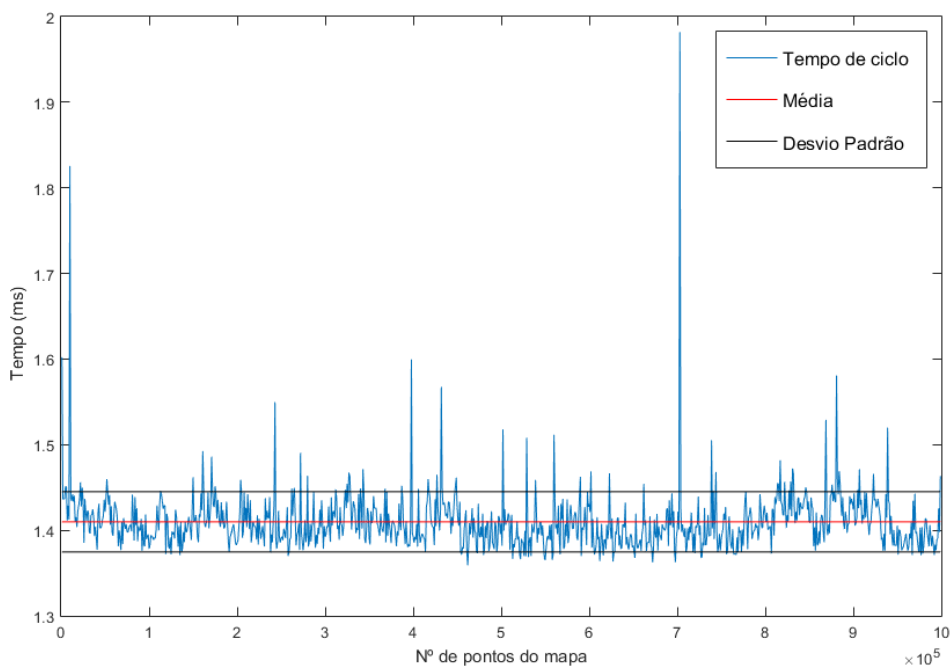


Figura 5.5: Gráfico da simulação da rampa com 10^6 pontos - tempo de ciclo vs número de pontos

Tabela 5.2: Resultados do teste da rampa com 10^6 pontos

Resultados				
<i>Mapa</i>	média (ms)	σ (ms)	<i>outliers</i>	n ^o mensagens recebidas
Rampa A	1.4098	0.0353	123	1000

Para este teste o tempo médio para cada ciclo foi de 1.4098ms, sendo o desvio padrão igual (σ) a 0.0353ms. Ao todo, dos 1000 ciclos de fusão que ocorreram, houve 123 que ultrapassaram os valores esperados tendo em consideração a dispersão dos tempos apresentados. No entanto, isto é provocado pela baixa variação entre os tempos de ciclo, o que resulta no baixo valor do σ , e não por valores de pico, que podem afetar o desempenho do sistema. É também importante referir que, neste caso, foram recebidas e processadas todas as 1000 mensagens enviadas obtendo-se uma reconstrução com 1 milhão de pontos, sendo que esta demorou aproximadamente 1 minuto.

À semelhança da rampa A, na tabela 5.3 são apresentados os pãrametros do teste para a reconstrução da segunda rampa. As suas características são semelhantes, sendo que a principal diferença é o número de mensagens enviadas. Esta alteração faz com que a distância percorrida seja de 25 metros e, como tal, a área mapeada será de $25m^2$.

Tabela 5.3: Pãrametros de teste da rampa com 5×10^6 pontos

Pãrametros								
<i>Mapa</i>	<i>f</i> (hz)	n ^o .M	n ^o .P	n ^o .TP	e.p(m)	e.pcl(m)	dist(m)	área(m ²)
Rampa B	25	5000	1000	5×10^6	0.001	0.005	25	25

A partir das imagens é possível observar: a distância total percorrida (25 metros) em 5.6, a largura da superfície reconstruída (aproximadamente 1 metro) na *top view* 5.7 e a altura da rampa (10 metros) na *top view* 5.8.

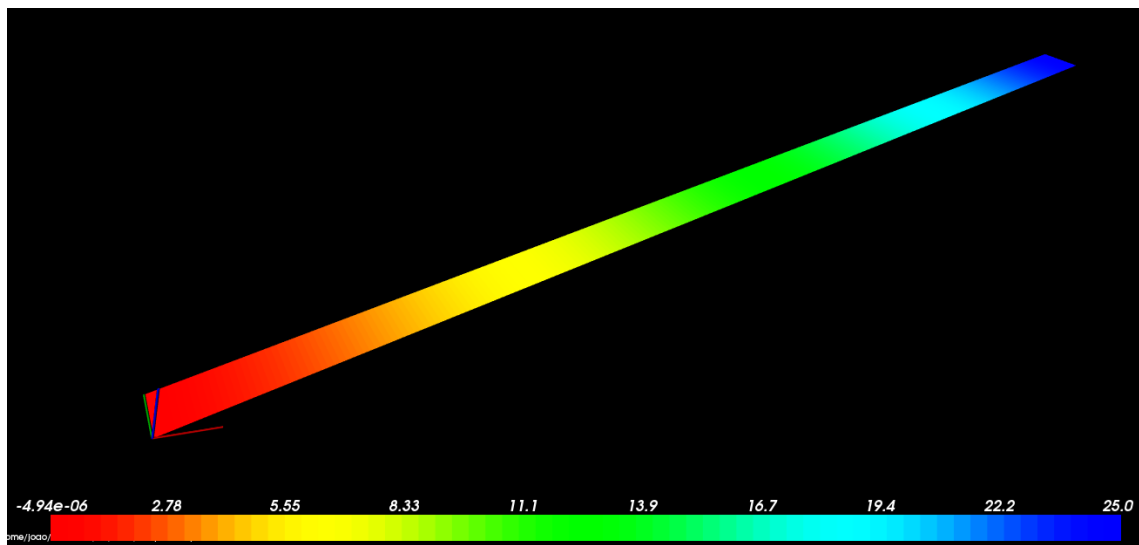


Figura 5.6: Simulação de uma rampa com 5×10^6 pontos

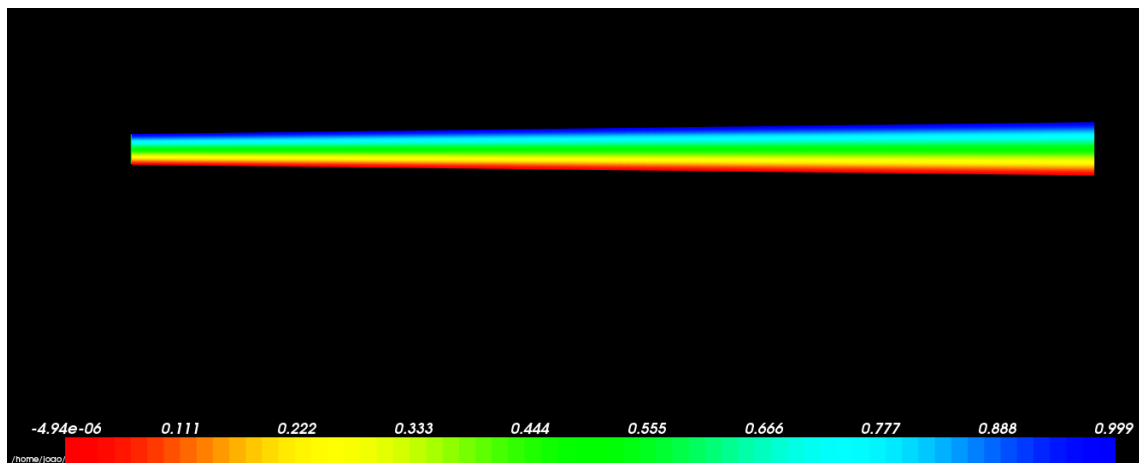


Figura 5.7: Simulação de uma rampa com 5×10^6 pontos (*top view*)

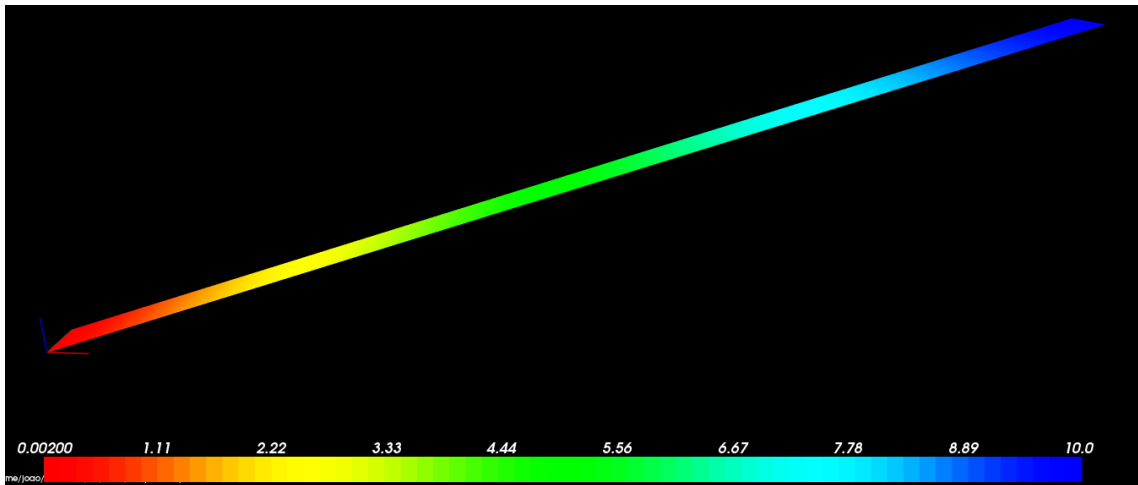


Figura 5.8: Simulação de uma rampa com 5×10^6 pontos (*side view*)

Com base nos gráficos 5.9 e 5.10 e na tabela 5.4, é notório que o aumento da quantidade de *pointclouds* no mapa não provoca alterações significativas aos tempos de ciclo do programa comparativamente ao caso da rampa A.

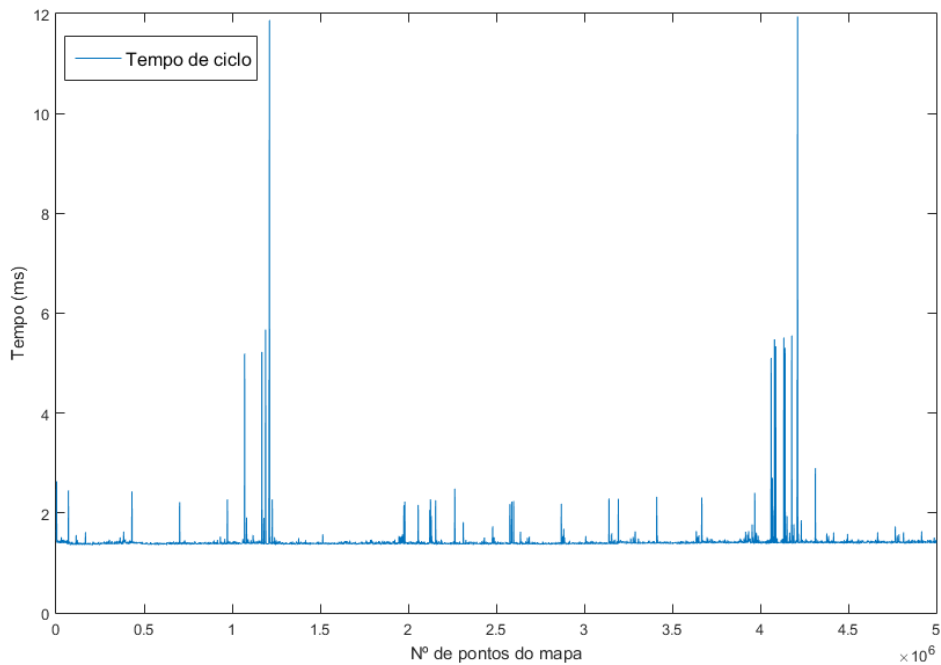


Figura 5.9: Gráfico da simulação da rampa com 5×10^6 pontos - tempo de ciclo vs número de pontos

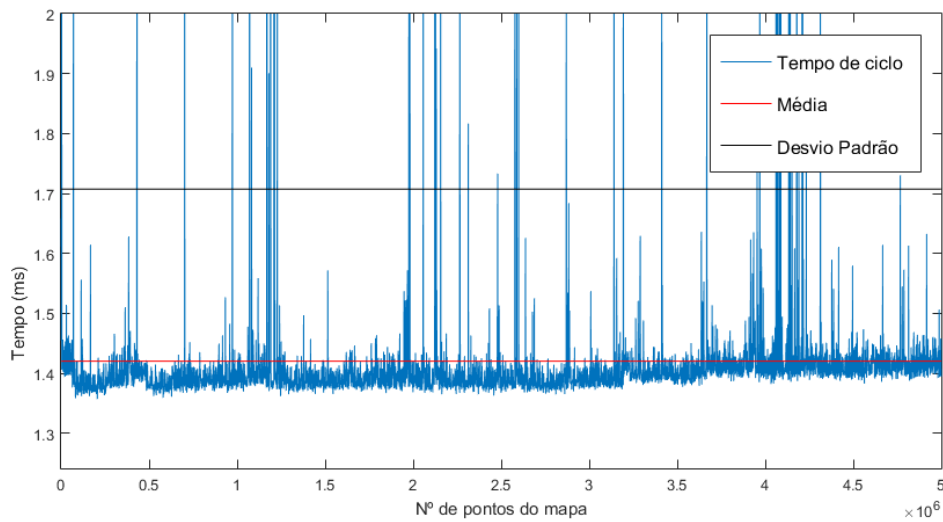


Figura 5.10: Gráfico da simulação da rampa com 5×10^6 pontos - tempo de ciclo vs número de pontos (*close view*)

Tabela 5.4: Resultados do teste da rampa com 5×10^6 pontos

Resultados				
<i>Mapa</i>	média (ms)	σ (ms)	<i>outliers</i>	nº mensagens recebidas
Rampa B	1.4205	0.2870	51	4999

Nesta simulação, o tempo médio de cada ciclo foi de 1.4205ms e o σ igual a 0.2870ms. Apesar dos picos observados serem superiores aos verificados em 5.5, apenas 51 ciclos podem ser considerados *outliers*. Como tal, pode-se inferir que o algoritmo foi executado dentro dos valores normais para este caso. No que diz respeito à robustez em termos de informação recebida e processada, foram ao todo usadas 4999 mensagens das 5000 enviadas em cerca de 3 minutos e 30 segundos.

O objetivo fundamental destas simulações era avaliar o impacto que o número de mensagens, com espaçamento entre *pointclouds* fixo, poderia ter na capacidade de processamento do sistema de reconstrução. Com base nos resultados obtidos, pode-se dizer que a influência é mínima.

Em ambos os casos o tempo para cada ciclo rondou os 1.4ms, estando a principal diferença relacionada com a dispersão dos seus valores. No caso da rampa B o desvio padrão é maior do que no caso da rampa A, o que indica que os tempos de ciclo estão espalhados por uma gama de valores superior.

Apesar disso, em ambas as situações não se verificaram problemas relativamente à consistência das reconstruções efetuadas, sendo que os resultados obtidos em termos da imagem corresponderam ao que era esperado que fosse observado.

5.1.2 Simulação de ondas

A segunda simulação passa pela criação de superfícies curvas, neste caso em forma de onda sinusoidal, sendo que para replicar esse desnível de revelo se utilizou uma função seno com um passo de $\frac{1}{4}$ de grau por cada iteração do nó de teste. O objetivo fundamental destes testes é poder analisar a influência que o número de pontos presentes na *pointcloud*, criada a partir da extração de uma linha laser, pode ter na capacidade de processamento do sistema. Para isso, são realizados 2 testes onde o número de mensagens publicadas é mantido fixo e o número de pontos por *pointcloud* é duplicado num dos casos.

Além disso, tal como na subsecção anterior (5.1.1), estas simulações permitem verificar se existe perda de mensagens ou falhas na reconstrução durante a operação de mapeamento.

A tabela 5.5 contém algumas das características da primeira onda. No total foram publicadas 1440 mensagens assumindo que a distância percorrida entre aquisições é de 0.005 metros, sendo percorrida uma distância de 7.2 metros. Para esta primeira simulação foi considerado que para cada *pointcloud* são extraídos 1000 pontos espaçados de 0.001 metros.

Tabela 5.5: Pârametros de teste da onda com 1.44×10^6 pontos

Mapa	Pârametros							
	$f(\text{hz})$	n°.M	n°.P	n°.TP	e.p(m)	e.pcl(m)	dist(m)	área(m ²)
Onda A	25	1000	1440	1.44×10^6	0.001	0.005	7.20	7.20

Nas figuras 5.11, 5.12 e 5.13 é apresentada a forma de onda reconstruída com recurso à informação presente na tabela 5.5 e as respetivas dimensões de comprimentos (7.20m), largura (1m) e amplitude (2m).

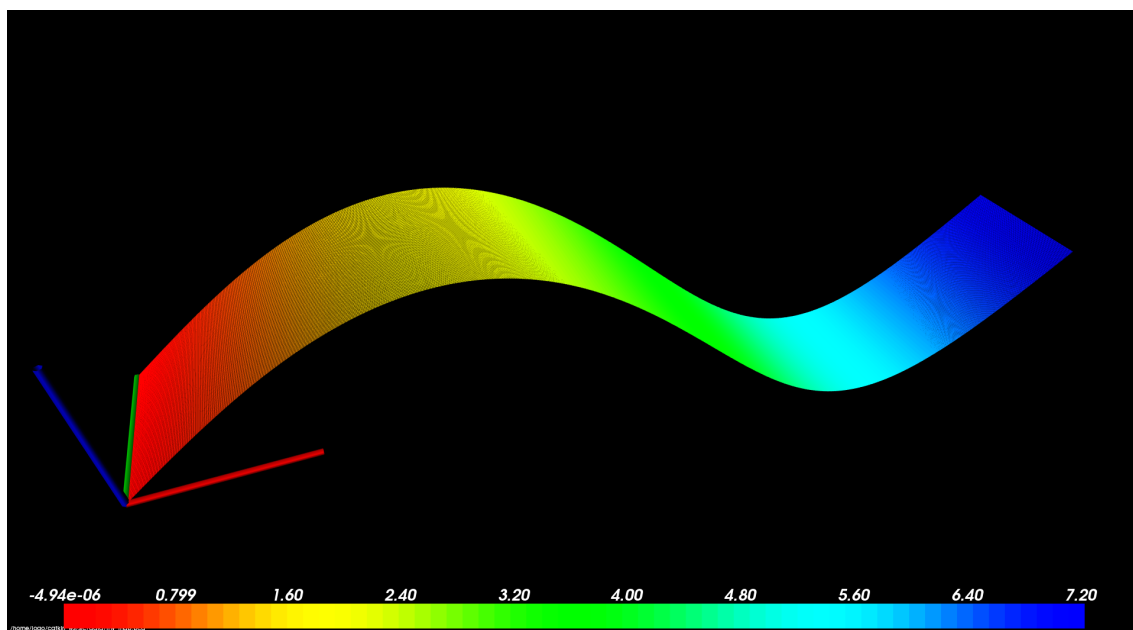


Figura 5.11: Simulação de uma onda com 1.44×10^6 pontos

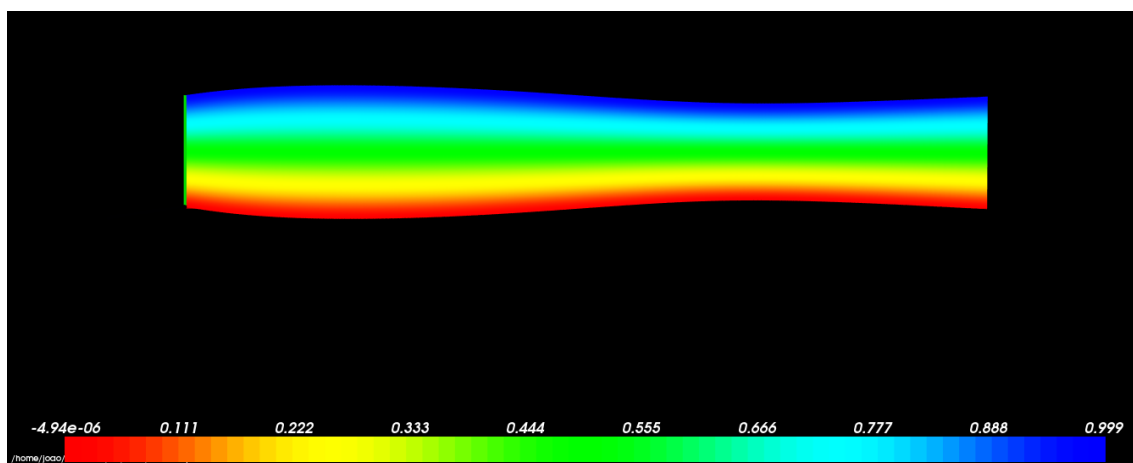


Figura 5.12: Simulação de uma onda com 1.44×10^6 pontos (*top view*)

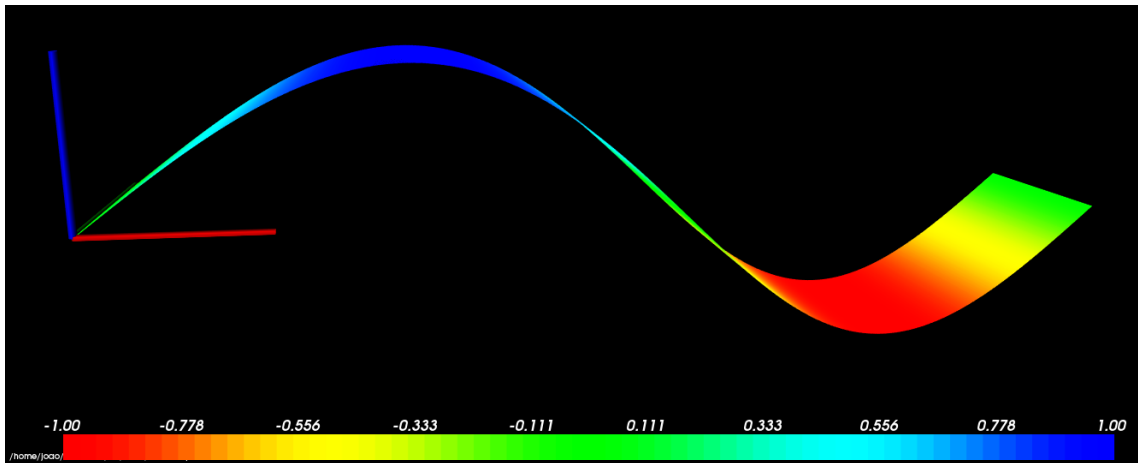


Figura 5.13: Simulação de uma onda com 1.44×10^6 pontos (*side view*)

Os resultados obtidos a partir dos ficheiros de *log*, gravados durante a execução do nó, permitiram a criação do gráfico 5.14 e da tabela 5.6. Com base nesses dados, é possível analisar o comportamento do nó durante o mapeamento.

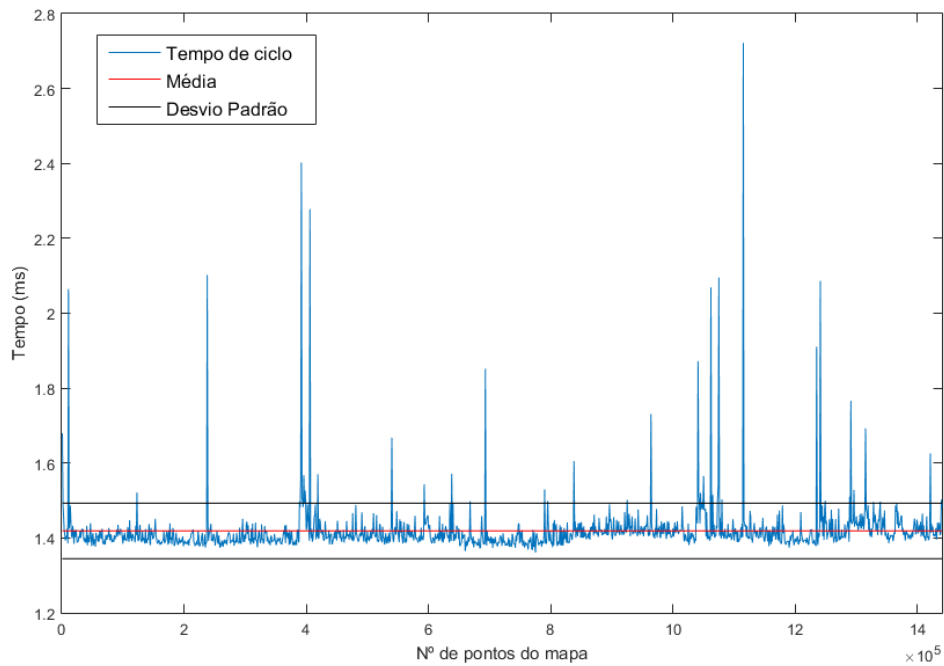


Figura 5.14: Gráfico da simulação da onda com 1.44×10^6 pontos - tempo de ciclo vs número de pontos

Tabela 5.6: Resultados do teste da onda com 1.44×10^6 pontos

Resultados				
<i>Mapa</i>	média (ms)	σ (ms)	<i>outliers</i>	nº mensagens recebidas
Onda A	1.4188	0.0742	48	1440

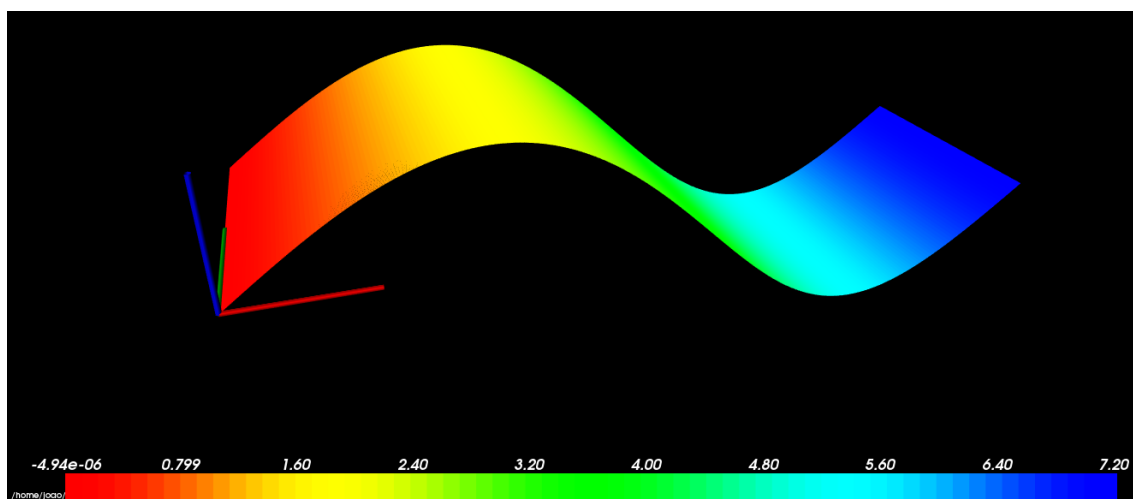
Em termos médios, o tempo de ciclo desta simulação assemelha-se ao dos casos das rampas analisadas na secção anterior (5.1.1). Sucintamente, é fácil observar que este teste foi caracterizado pela baixa variabilidade do tempo de ciclo, apesar de se observarem alguns picos pontuais no gráfico 5.14. Em relação às mensagens recebidas não se observaram perdas de dados.

Na tabela 5.7 são apresentados os parâmetros do segundo teste. Neste, foram de novo enviadas 1440 mensagens onde o valor de posição é incrementado em 0.005 metros a cada mensagem de odometria recebida. Ao contrário do que ocorre nas outras simulações, nesta foi considerado que o sistema ótico câmara-laser é capaz de extrair dados a partir de uma linha laser com 2 metros em vez de apenas 1. Na tabela 5.7 são apresentadas todas as características deste mapa.

Tabela 5.7: Parâmetros de teste da onda com 2.88×10^6 pontos

Parâmetros								
<i>Mapa</i>	f (hz)	nº.M	nº.P	nº.TP	$e.p(m)$	$e.pcl(m)$	dist(m)	área(m ²)
Onda B	25	1440	2000	2.88×10^6	0.001	0.005	7.20	14.40

Comparando as figuras 5.15, 5.16 e 5.17 com as da onda A, observa-se que esta tem o mesmo comprimento (7.20m) e amplitude (2m), mas o dobro da largura (2m).

Figura 5.15: Simulação de uma onda com 2.88×10^6 pontos

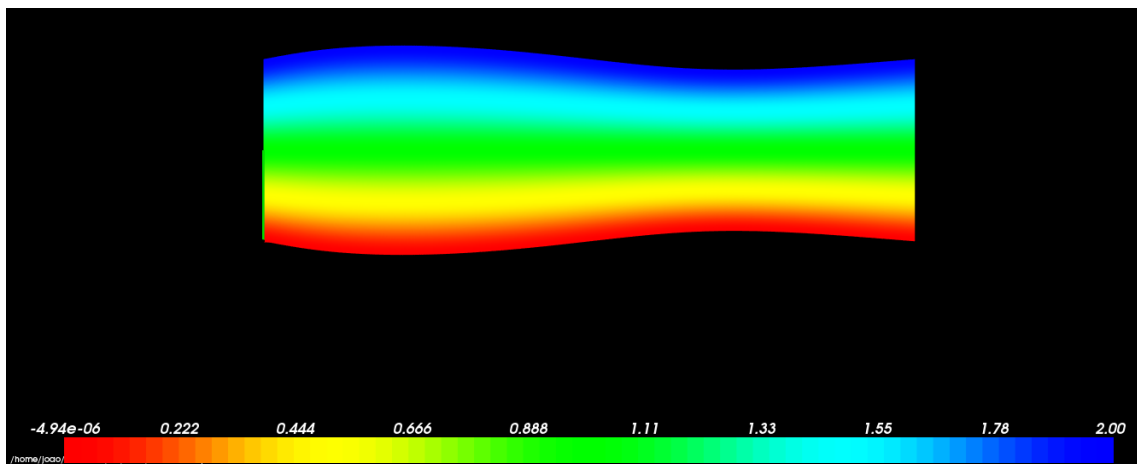


Figura 5.16: Simulação de uma onda com 2.88×10^6 pontos (*top view*)

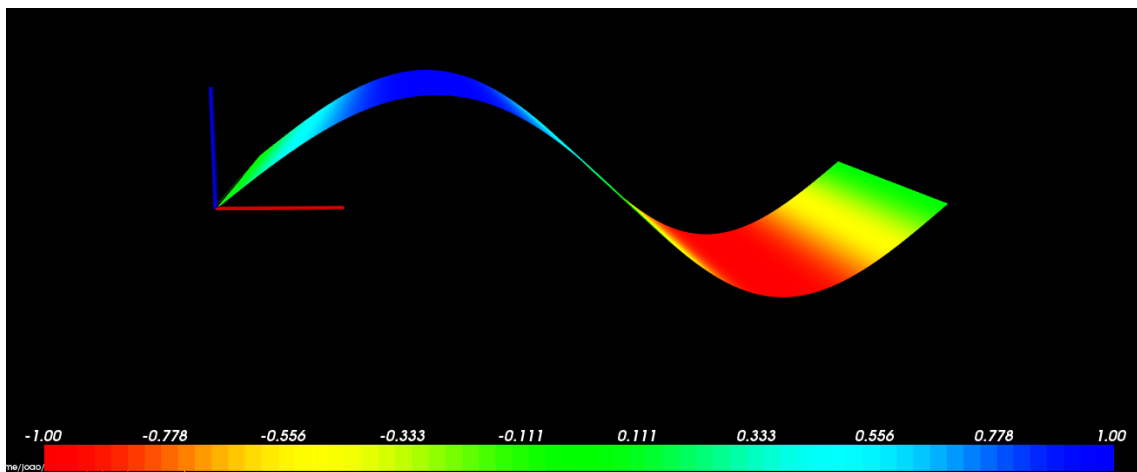


Figura 5.17: Simulação de uma onda com 2.88×10^6 pontos (*side view*)

A partir do gráfico 5.18 e da tabela 5.8 é possível perceber qual a diferença de aumentar a largura da extração de 1 metro para 2.

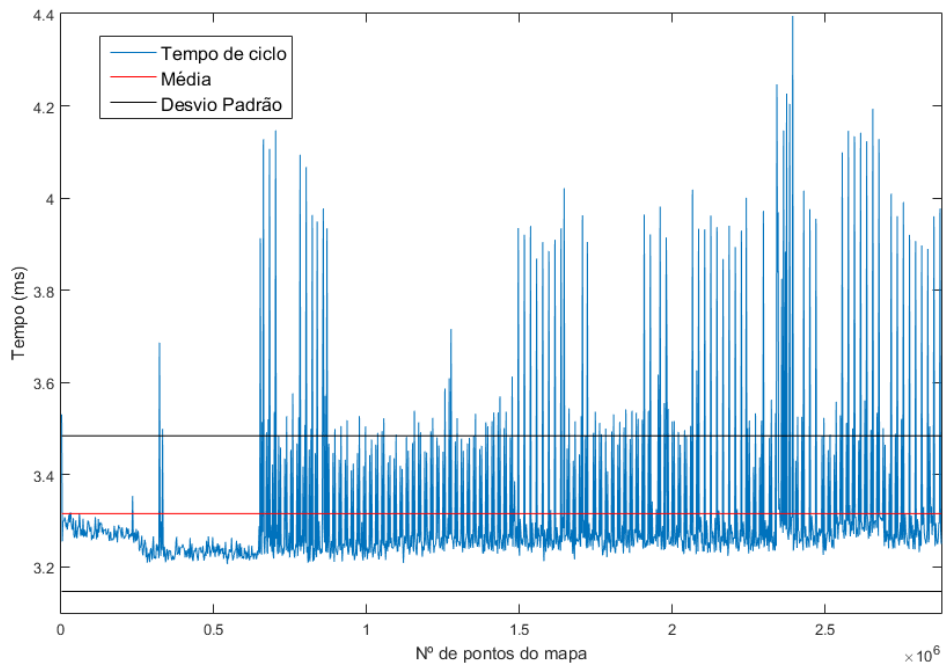


Figura 5.18: Gráfico da simulação da onda com 2.88×10^6 pontos - tempo de ciclo vs número de pontos

Tabela 5.8: Resultados do teste da onda com 2.88×10^6 pontos

Resultados				
<i>Mapa</i>	média (ms)	σ (ms)	<i>outliers</i>	nº mensagens recebidas
Onda B	3.3153	0.1689	158	1440

Ao duplicar o número de pontos de cada *pointcloud* publicada de 1000 para 2000, observa-se que o tempo médio de cada ciclo aumenta para mais do dobro e que a variabilidade da duração de cada ciclo também aumenta. Apesar disso, o mapeamento é feito de forma normal, não se verificando redução no número de mensagens recebidas em relação às enviadas.

O principal objetivo dos testes efetuados nesta subsecção era analisar de que forma o aumento ou redução do número de pontos das *pointclouds* publicadas afetava o tempo de ciclo do algoritmo desenvolvido e a qualidade dos mapas por ele criados. Comparando os resultados da onda A com a onda B, observa-se que existe um aumento do tempo de ciclo ao duplicar o número de pontos por *pointcloud* e, por conseguinte, a carga computacional exercida sobre o sistema é maior.

Este fenómeno ocorre devido ao facto de existir uma maior quantidade de pontos em cada *pointclouds* para processar. Outro motivo é o facto de os pontos de cada *pointcloud* publicada terem uma resolução de 0.001 metros, enquanto que o espaçamento entre *pointclouds* é de 0.005 metros. Ao duplicar o número de pontos por *pointcloud*, o sistema é obrigado a realizar mais divisões recursivas nos *voxels* da *octree* para acomodar esses pontos com resolução de 0.001 metros.

Em suma, são necessárias menos divisões dos *voxels* para pontos distanciados a 0.005m do que a 0.001m. Como tal, aumentar a profundidade de pesquisa da *octree* provoca um aumento do tempo de processamento. Mesmo assim, o algoritmo continua a não registar perdas de informação necessária à reconstrução dos mapas.

5.1.3 Simulação de um tubo

No último ensaio realizado, é simulado o mapeamento de uma estrutura semelhante a um tubo, que se encontra com metade da sua superfície enterrada. O objetivo desta simulação é perceber de que forma são utilizados os recursos do sistema, a nível do *CPU* e da memória *RAM*, à medida que a operação de mapeamento decorre e, consequentemente, à medida que o mapa aumenta. Para isso foi feito um teste de stress (controlado) ao sistema, monitorizando os valores do uso do *CPU* e da memória *RAM*.

Ao todo, foram publicadas 8000 mensagens, onde a cada mensagem de odometria é incrementada a posição do robô em 0.005 metros (em x) e em cada mensagem de *pointcloud* é enviada informação sobre a superfície externa do tubo, com uma resolução de 0.001 metros. Os parâmetros e característica da simulação realizada podem ser observados na tabela 5.9.

Tabela 5.9: Parâmetros de teste do tubo

<i>Mapa</i>	Parâmetros							
	$f(\text{hz})$	nº.M	nº.P	nº.TP	e.p(m)	e.pcl(m)	dist(m)	área(m ²)
Tubo	25	1000	8000	8×10^6	0.001	0.005	40	40

Na figura 5.19 é apresentada *top view* do tubo mapeado. Nesta é possível observar o comprimento total do tubo reconstruído (40 metros).

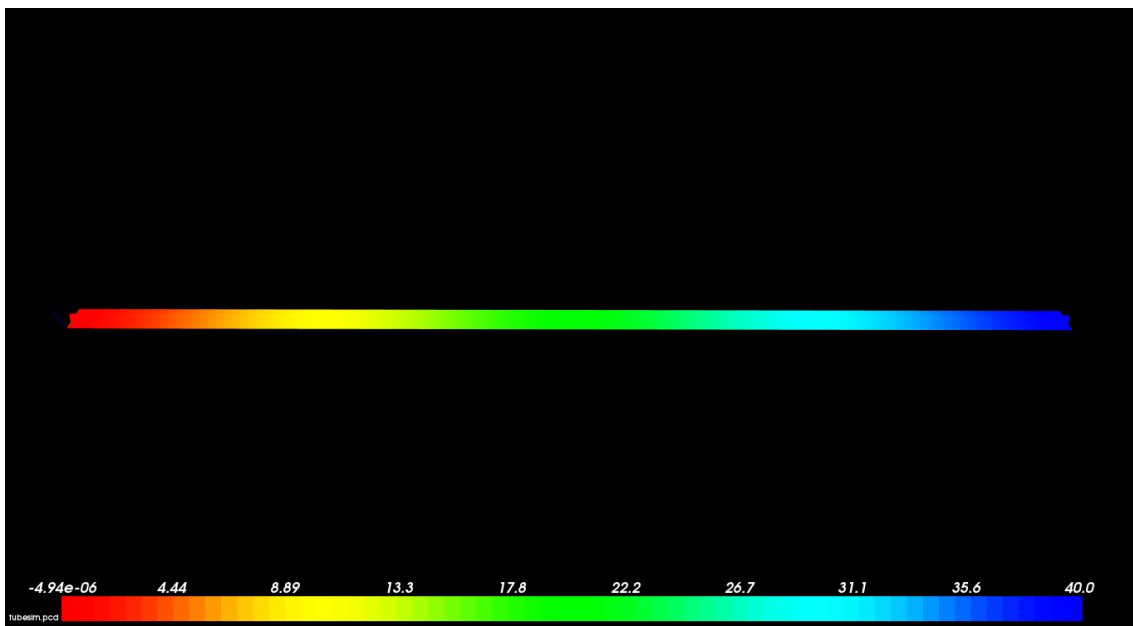


Figura 5.19: Simulação de um tubo com 8×10^6 pontos (*top view*)

Nas imagens 5.20 e 5.21, observa-se o relevo do tubo destacado de forma clara da superfície onde está inserido. Na *top view* é visto o tubo na íntegra, enquanto que no *close up* vê-se com mais detalhe a altura do mesmo, de 0.25 metros.

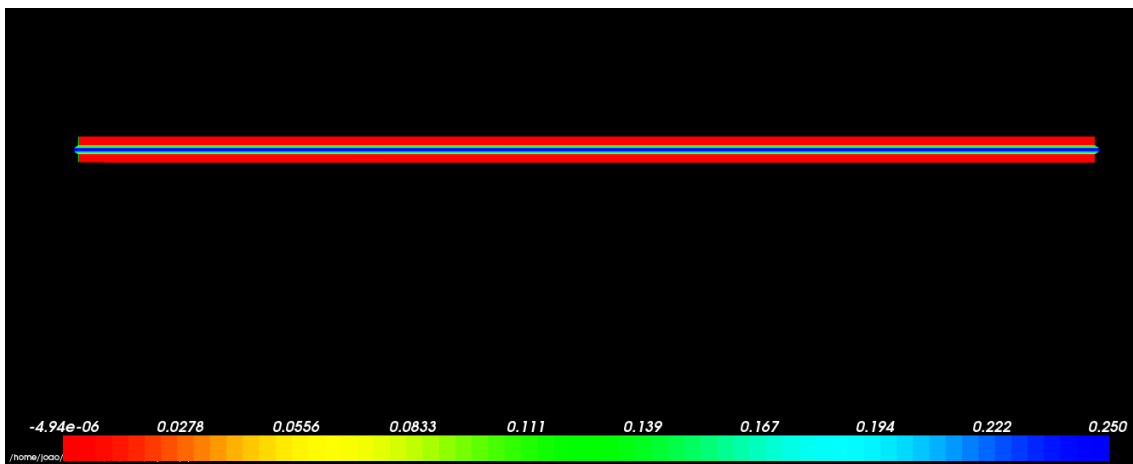


Figura 5.20: Simulação de um tubo com 8×10^6 pontos (*top view* com tubo destacado)

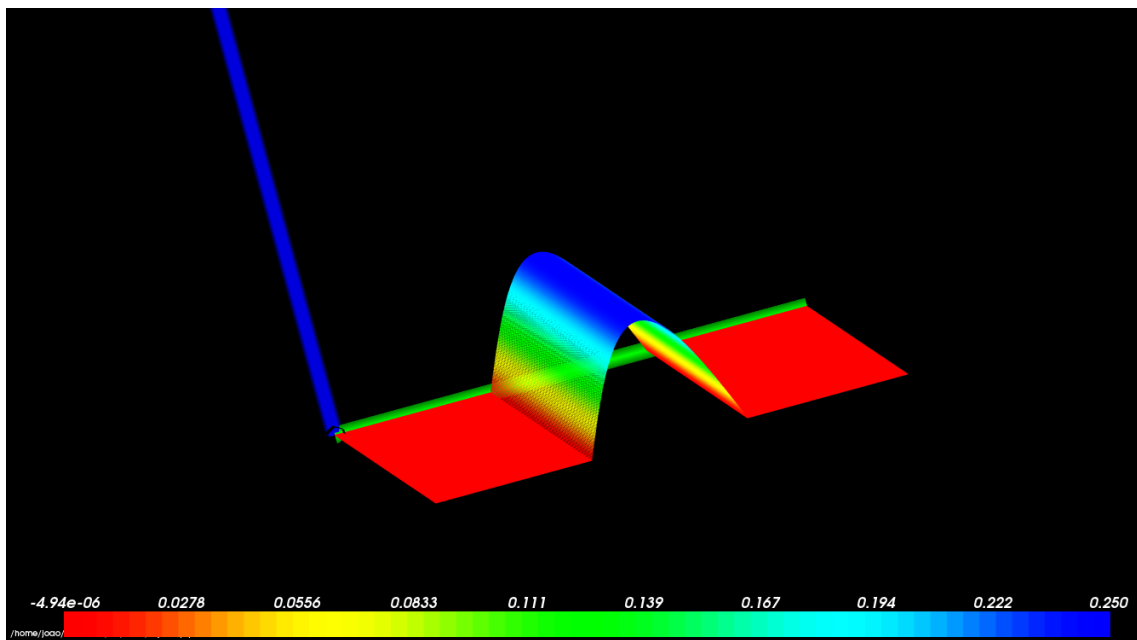


Figura 5.21: Simulação de um tubo com 8×10^6 pontos (*close up* com relevo destacado)

Por fim, na figura 5.22 é feita a análise da largura do varrimento efetuado (1 metro).

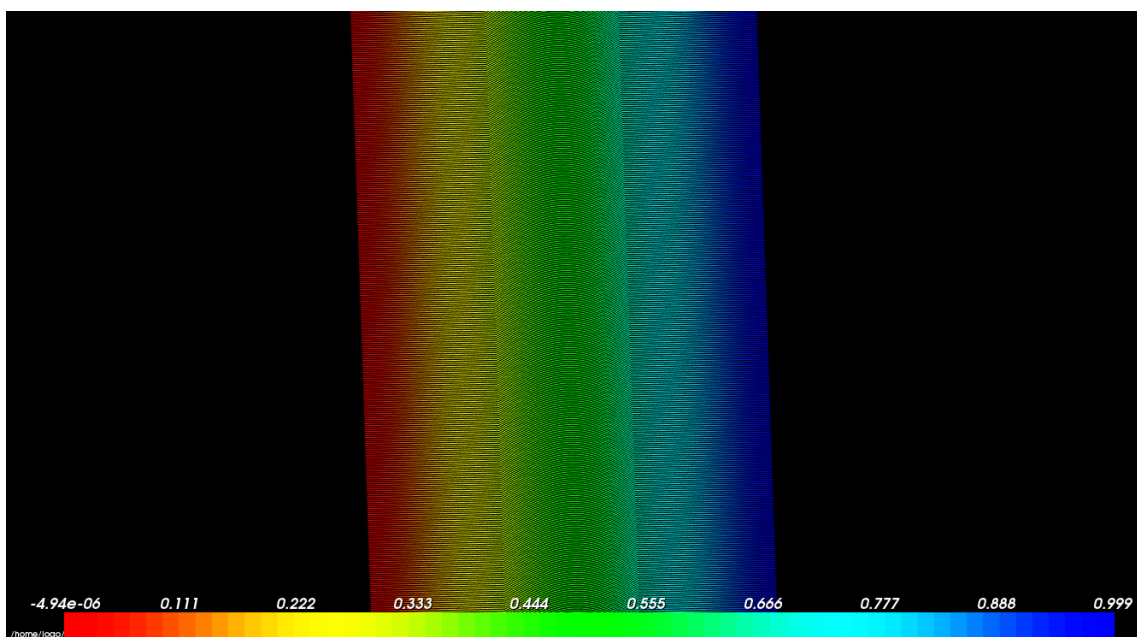


Figura 5.22: Simulação de um tubo com 8×10^6 pontos (*close up*)

No gráfico 5.23 estão representados os tempos de ciclo da execução do processo de reconstrução à medida que o tamanho do mapa aumenta. A partir deste, e com a ajuda da informação extraída dos ficheiros de *log* apresentados na tabela 5.10, é possível retirar algumas conclusões em relação ao desempenho do algoritmo durante esta simulação.

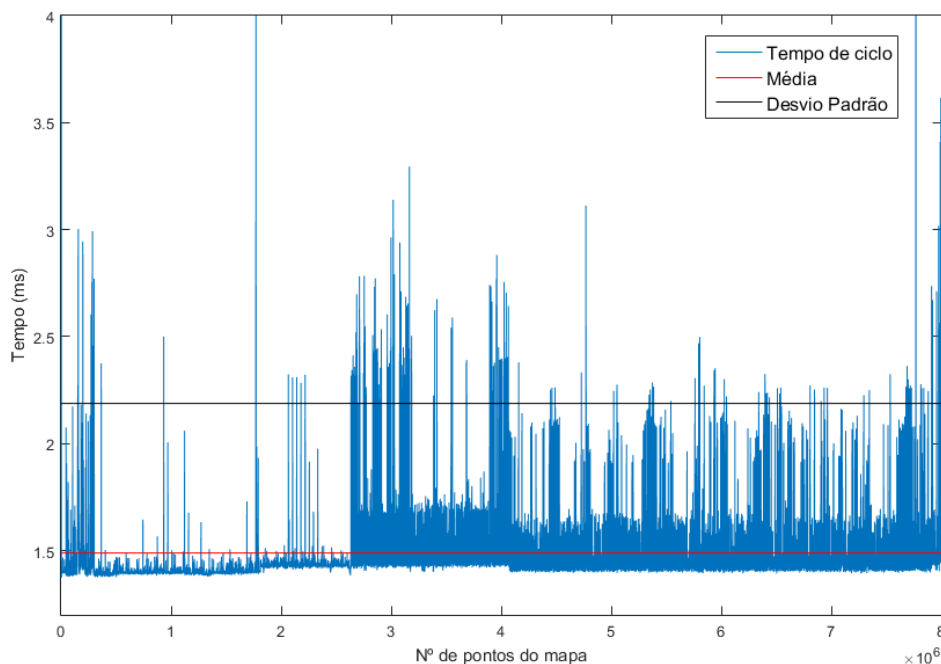


Figura 5.23: Gráfico da simulação de um tubo com 8×10^6 pontos - tempo de ciclo vs número de pontos

Tabela 5.10: Resultados do teste do tubo

Resultados				
<i>Mapa</i>	média (ms)	σ (ms)	<i>outliers</i>	nº mensagens recebidas
Tubo	1.4891	0.6984	166	7997

O algoritmo realizou toda a sua operação com um tempo de ciclo médio de 1.4891ms, sendo o desvio padrão σ observado igual a 0.6984 ms. Apesar da elevada variabilidade presente nos tempos de ciclo, apenas 166 podem ser considerados *outliers*, tendo-se perdido 3 mensagens das 8000 enviadas durante todo o processo de mapeamento. Em termos médios, relativamente ao tempo de ciclo, esta simulação, de maior dimensão, assemelha-se às analisadas nas subsecções 5.1.1 e 5.1.2.

Tal como foi mencionado no início desta análise, o objetivo desta simulação passa por compreender o usos dos recursos do sistema computacional durante a execução do mapeamento. Tendo isso em consideração, foi verificada a utilização do *CPU* e da memória *RAM* ao longo do processo. Nas figuras 5.24 e 5.25, é possível observar a percentagem de utilização dos recursos associados ao *CPU* e a quantidade de memória *RAM* usada no início e no final da execução do algoritmo.

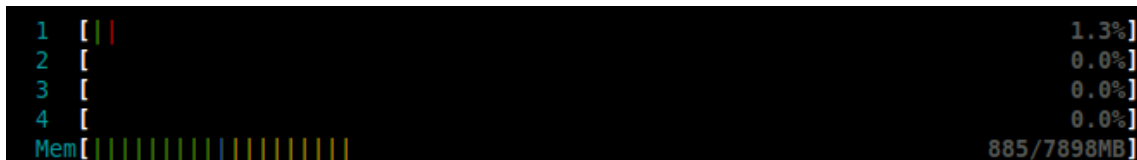


Figura 5.24: Utilização do *CPU* e da memória *RAM* no início do mapeamento



Figura 5.25: Utilização do *CPU* e da memória *RAM* no fim do mapeamento

O processador utilizado *Intel®Core™i5-5200U* é constituído por 2 núcleos. Devido à capacidade de *hyper-threading*, este processador consegue correr 2 *threads* em cada núcleo, comutando rapidamente entre os processos, em vez de apenas 1 [38]. Daí ser observada a percentagem de utilização de aparentemente 4 núcleos em vez de 2.

Como se pode verificar, no momento que precede o início do programa, a percentagem de utilização do processador em *idle* é baixa. Após se iniciar a execução do algoritmo, a utilização do processador manteve-se de forma constante nos valores observados na figura 5.25. Como tal, o aumento progressivo da dimensão do mapa, no que diz respeito à quantidade de pontos, não afetou a sua capacidade em termos de processamento de informação.

Relativamente à utilização de memória *RAM*, observa-se que à medida que o número de pontos no mapa aumenta, a quantidade de memória disponível diminui, sendo que esta só é libertada quando se finaliza a gravação do mapa. Neste caso, para o mapeamento dos 40 metros de tubo com 8 milhões de pontos, verificou-se que esta operação ocupou aproximadamente 75.5% da memória disponível.

O caso apresentado nesta secção tinha como principal objetivo avaliar de que forma são utilizados os recursos computacionais do sistema à medida que a operação de mapeamento decorre.

A partir da análise de todos os dados e figuras recolhidas pode concluir-se que o tempo de ciclo se mantém relativamente constante, não sendo observados picos durante o processamento capazes de deteriorar a qualidade do mapa criado. No que diz respeito à utilização de recursos, conclui-se que o fator que limita o tamanho dos mapas que se podem criar é a quantidade de memória *RAM* disponível e não o *CPU*, apesar deste também desempenhar um papel fundamental neste processo.

Este teste reforça a ideia de que o aumento da dimensão do mapa não tem um impacto significativo no tempo de cada ciclo de execução do algoritmo, a menos que se aumente a resolução das *pointclouds* criadas, e que o nó desenvolvido é capaz de operar a uma frequência de 25hz sem apresentar perdas relevantes de informação sobre *pointclouds*.

5.1.4 Eficiência dos métodos de concatenação

No decorrer da implementação foram considerados dois métodos de junção de *pointclouds*. O primeiro consistia num processo de concatenação, ou seja, à medida que as *pointclouds* chegavam ao nó de mapeamento (*Mapping Node* - figura 4.2) era alocado espaço para acomodar a informação. O segundo recorria ao uso de *octrees* que foram abordadas na secção 4.3.3.1 do capítulo 4.

Este teste foi realizado com o objetivo de confirmar que a utilização de *octrees* era mais benéfica para a criação dos mapas. Para isso foram enviadas 1000 mensagens a uma frequência de 10hz com *pointclouds* de 5000 pontos por forma a criar um degrau. Nas figuras 5.26 e 5.27 é possível observar o resultado dos mapas obtidos com os valores simulados:

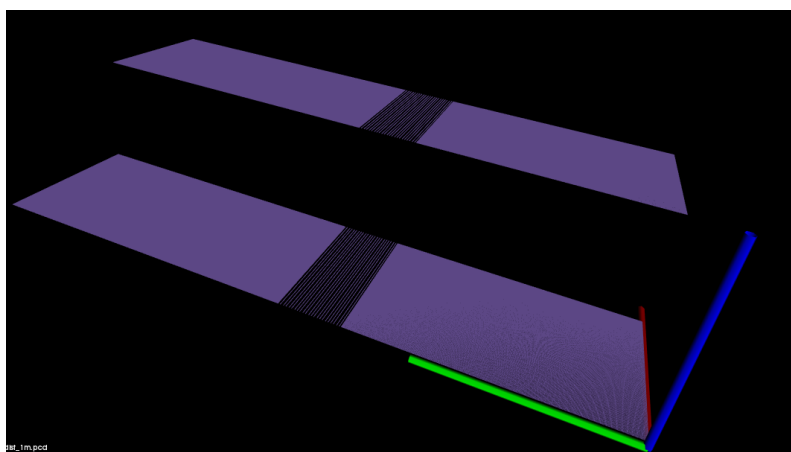


Figura 5.26: Mapa obtido através da concatenação



Figura 5.27: Mapa obtido através da octree

A estratégia de concatenação revelou-se extremamente ineficiente, comparativamente à *octree*. A execução do processo de fusão é extremamente lenta ocorrendo perda de informação, como se observa pelas falhas presentes na figura 5.26, com a agravante de demorar perto de 2 horas para concluir o teste realizado.

Resumidamente, o processo de alocação de memória para acomodar novas *pointclouds*, tornava-se computacionalmente pesado à medida que o tamanho do mapa aumentava. Isto provocava uma acumulação de *pointclouds* no *buffer* do *message handler* das *callbacks*, que era forçado a descartar informação para poder receber dados mais recentes.

Relativamente às *octrees*, processo implementado na versão final do algoritmo, cujo desempenho foi analisado nas secções 5.1.1, 5.1.2 e 5.1.3, não se observam perdas de informação e o tempo de execução foi muito menor, não tendo ultrapassado os 2 minutos.

Obviamente, é essencial ter em consideração a capacidade de processamento do equipamento que executa o programa. No entanto, perante as mesmas condições, a utilização de *octrees* revelou ser mais eficiente e robusta na criação dos mapas.

5.1.5 Conclusão

Ao longo desta secção foram estudadas várias simulações realizadas com o intuito de avaliar o desempenho do algoritmo. Para isso, foram produzidos cenários com diferentes características e observadas as imagens e os ficheiros de *log* resultantes.

Comparando as várias análises, é possível concluir que:

- A quantidade de pontos presentes no mapa e, por conseguinte, a área da superfície a mapear está limitada principalmente pela memória *RAM* do sistema que realiza esta tarefa. O aumento da dimensão do mapa criado pressupõe a alocação de mais espaço para guardar todos os pontos enviados. Como tal, a memória *RAM* vai sendo incrementalmente ocupada e só é libertada após a gravação do ficheiro que contém o mapa (*.pcd*).
- O tempo de cada ciclo de execução do algoritmo é influenciado pela quantidade de pontos presente em cada *pointcloud* e pela resolução escolhida para os mapas a criar. O tamanho definido para o *voxel* mais pequeno da *octree* utilizada, permite realizar reconstruções ao milímetro. Quanto mais precisa for a informação extraída dos objetos, mais próximos estarão os pontos dentro da *pointcloud* enviada ao sistema. Consequentemente, é necessário

realizar mais divisões recursivas de modo a que se possa acomodar todos os pontos na *octree*. Este processo requer maior esforço, por parte do processador, o que contribui para o aumento do tempo de ciclo do algoritmo.

- A utilização de *octrees* torna o processo de reconstrução extremamente eficiente e robusto. O número de mensagens perdidas, é insignificante quando comparado com o número de mensagens publicadas, o que poderá possibilitar o uso deste algoritmo em contexto real.

Tendo tudo isto em consideração, após esta iteração de testes, o algoritmo desenvolvido aparenta ter potencial aplicabilidade em missões reais.

5.2 Testes de validação realizados em laboratório

Após finalizados os testes em ambiente simulado e verificado que existem condições para passar à próxima etapa de validação do sistema, foram iniciados os testes em laboratório. Na primeira parte são apresentados os testes discretizados, sendo posteriormente analisados dois casos de reconstrução em *free run*.

O objetivo fundamental destas experiências é avaliar o comportamento do sistema quando os dados fornecidos provêm de um ambiente real controlado e, ao mesmo tempo, continuar a verificar o desempenho do algoritmo desenvolvido. Para estes testes, foi novamente usado o sistema com as especificações presentes na figura 5.1, tendo o processamento sido realizado em *offline*.

Inicialmente foi feita a calibração do sistema ótico câmara-laser. Este passo é extremamente importante para obter informação precisa das coordenadas das estruturas a mapear. Para isso, utilizou-se a plataforma em acrílico representada na figura 5.28.

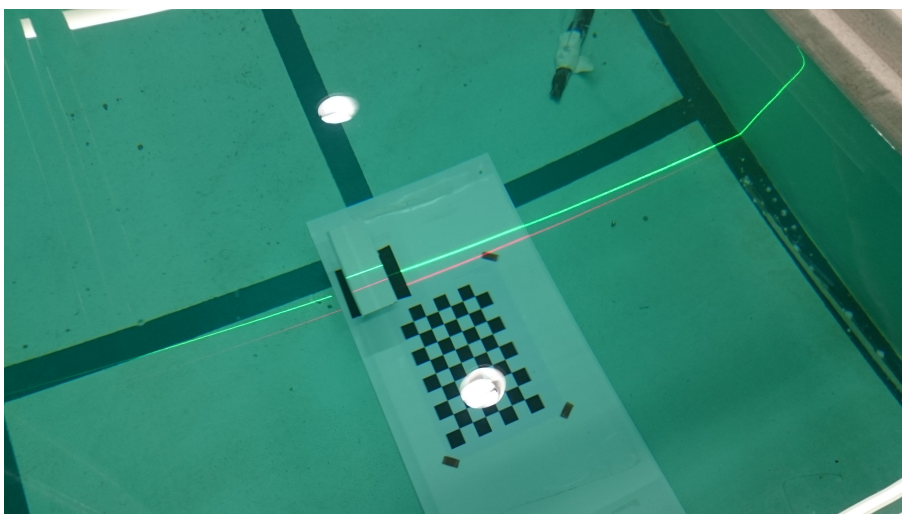


Figura 5.28: Plataforma usada para calibração do sistema câmara-laser

A calibração da câmara passa pela descoberta dos seus parâmetros intrínsecos. Estes são dados por f_x e f_y , que representam as distâncias focais, e por c_x e c_y , que correspondem às coordenadas do centro ótico em *pixels*. Para o cálculo destes parâmetros fizeram-se ensaios dentro de água, nos quais a câmara foi movida por cima do painel axadrezado, de modo a obter *frames* com várias perspetivas do mesmo. Os resultados dessa calibração são dados pela matriz:

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.633e3 & 0 & 6.573e2 \\ 0 & 1.638e3 & 4.701e2 \\ 0 & 0 & 1 \end{bmatrix}$$

A calibração do laser é feita através da sua projeção num objeto retangular, que introduz relevo à plataforma submersa, enquanto a câmara recolhe imagens. A partir destas, é obtida a seguinte equação do plano do laser:

$$1.425x + 9.033y + 8.091z + 9.212 = 0 \quad (5.1)$$

O cálculo dos valores das coordenadas dos pontos tridimensionais (x,y,z) , que pertencem à linha laser, a partir das coordenadas da imagem (x_{px}, y_{px}) , é feito através da resolução do seguinte sistema de equações:

$$\begin{cases} x_{px} = f_x \times \frac{x}{y} \\ y_{px} = f_y \times \frac{z}{y} \\ 1.425x + 9.033y + 8.091z + 9.212 = 0 \end{cases} \quad (5.2)$$

Terminada a calibração passou-se a recolha de dados. Estes foram adquiridos no laboratório e guardados em estruturas próprias (*rosbags*), para que as experiências pudessem ser replicadas o número de vezes que fosse necessário.

5.2.1 Reconstrução Discretizada

A dimensão da plataforma revelou ser impeditiva para a realização de experiências de mapeamento contínuo, pois a sua deslocação provocava demasiada agitação na água e, conseqüentemente, oscilações indesejadas no sistema de aquisição. Introduzindo demasiado ruído nas amostras recolhidas.

Como tal, foram realizadas aquisições intervaladas de 3 objetos com dimensões diferentes para que o algoritmo pudesse ser testado em conjunto com o sistema de aquisição ótico câmara-laser.

Com base nestes testes, foi possível avaliar se o algoritmo era capaz de reconstruir, de forma perceptível e com dimensões corretas, os mapas dos objetos usados. Assim sendo, nesta secção, ao contrário das outras, não é feita a análise do desempenho do algoritmo, mas sim avaliados os mapas no que diz respeito à sua concordância com as dimensões dos objetos medidos no mundo real.

Para que isso fosse possível, o procedimento adotado foi o seguinte:

- A cada 5 centímetros era realizada a obtenção de dados sobre a superfície submersa.
- Para cada aquisição, eram gravadas *rosbags* com 10 segundos de duração.
- Para a criação dos mapas em modo *offline*, a informação proveniente das *pointclouds* era fornecida pelas *rosbags*, enquanto que o valor do deslocamento era incrementado de 0.05 em 0.05 metros no nó de teste previamente criado até não existirem mais *rosbags* para prosseguir com a criação do mapa.
- Por fim, recorrendo ao programa de visualização de *pointclouds* (*pcl_viewer*) eram observados os mapas da reconstrução, de modo a verificar as suas dimensões.

Na figura 5.29 é possível observar o primeiro objeto mapeado (um tubo longo ligeiramente inclinado) e a plataforma usada. Neste teste foram adquiridas 21 *rosbags*.

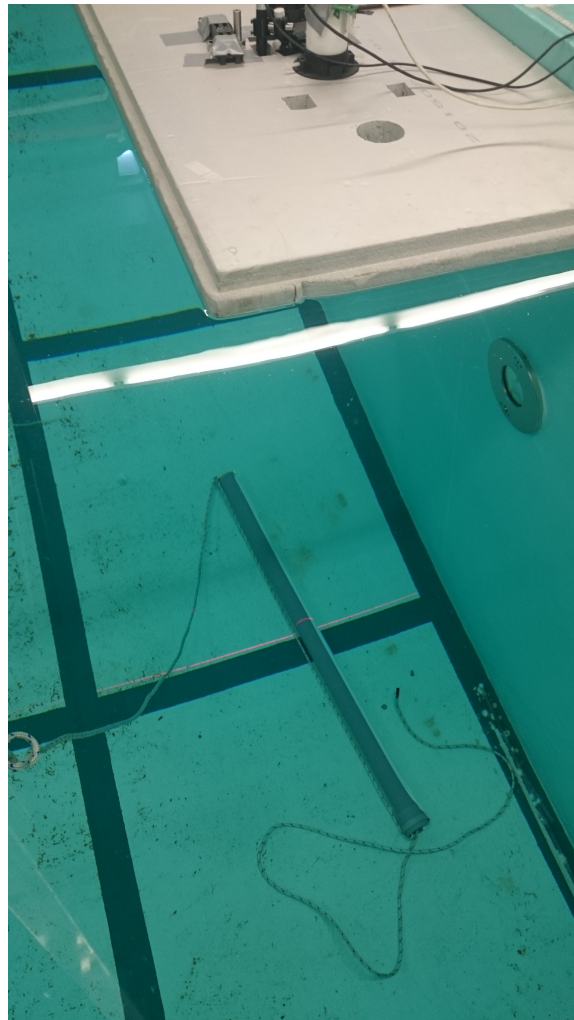


Figura 5.29: Fotografia do *setup* para mapeamento do tubo longo

Na tabela 5.11 é possível consultar as dimensões deste tubo.

Tabela 5.11: Dimensões do tubo longo

<i>Mapa</i>	Dimensões	
	comprimento (m)	diâmetro (m)
Tubo Longo	1	0.05

Através da observação do gradiente fornecido da figura 5.30 e 5.31, é possível observar que o comprimento do tubo na reconstrução se encontra entre os 0.9 metros e 1.0 metros e o seu diâmetro no intervalo entre os 0.05 a 0.07 metros.

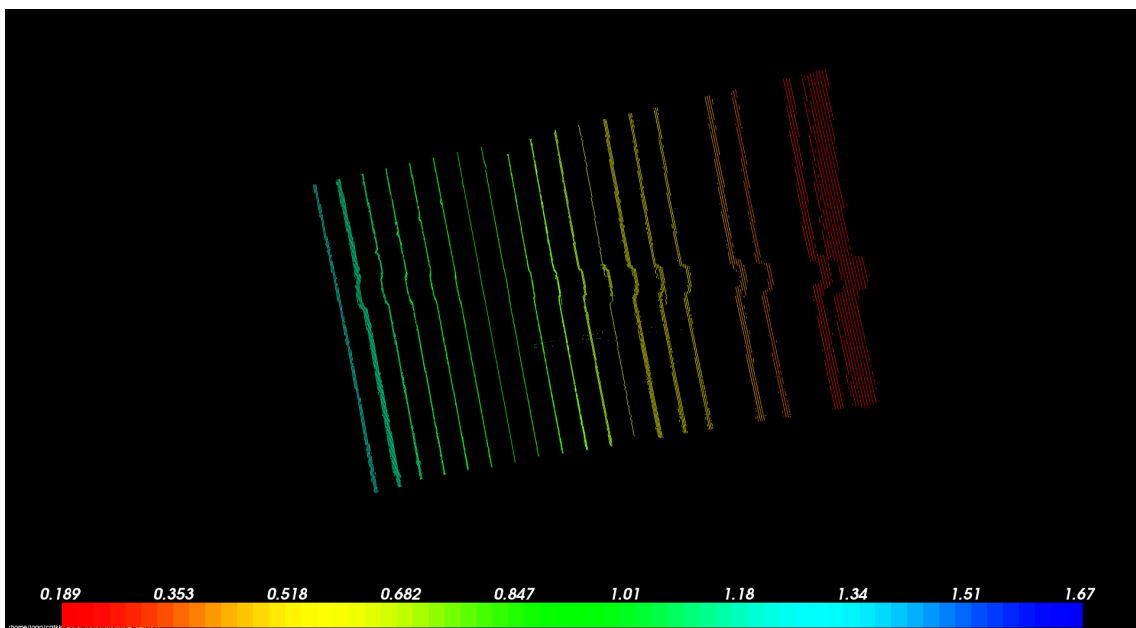


Figura 5.30: *Top view* da reconstrução do tubo longo - com escala de comprimento em metros

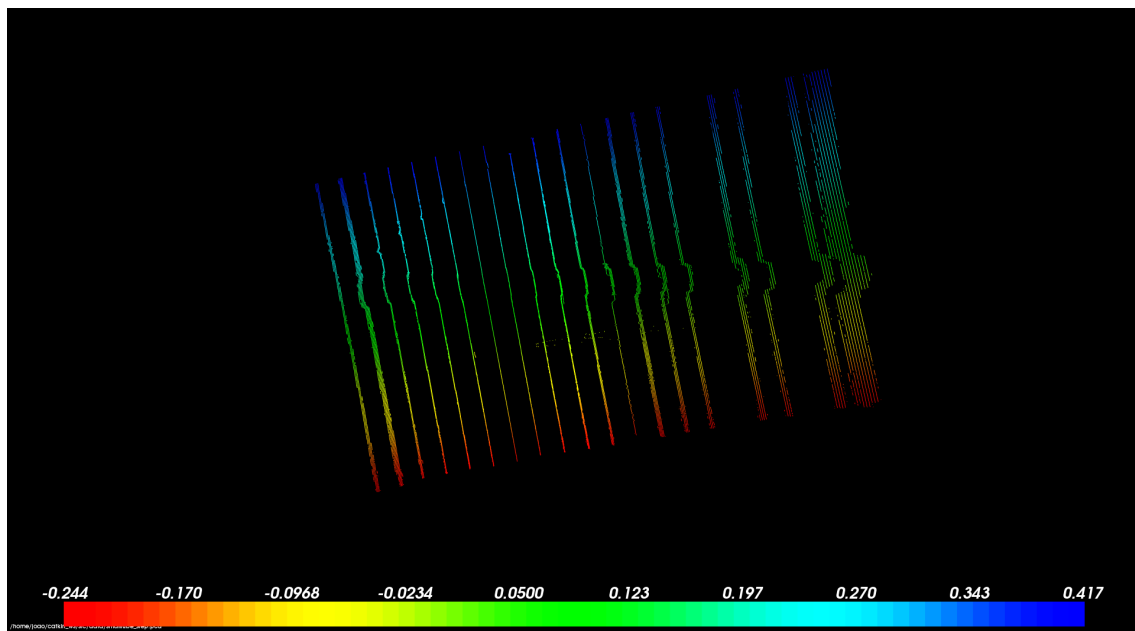


Figura 5.31: *Top view* da reconstrução do tubo longo - com escala de largura em metros

A segunda estrutura a ser mapeada foi um tubo curto com diâmetro superior ao comprimento. Como se pode observar na figura 5.32 e pelos valores da tabela 5.12.

Para a realização deste teste foram gravadas 6 *rosbags*.

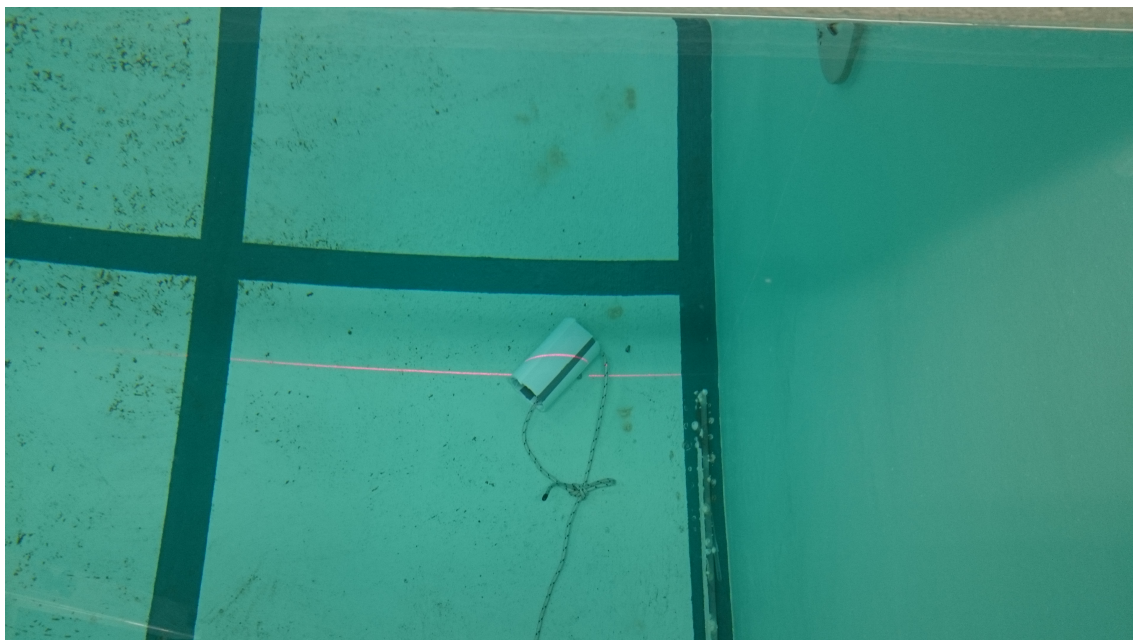


Figura 5.32: Fotografia do *setup* para mapeamento do tubo curto

Tabela 5.12: Dimensões do tubo curto

<i>Mapa</i>	Dimensões	
	comprimento (m)	diâmetro (m)
Tubo Curto	15.5	9.0

Com base na análise das figuras 5.33, 5.34 e 5.35 podemos concluir que:

- O comprimento do tubo se encontra entre os 0.148 e os 0.2 metros.
- O diâmetro do tubo está entre os 0.08 e os 0.12 metros.

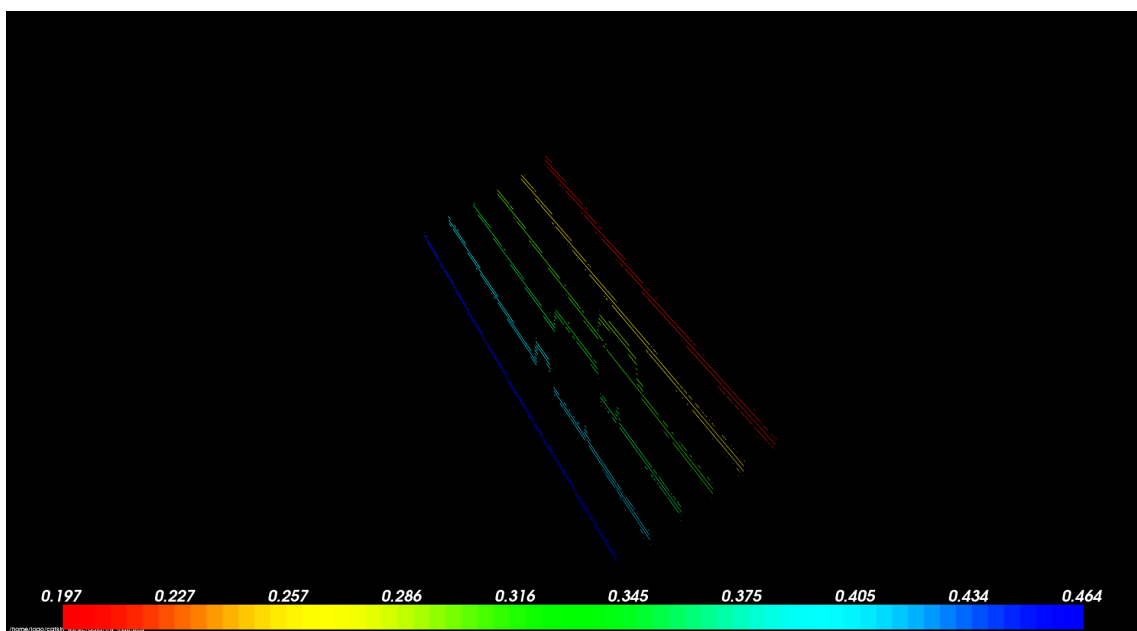


Figura 5.33: *Top view* da reconstrução do tubo curto - com escala de comprimento em metros

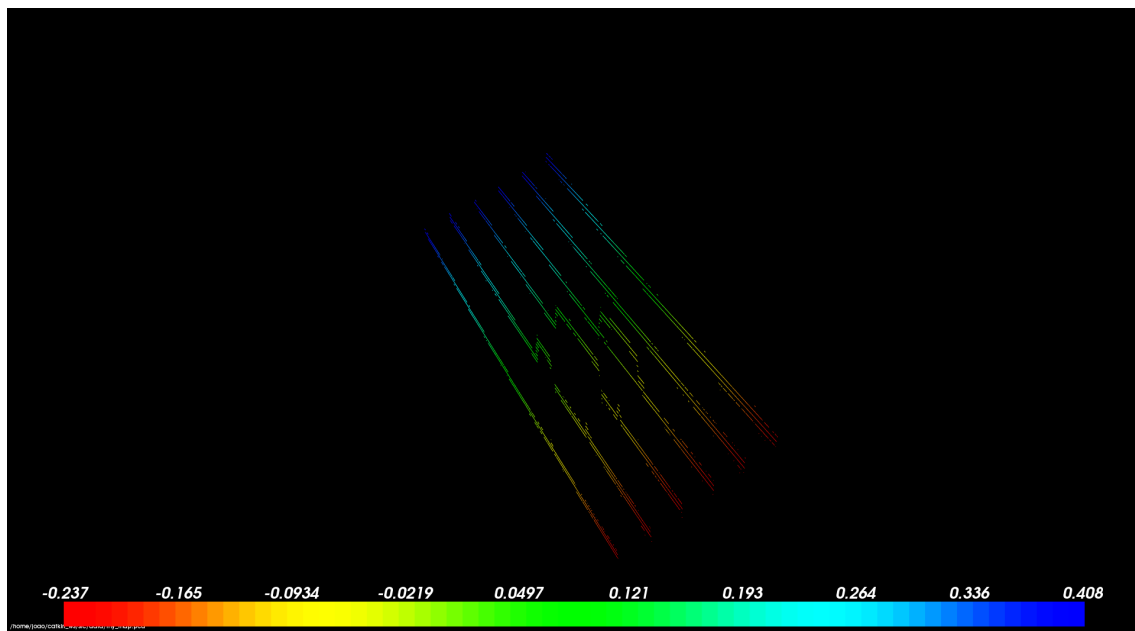


Figura 5.34: *Top view* da reconstrução do tubo curto - com escala de largura em metros

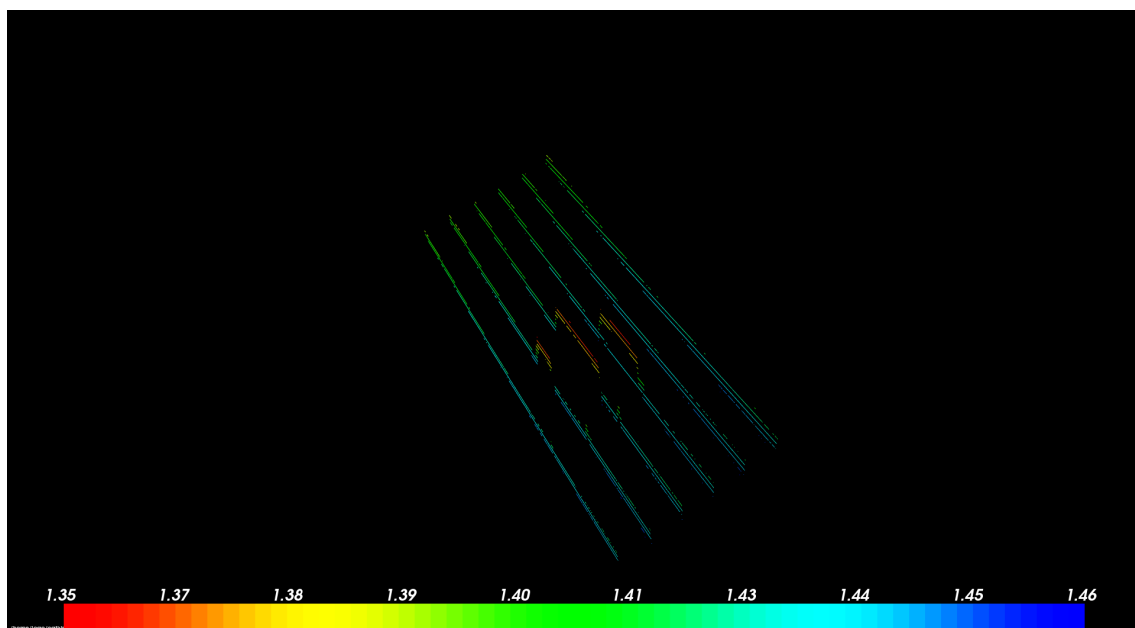


Figura 5.35: *Top view* da reconstrução do tubo curto - com escala de altura em metros

O último objeto a ser mapeado foi a caixa, apresentada na figura 5.36, com as dimensões mencionadas na tabela 5.13. Ao todo, foram adquiridas 17 *rosbags*.

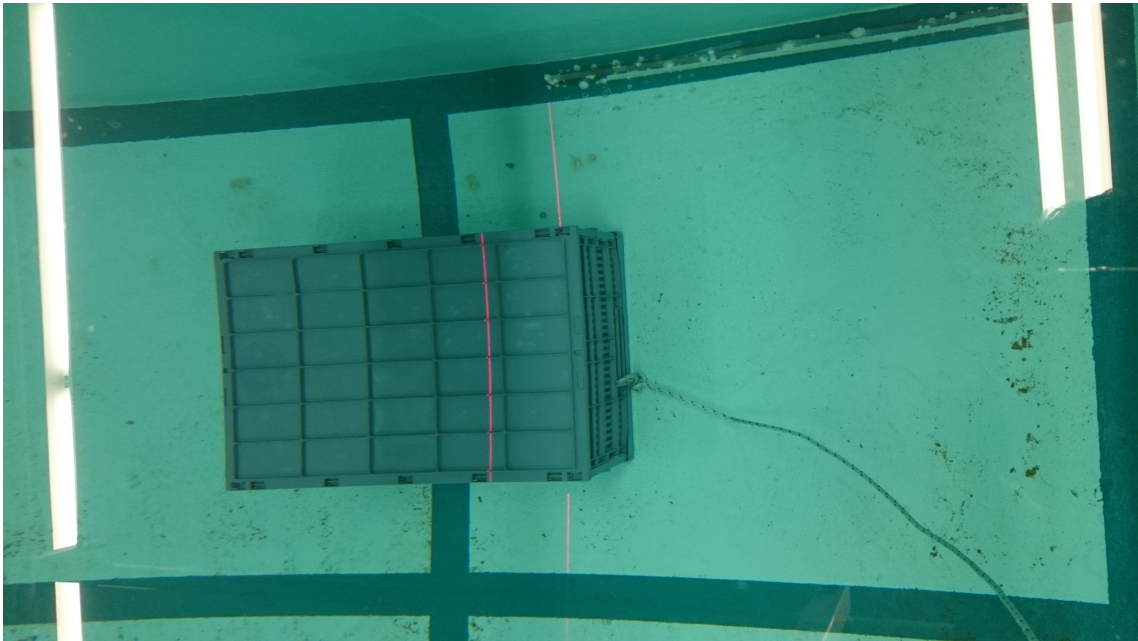
Figura 5.36: Fotografia do *setup* para mapeamento da caixa

Tabela 5.13: Dimensões da caixa

<i>Mapa</i>	Variáveis		
	comprimento (m)	altura (m)	largura (m)
Caixa	0.6	0.27	0.4

Através das imagens [5.37](#), [5.38](#) e [5.39](#) apresentadas pode-se observar que:

- O comprimento da caixa está próximo dos 0.65 metros.
- A largura da caixa é de aproximadamente 0.4 metros
- A altura da caixa está perto dos 0.33 metros.

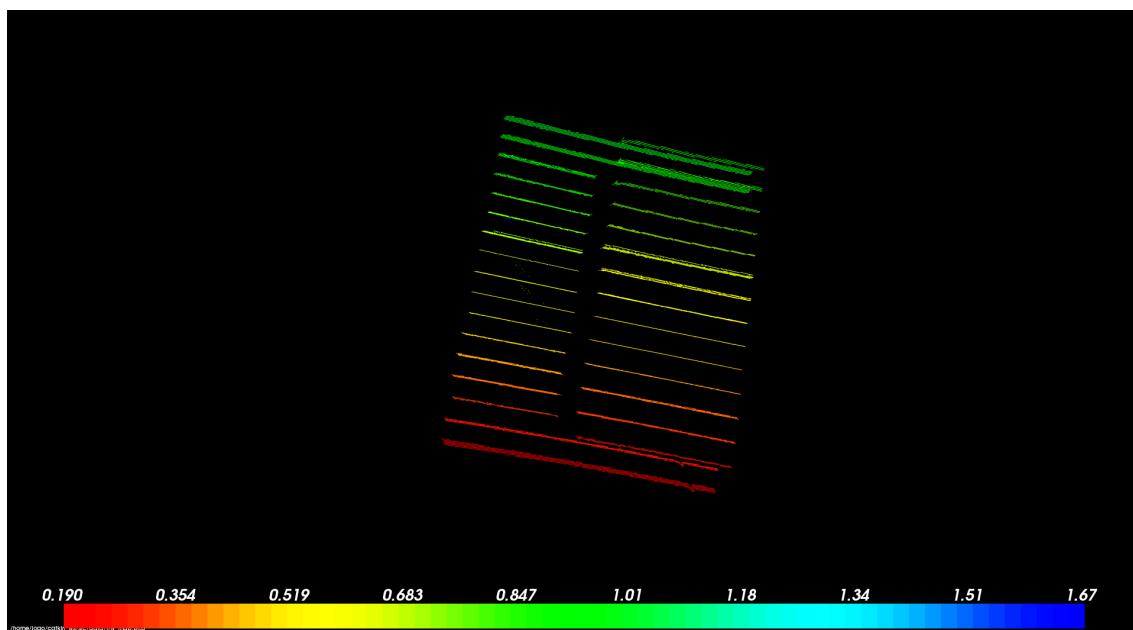


Figura 5.37: *Top view* da reconstrução da caixa - com escala de comprimento em metros

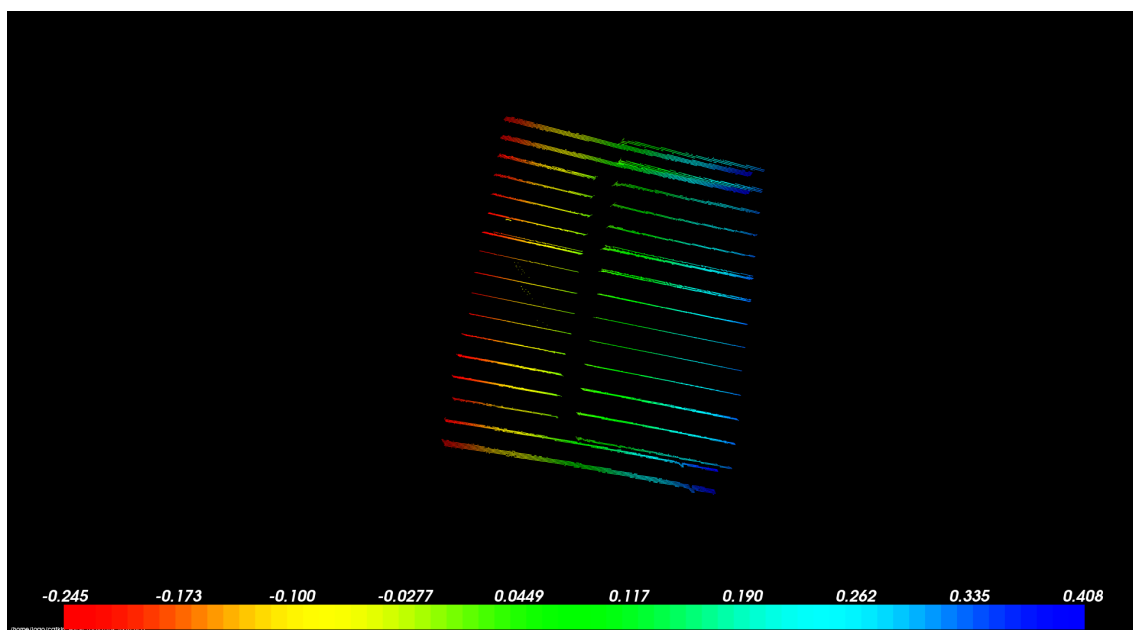


Figura 5.38: *Top view* da reconstrução da caixa - com escala de largura em metros

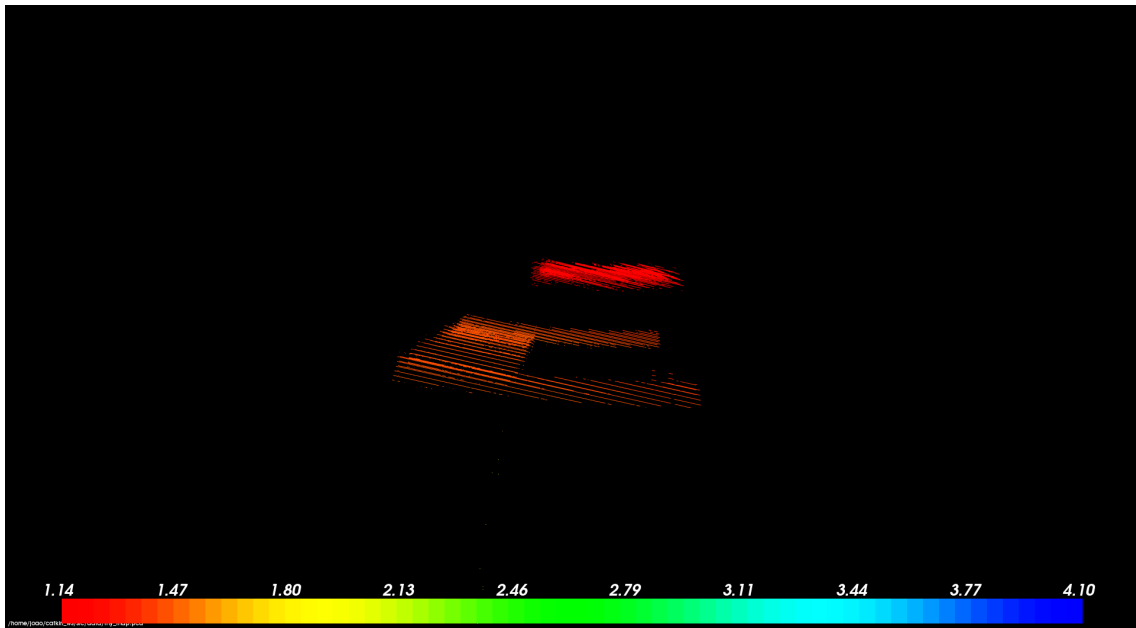


Figura 5.39: *Front view* da reconstrução da caixa - com escala de altura em metros

5.2.1.1 Conclusão

As imagens obtidas através desta aquisição intervalada de *pointclouds*, apresentaram bons resultados. Nestas, é possível perceber, com a ajuda dos gradientes fornecidos para cada imagem, qual o intervalo de valores mais prováveis para as suas dimensões, sendo que o erro dos valores observados ronda os 0.05 metros. Além disso, em todos os casos é fácil identificar qual dos 3 objetos se encontra mapeado em cada figura, pelo que a resolução definida atualmente para o algoritmo parece ser suficiente para este tipo de operações.

5.2.2 Reconstrução em *free run*

Os testes em *free run* assemelham-se ao comportamento que se pretende que o sistema tenha no futuro. Nestes, a plataforma desloca-se por cima das estruturas a mapear, enquanto o sistema câmara-laser realiza a aquisição, extração e segmentação da informação adquirida. Como tal, realizar este tipo de experiências é essencial para que se possa perceber se o sistema poderá vir a ter aplicabilidade em situações reais.

5.2.2.1 Mapeamento de um tubo

Em primeiro lugar, foi realizado o mapeamento de um tubo longo, representado na figura 5.29, cujas dimensões são apresentadas na tabela 5.11. Ao todo foram publicadas 373 mensagens com número de pontos variável (em média 670 pontos por mensagem ($\frac{2.5 \times 10^5}{373}$)).

Como se pode observar pelo gráfico 5.40, este teste teve a duração de aproximadamente 35 segundos, sendo que o mapa final é constituído por cerca de 250000 pontos. Para simular os valores da odometria, recorreu-se ao mesmo nó de teste utilizado na secção 5.1.1 e assumiu-se que a velocidade de deslocamento linear era de aproximadamente $0.04m/s$ ($\frac{1.46(m)}{35(s)}$), sendo que o espaçamento observado entre as *pointclouds* é de $0.004m$ ($\frac{1.46(m)}{373}$).

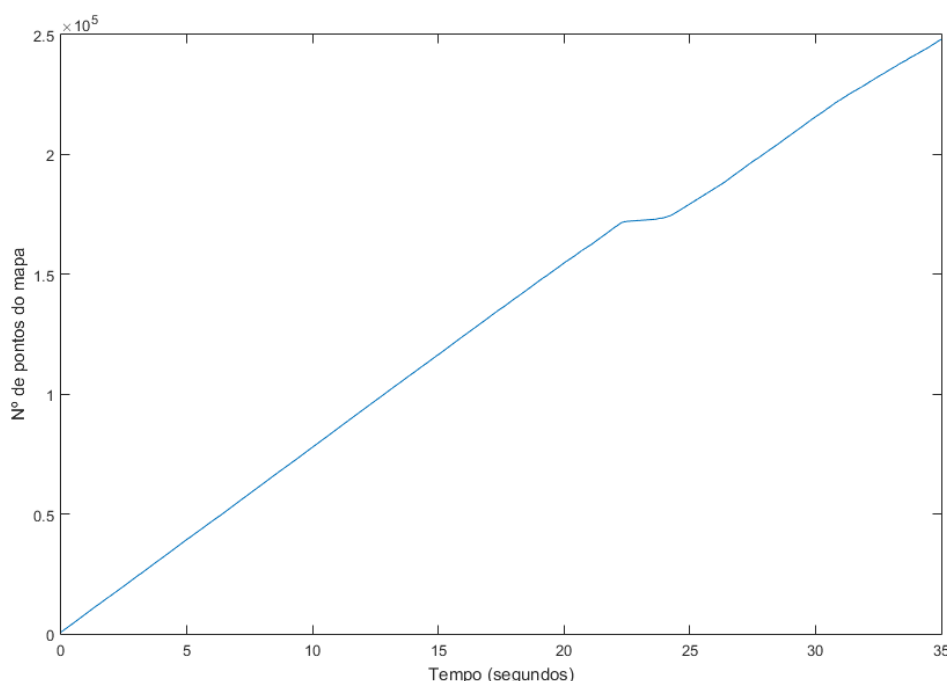


Figura 5.40: Número de pontos mapeados ao longo do tempo - reconstrução do tubo longo

Nas figuras que se seguem, é possível observar o resultado deste mapeamento. A escala presente na imagem 5.41 representa o comprimento do mapa obtido (1.46 metros).

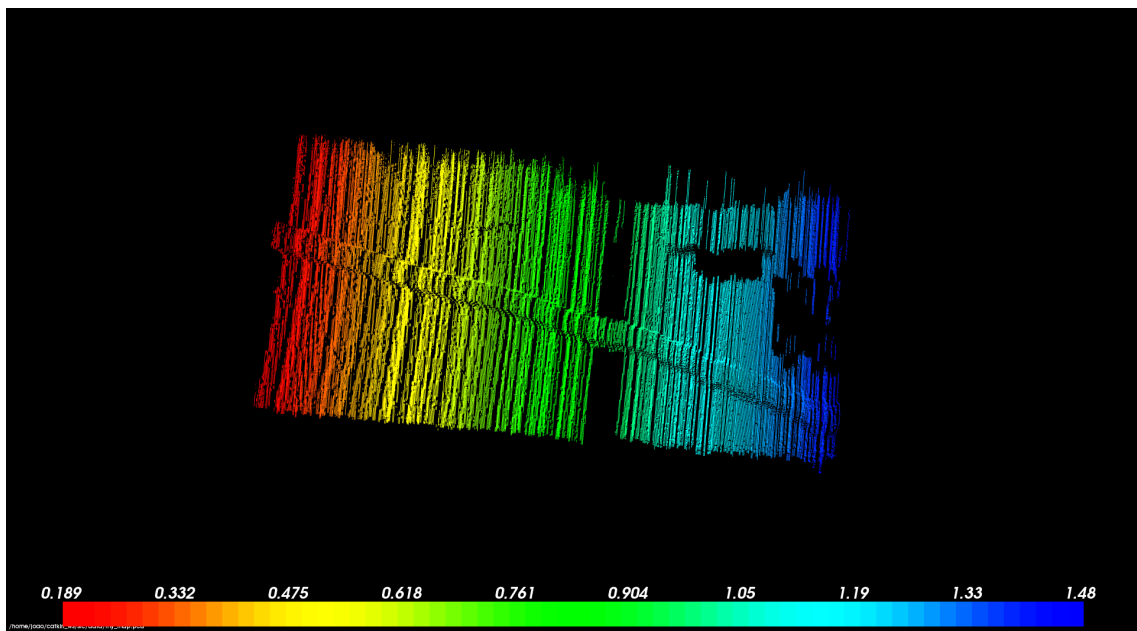


Figura 5.41: *Top view* da reconstrução em *free run* do tubo longo - com escala de comprimento

Através das escalas do gradiente observado nas figuras 5.42 e 5.43 é possível ficar com uma ideia do valor do diâmetro do tubo (entre 0.07 a 0.10 metros)

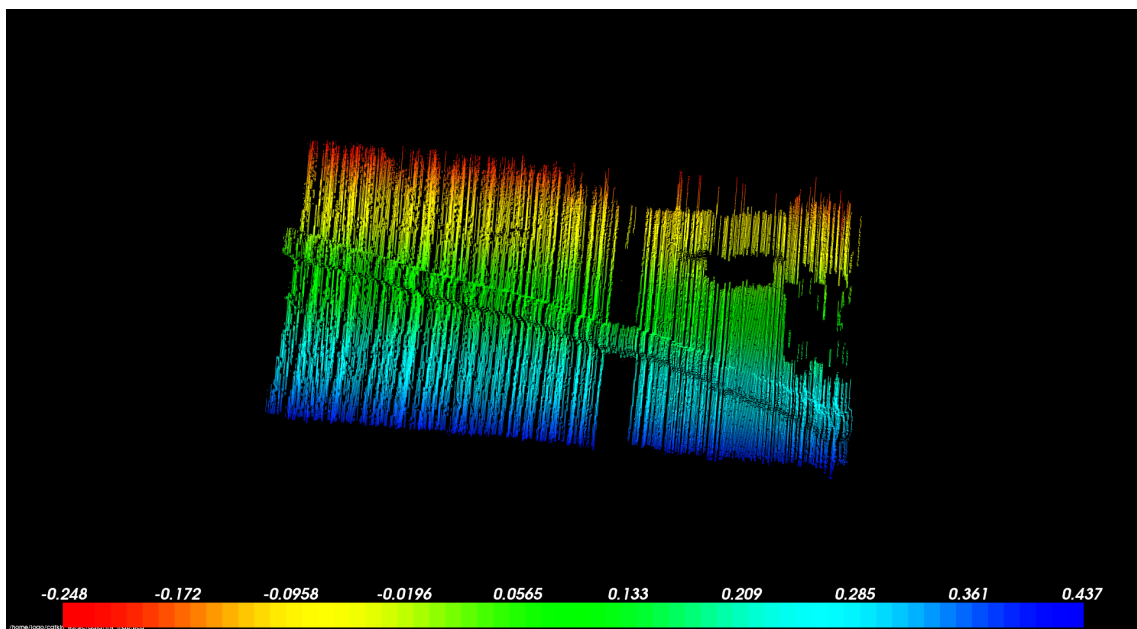


Figura 5.42: *Top view* da reconstrução em *free run* do tubo longo - com escala de largura em metros

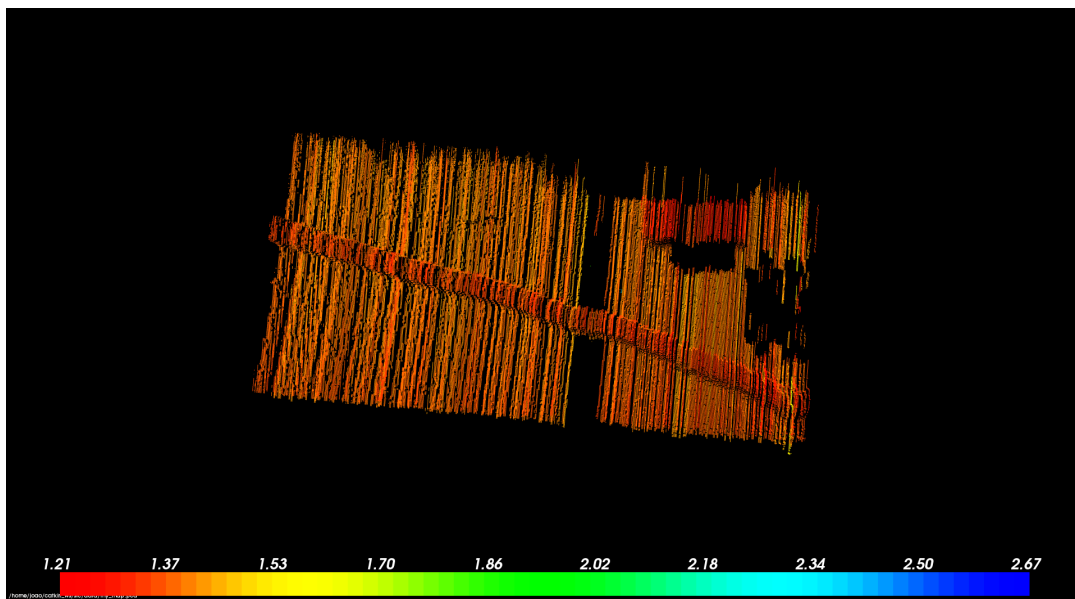


Figura 5.43: *Top view* da reconstrução em *free run* do tubo longo - com escala de altura em metros

À semelhança do que foi feito na secção dos testes simulados (5.1), com recurso à análise dos ficheiros de *log*, foi criado um gráfico (5.44) e uma tabela (5.14). A partir sua observação é possível perceber o desempenho do algoritmo desenvolvido, quando aplicado neste ambiente de teste controlado.

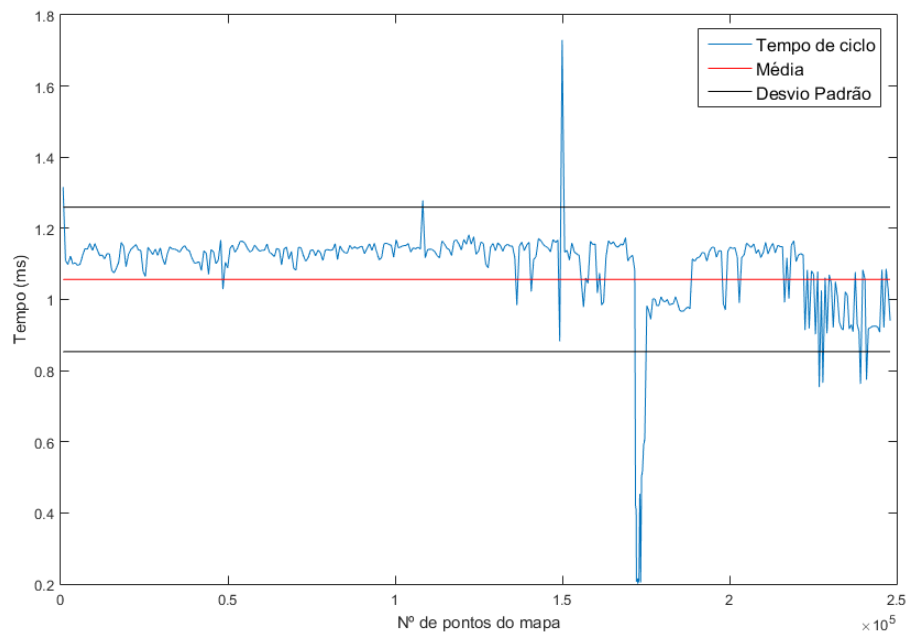


Figura 5.44: Gráfico do desempenho da reconstrução do tubo longo - tempo de ciclo vs número de pontos

Tabela 5.14: Resultados do mapeamento do tubo longo

<i>Mapa</i>	Resultados			
	média (ms)	σ (ms)	<i>outliers</i>	nº mensagens recebidas
Tubo	1.0563	0.2031	28	373

Em termos médios, o algoritmo foi relativamente mais rápido a executar as suas operações do que durante os testes simulados. Isto deve-se ao facto do número de pontos de cada *pointcloud* ser inferior.

Em relação à qualidade da reconstrução observada, esta não aparenta ser afetada por nenhum parâmetro relacionado com o algoritmo. Os erros da reconstrução e medição observados nos mapas devem-se, essencialmente, ao deslocamento da massa de água provocada pela plataforma de navegação, que provoca a oscilação da linha laser, e ao facto dos valores associados ao deslocamento do sistema (e.g velocidade e distância) serem determinados de forma empírica e não rigorosamente medidos.

5.2.2.2 Mapeamento de uma caixa

Em segundo lugar, foi mapeada parte de uma caixa, cujas dimensões podem ser observadas na tabela 5.13. Ao todo foram publicadas 127 mensagens com número de pontos variável (em média 787 pontos por mensagem ($\frac{10 \times 10^4}{127}$)).

Como é possível verificar através do gráfico 5.45, este teste durou cerca de 14 segundos, sendo que o mapa final é constituído por aproximadamente 100000 pontos. Para simular o deslocamento recorreu-se novamente ao nó de teste e considerou-se que a velocidade de deslocamento era de aproximadamente $0.095m/s$ ($\frac{1.33(m)}{14(s)}$), sendo que o espaçamento observado entre as *pointclouds* é de $0.01m(\frac{1.33(m)}{127})$.

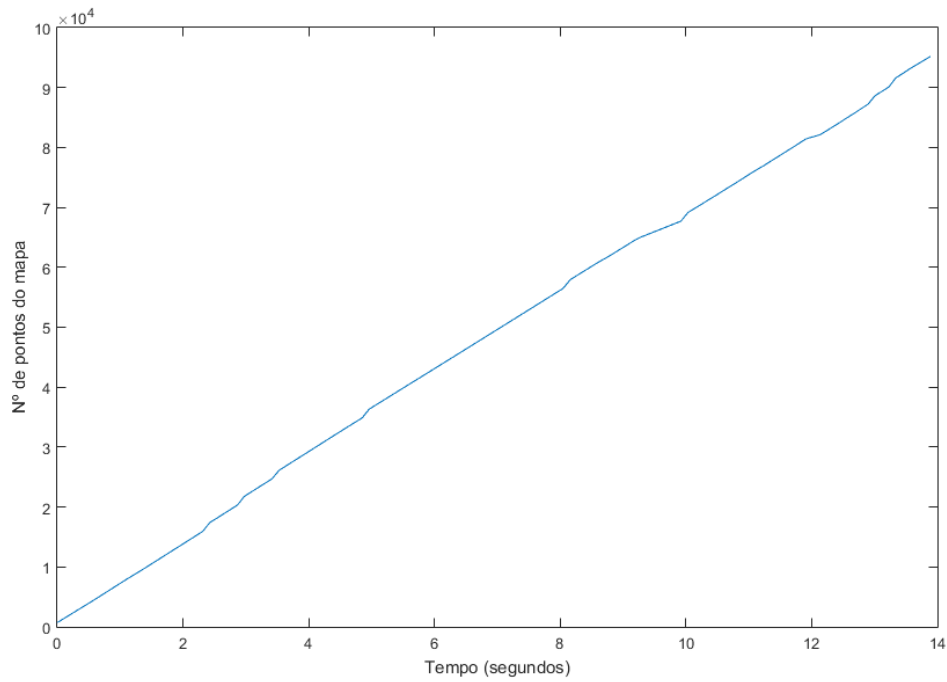


Figura 5.45: Número de pontos mapeados ao longo do tempo - reconstrução da caixa

A partir das figuras resultantes, podem observar-se as dimensões, ainda que aproximadas, do mapa e da caixa nele contido. Observando a escala da imagem 5.46 conclui-se que o comprimento do mapa obtido é 1.33 metros e que o comprimento da caixa é próximo de 0.55 metros.

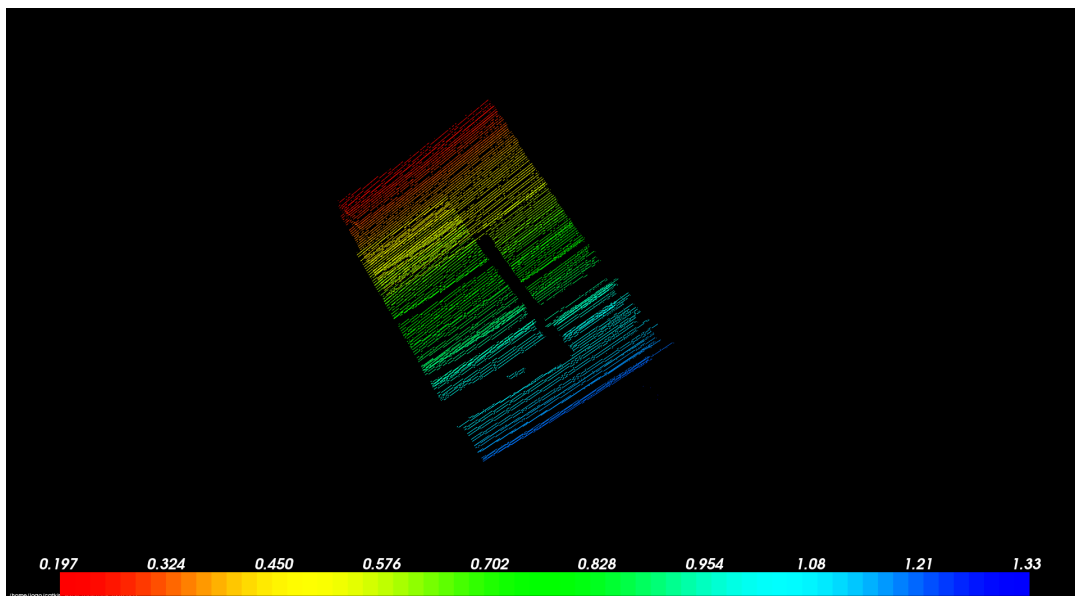


Figura 5.46: Top view da reconstrução em free run da caixa - com escala de comprimento em metros

Na imagem 5.47 é possível observar a largura do mapa obtido (aproximadamente 0.7 metros) e da caixa mapeada (aproximadamente 0.4 metros).

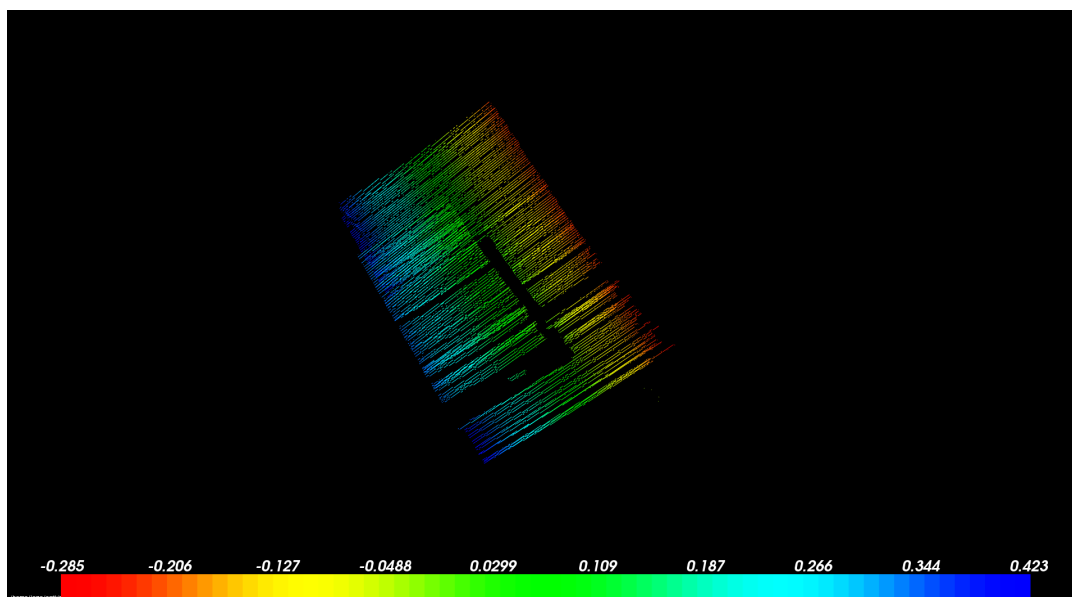


Figura 5.47: *Top view* da reconstrução em *free run* da caixa - com escala de largura em metros

Por fim, com base no gradiente observado nas figuras 5.48 e 5.49 é possível ficar com uma ideia do valor da altura da caixa (entre 0.20 a 0.30 metros).

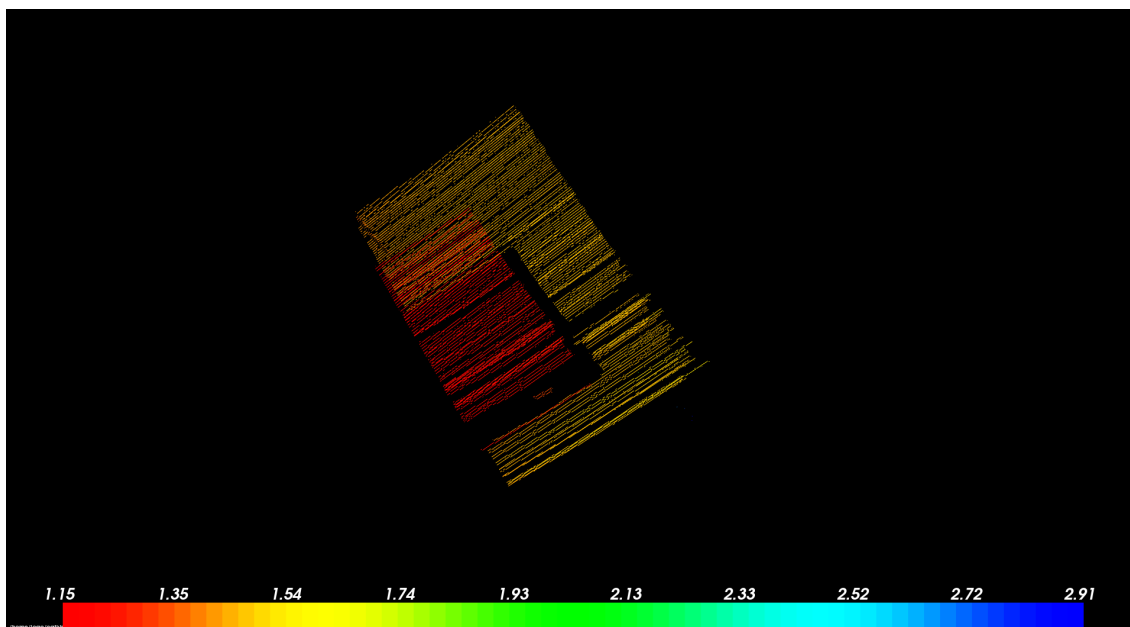


Figura 5.48: *Top view* da reconstrução em *free run* da caixa - com escala de altura em metros

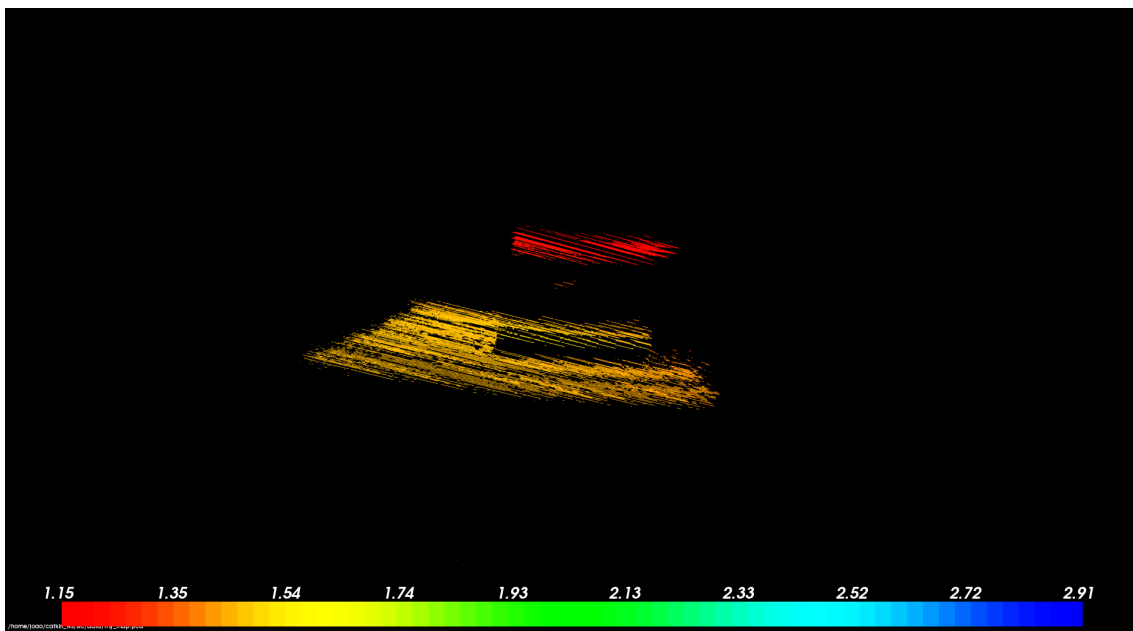


Figura 5.49: Side view da reconstrução em *free run* da caixa - com escala de altura em metros

Com o auxílio do gráfico 5.50 e da tabela 5.15 é feita a última avaliação da execução do algoritmo desenvolvido nesta dissertação.

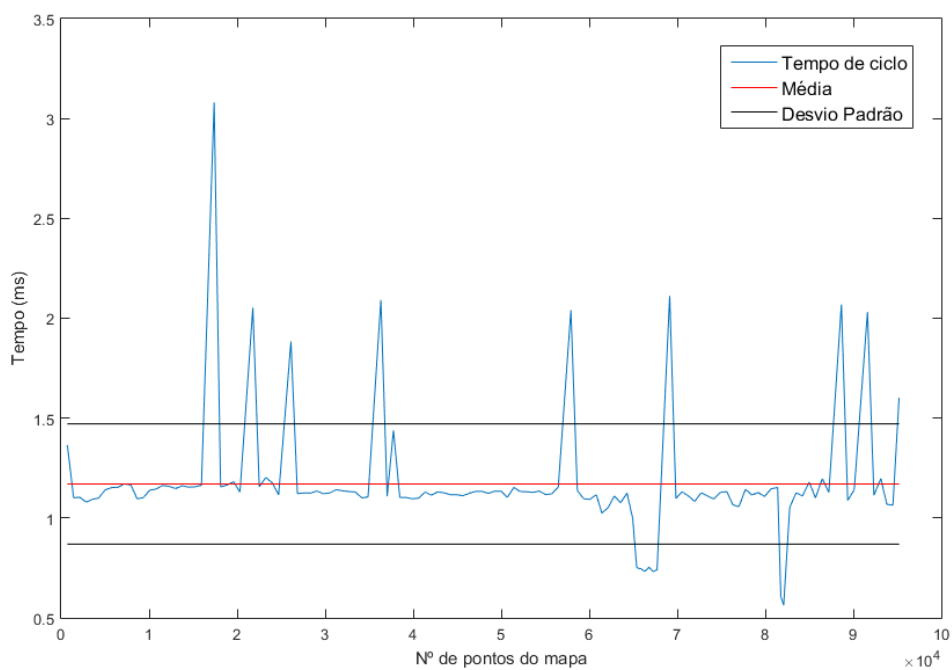


Figura 5.50: Gráfico do desempenho da reconstrução da caixa - tempo de ciclo vs número de pontos

Tabela 5.15: Resultados do mapeamento da caixa

Resultados				
<i>Mapa</i>	média (ms)	σ (ms)	<i>outliers</i>	n ^o mensagens recebidas
Caixa	1.1705	0.3010	17	127

Mais uma vez, neste ambiente controlado, o algoritmo demonstrou bons resultados. O tempo de ciclo manteve-se dentro do esperado para o volume de pontos de cada *pointcloud*. Todos os valores de pico observados não revelaram ter um impacto significativo no mapa final, tendo o mapeamento da caixa corrido dentro da normalidade.

5.3 Conclusão

Ao longo deste capítulo foram apresentados vários testes que tiveram como objetivo avaliar a performance do algoritmo desenvolvido e a sua capacidade de representar de forma precisa as estruturas submersas. De um modo geral os resultados revelaram ser positivos, apesar de ainda existirem otimizações que podem ser implementadas (e.g filtração de pontos dispersos).

Com base em todos os testes efetuados, a observação mais importante a fazer é, que o *hardware*, quer seja ele responsável pela aquisição ou pelo processamento de informação, desempenha um papel fundamental na qualidade dos mapas criados. A melhoria da tecnologia usada na aquisição pode proporcionar a extração de mais detalhes do meio, com menos ruído, enquanto que a utilização de unidades de processamento mais capazes permite que a operação de mapeamento se prolongue durante mais tempo, proporcionando maior autonomia aos veículos aquáticos autónomos que as realizam.

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Conclusão

Atualmente os métodos baseados na propagação do som apresentam limites que restringem o seu uso para algumas aplicações no meio aquático, principalmente na aquisição de informação a curta distância do objeto alvo.

Como tal, de forma a explorar novas soluções que permitam a extração de informação deste meio, nesta dissertação, é proposta uma solução de aquisição e processamento de informação visual, baseada num sistema ótico câmara-laser, que permite a criação de mapas de fundos aquáticos ou estruturas submersas com elevada definição.

O início desta tese passou pela compreensão da situação atual da tecnologia usada nas operações de mapeamento. Por conseguinte, em primeiro lugar, foi feito o estudo do estado da arte de algumas técnicas e sistemas, que são atualmente utilizados na extração de imagem, ótica ou acústica, para a reconstrução de superfícies e auxílio de navegação. A partir desta, é possível deprender que, hoje em dia, existe sem dúvida lugar para a utilização de sistemas como o que é proposto nesta dissertação.

De seguida, foi feita e apresentada a análise da arquitetura do sistema. Nesta, foi analisada a disposição física dos equipamentos de aquisição (câmara-laser), com o objetivo de compreender e propor possíveis parâmetros de instalação (e.g ângulo de incidência do laser na superfície) ideais a uma aquisição precisa de dados em *close range navigation* (0.5 a 2 metros). As restrições consideradas durante esta avaliação correspondem ao limites físicos impostos à instalação deste num veículo aquático autónomo, já disponível na Faculdade de Engenharia da Universidade do Porto. Com base nos resultados obtidos, concluiu-se que, os parâmetros devem ser selecionados tendo em atenção os objetivos da missão de mapeamento. No entanto, é proposto o uso de um ângulo, entre o referencial da câmara e o referencial do laser (α), compreendido entre os 0.55 e os 0.9 radianos e a utilização do comprimento máximo disponível para o afastamento entre a câmara e o laser, neste caso 0.25 metros.

Finalmente, foi concebido um algoritmo baseado em *octrees* que realiza a reconstrução dos

mapas recorrendo à informação extraída pelo sistema de aquisição ótico (*pointclouds*) e outra informação sensorial, neste caso a odometria. Este, foi criado com recurso à *framework* ROS, por forma a facilitar a sua integração com os outros nós já existentes. Para capacitar o programa desenvolvido de funções de manipulação e processamento de *pointclouds*, recorreu-se a uma biblioteca *open source*, a *Point Cloud Library* (PCL). O comportamento do algoritmo foi avaliado através de testes com recurso a valores simulados e valores adquiridos em laboratório.

Os resultados obtidos, revelam que existe potencial aplicabilidade deste sistema em contexto real, sendo que o algoritmo implementado foi capaz de realizar a reconstrução dos mapas quer em ambiente simulado, quer em ambiente laboratorial com boa resolução .

Em suma, os objetivos para esta dissertação foram cumpridos. Tendo sido realizadas, de forma sequencial, todas etapas necessárias à análise, conceção e implementação de um sistema de aquisição ótico com recurso a triangulação laser para mapeamento de estruturas submersas com alta definição.

6.2 Trabalho Futuro

O trabalho desenvolvido nesta tese cumpriu os objetivos inicialmente propostos. Apesar disso, ainda existem melhorias que podem ser feitas de modo a possibilitar a utilização do sistema em contexto real e para que os mapas criados tenham melhor qualidade. Tendo isto em consideração, o trabalho futuro deverá passar por:

- Desenvolver um nó que, com base em informação proveniente de sensores instalados no AUV (*encoders*, giroscópios, acelerómetros, etc), seja capaz de fornecer informação sobre localização relativa (posição e orientação) para que o algoritmo desenvolvido nesta dissertação tenha acesso à informação necessária para realizar as transformações geométricas sobre a *pointcloud*.
- Implementar técnicas de filtragem no algoritmo. As reconstruções realizadas com recurso à informação obtida nos testes de laboratório apresentam níveis de ruído introduzido pelo sistema de aquisição. Por conseguinte, a implementação de métodos de filtragem de *outliers* e de ruído é a próxima etapa a ser explorada para o aperfeiçoamento do algoritmo implementado.
- Implementar métodos de suavização de superfícies de modo a melhorar aspeto dos mapas criados.
- Realizar testes em ambiente real, para terminar o processo de validação do sistema desenvolvido.

Referências

- [1] F. Bruno, G. Bianco, M. Muzzupappa, S. Barone, e A.V. Razionale. Experimentation of structured light and stereo vision for underwater 3D reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(4):508–518, 2011. doi:10.1016/j.isprsjprs.2011.02.009.
- [2] Underwater Image Processing: State of the Art of Restoration and Image Enhancement Methods. *EURASIP Journal on Advances in Signal Processing*, 2010:1–15, 2010. doi:10.1155/2010/746052.
- [3] Alberto Elfes. Sonar-Based Real-World Mapping and Navigation. *IEEE Journal on Robotics and Automation*, 3(3):249–265, 1987. doi:10.1109/JRA.1987.1087096.
- [4] Kalyan K. Halder, Murat Tahtali, e Sreenatha G. Anavatti. Turbulence mitigation and moving object detection for underwater imaging. *SPIE 9622, 2015 International Conference on Optical Instruments and Technology: Optoelectronic Imaging and Processing Technology*, 9622:96220C, 2015. doi:10.1117/12.2191108.
- [5] S. Goode W. Jarosz E. Hou, W. Woods e A. Weidemann. Impacts of optical turbulence on underwater imaging. *Proc. SPIE*, 8030:803009, 2011. doi:10.1117/12.883114.
- [6] Advances in underwater mapping and their application potential for Safety, Security, and Rescue Robotics (SSRR). *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2012*, 00(c):12–14, 2012. doi:10.1109/SSRR.2012.6523907.
- [7] Andrea Petroni, Stefano Pergoloni, Ieee Student, Mauro Biagi, Ieee Senior, Stefania Colonese, Ieee Member, Roberto Cusani, e Gaetano Scarano. Acoustic Bathymetric Mapping in very Shallow Water Reservoir. páginas 0–3, 2015.
- [8] Chris Wilcox e Marc Los Huertos. A simple , rapid method for mapping bathymetry of small wetland basins. 301:29–36, 2005. doi:10.1016/j.jhydrol.2004.06.027.
- [9] Marcus Hammond e Stephen M Rock. A SLAM-based Approach for Underwater Mapping using AUVs with Poor Inertial Information. *2014 IEEE/OES Autonomous Underwater Vehicles (AUV) Oxford, MS*, páginas 1–8, 2014. doi:10.1109/AUV.2014.7054419.
- [10] Halleh Mortazavi, John P. Oakley, e Braham Barkat. Mitigating the effect of optical back-scatter in multispectral underwater imaging. *IST 2012 - 2012 IEEE International Conference on Imaging Systems and Techniques, Proceedings*, (2):633–636, 2012. doi:10.1109/IST.2012.6295591.
- [11] Stein M. Nornes, Martin Ludvigsen, Øyvind Ødegard, e Asgeir J. Sørensen. Underwater Photogrammetric Mapping of an Intact Standing Steel Wreck with ROV. *IFAC-PapersOnLine*, 48(2):206–211, 2015. doi:10.1016/j.ifacol.2015.06.034.

- [12] L. Bjorno. Developments in sonar technologies and their applications. 2013 *IEEE International Underwater Technology Symposium, UT 2013*, páginas 2–9, 2013. doi:10.1109/UT.2013.6519839.
- [13] Hyeonwoo Cho, Jeonghwe Gu, Hangil Joe, Akira Asada, e Son Cheol Yu. Acoustic beam profile-based rapid underwater object detection for an imaging sonar. *Journal of Marine Science and Technology (Japan)*, 20(1):180–197, 2015. doi:10.1007/s00773-014-0294-x.
- [14] G. Montereale Gavazzi, F. Madricardo, L. Janowski, A. Kruss, P. Blondel, M. Sigovini, e F. Foglini. Evaluation of seabed mapping methods for fine-scale classification of extremely shallow benthic habitats – Application to the Venice Lagoon, Italy. *Estuarine, Coastal and Shelf Science*, 170:45–60, 2015. doi:10.1016/j.ecss.2015.12.014.
- [15] Edward Chen e Jenhwa Guo. Real time map generation using sidescan sonar scan-lines for unmanned underwater vehicles. *Ocean Engineering*, 91:252–262, 2014. doi:10.1016/j.oceaneng.2014.09.017.
- [16] Mohammadreza Bayat e A Pedro Aguiar. Underwater localization and mapping: observability analysis and experimental results. *Industrial Robot-an International Journal*, 41(2):213–224, 2014. doi:10.1108/ir-09-2013-398.
- [17] G. Suresh, Shanavas A. Natarajan, V., e Naval Physical & Oceanographic Laboratory Thrikkakara Kochi India. Underwater imaging using acoustic lens. 2015 *IEEE Underwater Technology, UT 2015*, páginas 1–3, 2015. doi:10.1109/UT.2015.7108216.
- [18] J. Derek Tucker e Mahmood R. Azimi-Sadjadi. Coherence-based underwater target detection from multiple disparate sonar platforms. *IEEE Journal of Oceanic Engineering*, 36(1):38–52, 2011. doi:10.1109/JOE.2010.2094230.
- [19] Sir David Brewster. Chapter 2 - Principles of Stereoscopic Depth Perception and Reproduction. *Reproduction*, páginas 51–82, 2007.
- [20] L. Repola, R. Memmolo, e D. Signoretti. Instruments and Methodologies for the Underwater Tridimensional Digitization and Data Musealization. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-5/W5(April):183–190, 2015. doi:10.5194/isprsarchives-XL-5-W5-183-2015.
- [21] Ricard Prados, Rafael Garcia, Javier Escartin, e Lazlo Neumann. Challenges of close-range underwater optical mapping. *OCEANS 2011 IEEE - Spain*, 2011. doi:10.1109/Oceans-Spain.2011.6003501.
- [22] Stefan May e Kai Pervoelz. 3D cameras: 3D computer vision of wide scope. *Vision Systems: Applications*, (June):181–202, 2007.
- [23] Trinocular stereo for non-parallel configurations. *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, 1:0–3, 2000. doi:10.1109/ICPR.2000.905401.
- [24] Bing Zheng, Hao Zhang, Haiyong Zheng, e T. Aaron Gulliver. Underwater imaging based on inhomogeneous illumination. *IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing - Proceedings*, páginas 873–876, 2011. doi:10.1109/PACRIM.2011.6033010.
- [25] Yanan Wen, Weijian Tian, Bing Zheng, Guozun Zhou, Hui Dong, e Qiong Wu. New approach for underwater imaging and processing. 9100:91000S, 2014. doi:10.1117/12.2050139.

- [26] Yuzhang Chen, Kecheng Yang, Xiaohui Zhang, Min Xia, e Wei Li. Modelling of beam propagation and its applications for underwater imaging. *Frontiers of Optoelectronics in China*, 4(4):398–406, 2011. doi:10.1007/s12200-011-0219-9.
- [27] Huachuan Huang, Rongbo Wang, Keding Yan, Zhengang Yan, Shouyu Wang, Zhenhua Li, e Zeren Li. High quality underwater imaging platform with laser range gated technique combining with image denoising and restoration. 2014. doi:10.1117/12.2071247.
- [28] Jie-fei Han, Tao Luo, Li-ying Sun, Chi-zhu Ding, Min Xia, e Ke-cheng Yang. Research of application of high-repetition-rate green laser in underwater imaging system. 2013. doi:10.1117/12.2034688.
- [29] Donna M. Kocak, Fraser R. Dagleish, Frank M. Caimi, e Yoav Y. Schechner. A Focus on Recent Developments and Trends in Underwater Imaging. *Marine Technology Society Journal*, 42(1):52–67, 2008. doi:10.4031/002533208786861209.
- [30] Ooceantools. <http://www.oceantools.co.uk/>. Acedido a 20 de Junho de 2016.
- [31] Bryan Poling. A Tutorial On Camera Models. páginas 1–10, University of Minnesota.
- [32] Ros. <http://www.ros.org/>. Acedido a 20 de Junho de 2016.
- [33] Radu Bogdan Rusu e Steve Cousins. 3D is here: Point Cloud Library (PCL). Em *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
- [34] J. Barbosa e Sousa. A. Transformações geométricas 3d. *Computação Gráfica*, 2003.
- [35] Pcl - point cloud library. <http://pointclouds.org/>. Acedido a 20 de Junho de 2016.
- [36] Wikipedia. <https://www.wikipedia.org/>. Acedido a 20 de Junho de 2016, utilizado para obter imagens.
- [37] Mathworks. <https://www.mathworks.com/>. Acedido a 20 de Junho de 2016, utilizado para obter imagens.
- [38] Intel. <http://ark.intel.com/>. Acedido a 20 de Junho de 2016.