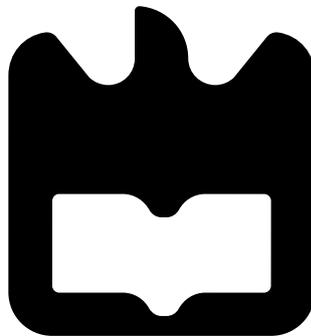




Miguel
da Mota Rodrigues

Mapas de navegabilidade para condução
autónoma no Campus UA





**Miguel
da Mota Rodrigues**

**Mapas de navegabilidade para condução
autónoma no Campus UA**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestrado em Engenharia Mecânica, realizada sob a orientação científica de Vitor Manuel Ferreira dos Santos, Professor Associado do Departamento de Engenharia Mecânica da Universidade de Aveiro.

o júri / the jury

presidente / president

Professor Doutor Jorge Augusto Fernandes Ferreira

Professor Auxiliar da Universidade de Aveiro - Departamento de Engenharia
Mecânica

vogais / examiners committee

Professor Doutor Vitor Manuel Ferreira dos Santos

Professor Associado da Universidade de Aveiro - Departamento de Engenharia
Mecânica (orientador)

Professor Doutor Carlos Baptista Carneira

Professor Auxiliar da Universidade de Lisboa - Instituto Superior Técnico

**agradecimentos /
acknowledgements**

Gostava de deixar aqui os meus mais sinceros agradecimentos ao Professor Doutor Vitor Santos e ao Doutor Ricardo Pascoal, pela orientação, o entusiasmo e motivação que sempre me transmitiram ao longo desta dissertação. Um especial agradecimento a Jorge Almeida e a João Torrão pelo apoio e troca de opiniões essenciais ao desenvolvimento deste projecto. Agradeço ainda à minha namorada Bárbara Pereira, por todo o apoio, compreensão e paciência que teve para comigo. Aos meus colegas que partilharam o Laboratório de Automação e Robótica ao longo destes meses, pela boa disposição e pelos momentos de descontração proporcionados. Finalmente, quero deixar um agradecimento a toda a minha família.

Palavras-Chave

Grupo Atlas; GPS; ROS; Nuvem de Pontos; Varrimento Laser; Navegabilidade; Reconstrução 3D.

Resumo

Este trabalho propõe um sistema para levantamento e avaliação da navegabilidade de uma zona de interesse. Numa primeira fase, usando o robô Atlas2000 pertencente ao grupo Atlas, do Laboratório de Automação e Robótica (LAR) do Departamento de Engenharia Mecânica (DEM) da Universidade de Aveiro, é realizado o levantamento da nuvem de pontos com o auxílio de dois lasers 2D. Um dos lasers realiza um varrimento horizontal para obter uma nuvem de pontos de obstáculos positivos, nomeadamente postes, bancos, entre outros. O segundo laser, com um varrimento num plano inclinado, realiza uma reconstrução 3D da área de passagem, com a acumulação de nuvens de pontos. Posteriormente, será possível realizar uma avaliação do local de passagem para verificar a possibilidade de navegabilidade do veículo. Finalmente, estando a fase anterior concluída, é criada uma representação da navegabilidade num formato possível de visualizar como uma camada no GoogleMaps.

Keywords

Atlas Group; GPS; ROS; Pointcloud; Laser scan; ; 3D Reconstruction.

Abstract

This work proposes a system that will be surveying and evaluating an area of interest. In a first phase, using the robot Atlas 2000 that belongs to the research and development team Atlas, of the Automation and Robotics Laboratory from the Mechanical Engineering Department at Aveiro University, a survey is obtained using two 2D lasers. One of the lasers by doing an horizontal scan obtains a pointcloud in which if is possible to detect the positive obstacles, like benches, poles and others. The second laser, with an inclined scan, makes a 3D reconstruction that is obtained by the accumulation of several lasers scans. In the second stage if is possible to evaluate the area of interest that the Atlas2000 will be passing through and determine if the area is navigable or not. Finally a file will be created with the final representation that can be visualized in GoogleMaps.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	v
1 Introdução	1
1.1 Enquadramento e Descrição do Problema	1
1.2 Projeto ATLAS	1
1.2.1 Introdução ao Projecto	1
1.2.2 Atlas 2000	2
1.3 Plataforma de Desenvolvimento de Software	3
1.4 Objetivos Principais	3
1.5 Estrutura da Dissertação	4
2 Estado da Arte	5
2.1 Localização e Mapeamento	5
2.2 Mapas para Navegação	7
2.2.1 Modelo 2.5D	8
2.2.2 Modelo 3D	12
2.3 Considerações Adicionais	14
3 Localização do Atlas 2000	15
3.1 Sistema de Coordenadas do Atlas 2000	15
3.2 Recurso ao modelo mecânico do Atlas 2000	16
3.3 Hector SLAM	18
4 Alterações Realizadas no Atlas 2000	21
4.1 Software	21
4.2 Intervenção Elétrica	21
4.3 Mudanças da Plataforma Física	23
5 Mapeamento e Representação Global	25
6 Resultados	33
6.1 Dados do Sistema Laser	33
6.2 Mapas de Navegabilidade	38

7 Conclusões e Trabalhos Futuros	41
Referências	43
8 Anexos	47
8.1 Anexo A	47

Lista de Figuras

1.1	Robôs para competição de condução autónoma do projeto ATLAS	2
1.2	Troca de rodas para uso do Atlas 2000 em <i>outdoors</i>	2
1.3	Recetor GPS e sistema auxiliar do Atlas 2000	3
2.1	Exemplo de reconstrução com uma <i>Kinect</i> [9]	7
2.2	Mapas de elevação e limitações inerentes.	8
2.3	a) Exemplo de várias células num mapa MLS. A célula <i>A</i> mostra que podem existir mais que um grupo de valores de media e variação. A célula <i>B</i> mostra que todas elas podem ter profundidade. A célula <i>C</i> representa uma que não tem profundidade; b) Mapa MLS que representa correctamente as elevações . . .	9
2.4	Mapa MLS. A amarelo são as zonas navegáveis, a azul as zonas não navegáveis e a vermelho as superfícies verticais	9
2.5	Nuvem de pontos após a aplicação do filtro que limita altura e comprimento . .	10
2.6	Determinação das normais de uma nuvem de pontos	10
2.7	Mapa resultante dos produtos das acessibilidades	12
2.8	Representação generalizada de uma octree	12
2.9	Resultado da limitação de profundidade das células nas octree. Diferentes resoluções, da esquerda para a direita, para profundidade de 0.08m, 0.64m e 1.28m, respetivamente.	13
2.10	Possíveis resultados do processamento de nuvens de pontos 3D de dados laser .	13
2.11	GPS - Constelação de satélites [17]	14
2.12	Mapa Google resultante da aplicação de uma camada em KML	14
3.1	Sistema de referenciais adotados para o Atlas 2000	16
3.2	Modelo de movimento utilizado para o Atlas 2000	16
3.3	Árvore de transformações do laser horizontal (<i>laser1</i>) e do laser inclinado (<i>laser2</i>)	18
3.4	Esquema de funcionamento das grelhas de ocupação do pacote do Hector SLAM	19
3.5	Árvore de transformações do laser horizontal (<i>laser1</i>) e do laser inclinado (<i>laser2</i>)	20
4.1	Aspetto inicial dos cabos nas placas de controlo do veículo.	22
4.2	Aspetto do circuito elétrico após alterações.	22
4.3	Lasers usados no trabalho relatado na dissertação	23
4.4	Lasers usados no trabalho relatado na dissertação	24
4.5	Reforço de apoios para a zona de acomodação do computador de bordo	24
5.1	Esquema de processamento de uma nuvem de pontos 3D para a sua representação no referencial global	25

5.2	Resultados da aplicação de algoritmos <i>ConcaveHull</i> e <i>ConvexHull</i> . a)As linhas vermelhas representam o resultado da utilização do algoritmo <i>ConcaveHull</i> a um conjunto de pontos; b)As linhas verdes representam o resultado da utilização do algoritmo <i>ConvexHull</i> a um conjunto de pontos [26]	26
5.3	Exemplo da aplicação da estrutura KML no Anexo A num mapa da Google.	27
5.4	Ambiente exterior usado para realizar os testes iniciais de utilização do método proposto	27
5.5	Reconstrução 3D do ambiente exterior apresentado na figura 5.4	28
5.6	Nuvem de pontos 3D com as zonas onde se encontram os obstáculos	28
5.7	Nuvem de pontos projetada das zonas onde se encontram os obstáculos	29
5.8	Estado inicial e final da aplicação do algoritmo <i>concave hull</i> a um obstáculo	29
5.9	Obstáculos delimitados pelas extremidades com a aplicação do algoritmo <i>ConvexHull</i> que são os pilares da figura 5.4	30
5.10	Representação da nuvem de pontos 2D num mapa da Google	31
6.1	Primeira zona de testes para realizar reconstrução 3D	34
6.2	Resultado do pós-processamento realizado à primeira zona com pessoas a passar. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis.	34
6.3	Resultado do pós-processamento realizado à sem movimentação de pessoas. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis.	35
6.4	Segunda zona de testes para realizar reconstrução 3D	35
6.5	Resultado do pós-processamento realizado à segunda zona com a influencia da passagem de pessoas. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis.	36
6.6	Resultado do pós-processamento realizado à segunda zona sem movimentação de pessoas. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis.	36
6.7	Terceira zona de testes para realizar reconstrução 3D	37
6.8	Resultado do pós-processamento realizado à terceira zona. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis.	37
6.9	Mapa de navegabilidade para a primeira zona com pessoas a passar. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis	38
6.10	Mapa de navegabilidade para a primeira zona sem movimentação de pessoas. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis	39
6.11	Mapa de navegabilidade para a segunda zona com a influencia da passagem de pessoas. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis	39
6.12	Mapa de navegabilidade para a segunda zona. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis	40
6.13	Mapa de navegabilidade para a terceira zona. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis	40

Lista de Tabelas

2.1	Resumo dos possíveis métodos SLAM (<i>Simultaneous Localization And Mapping</i>)	6
-----	------------------------------------------------------------------------------------	---

Capítulo 1

Introdução

1.1 Enquadramento e Descrição do Problema

Esta dissertação enquadra-se na área de condução autónoma do Laboratório de Automação e Robótica, do Departamento de Engenharia Mecânica da Universidade de Aveiro.

A existência de condução autónoma pressupõe, entre outras coisas, saber-se onde é possível um veículo navegar. Para condução em estradas públicas destinadas ao tráfego automóvel, os mapas proporcionados pela Google são adequados como base de trabalho, mas para condução em outras áreas, e.g., em campus universitários ou em zonas privadas de empresas, isso não é possível.

Tornou-se necessária a existência de uma ferramenta que possibilita a criação de mapas onde conste informação sobre certas áreas, fora das estradas públicas destinadas ao tráfego automóvel, que permita inferir se são compatíveis com a realização de rotas para a navegação autónoma de veículos. Com isto torna-se possível o planeamento de visitas guiadas sem condutores e transporte automático em áreas privadas. Neste projeto é necessário fazer-se o levantamento tridimensional de um espaço e para tal há que efetuar a aquisição e análise de dados provenientes de vários equipamentos. A plataforma para recolha de dados consiste no veículo robótico Atlas 2000 em modo de teleoperação. Já existem muitos desenvolvimentos nesta área, mas um sistema que engloba todas as partes necessárias para a construção de mapas de navegabilidade ainda não está disponível. Como referido anteriormente, é necessário recolher, avaliar e representar dados vindos de vários sensores para:

- Detecção de obstáculos positivos como, por exemplo, postes e bancos de jardim;
- Detecção de obstáculos negativos como, por exemplo, ravinas e desníveis de pavimento;

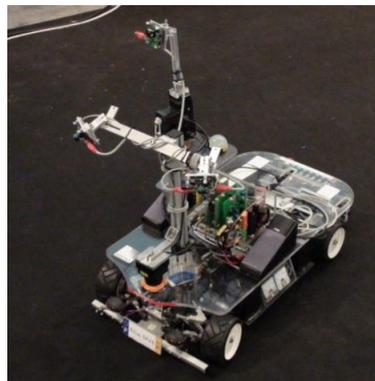
1.2 Projeto ATLAS

1.2.1 Introdução ao Projecto

O robô utilizado nesta dissertação, nomeadamente o Atlas 2000, foi desenvolvido no contexto do projeto ATLAS. Este projeto foi criado com o objetivo de participar em competições nacionais de condução autónoma, no qual, por várias vezes, alcançou o primeiro e segundo lugar. Os dois robôs que participavam nestas competições denominam-se de Atlas 2000 e Atlas MV apresentados na figura 1.1a e 1.1b, respetivamente.



(a) Atlas 2000



(b) Atlas MV

Figura 1.1: Robôs para competição de condução autónoma do projeto ATLAS

1.2.2 Atlas 2000

O Atlas 2000 sofreu, ao longo do tempo, várias alterações. Inicialmente a sua estrutura estava preparada para as competições de navegação *indoors*, mas devido a uma intervenção anteriormente executada por Eugénio Costa [1], este passou a poder navegar em ambientes exteriores. As intervenções ocorreram nas seguintes frentes:

- **Rodas** - Troca das rodas para pista de velocidade de escala 1:4 para umas rodas com pneu pneumático regulável de 8 polegadas de diâmetro, representadas na figura 1.2a e 1.2b respetivamente.



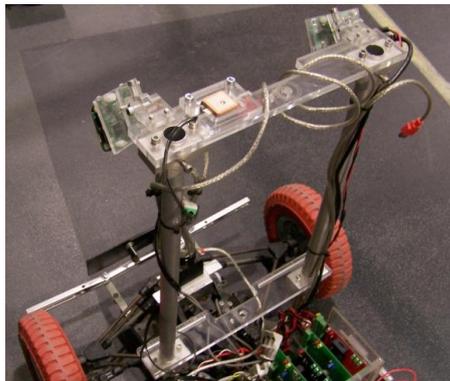
(a) Rodas para pista de velocidade



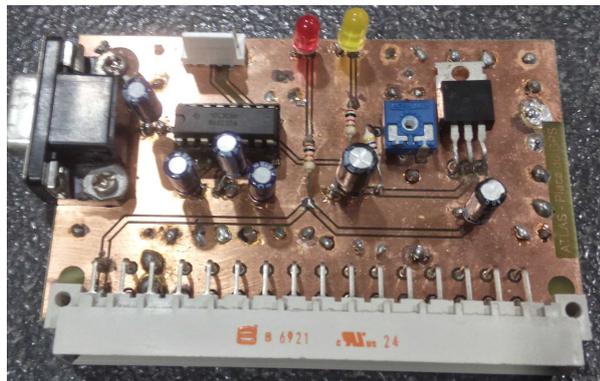
(b) Rodas com pneu pneumático

Figura 1.2: Troca de rodas para uso do Atlas 2000 em *outdoors*

- **Receptor GPS** - Para determinar a posição geográfica aproximada em que o veículo se encontra, foi adicionado um módulo receptor GPS e criada a placa de controlo e alimentação do mesmo. A figura 1.3 mostra o receptor GPS na estrutura de suporte fixa ao veículo e apresenta também a placa de controlo e alimentação.



(a) Recetor GPS e sua posição no carro



(b) Placa de monitorização e alimentação do recetor GPS

Figura 1.3: Recetor GPS e sistema auxiliar do Atlas 2000

- **Estrutura** - Foi realizado um reforço à estrutura existente bem como feita a troca de peças danificadas.

1.3 Plataforma de Desenvolvimento de Software

Optou-se por utilizar como plataforma de desenvolvimento o ecossistema *Robot Operating System* (ROS), mostrado no trabalho de Quigley, Conley, Gerkey et al. [2]. Esta plataforma encontra-se estabelecida como preferencial nos desenvolvimentos executados no Laboratório de Automação e Robótica, e tem como características principais a facilidade de reutilização do código desenvolvido, de integração do software compatível com o *hardware*, entre outras.

Esta plataforma tem um carácter bastante flexível para a criação de *software* diretamente relacionado com a área da robótica. Neste momento, o sistema proporciona, de forma gratuita e com código aberto, um número considerável de ferramentas, *drivers* e bibliotecas padrão. Encontra-se ainda em expansão, sendo a comunidade de investigadores da área os principais responsáveis por tal crescimento.

O ROS aplica uma estrutura modular constituída por *packages*, em que usa o método de publicação/subscrição para comunicação. De uma forma resumida, um módulo publicador gera informação que é encapsulada numa mensagem de um certo tipo e publicada num tópico. Todos os módulos interessados, que estejam subscritos ao referido tópico, terão acesso aos dados.

1.4 Objetivos Principais

Esta dissertação tem como objetivo principal a criação de um sistema, estrutura e software que permita a reconstrução e avaliação da navegabilidade de ambientes exteriores. Isto será possível com a utilização de dois distanciómetros *laser*, um 2D Hokuyo UTM-30LX e um 2D URG-04LX-UG01, mas também com a criação de novos módulos em conjugação com a utilização de módulos já existentes em ROS.

Os módulos usados terão de ser capazes de obter nuvens de pontos 3D, fazer o processamento dessa informação para determinar o que é navegável ou não em função da existência de

obstáculos, e por fim, representar a informação relevante num formato standard denominado de KML para ser utilizado como camada no GoogleMaps.

1.5 Estrutura da Dissertação

Esta dissertação está dividida, e organizada, da seguinte forma:

- Capítulo 2 - Estado da Arte
Apresentação das soluções já existentes na área da localização e mapeamento, bem como na criação de mapas para navegação.
- Capítulo 3 - Localização do Atlas 2000
Descrição de possíveis métodos de localização para o Atlas 2000
- Capítulo 4 - Alterações Realizadas ao Atlas 2000
Alterações realizadas à estrutura do Atlas 2000, bem como ao seu software.
- Capítulo 5 - Mapeamento e Representação
Explicação do processo usado desde a utilização da informação proveniente dos sensores usados até à sua representação num mapa da Google.
- Capítulo 6 - Resultados
Apresentação e comentário dos resultados obtidos
- Capítulo 7 - Conclusões e Trabalho Futuro
Neste capítulo são explicadas as conclusões e possíveis trabalhos futuros.

Capítulo 2

Estado da Arte

De maneira a suportar o desenvolvimento da ferramenta geradora de mapas de navegabilidade, fez-se uma pesquisa inicial sobre sistemas de localização e mapeamento existentes para criar mapas locais, métodos de tratamento e sua representação.

2.1 Localização e Mapeamento

Desde há uns anos até à atualidade, a área da robótica dedicada à localização e mapeamento tem ganho bastante atenção. Tem havido grandes avanços, mas as limitações ainda são consideráveis, nomeadamente em zonas extensas e com uma dinâmica elevada, e é ainda um problema a resolver para obter um veículo verdadeiramente autónomo [3].

É de uma importância elevada um veículo autónomo saber a sua própria localização para poder realizar planeamento de caminhos ou evitar obstáculos. Sem este tipo de informação, um veículo estará limitado ao facto de reagir ao que lhe poderá aparecer, e não planear para o futuro [4].

Serão mencionadas várias formas de adquirir a localização de um veículo:

- **Navegação estimada** - Esta forma de localização, mais conhecida por *Dead reckoning*, tem como objetivo estimar a posição atual tendo a informação da posição e direção numa iteração anterior, bem como dados de sensores relativos à propriocepção. O veículo está equipado com vários sensores, entre eles um codificador para as rotações do motor e um potenciómetro no sistema de direção dianteiro. Sabendo o diâmetro das rodas é possível determinar o espaço percorrido pelo veículo. Um problema grave com este método é que a cada iteração o erro associado é acumulado, não havendo limite para o erro total (*boundless*) por ausência de referência externa conhecida.
- ***A priori map localization*** - Tendo o robô sensores para obter informação sobre o ambiente que o rodeia, poderá ser obtida uma posição absoluta no mundo, recorrendo a um mapa previamente conhecido, e então, ser feita a localização por comparação com pontos de interesse (*distinct landmarks*). Para a criação de um *a priori map* tem de se efetuar um levantamento da área em questão. Zonas novas implicam a geração de novos mapas, o que em espaços exteriores dinâmicos pode resultar em mapas de pouca qualidade.
- **Localização e Mapeamento Simultâneos (SLAM)** - Nestes métodos é necessário armazenar a informação recolhida enquanto um robô se move. Nesta abordagem deixa

de ser necessário um *a priori map*, o que, por consequência, permite a construção de mapas de elevada complexidade. À medida que os pontos de referência vão acumulando, a posição estimada do robô é melhorada, bem como o mapeamento. É de mencionar que à medida que os pontos de referência são revistos mais vezes, o mapa começa a ganhar mais o aspeto de um "*a priori map*".

SLAM é uma opção ideal para navegação em ambientes completamente desconhecidos, porque o robô pode ser deixado sem qualquer tipo de intervenção humana para realizar o levantamento do espaço.

Na tabela 2.1, de forma resumida, são descritos alguns dos métodos SLAM existentes [5].

Tipo de Laser	Dimensionalidade da Representação da Posição e Orientação	
	2D	6D
2D	Mapeamento 2D Planar com um sonar ou laser 2D [3]	Slice-wise 6D SLAM Mapeamento 3D com a localização a considerar x, y, z , roll, yaw e pitch [6]
3D	Mapeamento 3D Planar usando método de localização planar com laser scan com varrimento vertical ou um 3D scanner	Verdadeiro 6D SLAM Mapeamento 3D com laser 3D ou um sistema stereo em que a posição é estimada com a informação do sensor [5]

Tabela 2.1: Resumo dos possíveis métodos SLAM (*Simultaneous Localization And Mapping*)

É ainda de mencionar que para realizar apenas uma reconstrução 3D parcial de um mapa local, pode ser usado o método descrito na dissertação de Pedro Salvado [7]. Este trabalho usa um distanciômetro laser suportado por uma plataforma rígida com uma ligeira inclinação, de forma a que este se encontre apontado para superfície do solo à frente do veículo. Neste trabalho foi desenvolvido um módulo em ROS que estava responsável pela acumulação de várias nuvens de pontos, com o auxílio de uma biblioteca denominada de *Point Cloud Library* [8]. Este módulo poderia subscrever a informação proveniente de câmaras *Bumblebee XB3* e *Kinect* e de distanciômetros *laser*.

A figura 2.1 ilustra o resultado de uma reconstrução com a utilização da câmara *Kinect* desenvolvido pelo trabalho de Joel Pereira [9].

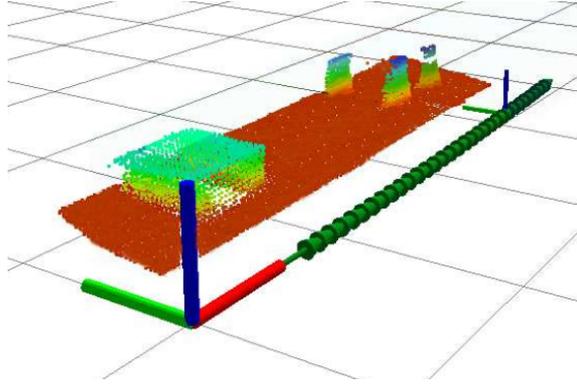


Figura 2.1: Exemplo de reconstrução com uma *Kinect* [9]

Para o caso do uso de um distanciômetro *laser* planar, a informação disponibilizada encontra-se em coordenadas polares, i.e., fornece a distância e o ângulo a que um obstáculo (ponto) se encontra relativamente ao sensor. Para a criação de uma nuvem tridimensional, representada em coordenadas cartesianas, é necessário que os dados em unidades polares sejam primeiro convertidos com o auxílio do sistema 2.1.

$$\begin{cases} x_i = r_i \cos(\theta_i) \\ y_i = r_i \sin(\theta_i) \\ z_i = 0 \end{cases} \quad (2.1)$$

onde r_i é a distância medida pelo sensor e θ_i o ângulo de varrimento de cada ponto. Depois são rodados e transacionados de acordo com as estimativas dos movimentos da plataforma para acumulação em nuvem de pontos. Isto é possível com a concatenação (operação $+=$ da *Point Cloud Library*) entre nuvens de pontos.

Após esta fase, é necessário transformar os valores obtidos, que se encontravam no referencial do sensor para o referencial de origem. Que, neste caso, foi considerado como referencial de origem o centro do para-choques frontal, sendo mostrado na equação 2.2 a transformação mencionada.

$$f(x) = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{center_bumper} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{sensor} \quad (2.2)$$

2.2 Mapas para Navegação

Mesmo depois de obter informação proveniente de *lasers* ou câmaras e realizar o mapeamento, momento em que estes mapas poderão servir de base ao planeamento de rotas, verifica-se que este processo apresenta um grau elevado de dificuldade e, conseqüentemente, esta área tem recebido bastante atenção e o assunto tem sido bastante investigado. Neste contexto, a representação geométrica destes mapas (mapas de elevação) têm sido populares, sejam os modelos 3D [10] ou os 2.5D [11], sendo necessário um nível elevado de processamento para processar e construir estes modelos 3D.

2.2.1 Modelo 2.5D

Os mapas de elevação consistem em grelhas de ocupação [12] [13] em que cada célula inclui o valor de altura. É de notar que a caracterização discreta do ambiente em zonas como túneis ou passagens sob pontes seriam representadas erradamente como obstáculos, conforme apresentado na figura 2.2.

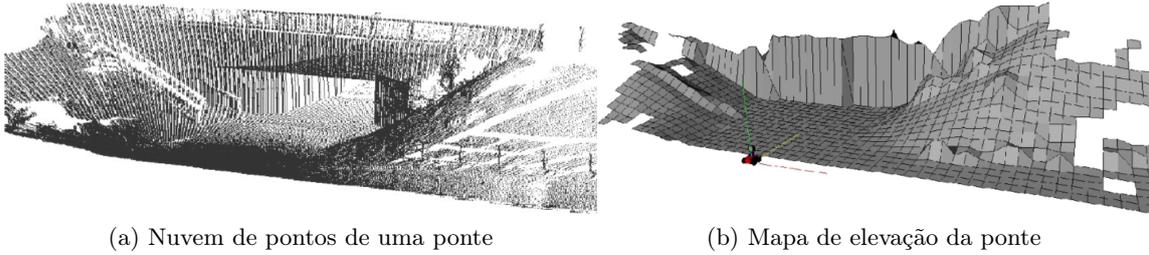


Figura 2.2: Mapas de elevação e limitações inerentes.

No trabalho de Triebel, Pfaff e Burgard [11], foi apresentada uma extensão aos mapas de elevação que é denominada por *MLS Maps (Multi Level Surface Maps)*. Estes mapas consistem numa grelha 2D de tamanho variável em que cada célula representa cada uma das superfícies com a média e a variação da altura registada. Estes parâmetros, ao serem processados, determinam o facto de a célula em questão ser acessível àquela altura, e é representada pela média μ_{ij}^k , e a incerteza representada pela variação σ_{ij}^k . Na figura 2.3a podem ser visualizados alguns exemplos de representações de células num MLS.

Um mapa do tipo MLS pode ser criado por dois processos, por uma nuvem de pontos ou juntando dois mapas MLS já existentes. Conclui-se que, se um mapa m_1 for criado de uma nuvem de pontos C_1 e um mapa m_2 de uma nuvem de pontos C_2 , então um mapa m_3 resultante de juntar m_1 e m_2 é idêntico de um criado de uma nuvem de pontos $C_3 = C_1 + C_2$. A criação de um MLS dada uma nuvem de pontos C segue os seguintes pontos:

- Cada célula do mapa com índices (i, j) armazena todos os pontos $P = (x, y, z)$ a ela correspondentes, isto é:

$$s(i) \leq x \leq s(i + 1) \quad (2.3)$$

$$s(j) \leq y \leq s(j + 1) \quad (2.4)$$

em que s é a largura da célula.

- Para cada célula é calculado um intervalo de alturas com base nas alturas dos pontos da mesma. Se o espaçamento entre dois intervalos consecutivos forem menores que um valor inicialmente estabelecido, γ , isto significa que os mesmos pertencem de facto ao mesmo intervalo. Este espaçamento deve ser escolhido de forma que o robô que pretenda navegar consiga passar nesse espaço.
- Os intervalos são classificados de horizontais ou verticais se, respetivamente, a altura dos mesmo for inferior ou superior a um certo valor limite.
- Para cada intervalo classificado como vertical, é armazenado o valor da média e a maior variação, pois estabeleceu-se que eram os parâmetros importantes para determinar se

seria navegável. É memorizado também o comprimento do intervalo, denominado de profundidade de uma célula (d), pois este é importante para o caso de juntar dois mapas MLS.

- Para cada intervalo classificado como horizontal, é calculada a média, μ , e o desvio padrão, σ , para todas as medições no mesmo. De mencionar que é atribuído um valor nulo à profundidade das células horizontais.

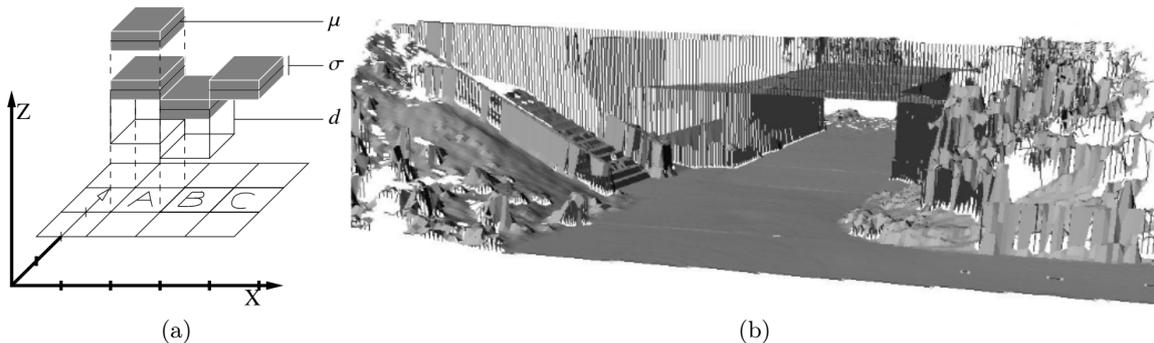


Figura 2.3: a) Exemplo de várias células num mapa MLS. A célula A mostra que podem existir mais que um grupo de valores de média e variação. A célula B mostra que todas elas podem ter profundidade. A célula C representa uma que não tem profundidade; b) Mapa MLS que representa correctamente as elevações

Por fim, após a classificação das células em horizontais ou verticais, é verificado para cada uma se tem entre 5 a 8 células vizinhas e se a diferença de altura entre a célula em questão e as células vizinhas é inferior a 10 cm. Se estas condições forem verificadas, uma célula é classificada como navegável. Na figura 2.4 pode-se ver o resultado após classificação e representação.

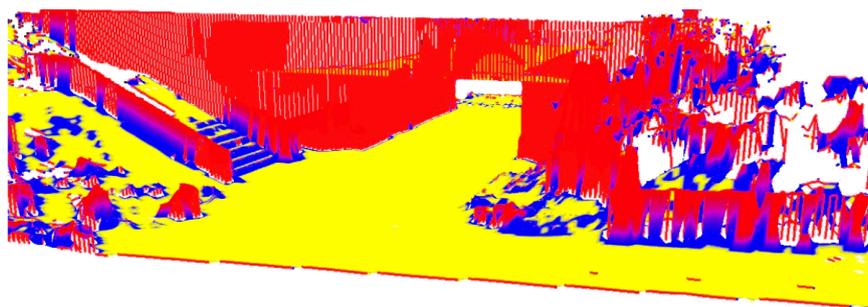


Figura 2.4: Mapa MLS. A amarelo são as zonas navegáveis, a azul as zonas não navegáveis e a vermelho as superfícies verticais

Uma variação deste método foi usado na dissertação realizada por Diogo Matos [14]. Após a recolha de uma nuvem de pontos, é-lhe aplicado um pré-processamento que consistiu em usar um filtro que descartava pontos acima dos 2 metros e a uma distância superior a 25 metros (figura 2.5). Além deste filtro, foram estimadas as normais da nuvem de pontos, aplicando o produto externo aos vetores definidos entre um ponto e os seus vizinhos. Isto é possível com

o auxílio da biblioteca da *PointCloud*. Nesta biblioteca existem duas formas de determinar as normais, em que o método usado considerava os vizinhos num determinado raio do ponto a processar.

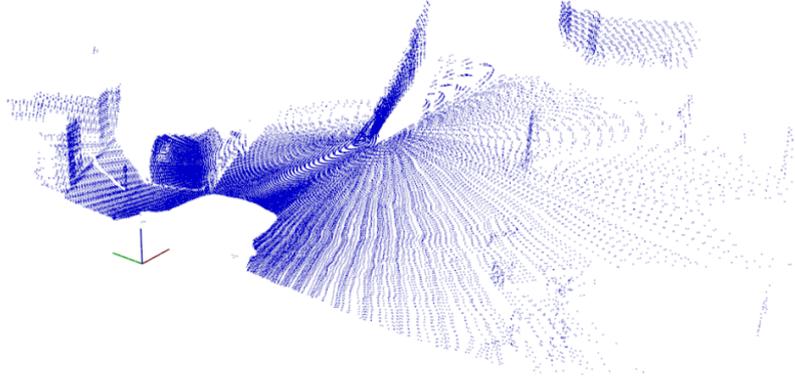


Figura 2.5: Nuvem de pontos após a aplicação do filtro que limita altura e comprimento

Para evitar variações muito grandes induzidas pela discretização da superfície, foi usado um raio com valor suficiente para que a normal determinada seja coerente em relação ao que a rodeia. O problema de determinar as normais com o raio da vizinhança, é que este raio tem de ser suficientemente grande para que as normais em cada ponto não sejam influenciadas por erros de medição aos pontos vizinhos mais próximos, mas também que a sua dimensão oculte variações reais com interesse. Este problema pode ser visualizado na figura 2.6.

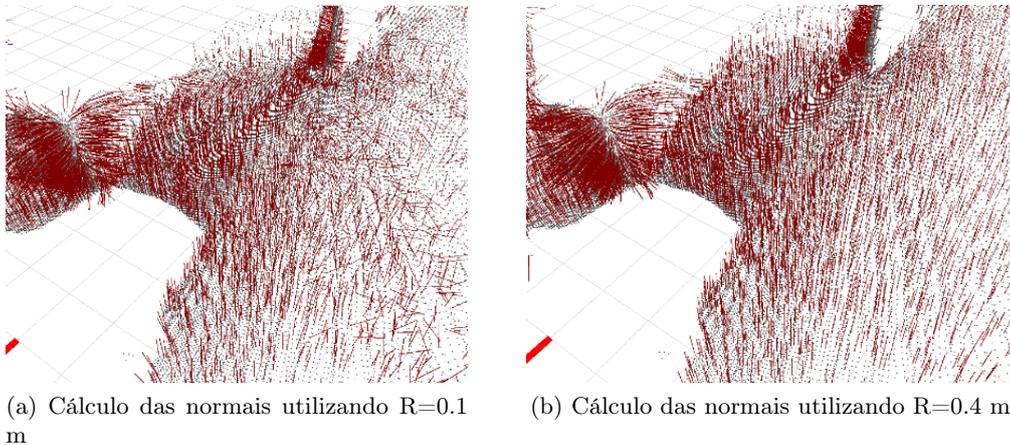


Figura 2.6: Determinação das normais de uma nuvem de pontos

Após esta fase é inicializada uma grelha de ocupação, são processados os pontos da nuvem e é armazenada a informação para cada célula que lhes corresponde. A equação 2.5 mostra como é calculada a célula $C(i, j)$ onde o ponto \vec{p}_n , pertence.

$$\vec{p}_n = (p_x, p_y) \hookrightarrow C(i, j) : \begin{cases} i = \text{round}\left(\frac{p_x}{S_x}\right) \\ j = \text{CARaxis_col} + \text{round}\left(\frac{p_y}{S_y} + k\right) \end{cases} \quad (2.5)$$

onde p_x e p_y representam as coordenadas x e y do ponto \vec{p}_n .

Para ser determinada a coluna à qual um ponto pertence é levado em conta a posição central da grelha de ocupação, para que esta fique definida em torno da origem do referencial da nuvem de pontos. Logo, k na equação 2.5, toma os valores de $+0.5$ ou -0.5 para o caso de p_y ser positivo ou negativo.

Na fase seguinte são calculadas as médias aritméticas da coordenada z e os ângulos associados às normais nos pontos de uma célula, μ_z , μ_α , μ_β e μ_ψ , respetivamente. Com as equações 2.6, 2.7, 2.8 e 2.9 são determinados valores de confiança ν associados a cada célula para os valores de cada média, que representam o grau de representatividade dos valores contidos na célula, para posteriormente ser usado no cálculo da acessibilidade.

$$\nu_z = 1 - \frac{\sigma_z}{\sigma_{0z}} \quad (2.6) \qquad \nu_\alpha = 1 - \frac{\sigma_\alpha}{\sigma_{0\alpha}} \quad (2.7)$$

$$\nu_\beta = 1 - \frac{\sigma_\beta}{\sigma_{0\beta}} \quad (2.8) \qquad \nu_\psi = 1 - \frac{\sigma_\psi}{\sigma_{0\psi}} \quad (2.9)$$

A confiança ν , que é calculada com base no desvio padrão corrigido, σ , e de uma constante σ_0 para normalizar os valores da confiança entre 0 e 1. É de mencionar que estes cálculos só são válidos para células com mais do que um ponto, em que, para o caso de existir apenas um ponto numa célula, é-lhe atribuído uma confiança de 0.5.

Nesta fase, cada célula é classificada com 8 variáveis, ν_z , ν_α , ν_β , ν_ψ , μ_z , μ_α , μ_β e μ_ψ , em que 4 delas são as médias geométricas das confianças e as outras quatro são os valores de confiança. Com as equações 2.10, 2.11, 2.12 e 2.13 são calculados os valores de disparidade para cada uma das propriedades que significam o quanto uma célula se difere das suas células vizinhas, em que k representa o número de vizinhos de uma célula.

$$\vartheta_z^n = \frac{\sum_{i=1}^k \frac{|\mu_z^i - \mu_z^n|}{\sqrt{\nu_z^i \cdot \nu_z^n}}}{k} \quad (2.10) \qquad \vartheta_\alpha^n = \frac{\sum_{i=1}^k \frac{|\mu_\alpha^i - \mu_\alpha^n|}{\sqrt{\nu_\alpha^i \cdot \nu_\alpha^n}}}{k} \quad (2.11)$$

$$\vartheta_\beta^n = \frac{\sum_{i=1}^k \frac{|\mu_\beta^i - \mu_\beta^n|}{\sqrt{\nu_\beta^i \cdot \nu_\beta^n}}}{k} \quad (2.12) \qquad \vartheta_\psi^n = \frac{\sum_{i=1}^k \frac{|\mu_\psi^i - \mu_\psi^n|}{\sqrt{\nu_\psi^i \cdot \nu_\psi^n}}}{k} \quad (2.13)$$

Por fim usando as equações 2.14, 2.15, 2.16 e 2.17 é calculado a acessibilidade das células, ϖ , comparando a disparidade com um valor limite, Th . Na figura 2.7 pode ser visualizado o resultado do produto das acessibilidades, usando um Th_z de 10 cm, $Th_\alpha = Th_\beta = Th_\psi$ de 0.2 radianos e para o cálculo da confiança foi usado $\sigma_0 = 2 \cdot Th$. De mencionar que nesta figura a cor azul corresponde a uma acessibilidade de 1 e a cor vermelha a uma de valor igual a 0

$$\varpi_z^n = 1 - \frac{\vartheta_z^n}{Th_z} \quad (2.14) \qquad \varpi_\alpha^n = 1 - \frac{\vartheta_\alpha^n}{Th_\alpha} \quad (2.15)$$

$$\varpi_{\beta}^n = 1 - \frac{\vartheta_{\beta}^n}{Th_{\beta}} \quad (2.16)$$

$$\varpi_{\psi}^n = 1 - \frac{\vartheta_{\psi}^n}{Th_{\psi}} \quad (2.17)$$

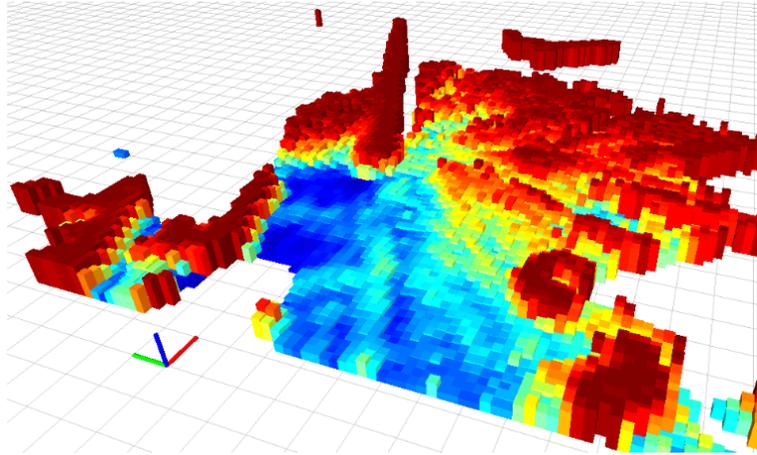


Figura 2.7: Mapa resultante dos produtos das acessibilidades

2.2.2 Modelo 3D

Na investigação apresentada por Hornung, Wurm, Bennewitz et al. [10] foi criada uma representação 3D de um ambiente que considera não só as zonas ocupadas mas também as zonas livres ou desconhecidas. Este modelo usa a filosofia das *Octrees* que, na base, é uma estrutura de árvore em que cada célula da octree representa um cubo em espaço físico (2.8).

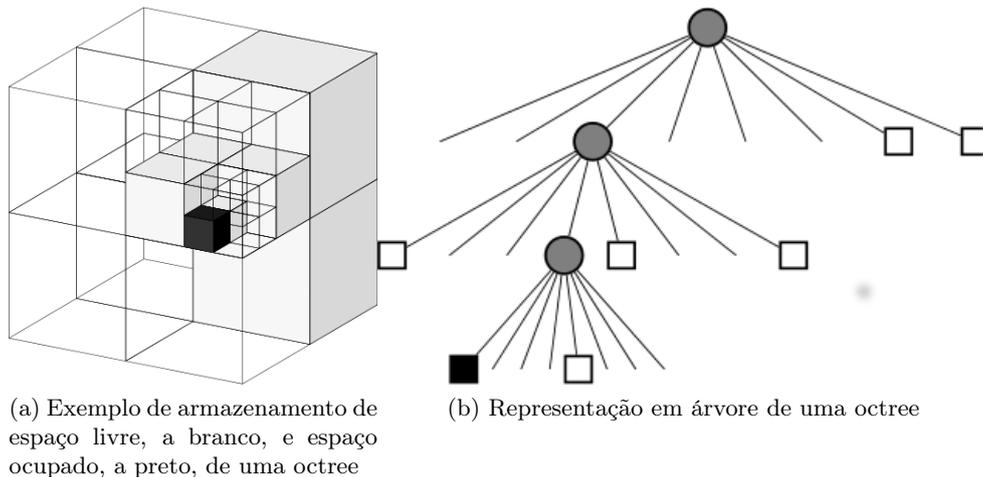


Figura 2.8: Representação generalizada de uma octree

Este modelo de octree tem como características principais os seguintes pontos:

- **Modelo *Full 3D*** - O mapa modela o ambiente sem qualquer tipo de simplificações, em que, também tem de representar o espaço livre e desconhecido. De notar que o espaço livre e ocupado é importante para planeamento, e o desconhecido para a exploração.

- **Atualizável** - Este modelo permite a adição de novas informações às já existentes. As atualizações e modelações levam em consideração os erros dos sensores usados e mudanças no ambiente para levar em conta a dinâmica do espaço. Existe também a possibilidade de usar vários robôs a atualizar um mesmo mapa.
- **Flexível** - Não necessita de conhecimento *a priori* do tamanho do mapa resultante, pois este permite uma expansão dinâmica. Ele também permite a representação em várias resoluções para o tratamento de informação ser otimizável.
- **Compacto** - O mapa é armazenado de forma compacta para ser possível a sua troca entre robôs e para minimizar o espaço utilizado em disco. Variando a resolução da octree, ilustrado na figura 2.9, a quantidade de dados poderá variar.

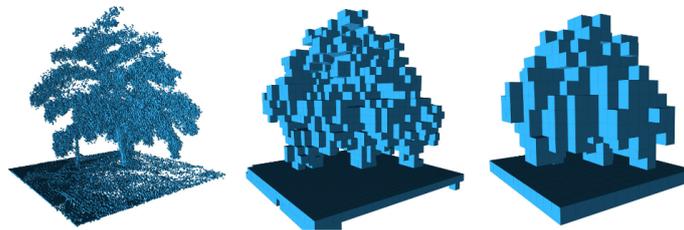
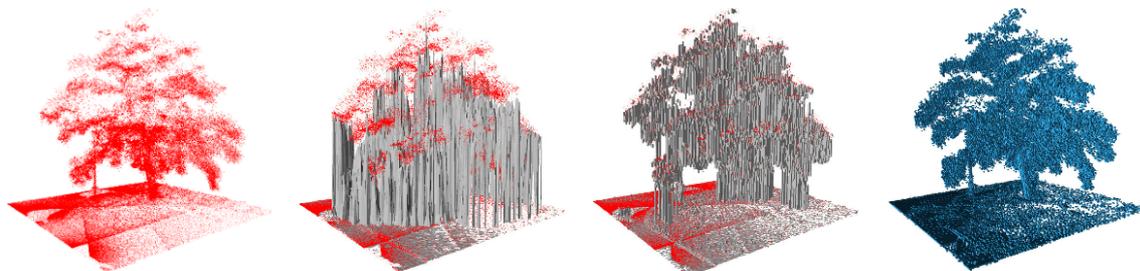


Figura 2.9: Resultado da limitação de profundidade das células nas octree. Diferentes resoluções, da esquerda para a direita, para profundidade de 0.08m, 0.64m e 1.28m, respetivamente.

Comparando o resultado do modelo 3D mostrado na figura 2.10d, com os resultados de outros métodos tipicamente usados, concluí-se que de uma nuvem de pontos 3D, sem qualquer processamento prévio (2.10a), pode ser obtido também um mapa de elevação para os pontos em questão (2.10b) [15] e um mapa MLS (2.10c), referido no sub-capítulo 2.2.1.



(a) Nuvem de pontos de um laser sem qualquer tipo de tratamento (b) Mapa de elevação dessa mesma nuvem de pontos (c) Mapa MLS (Multi-level surface map) (d) Mapa resultante usando a filosofia de Octrees falada

Figura 2.10: Possíveis resultados do processamento de nuvens de pontos 3D de dados laser

2.3 Considerações Adicionais

Após realizado o levantamento do estado da arte, conclui-se que para criar mapas que representassem os obstáculos, num sistema global, poderiam ser usados como base o mapas da Google. Teriam de ser recolhidos dados da área proposta nesta dissertação (Universidade de Aveiro) e para existir uma passagem desses dados de nível local para um global, serão usados recetores GPS.

O sistema de posicionamento global (GPS) oferece, com precisão e em qualquer parte do globo, uma posição a três dimensões e a velocidade a um utilizador que tenha um recetor apropriado. Este posicionamento é realizado com o auxílio de uma constelação de 24 satélites organizados em 6 órbitas com 4 satélites por órbita (figura 2.11), utilizando um tempo universal sincronizado entre eles (UTC). Pode-se ainda referir que o protocolo para comunicar com o recetor é o NMEA 0183 [16].



Figura 2.11: GPS - Constelação de satélites [17]

Determinou-se que para representação nos mapas da Google era possível a utilização de camadas para o efeito. O KML ou *Keyhole Markup Language* [18] é um formato de ficheiros usado para representar dados geográficos em aplicações que usem mapas, logo com a ajuda da biblioteca KML será possível a criação de polígonos representativos dos obstáculos. É ainda de mencionar que este é um formato standard gerido pela Open Geospatial Consortium, Inc.

Um possível resultado com este processo de representação está ilustrado na figura 2.12.



Figura 2.12: Mapa Google resultante da aplicação de uma camada em KML

Capítulo 3

Localização do Atlas 2000

Para o desenvolvimento dos mapas para navegação de veículos será necessário conhecer a posição e orientação do veículo que executa o levantamento, os quais seriam obtidos a partir dos vários sensores que existem no Atlas 2000, nomeadamente lasers, codificadores e recetores GPS.

Serão apresentadas as duas metodologias consideradas para obtenção da informação necessária, proveniente do Atlas 2000 e dos seus sensores, para ser determinada a posição e orientação do mesmo.

3.1 Sistema de Coordenadas do Atlas 2000

Além da instalação de sensores, é necessário relacioná-los entre si e transmitir essa informação de posicionamento relativo para um sistema responsável pela recolha e tratamento desses dados. Esta informação é disponibilizada por um tipo de ficheiro com extensão URDF (Unified Robot Description Format), que permite descrever a posição e orientação dos sensores relativamente a um referencial à escolha [2]. De notar que nem todos os referenciais estão relacionados com um sensor, como por exemplo `\base_link` (que diz respeito ao referencial de movimento do veículo) e o `\world` (referencial mundo com origem onde se deu o início da recolha de informação). Esta relação é necessária para possibilitar a realização de transformações e cálculos em relação ao robô. É importante a utilização destes referenciais para que a informação proveniente de equipamentos diferentes esteja sobre o mesmo referencial final.

A figura 3.1 mostra um esquema onde estão representados os referenciais associados a cada sensor.

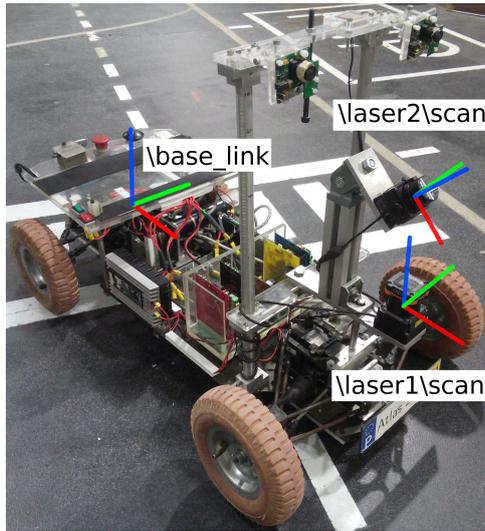


Figura 3.1: Sistema de referenciais adotados para o Atlas 2000

3.2 Recurso ao modelo mecânico do Atlas 2000

O modelo de movimento do Atlas 2000 segue uma simplificação do modelo mecânico do veículo de Ackerman, denominado de Modelo Bicicleta. Este modelo, com o objetivo de apenas monitorizar o caminho percorrido, simplifica um veículo de quatro rodas num de duas, combinando as duas rodas dianteiras e as duas rodas traseiras, usando assim um veículo de duas rodas [19]. A figura 3.2b ilustra a simplificação do modelo mecânico usado no Atlas 2000.

O cálculo da hodometria do robô necessita de vários parâmetros, nomeadamente a distância entre eixos, a relação de transmissão, a velocidade e o ângulo da direção do veículo. Com o auxílio do algoritmo de egomotion pertencente ao *software* do Atlas 2000, pode ser determinada a orientação (ψ) e posição (coordenadas x e y) do carro. Na figura 3.2a é apresentado um diagrama com os parâmetros de entrada necessários para serem determinados a orientação e posição anteriormente referidas.

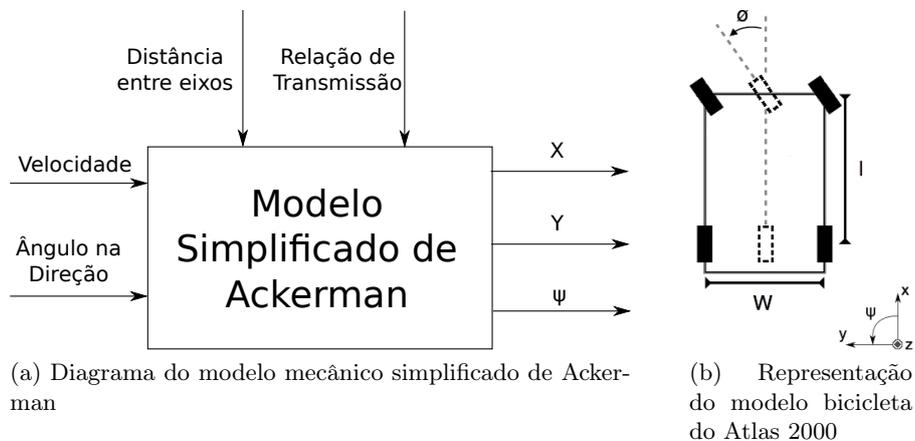


Figura 3.2: Modelo de movimento utilizado para o Atlas 2000

Foi necessário definir o vetor de estado a utilizar, que é constituído pelas coordenadas do veículo no plano xy e a sua orientação ψ , o valor da velocidade atual v e a orientação das rodas ϕ .

$$f(x) = \begin{bmatrix} x \\ y \\ \psi \\ v \\ \phi \end{bmatrix} \quad (3.1)$$

Para se determinar a posição e orientação do veículo, têm de ser obtidas as variações das coordenadas no plano x e y do carro, identificadas como Δx e Δy respetivamente, mas também a variação da orientação do carro em torno do eixo z , denominada de $\Delta\psi$. Com o apoio de fórmulas descritas no trabalho de Latombe [19], e apresentadas nas equações 3.2, 3.3 e 3.4, é possível determinar estas variações.

$$\Delta\psi = \sin(\phi) \frac{v}{l} \Delta t \quad (3.2)$$

$$\Delta x = \cos(\phi) \cos(\phi) v \Delta t \quad (3.3)$$

$$\Delta y = \sin(\phi) \cos(\phi) v \Delta t \quad (3.4)$$

Após determinadas as variações das coordenadas no plano local ao veículo e a variação da orientação, com o auxílio das equações 3.5, 3.6 e 3.7, é calculada a posição e orientação real do Atlas 2000.

$$\psi_i = \psi_{i-1} + \Delta\psi \quad (3.5)$$

$$x_i = x_{i-1} + \Delta x \quad (3.6)$$

$$y_i = y_{i-1} + \Delta y \quad (3.7)$$

Nesta fase do trabalho, as informações provenientes dos sensores não se encontram relacionadas entre si, logo, utiliza-se o pacote em ROS (*atlas2000_egomotion*) que tem como função publicar as transformações entre referenciais (*frames*), e caracterizar o carro em relação ao ambiente que o rodeia, relacionando, assim, todos os referenciais a um principal, nomeadamente o *frame \ world*.

A figura 3.3 ilustra a árvore de transformações do referencial do laser horizontal para o referencial mundo e a árvore de transformações do referencial do laser inclinado até ao referencial mundo.

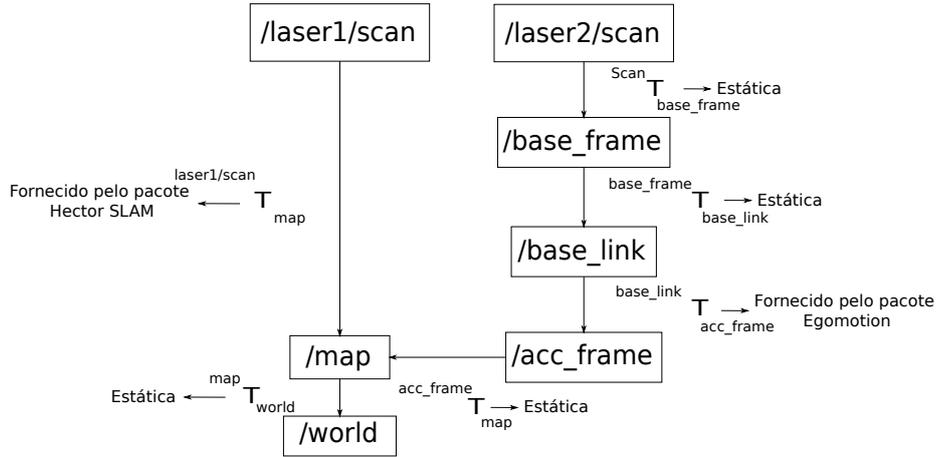


Figura 3.3: Árvore de transformações do laser horizontal (laser1) e do laser inclinado (laser2)

3.3 Hector SLAM

Para realizar a localização do Atlas 2000, poderá ser usado o pacote ROS denominado de *hector_slam*. No âmbito desta dissertação será utilizado o *hector_mapping* que permite a localização e mapeamento simultâneo (SLAM). Este método é denominado por um *SLAM frontend* uma vez que estima o movimento do robô em tempo real. O algoritmo depende apenas da existência de um distanciómetro planar de alta frequência para poder realizar o alinhamento entre varrimentos.[20] [21]

Este pacote subscreve o tópico de um laser e publica, principalmente, uma grelha de ocupação do espaço que o rodeia e as transformações necessárias para saber a posição e orientação em que o laser se encontra em qualquer momento e onde foi iniciado o pacote.

Para a criação de um mapa e por consequência a sua grelha de ocupação, usando o *hector_mapping*, é necessário realizar um passo denominado de alinhamento. É de mencionar que este alinhamento é baseado numa aproximação Gauss-Newton inicialmente apresentada no estudo de Lucas e Kanade [22]. Para este alinhamento, entre varrimentos diferentes ou de varrimentos novos com um mapa existente, é necessário determinar a transformação representada por $\xi = (p_x, p_y, \psi)^T$ que minimiza a equação 3.8.

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^n [1 - M(S_i(\xi))]^2 \quad (3.8)$$

em que $S_i(\xi)$ são as coordenadas no mundo dos pontos recolhidos por um varrimento laser e $M(S_i(\xi))$ determina a ocupação nesse ponto. Nesta fase, é relevante mencionar que dada a natureza discreta da grelha de ocupação resultante do *hector_mapping*, existe um limite à precisão que pode ser obtida, mas também não permite o processamento direto dos valores interpolados ou suas derivadas parciais. Logo, é necessário a utilização de uma sub-grelha para aplicar uma interpolação bilinear (figura 3.4a) para estimar a probabilidade de ocupação e suas derivadas (gradiente de ocupação) mostrado na figura 3.4b.

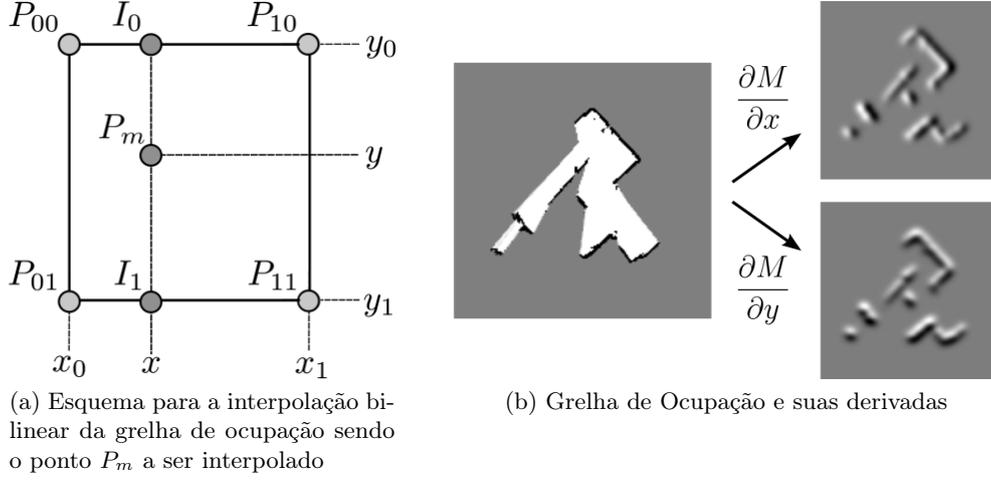


Figura 3.4: Esquema de funcionamento das grelhas de ocupação do pacote do Hector SLAM

De notar que a resolução considerada nesta grelha é igual a 1.

Dado um mapa contínuo de coordenadas P_m , o valor de ocupação $M(P_m)$ e o seu gradiente $\nabla M(P_m) = (\frac{\delta M}{\delta x}(P_m), \frac{\delta M}{\delta y}(P_m))$, podem ser aproximados usando as equações 3.9, 3.10 e 3.11 respetivamente.

$$M(P_m) \approx \frac{y - y_0}{y_1 - y_0} \left(\frac{x - x_0}{x_1 - x_0} M(P_{11}) + \frac{x_1 - x}{x_1 - x_0} M(P_{01}) \right) + \frac{y_1 - y}{y_1 - y_0} \left(\frac{x - x_0}{x_1 - x_0} M(P_{10}) + \frac{x_1 - x}{x_1 - x_0} M(P_{00}) \right) \quad (3.9)$$

$$\frac{\delta M}{\delta x}(P_m) \approx \frac{y - y_0}{y_1 - y_0} (M(P_{11}) - M(P_{01})) + \frac{y_1 - y}{y_1 - y_0} (M(P_{10}) - M(P_{00})) \quad (3.10)$$

$$\frac{\delta M}{\delta y}(P_m) \approx \frac{x - x_0}{x_1 - x_0} (M(P_{11}) - M(P_{01})) + \frac{x_1 - x}{x_1 - x_0} (M(P_{10}) - M(P_{00})) \quad (3.11)$$

em que P_{00} , P_{01} , P_{10} e P_{11} são os valores inteiros, das coordenadas no mapa, e P_m a coordenada do ponto interior à célula.

Dada uma estimacão inicial da transformacão, é necessário determinar $\Delta\xi$ que otimiza o erro medido com base na equacão 3.12 e, por consequência, devido à primeira ordem da expansão de Taylor de $M(S_i(\xi + \Delta\xi))$ é deduzida a equacão 3.13.

$$\sum_{i=1}^n [1 - M(S_i(\xi + \Delta\xi))]^2 \rightarrow 0 \quad (3.12)$$

$$\sum_{i=1}^n \left[1 - M(S_i(\xi)) - \nabla M(S_i(\xi)) \frac{\delta S_i \xi}{\delta(\xi)} \Delta\xi \right]^2 \rightarrow 0 \quad (3.13)$$

Na figura 3.5 é mostrada a árvore de transformacões usada e que relaciona o referencial do laser horizontal para o mundo e o referencial do laser inclinado para o mundo.

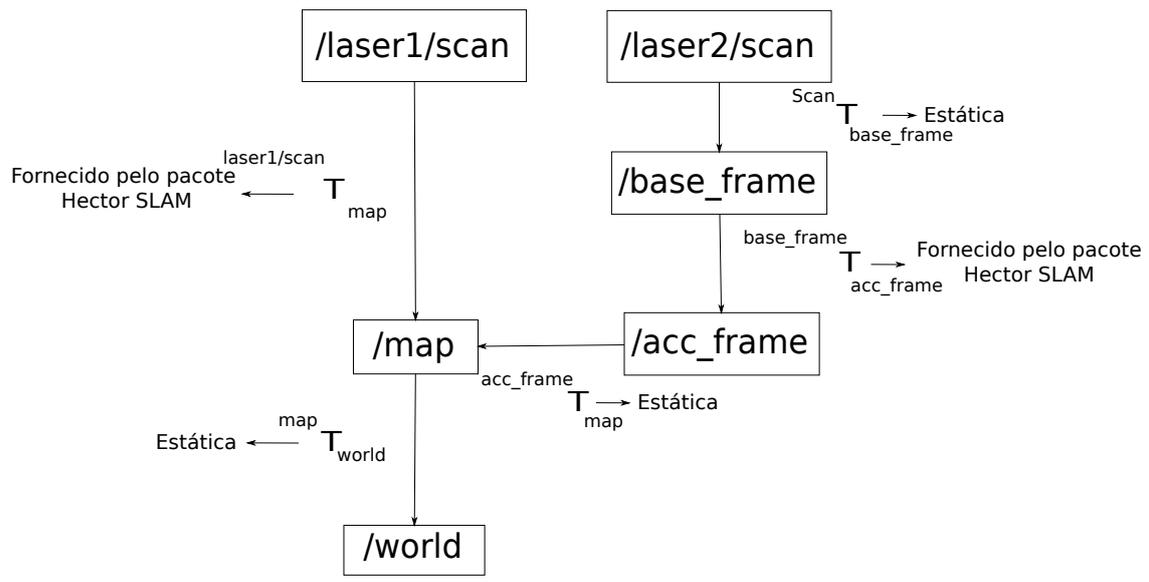


Figura 3.5: Árvore de transformações do laser horizontal (laser1) e do laser inclinado (laser2)

Capítulo 4

Alterações Realizadas no Atlas 2000

Para preparar o robô Atlas 2000 para os trabalhos de levantamento e criação dos mapas de navegabilidade, houve necessidade de realizar algumas intervenções no sistema mecânico, no sistema elétrico e ao nível do *software* base. Todas as alterações serão descritas com mais detalhe ao longo deste capítulo.

4.1 Software

No início desta dissertação o veículo encontrava-se a operar com uma coleção de *software open-source* denominado CARMEN (*Carnegie Mellon Robot Navigation Toolkit*) [23] que entretanto foi substituída pela plataforma ROS já apresentada na secção 1.3. Explicando um pouco a filosofia da CARMEN, trata-se de uma coleção modular, expansível, programada principalmente na linguagem C, com módulos básicos de navegação, localização e outros do interesse da robótica móvel. A comunicação entre processos escolhida para a coleção trata-se de um pacote denominado de IPC (*Inter-process Communication*) [24] que é estabelecida com sockets TCP/IP e sockets Unix.

4.2 Intervenção Elétrica

Ao longo de vários trabalhos anteriores foram realizadas alterações pontuais ao sistema elétrico que não respeitaram uma estrutura de ligações elétricas organizada e segura, nomeadamente:

- Condutores sob tensão com pontas soltas;
- Circuitos constituídos por troços condutores concatenados sem consistência entre cores do isolamento;
- Ausência total de etiquetagem, e.g., que tornasse mais evidente a forma correta de ligação dos condutores às baterias;
- Interruptor geral com ligações incorretas interrompia apenas metade do circuito.

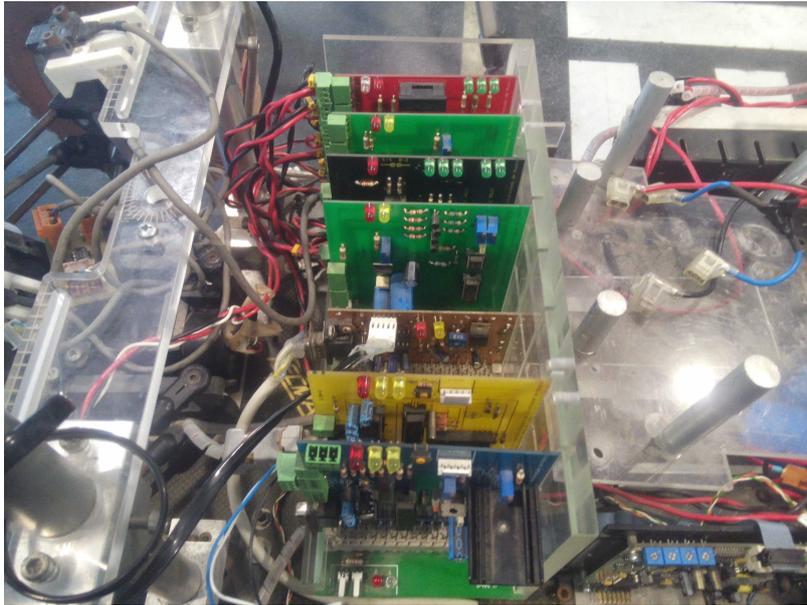
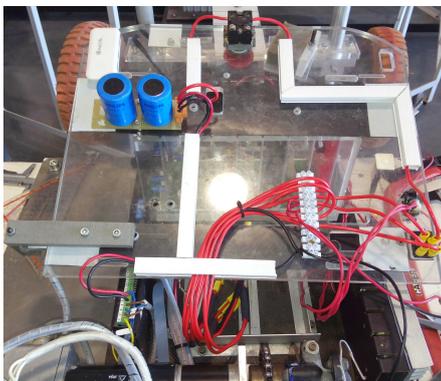
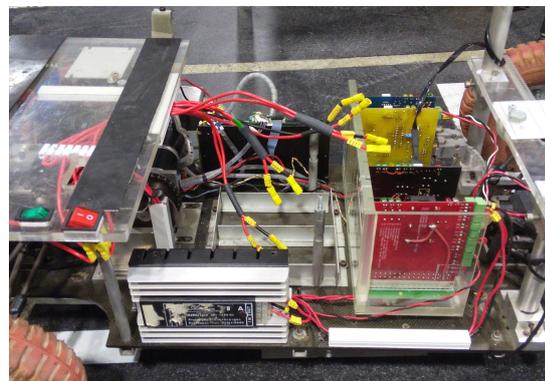


Figura 4.1: Aspecto inicial dos cabos nas placas de controlo do veículo.

O aspecto do circuito, antes intervenção nas placas de controlo e na zona das baterias, pode ser visto na figura 4.1. Para melhor organizar a cablegem utiliza-se a zona inferior da placa acrílica de cobertura da zona traseira do veículo para realizar uma grande parte do suporte para o circuito apresentado na figura 4.2a. É de mencionar também que o esquema de cores foi decidido de tal forma que apenas os condutores ligados ao terminal de referência da alimentação possuem isolamento preto, isto é, o terminal indicado como negativo da primeira bateria, o zero do servo-amplificador e o zero do regulador de tensão. Todos os restantes, que apresentam diferença de potencial (12 Volt, 24 Volt e 48 Volt) ficaram com isolamento vermelho e encontram-se etiquetados com a tensão correspondente das ligações das baterias, como pode ser visto na figura 4.2b.



(a) Organização do circuito elétrico após intervenção.



(b) Etiquetação dos cabos terra e com tensão.

Figura 4.2: Aspecto do circuito elétrico após alterações.

4.3 Mudanças da Plataforma Física

Para realizar a recolha de informação do ambiente onde o veículo irá percorrer, foram escolhidos dois lasers Hokuyo, modelos URG-04LX-UG01 e UTM-30LX, mostrados respetivamente nas figuras 4.3a e 4.3b. A razão da escolha da utilização destes dois modelos foi a limitação de material disponível e adequabilidade para o trabalho.

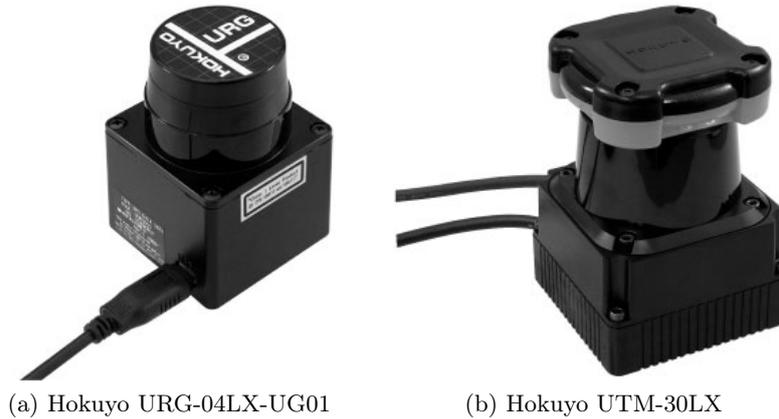


Figura 4.3: Lasers usados no trabalho relatado na dissertação

É ainda de mencionar que o modelo URG-04LX-UG01 tem um campo de visão de 240° , um alcance máximo de 5.6 metros, uma precisão de ± 30 milímetros e funciona a uma frequência de amostragem de 10 hertz de um varrimento completo. Relativamente ao modelo UTM-30LX este tem como características principais o facto de ter um campo de visão de 270° , um alcance máximo de 30 metros, uma precisão de ± 30 milímetros e funciona a uma frequência de amostragem de 40 hertz para o varrimento completo .

Para fazer a reconstrução 3D, os lasers tiveram de ser colocados num suporte que permitisse fixar um deles com um plano de varrimento aproximadamente horizontal e o outro com um plano de varrimento com uma certa inclinação. Esta inclinação foi decidida tomando em atenção o facto da radiação solar na gama do infravermelho perturbar o sistema de deteção, podendo mesmo saturá-lo, mas que mesmo assim permitisse a visualização do estado do espaço à frente do veículo. Na figura 4.4a ilustra-se a primeira versão do suporte para os dois sensores, que permitia uma maior liberdade de posicionamento dos lasers mas, por consequência, sofria de algumas instabilidades, o que inviabilizava uma calibração permanente das transformações extrínsecas. Numa segunda versão (figura 4.4b), foi levado em atenção esse problema, e optou-se pela criação de um suporte mais rígido composto por chapa de alumínio e um perfil bosch.



(a) Suporte para os *lasers* em acrílico



(b) Suporte para os *lasers* em alumínio

Figura 4.4: Lasers usados no trabalho relatado na dissertação

Também foi realizado um reforço ao suporte do computador de bordo para evitar danos de choque ou queda, que se tornaram mais prováveis devido aos movimentos que o veículo irá executar em piso irregular. Na figura 4.5 é apresentado o estado final da zona em questão.

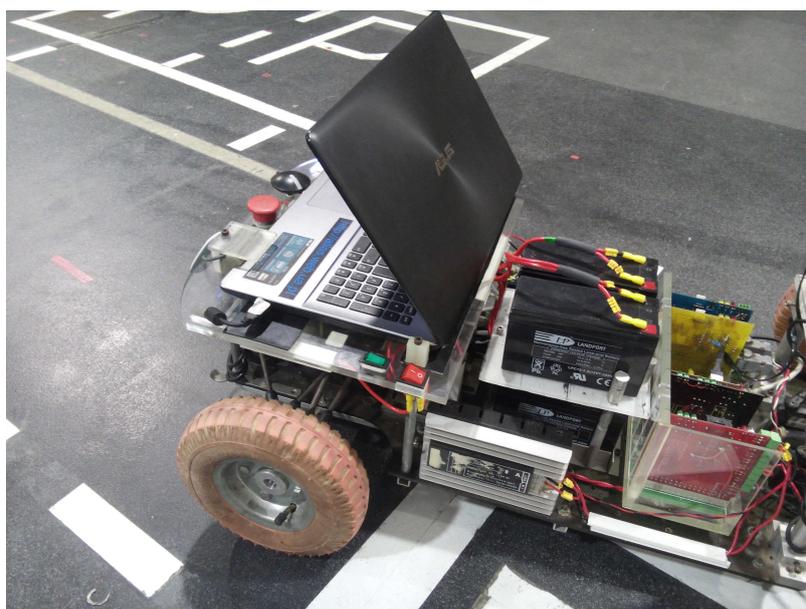


Figura 4.5: Reforço de apoios para a zona de acomodação do computador de bordo

Capítulo 5

Mapeamento e Representação Global

Neste capítulo será demonstrada a sequência de algoritmos usados para o processamento da nuvem 3D inicial até à sua representação num mapa da Google maps. Esta sequência de algoritmos é ilustrada na figura 5.1 e cada um será explicado de uma forma breve .

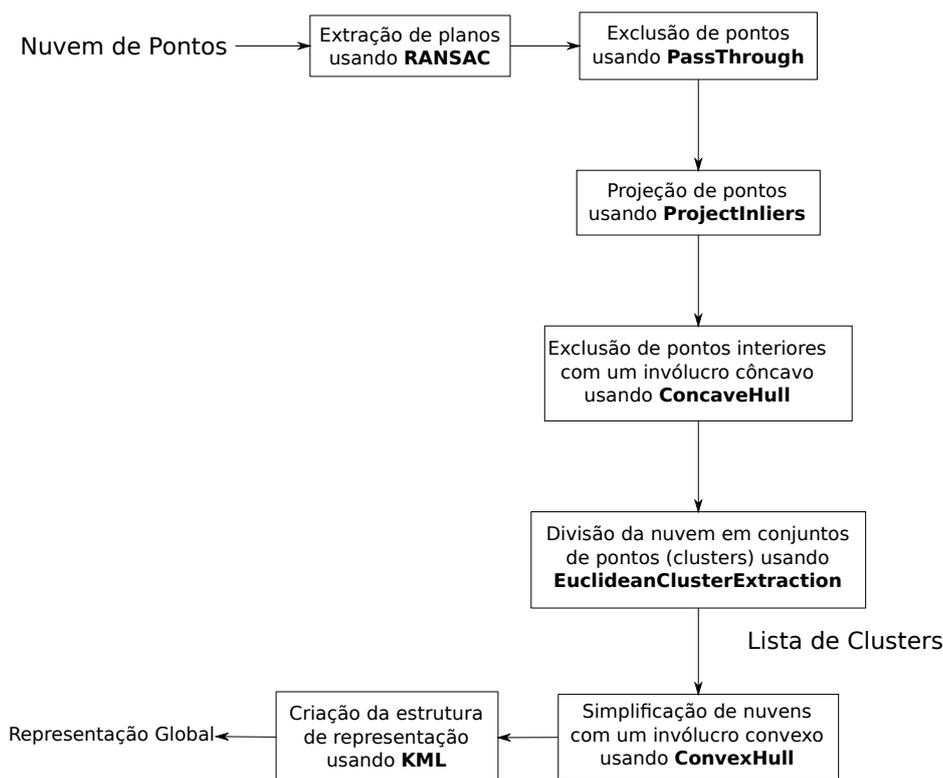


Figura 5.1: Esquema de processamento de uma nuvem de pontos 3D para a sua representação no referencial global

Os algoritmos usados pertencentes à biblioteca da pointcloud:

- *RANSAC* (*RAN*d*om SA*m*ple C*o*nsensus*) - Baseia-se num método iterativo para estimar parâmetros de um modelo matemático de informação previamente recolhida. Inicialmente esta técnica foi apresentada por Fischler e Bolles [25]. Neste caso o modelo

oferecido é o plano e os dados são uma nuvem de pontos.

- *PassThrough* - Filtro que processa pontos de uma nuvem e aplica limites numéricos, p.e., numa dada coordenada.
- *ProjectInlier* - Usando um *sample consensus model* e com base nos coeficientes determinados pelo *RANSAC*, é possível projetar pontos de uma nuvem num plano.
- *EuclideanClusterExtraction* - Sendo fornecido uma nuvem de pontos e com base na distância euclidiana, torna-se possível decompor o espaço em conjuntos separados de pontos (*clusters*).
- *ConcaveHull* e *ConvexHull* - Algoritmos para determinar conjuntos de pontos num espaço em que o “fecho” dos mesmos poderá ser, respetivamente, côncavo ou convexo. Na figura 5.2a é ilustrado com linhas vermelhas um invólucro côncavo de um conjunto de pontos e na figura 5.2b é demonstrado com linhas verdes um invólucro convexo .

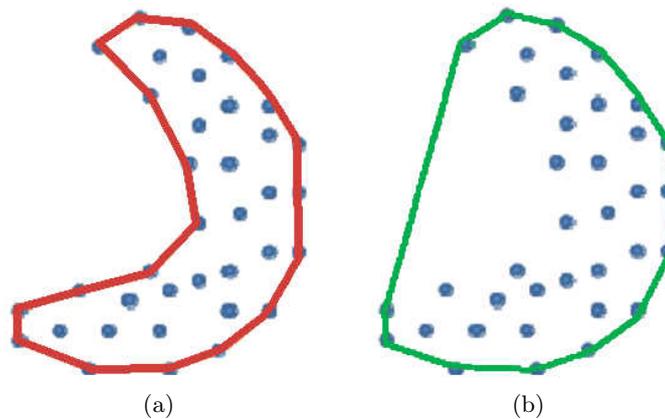


Figura 5.2: Resultados da aplicação de algoritmos *ConcaveHull* e *ConvexHull*. a)As linhas vermelhas representam o resultado da utilização do algoritmo *ConcaveHull* a um conjunto de pontos; b)As linhas verdes representam o resultado da utilização do algoritmo *ConvexHull* a um conjunto de pontos [26]

Keyhole Markup Language (KML) utiliza uma estrutura de tags com elementos e atributos, que se assemelha à estrutura usada pelo XML. No Anexo A é demonstrada uma estrutura exemplo resultante da criação de um polígono, com as coordenadas definidas na tag `<coordinates>`, com um estilo personalizado, nomeadamente, as linhas a vermelho definidas pela tag `<color>` da `<LineStyle>` e um preenchimento a vermelho definido nas tags `<color>` e `<fill>` da `<PolyStyle>`. A figura 5.3 ilustra o resultado da aplicação deste código a um mapa Google.

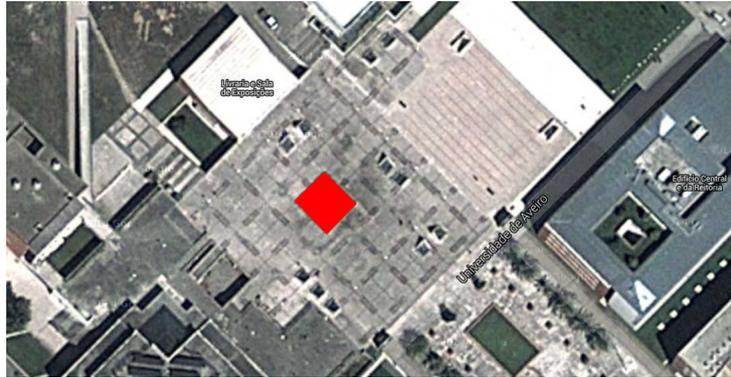


Figura 5.3: Exemplo da aplicação da estrutura KML no Anexo A num mapa da Google.

Aquisição e Processamento

Para criar um mapa local utilizaram-se os dados do laser horizontal (4.3b) para a utilização do pacote *hector_mapping*, usando-o como hodometria na reconstrução da nuvem de pontos. Usaram-se também os dados do laser inclinado, apresentado na figura 4.3a para determinar a distância ao solo e em especial, os obstáculos positivos e negativos. Para realmente realizar a reconstrução 3D do ambiente recorreu-se a um pacote em ROS, denominado de *pc_accumulation* desenvolvido por Pedro Salvado [7]. Este módulo é responsável pela acumulação de nuvens de pontos. A figura 5.5 ilustra uma nuvem de pontos 3D (do ambiente exterior mostrado na figura 5.4) que representa uma acumulação de varrimentos consecutivos obtida com o pacote referido.



Figura 5.4: Ambiente exterior usado para realizar os testes iniciais de utilização do método proposto

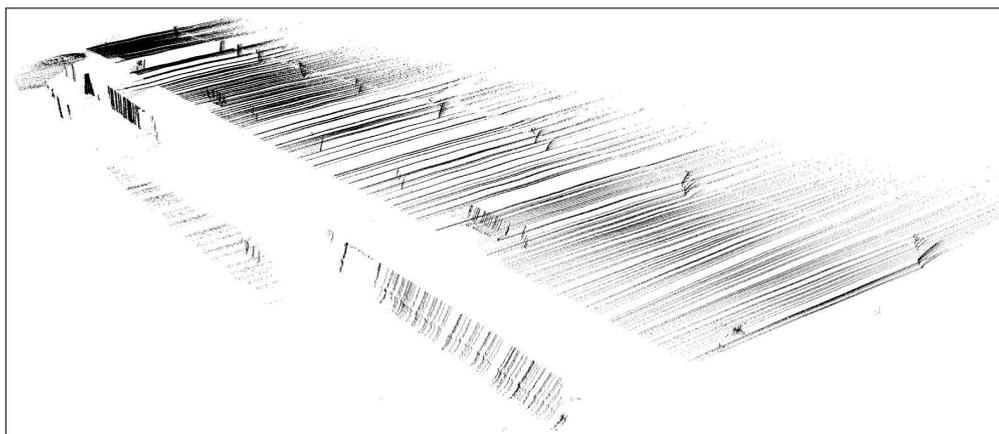


Figura 5.5: Reconstrução 3D do ambiente exterior apresentado na figura 5.4

Após recolhida a nuvem de pontos, é necessário processá-la para realizar a extração e representação dos obstáculos positivos e negativos com diferença de cota fora do limiar permitido (zonas que são de impossível navegação). Para este efeito foi criado um pacote em ROS, denominado de *obstacle_extraction*, que usa funções da biblioteca *pointcloud* para processamento do mapa 3D, e da biblioteca KML, para representação de polígonos num mapa da Google, respetivamente.

Numa primeira fase de processamento, e como o uso do veículo será apenas em pisos praticamente planos, será usado o algoritmo de *RANSAC* presente na biblioteca da *pointcloud* para determinar o plano do chão e determinar os coeficientes desse plano.

Após esta fase, é usado o filtro de *PassThrough* para separar todos os pontos cuja cota diste mais do que 15 cm do plano, assim obtém-se uma nuvem resultante apenas com as zonas onde a passagem é impossibilitada por algum obstáculo, positivo ou negativo (figura 5.6). Foi escolhido este valor devido a ser a limitação de altura que o carro consegue subir ou descer em segurança.

Os coeficientes do plano do chão, anteriormente determinados, tornam possível projetar a nuvem 3D (figura 5.7) no referido plano.

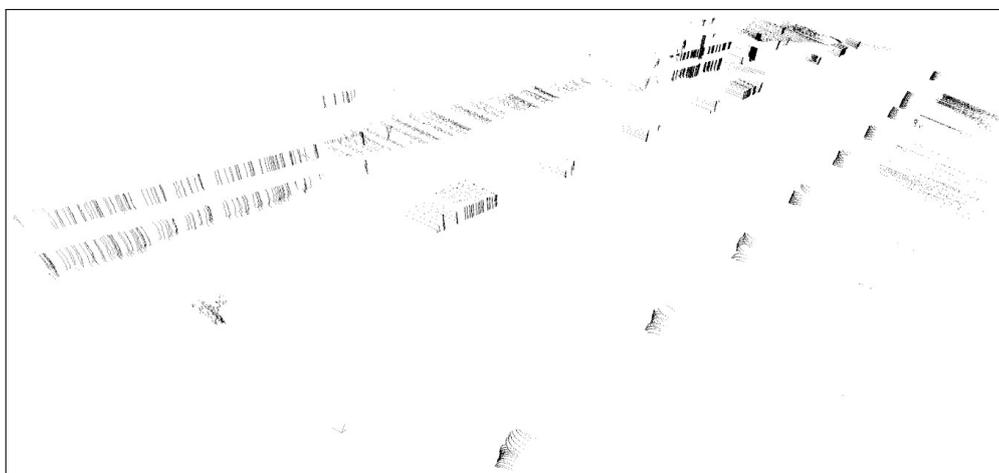


Figura 5.6: Nuvem de pontos 3D com as zonas onde se encontram os obstáculos

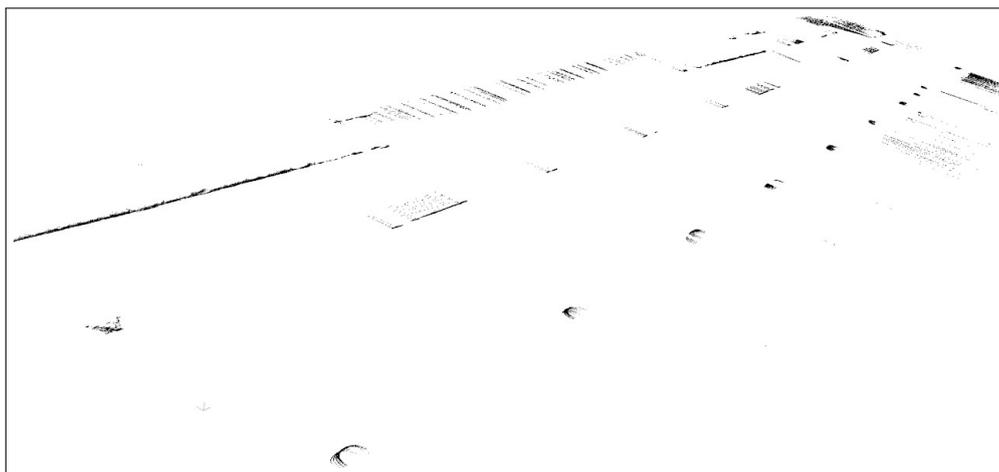
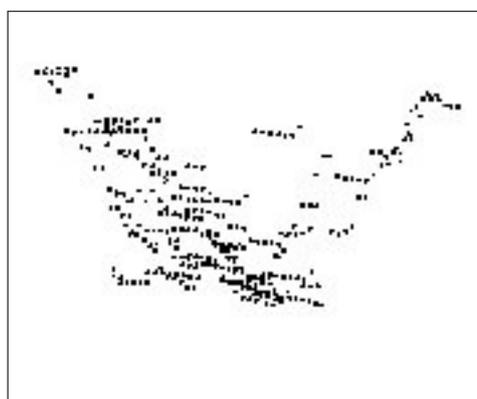
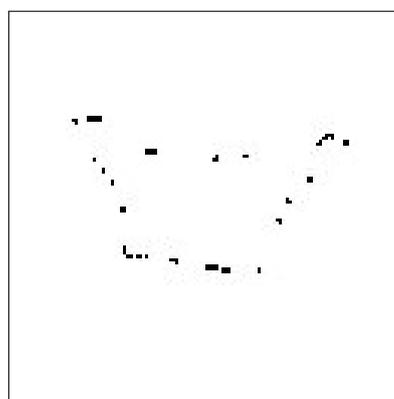


Figura 5.7: Nuvem de pontos projetada das zonas onde se encontram os obstáculos

Por fim, ao aplicar o algoritmo de *concave hull* é possível delimitar de uma forma côncava todos os obstáculos. Na figura 5.8 pode ser visualizado a diferença de situações entre o antes e o depois de aplicar o algoritmo mencionado a um dos obstáculos.



(a) Obstáculo projetado



(b) Resultado do algoritmo de *concave hull* aplicado a obstáculo projetado aplicado

Figura 5.8: Estado inicial e final da aplicação do algoritmo *concave hull* a um obstáculo

Por fim, foi tomada a decisão de realizar os polígonos dos obstáculos de uma forma conservadora, logo foi usado o algoritmo de *ConvexHull* em cada um deles para delimitar as suas extremidades. Para isso acontecer a cada obstáculo, o *ConvexHull* não poderá ser aplicado à nuvem de pontos como ela se encontra, logo foi usado o algoritmo de *EuclideanClusterExtraction* para extrair cada obstáculo para a sua nuvem individual e então suceder à aplicação do *ConvexHull*. A parte da nuvem de pontos resultante deste processo pode ser visualizada na figura 5.9.

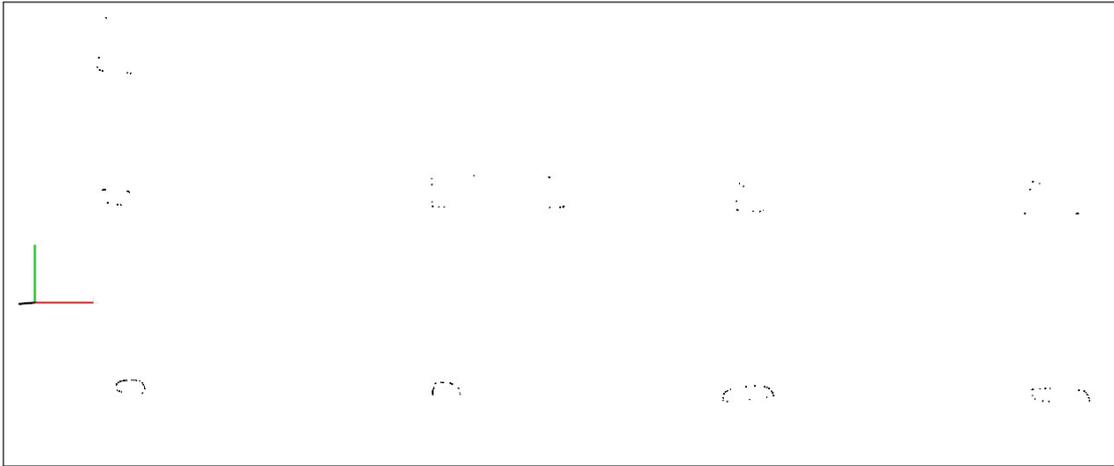


Figura 5.9: Obstáculos delimitados pelas extremidades com a aplicação do algoritmo *ConvexHull* que são os pilares da figura 5.4

Transformação do Mapa Local para Global

Após esta fase, e com o registo geográfico do local onde foi iniciada a reconstrução do ambiente, obtido via recetor GPS, é possível transformar as coordenadas cartesianas do mapa x e y para latitude e longitude. De mencionar que, numa área pequena é válida a utilização de uma transformação com elemento de arco esférico num plano. Com o auxílio da função 5.1 é possível determinar as coordenadas globais iniciais x (Y_Global_Init) e y (X_Global_Init).

$$\begin{cases} Y_Global_Init = LatInit \times EarthR \\ X_Global_Init = LongInit \times EarthR \times \cos(LatInit) \end{cases} \quad (5.1)$$

onde a $LatInit$ é a latitude inicial, a $LongInit$ é a longitude inicial e $EarthR$ é o raio médio terrestre em quilómetros. A latitude e longitude inicial provenientes do módulo recetor GPS estão em graus, sendo necessário realizar a conversão para radianos.

Para cada obstáculo é conhecida a coordenada local na nuvem de pontos, assim, com o auxílio do sistema 5.2, é possível determinar a variação em x e y das coordenadas globais. Note-se no entanto que os eixos do referencial do mapa local e do global não estão coincidentes.

$$\begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) \\ -\sin(\gamma) & \cos(\gamma) \end{bmatrix} \times \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (5.2)$$

onde ΔX e ΔY são as variações das coordenadas X e Y do referencial global, Δx e Δy são as variações das coordenadas x e y do referencial local e γ é o ângulo necessário para o mapa global fique alinhado com mapa mundo. O ângulo foi determinado de forma heurística.

Após determinadas as variações das coordenadas globais dos pontos de cada obstáculo, é necessário aplicar o sistema 5.3 para serem determinadas as latitudes e longitudes respetivas a cada ponto. De notar que estes resultados estarão em radianos e será necessário realizar a conversão para graus.

$$\begin{cases} Lat_Point = (Y_Global_Init + \Delta Y)/EarthR \\ Long_Point = (X_Global_Init + \Delta X)/(EarthR \times \cos(Lat_Point)) \end{cases} \quad (5.3)$$

Nesta fase, e com o auxílio da biblioteca KML, é possível criar um algoritmo que realiza estes cálculos e cria os polígonos em KML. O resultado deste processo pode ser visualizado na figura 5.9

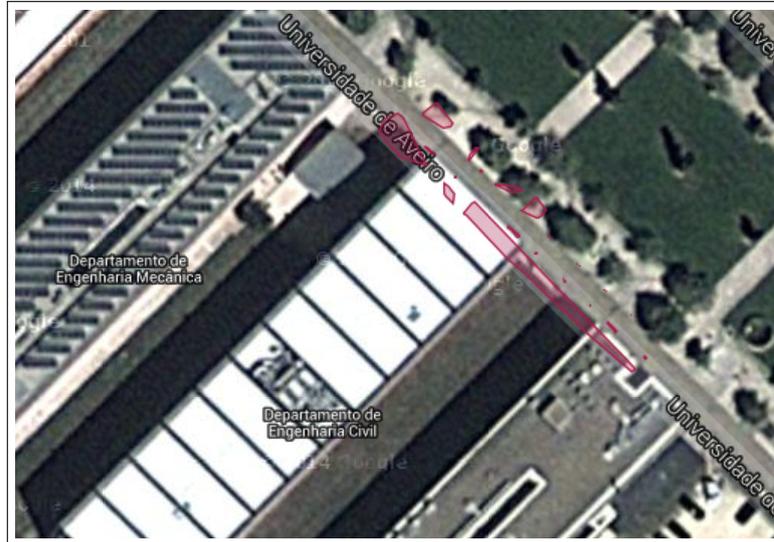


Figura 5.10: Representação da nuvem de pontos 2D num mapa da Google

Capítulo 6

Resultados

Neste capítulo serão apresentados os resultados obtidos, para três zonas do campus da Universidade de Aveiro. Serão apresentadas numa primeira fase as nuvens de pontos recolhidas em três zonas distintas, e numa segunda fase o resultado do pós-processamento que culmina num ficheiro KML. Este pós-processamento encontra-se descrito no capítulo 5. Nas figuras seguintes estão ilustradas a verde as zonas navegáveis e a vermelho as zonas não navegáveis.

6.1 Dados do Sistema Laser

A figura 6.1 mostra a primeira zona de testes que foi escolhida, devido à existência de bastantes obstáculos, de modo a que o pacote de localização usado tivesse referencias suficientes e acumulasse o mínimo de erros. A figura 6.4 mostra a segunda zona de testes que foi escolhida devido ao facto de estar presente um plano inclinado. Por fim, a figura 6.7 demonstra a terceira zona de testes que foi escolhida por proporcionar um ambiente intermédio entre conter obstáculos que poderão servir de referencia, para a localização, e zonas sem referencias. As figuras 6.2 e 6.3, 6.5 e 6.6, 6.8 ilustram respetivamente a primeira, a segunda e a terceira zona de testes com o pós-processamento referenciado.

Na figura 6.1 é possível visualizar um extenso plano do chão, pilares de seção circular e retangular, palmeiras e parte de um edifício. Nas nuvens de pontos processadas, figuras 6.2 e 6.3, é possível verificar que os pilares, troncos e parede do edifício, encontram-se a vermelho e o plano do chão e todas as partes que distam menos de 15 cm dele, encontra-se a verde. É de notar que parte da parede encontra-se a verde devido a esta se intersecar com o plano do chão. Estas duas reconstruções de pontos foram efetuadas quando, respetivamente, se encontravam pessoas a circular e sem pessoas.



Figura 6.1: Primeira zona de testes para realizar reconstrução 3D

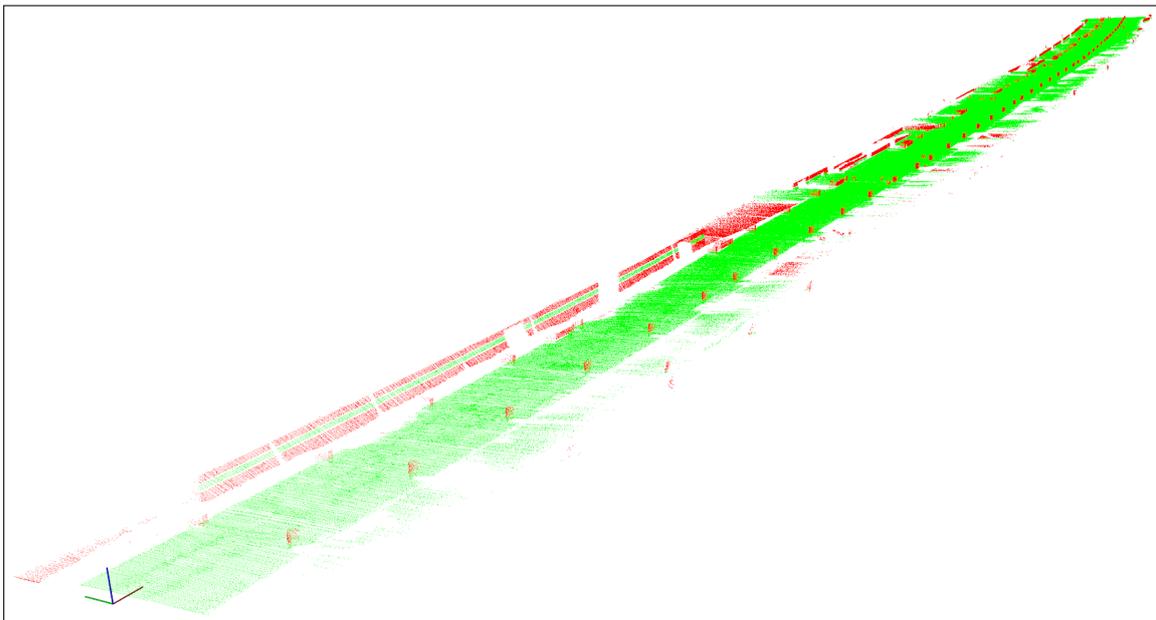


Figura 6.2: Resultado do pós-processamento realizado à primeira zona com pessoas a passar. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis.

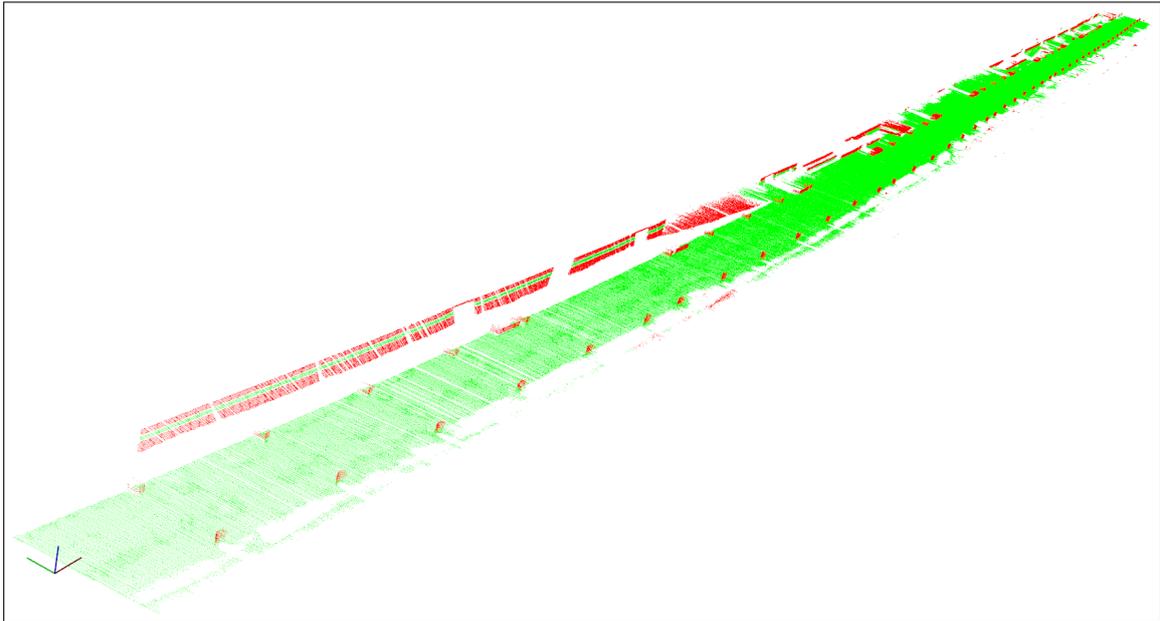


Figura 6.3: Resultado do pós-processamento realizado à sem movimentação de pessoas. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis.

Na figura 6.4 é possível visualizar um curto plano do chão inclinado e parte de um edifício. Nas nuvens de pontos processadas, figuras 6.5 e 6.6, é possível verificar que o plano do chão percorrido encontra-se a verde e o plano do relvado e da zona superior do lado esquerdo encontram-se a vermelho. Estas duas reconstruções de pontos foram efetuadas quando, respetivamente, se encontravam pessoas a circular e sem pessoas.



Figura 6.4: Segunda zona de testes para realizar reconstrução 3D

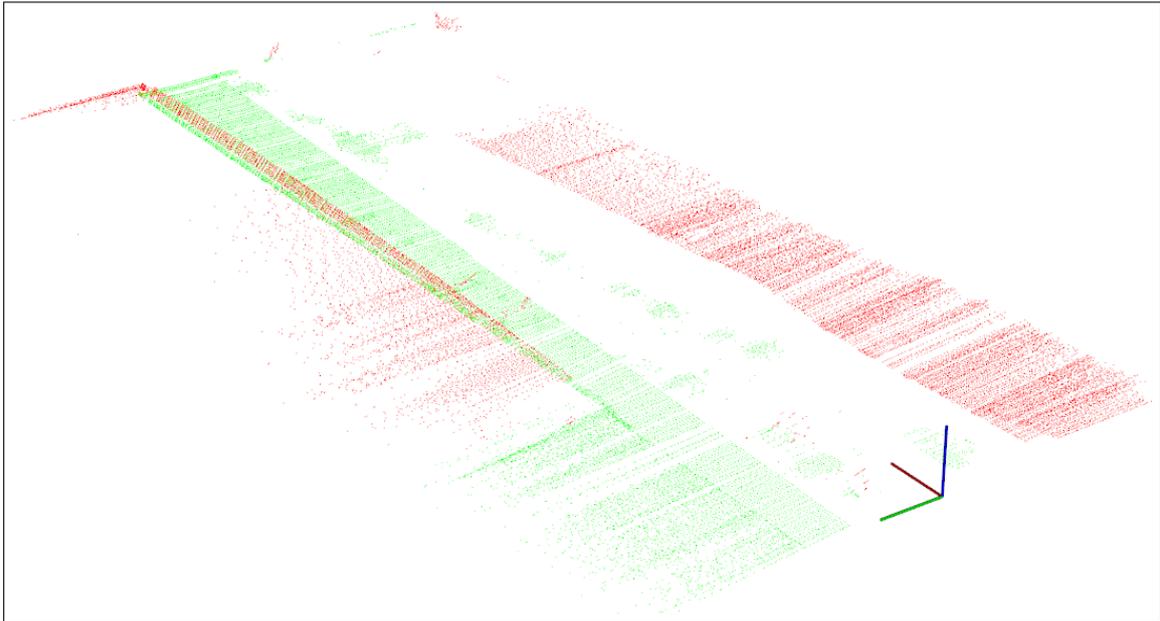


Figura 6.5: Resultado do pós-processamento realizado à segunda zona com a influencia da passagem de pessoas. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis.

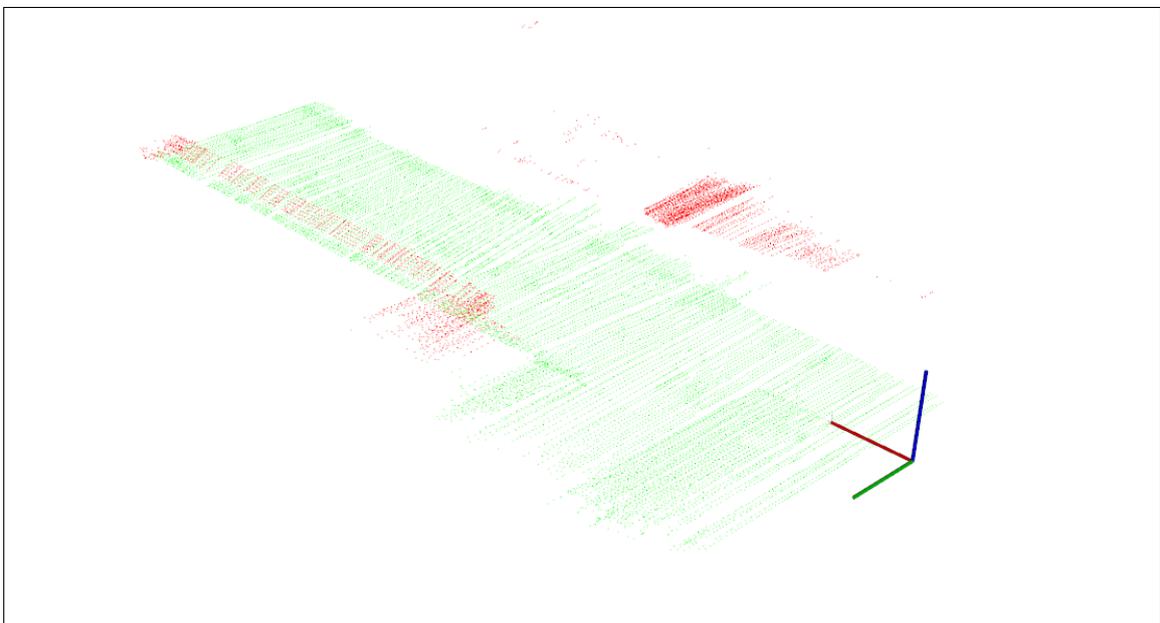


Figura 6.6: Resultado do pós-processamento realizado à segunda zona sem movimentação de pessoas. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis.

Na figura 6.7 é possível visualizar o plano do chão, postes de iluminação, parte de um edifício e grandes obstáculos no meio do espaço. Na nuvem de pontos processada, figura 6.8, é possível verificar que o plano do chão percorrido encontra-se a verde e, os grandes obstáculos

no meio do espaço, parte do edifício e a escadaria, encontram-se a vermelho.



Figura 6.7: Terceira zona de testes para realizar reconstrução 3D

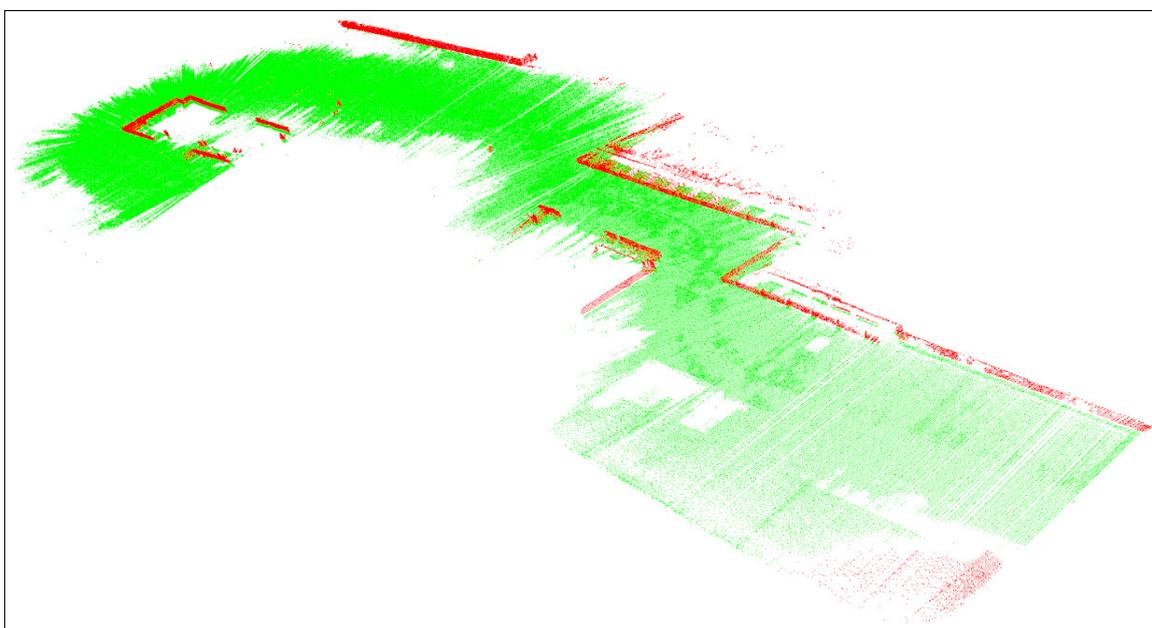


Figura 6.8: Resultado do pós-processamento realizado à terceira zona. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis.

6.2 Mapas de Navegabilidade

Após a fase inicial de pós-processamento das nuvens de pontos e usando dados recolhidos pelo recetor GPS, poderá ser concluída a análise das nuvens de pontos para realizar os mapas de navegabilidade. Os resultados deste processo final podem ser visualizados nas figuras 6.9 e 6.10 para o primeira zona, nas figuras 6.11 e 6.12 para o segunda zona e a figura 6.13 para o terceira zona. As zonas navegáveis são a interceção do alcance do laser e das zonas definidas como não navegáveis.

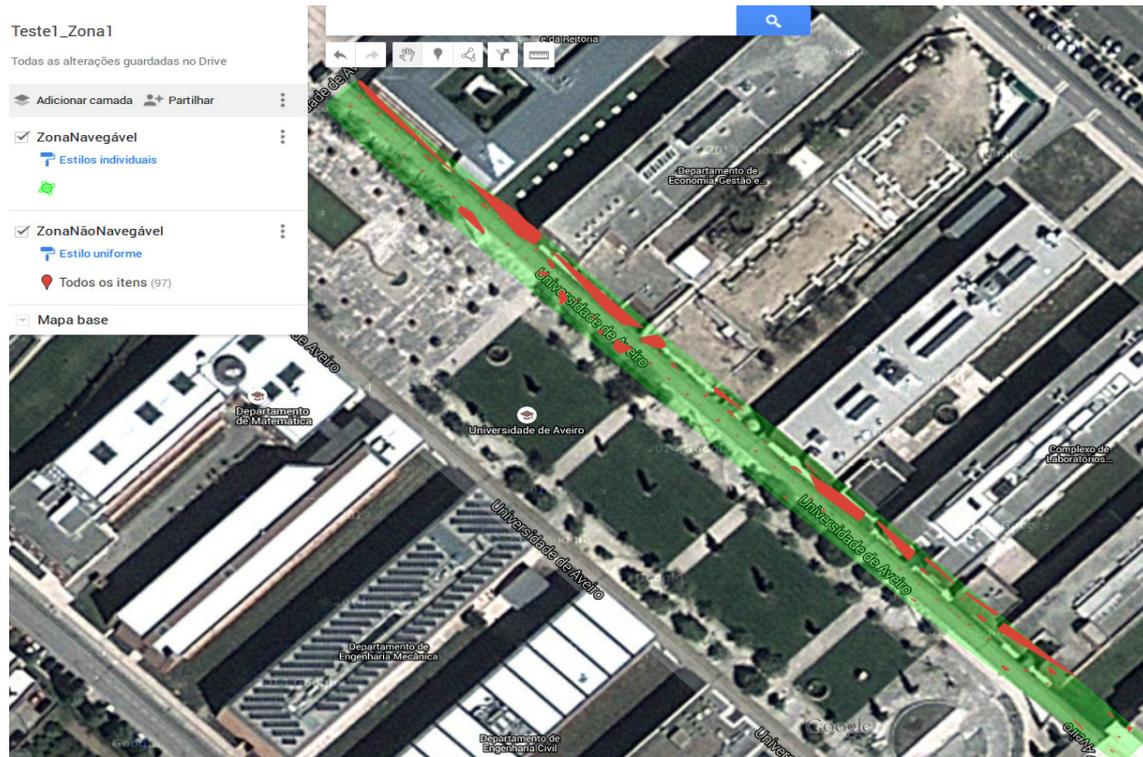


Figura 6.9: Mapa de navegabilidade para a primeira zona com pessoas a passar. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis

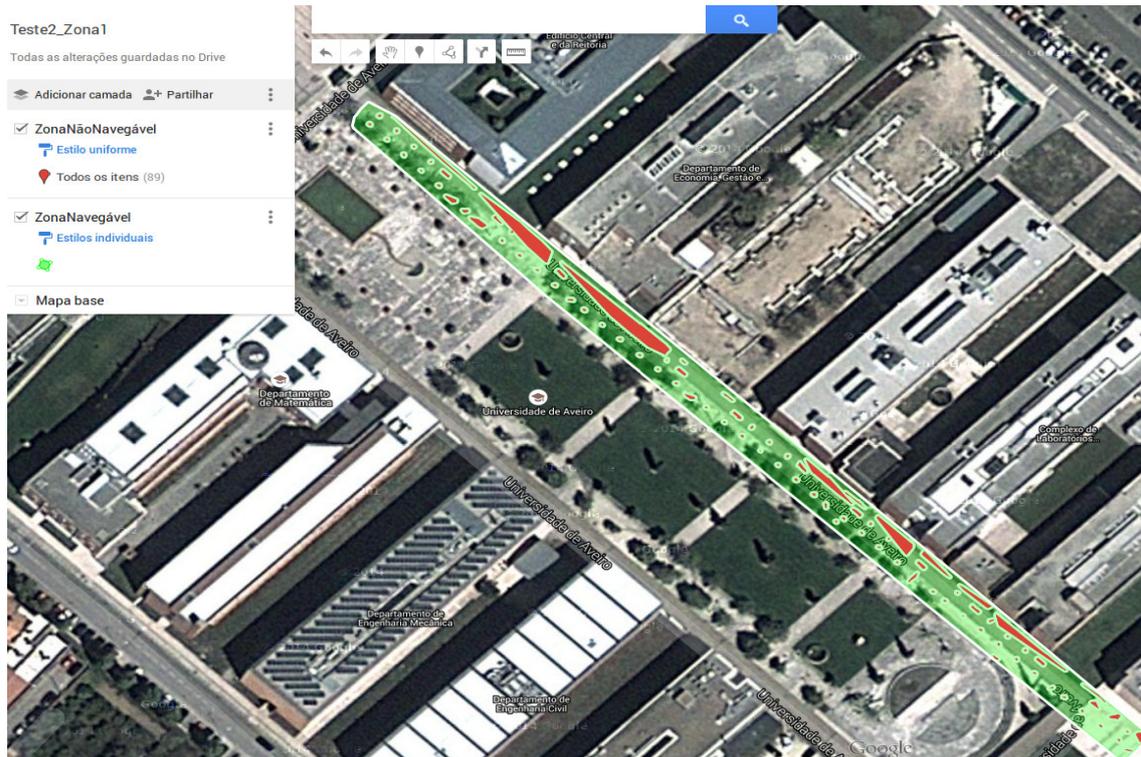


Figura 6.10: Mapa de navegabilidade para a primeira zona sem movimentação de pessoas. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis

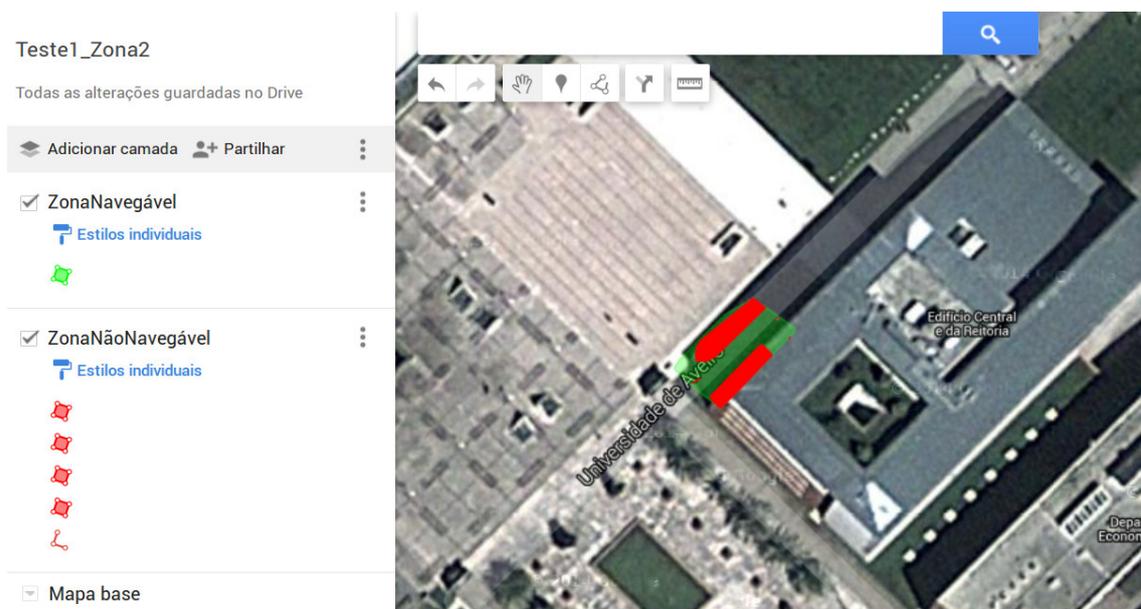


Figura 6.11: Mapa de navegabilidade para a segunda zona com a influencia da passagem de pessoas. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis

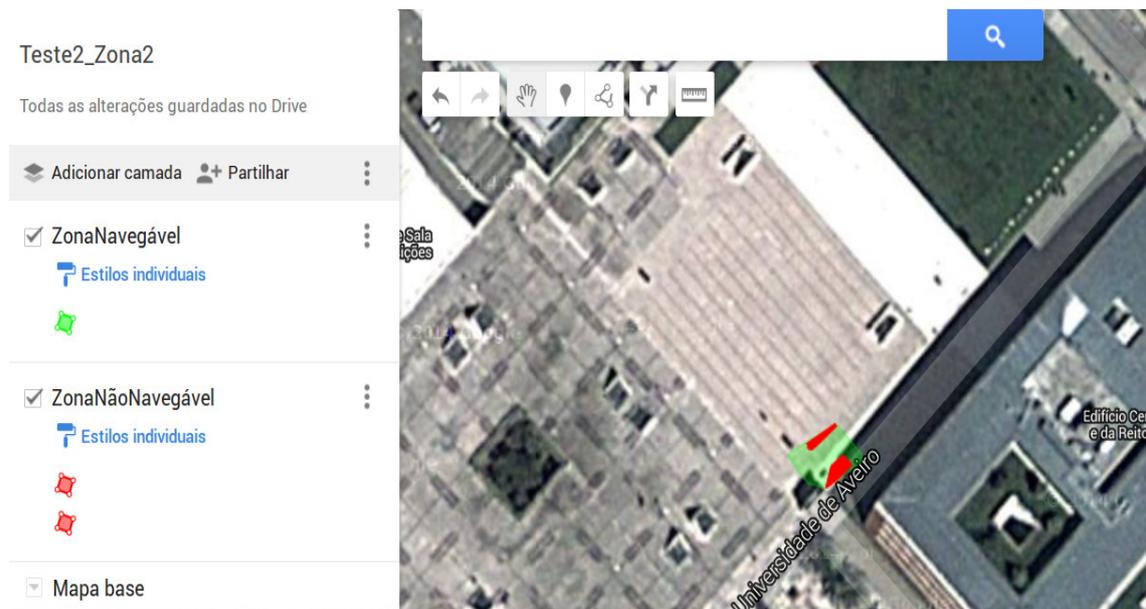


Figura 6.12: Mapa de navegabilidade para a segunda zona. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis

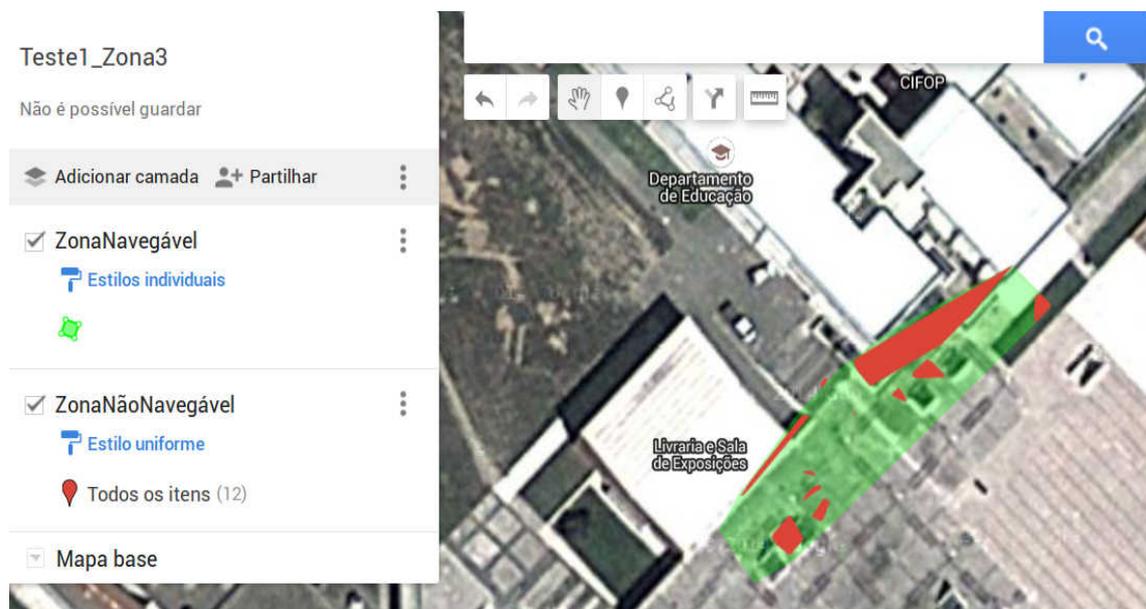


Figura 6.13: Mapa de navegabilidade para a terceira zona. A cor verde representa as zonas navegáveis e a vermelho as zonas não navegáveis

Capítulo 7

Conclusões e Trabalhos Futuros

Este trabalho teve como objetivo criar um sistema de dois *lasers* 2D e um GPS, em que fosse possível realizar uma recolha de dados de ambientes exteriores, nomeadamente do campus da Universidade de Aveiro e criar mapas de navegabilidade. Neste capítulo será feita uma análise sobre o trabalho desenvolvido e os objetivos cumpridos. Por fim, serão enumeradas algumas propostas para trabalhos futuros.

Conclusões

A migração do software para ROS foi realizada com sucesso, resultando num programa que, de forma precisa, lida com o fluxo de informação para o controlo do Atlas 2000. O sistema baseado em ROS combina as vantagens já existentes de ser modular, com o facto de ser possível alterar código que controla tarefas específicas sem que estas influenciem outras funções, mas também com a filosofia de tornar o sistema mais organizado, simplificando, a inserção de novas funções.

A criação do suporte final para o hardware foi realizada com sucesso, sendo composto por peças em alumínio e aço. Tornando-se assim mais vantajoso pelo facto de permitir manter a calibração inicial válida, i.e., sem que vibrações provocadas pelo movimento do veículo em pisos não regulares, alterem a calibração inicial.

O sistema desenvolvido para a recolha de dados, e que usa o pacote *Hector Mapping* como auxílio na localização do veículo, foi realizado com sucesso. Este sistema não tem a capacidade de mapear todos os casos impostos por um ambiente exterior mas, de uma forma genérica, para as zonas apresentadas, foram obtidos resultados visualmente aceitáveis. Isto pode ser verificado na comparação entre os mapas da google com os mapas de navegabilidade resultantes. O facto do sistema desenvolvido não resolver todos os casos é devido à forma simplificada como se obteve a localização do veículo. Foi usado o *Hector_Mapping* sem terem sido usados mais sensores para auxiliar esta localização, nomeadamente para determinar a pose do veículo.

Com os resultados obtidos pode-se concluir que, para a primeira zona de testes em que a localização se realizou com um maior sucesso, a passagem de pessoas influencia de forma negativa e progressiva na acumulação de erros da nuvem de pontos, algo que se visualiza no canto superior direito da figura 6.3, com a curvatura existente na nuvem de pontos. Mesmo assim, verificou-se que para este caso, num espaço de 100 metros percorridos, a precisão do mapa realizado é visualmente aceitável, isto comparando com o mapa da Google respetivo.

Para o segundo e terceiro caso, devido à falta de obstáculos positivos, o pacote

Hector_Mapping não conseguiu, em alguns casos, efetuar o alinhamento correto dos varrimentos laser, sendo que as novas localizações publicadas fiquem incorretas. Neste caso foram obtidos melhores resultados no teste em que existia passagem de pessoas. Mesmo que normalmente a movimentação de pessoas influencie negativamente no pacote usado para a localização, quando existem poucos ou nenhuns obstáculos que sejam usados como referencia para a mesma, a presença de pessoas tem o efeito contrario.

Isto pode ser verificado com o facto de, no segundo caso, as nuvens de pontos resultarem com tamanhos diferentes. O problema é visualizado com maior facilidade nos mapas de navegabilidade destes dois testes, mostrados nas figuras 6.11 e 6.12. Para o terceiro caso, este problema deve-se ao facto de existir uma falha de pontos no centro da imagem, em que no mapa de navegabilidade para este teste (figura 6.13) isto pode ser visualizado com o facto de existirem pontos dispersos no mapa.

Foram encontrados valores com baixa precisão para os dados provenientes do recetor GPS. Este problema pode ser visualizado nas figuras onde os polígonos que representam os obstáculos não estarem corretamente alinhados com a sua posição real no mapa da Google(figura 6.13).

Com isto, pode ser concluído que os objetivos iniciais de criar um sistema que possibilitasse a recolha de informação do ambiente exterior do campus da Universidade de Aveiro, com respetiva avaliação da informação para criar mapas de navegabilidade, foram alcançados. É ainda de mencionar que o sistema aplicado em ambientes com obstáculos tem um funcionamento bastante bom, o mesmo já não se pode dizer para espaços amplos desobstruídos. Este trabalho foi concluído com potencial para ser continuado em trabalhos futuros.

Trabalhos Futuros

Com o objetivo de resolver os problemas encontrados ao longo desta dissertação, ou serem adicionadas novas funcionalidades, são apresentadas algumas propostas de trabalhos futuros.:

- Substituição do servo da direção e da sua placa de controlo;
- Melhorar o sistema de tração do carro para este realizar subidas de maior inclinação;
- Melhorar o sistema de localização usado para realizar as reconstruções de pontos 3D;
- Melhorar o pós processamento das nuvens de pontos para que se realize uma identificação mais refinada de obstáculos positivos e negativos;
- Implementar um sistema de calibração automática para o posicionamento dos lasers;

Referências

- [1] E. da Costa, “Navegação em Ambientes Exteriores do Robô da Série Atlas 2000”, tese de mestrado, Universidade de Aveiro, 2012. endereço: http://lars.mec.ua.pt/public/LAR%20Projects/RobotNavigation/2012_EugenioCosta/Tese_EugenioCosta_FINAL.pdf (acedido em 27/05/2015).
- [2] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler e A. Y. Ng, “ROS: an open-source Robot Operating System”, *ICRA workshop on open source software*, vol. 3, nº 3.2, p. 5, 2009.
- [3] S. Thrun, “Exploring Artificial Intelligence in the New Millennium”, em, G. Lakemeyer e B. Nebel, eds., San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, cap. Robotic Mapping: A Survey, pp. 1–35, ISBN: 1-55860-811-7. endereço: <http://dl.acm.org/citation.cfm?id=779343.779345>.
- [4] T. Bailey, “Mobile Robot Localisation and Mapping in Extensive Outdoor Environments”, tese de doutoramento, ACFR - University of Sydney, 2002.
- [5] A. Nuchter, K. Lingemann, J. Hertzberg e H. Surmann, “6D SLAMdash;3D Mapping Outdoor Environments: Research Articles”, 8-9, vol. 24, Chichester, UK: John Wiley e Sons Ltd., 2007, pp. 699–722. DOI: 10.1002/rob.v24:8/9. endereço: <http://dx.doi.org/10.1002/rob.v24:8/9>.
- [6] M. Bosse e R. Zlot, “Continuous 3D Scan-matching with a Spinning 2D Laser”, em *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, sér. ICRA’09, Kobe, Japan: IEEE Press, 2009, pp. 4244–4251, ISBN: 978-1-4244-2788-8. endereço: <http://dl.acm.org/citation.cfm?id=1703775.1704130>.
- [7] P. Salvado, “Reconstrução Dinâmica de Mapa Local para o Atlascar”, tese de mestrado, Universidade de Aveiro, 2012. endereço: http://lars.mec.ua.pt/public/LAR%20Projects/Perception/2012_PedroSalvado/Pedro_Salvado_Final_Tese.pdf (acedido em 29/05/2015).
- [8] R. Rusu e S. Cousins, “3D is here: Point Cloud Library (PCL)”, em *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, Shanghai, China, 2011, pp. 1–4. DOI: 10.1109/ICRA.2011.5980567.
- [9] J. Pereira, “Estacionamento autónomo usando perceção 3D”, tese de mestrado, Universidade de Aveiro, 2012. endereço: http://lars.mec.ua.pt/public/LAR%20Projects/RobotNavigation/2012_JoelPereira/Tese_escrita/Autonomous_parking_using_3D_perception.pdf (acedido em 05/06/2015).

- [10] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss e W. Burgard, “Octomap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees”, 3, vol. 34, Hingham, MA, USA: Kluwer Academic Publishers, 2013, pp. 189–206. DOI: 10.1007/s10514-012-9321-0. endereço: <http://dx.doi.org/10.1007/s10514-012-9321-0>.
- [11] R. Triebel, P. Pfaff e W. Burgard, “Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing”, em *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 06*, 2006.
- [12] H. Moravec e A. Elfes, “High resolution maps from wide angle sonar”, em *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, 1985, pp. 116–121. DOI: 10.1109/ROBOT.1985.1087316.
- [13] A. Elfes, “Using Occupancy Grids for Mobile Robot Perception and Navigation”, 6, vol. 22, Los Alamitos, CA, USA: IEEE Computer Society Press, 1989, pp. 46–57. DOI: 10.1109/2.30720. endereço: <http://dx.doi.org/10.1109/2.30720>.
- [14] D. Matos, “Deteção do Espaço Navegável para o ATLASCAR usando informação 3D”, tese de mestrado, Universidade de Aveiro, 2013. endereço: http://lars.mec.ua.pt/public/LAR%20Projects/Laser3D/2013_DiogoMatos/Tese%20Escrita/tese_diogomatos_46407.pdf (acedido em 27/05/2015).
- [15] M. Herbert, C. Caillas, E. Krotkov, I. Kweon e T. Kanade, “Terrain Mapping for a Roving Planetary Explorer”, em *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, vol. 2, Scottsdale, AZ, 1989, pp. 997–1002. DOI: 10.1109/ROBOT.1989.100111.
- [16] N. M. E. Association, *NMEA 0183, Standard for Interfacing Marine Electronic Devices*. NMEA National Office, 1995. endereço: <https://books.google.pt/books?id=irVScgAACAAJ>.
- [17] (). CitSci.org - Untitled Document, endereço: http://citsci.org/DH.php?WC=/WS/CitSci/Tutorials_Wisconsin/Tutorial2_Static.html (acedido em 25/06/2015).
- [18] D. Nolan e D. Lang, “Keyhole Markup Language”, English, em *XML and Web Technologies for Data Sciences with R*, sér. Use R!, Springer New York, 2014, pp. 581–618, ISBN: 978-1-4614-7899-7. DOI: 10.1007/978-1-4614-7900-0_17. endereço: http://dx.doi.org/10.1007/978-1-4614-7900-0_17.
- [19] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic Publishers, 1991, ISBN: 079239206X.
- [20] S. Kohlbrecher, O. von Stryk, J. Meyer e U. Klingauf, “A flexible and scalable SLAM system with full 3D motion estimation”, em *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, Kyoto, Japan, 2011, pp. 155–160. DOI: 10.1109/SSRR.2011.6106777.
- [21] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, O. von Stryk e U. Klingauf, “Hector open source modules for autonomous mapping and navigation with rescue robots”, em *RoboCup 2013: Robot World Cup XVII*, sér. Lecture Notes in Computer Science, vol. 8371, Springer, 2014, pp. 624–631.

- [22] B. D. Lucas e T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision”, em *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, sér. IJCAI’81, vol. 2, Vancouver, BC, Canada: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679. endereço: <http://dl.acm.org/citation.cfm?id=1623264.1623280>.
- [23] D. Montemerlo, N. Roy e S. Thrun, “Perspectives on Standardization in Mobile Robot Programming: The Carnegie Mellon Navigation (CARMEN) Toolkit”, em *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, 2003, pp. 2436–2441. DOI: 10.1109/IROS.2003.1249235.
- [24] L. Lamport, “On interprocess communication”, English, *Distributed Computing*, vol. 1, n° 2, pp. 77–85, 1986, ISSN: 0178-2770. DOI: 10.1007/BF01786227. endereço: <http://dx.doi.org/10.1007/BF01786227>.
- [25] M. A. Fischler e R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, 6, vol. 24, New York, NY, USA: ACM, 1981, pp. 381–395. DOI: 10.1145/358669.358692. endereço: <http://doi.acm.org/10.1145/358669.358692>.
- [26] J.-S. Park e S.-J. Oh, “A New Concave Hull Algorithm and Concaveness Measure for n-dimensional Datasets.”, *Journal of Information Science and Engineering*, vol. 28, n° 3, pp. 587–600, 2012. endereço: <http://dblp.uni-trier.de/db/journals/jise/jise28.html#Park012>.

Capítulo 8

Anexos

8.1 Anexo A

Código exemplo de uma estrutura KML

```
<?xml version='1.0' encoding='utf-8'?>
<kml>
  <Document>
    <Style id="Style_1">
      <LineStyle>
        <color>ff0000ff</color>
        <width>1</width>
      </LineStyle>
      <PolyStyle>
        <color>ff0000ff</color>
        <fill>1</fill>
        <outline>1</outline>
      </PolyStyle>
    </Style>
    <Placemark>
      <name>Obstáculo_1</name>
      <styleUrl>#Style_1</styleUrl>
      <Polygon>
        <tessellate>1</tessellate>
        <altitudeMode>relativeToGround</altitudeMode>
        <outerBoundaryIs>
          <LinearRing>
            <coordinates>
              -8.6587393,40.6311288,0
              -8.658632,40.6310433,0
              -8.6585328,40.6311207,0
              -8.6586401,40.6312021,0
              -8.6587393,40.6311287,0
            </coordinates>
          </LinearRing>
        </outerBoundaryIs>
      </Polygon>
    </Placemark>
  </Document>
</kml>
```