Universidade de Aveiro Departamento de Engenharia Mecânica 2013

Diogo Artur Fonseca Matos Deteção do Espaço Navegável para o ATLASCAR usando informação 3D



Universidade de Aveiro Departamento de Engenharia Mecânica 2013

Diogo Artur Fonseca Matos

Deteção do Espaço Navegável para o ATLASCAR usando informação 3D

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

O júri / The jury

Presidente / President	Prof. Doutor Jorge Augusto Fernandes Ferreira Professor Auxiliar da Universidade de Aveiro
Vogais / Committee	Prof. Doutor Paulo Miguel de Jesus Dias Professor Auxiliar da Universidade de Aveiro
	Prof. Doutor Vítor Manuel Ferreira dos Santos Professor Associado da Universidade de Aveiro (orientador)

Agradecimentos / Acknowledgements

Gostava de deixar aqui o meu agradecimento ao Professor Doutor Vítor Santos não só pela orientação e motivação neste projeto, mas também pela dedicação e entusiamo que coloca nas unidades curriculares que leciona, às quais tive o privilégio de assistir.

Agradeço ao Jorge Almeida e ao Miguel Oliveira pela disponibilidade e apoio ao desenvolvimento deste projeto.

Um especial agradecimento ao Ricardo Pascoal pela troca de opiniões e por todo o tempo depositado neste projeto, a sua ajuda foi essencial para que este projeto fosse concluído. Obrigado Ricardo.

Agradeço a todos os meus amigos e colegas de curso por terem me acompanhado ao longo destes 5 anos.

À minha namorada por todo o apoio e paciência não só durante o desenvolvimento deste projeto, mas por estar sempre presente nos momentos mais importantes durante estes 5 anos.

Finalmente, um agradecimento aos meus pais por todo o seu incansável apoio e dedicação durante a minha formação académica. Obrigado!

Palavras-chave AtlasCar; microcontrolador; percepção 3D; laser 2D; laser 3D; nuvem de pontos; navegabilidade Resumo Este trabalho descreve um método para a deteção do espaço navegável em torno de um veículo a partir da perceção de dados 3D. Estes dados 3D são obtidos a partir de um laser planar 2D em rotação continua. Na primeira parte, o trabalho contemplou uma fase de refinamento e requalificação do laser 3D para melhorar a qualidade dos dados em virtude do movimento rotacional da unidade laser planar; após essa etapa foi desenvolvido software que acumula os diversos varrimentos do laser 2D e constrói uma nuvem de pontos representativa do ambiente. Na segunda fase do trabalho foi desenvolvido software para deteção do espaço navegável, levando em linha de conta a elevação do terreno bem como as normais médias. O espaço foi dividido numa grelha bidimensional com células que contêm diversas propriedades que permitem calcular a sua acessibilidade média; o resultado é uma espécie de mapas de elevação estendidos que podem ser usados para calcular a navegabilidade de uma qualquer trajetória, e que foram designados mapas de acessibilidade. Os resultados indicam uma clara melhoria na qualidade dos dados 3D obtidos com a unidade laser 2D e foram também feitos estudos e análises da influência dos parâmetros dos algoritmos no cálculo dos mapas de acessibilidade.

Keywords AtlasCar; microcontroller; 3D perception; 2D laser; 3D laser; point cloud; navigability Abstract This work presents a method for detecting the navigability of the environment using 3D data of the vehicle's surroundings. In this work the 3D data is achieved by a 2D laser on continuous rotation. Initially the hardware of the 3D acquisition system was refined in order to achieve better data quality due to the rotational movement of the planar laser unit; afterwards, a software module to store the different 2D scans and to create a point cloud of the environment was developed. In the second stage, a software module for detecting the navigability of the environment, using the elevation of the terrain as well as the normal average was developed. The space was divided into a two-dimensional grid with cells that contain several properties in order to calculate the accessibility average, the result is a sort of extended elevation maps that may be used to calculate navigability of any trajectory, that were designated accessibility maps. The results indicate a clear improvement in the quality of the 3D data obtained with the planar laser unit and were also made studies and analyzes of the influence algorithms' parameters' for the calculation of accessibility maps.

Conteúdo

Co	onteí	ido		i
\mathbf{Li}	sta d	le Tabe	elas	iii
Li	sta d	le Figu	ras	\mathbf{v}
1	Intr	odução	0	1
	1.1	Projet	o ATLAS	1
	1.2	Aprese	entação do problema	2
	1.3	Objeti	vos	3
	1.4	Robot	Operating System	3
2	Esta	ado de	arte	5
	2.1	Sistem	as de aquisição 3D	5
		2.1.1	Velodyne HDL-64E LiDAR	5
		2.1.2	Câmaras 3D	6
		2.1.3	Câmaras estéreo	6
	2.2	Sensor	es LRF	6
	2.3	Mapas	para navegação	7
3	\mathbf{Sist}	ema de	e aquisição 3D	11
	3.1	Descri	ção do sensor	11
	3.2	Propri	edades da configuração	12
	3.3	Sincro	nismo do sistema	12
		3.3.1	Configuração do laser	13
		3.3.2	Implementação do pulso de sincronismo	13
	3.4	Recons	strução tridimensional	14
		3.4.1	Sistemas de coordenadas do AtlasCar	14
		3.4.2	Integração com o ROS	15
		3.4.3	Construção da nuvem de pontos	16
4	Maj	pas de	acessibilidade e de navegabilidade	23
	4.1	Pré-pr	ocessamento da nuvem de pontos	23
	4.2	Grelha	ι de ocupação	25
		4.2.1	Inicialização da grelha	25
		4.2.2	Cálculo de propriedades das células	27
		4.2.3	Interpolação de propriedades das células	28

		4.2.4 Cálculo da disparidade das células	29
	4.3	Mapas de acessibilidade	30
	4.4	Mapas de navegabilidade	32
5	Res	ultados e discussão	33
	5.1	Resultado das alterações no sistema de aquisição 3D	33
	5.2	Resultados na criação dos mapas de acessibilidade	40
6	Con	nclusões e trabalho futuro	55
	6.1	Sistema de aquisição 3D	55
	6.2	Deteção do espaço navegável	55
	6.3	Trabalho futuro	56
Re	eferê	ncias	57
A	Circ	cuito microcontrolador	59
в	Map	pas de elevação	63
С	C Launch file módulo navigability_map 6		

Lista de Tabelas

5.1	Resultados de análise de uma superfície planar horizontal. Nota - Sensor	
	L1, Laser 3D sem alterações; Sensor L2, Laser Velodyne; Sensor L3, Laser	
	3D com correção do <i>scan</i> utilizando os <i>time stamps</i> de ϕ	34
5.2	Resultados de análise de superfície planares horizontais, utilizando dife-	
	rentes metodologias de construção da nuvem de pontos utilizando os dados	
	do laser 3D. Nota - Superfície $S1$ representada a preto na figura 5.6 (a);	
	Superfície $S2$ representada a preto na figura 5.6 (b); Metodologia A, Laser	
	3D com correção do scan utilizando os time stamps de ϕ ; Metodologia B ,	
	Com correção do scan utilizando o time stamp de inicio do scan; Metodo-	
	logia C, Sem correção do scan. \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots	34
5.3	Resultado da eficiência do algoritmo variando as dimensões de células qua-	
	dradas, $Sx = Sy$	46
5.4	Resultado da eficiência do algoritmo variando as dimensões de células não	
	quadradas, $Sx \neq Sy$	46
5.5	Resultado da eficiência do algoritmo variando o raio, R, de procura da	
	vizinhança para o cálculo das normais da nuvem de pontos. Nota - Raio,	
	R, em metros \ldots	47
5.6	Resultado da eficiência do algoritmo variando a constante normalizadora,	
	σ_0 , do valor de confiança, ν	47
5.7	Tempos de processamento do algoritmo na construção dos mapas de aces-	
	sibilidade	48

Lista de Figuras

1.1	Robôs ATLAS para condução autónoma. [2]	1
1.2	$AtlasCar. [1] \dots \dots \dots \dots \dots \dots \dots \dots \dots $	2
1.3	Exemplo de troca de mensagens entre três nodos. a) troca de mensagens, nodos sinalizados como elipses e tópicos como quadrados; b) tráfego das	
	mensagens para cada tópico, escala horizontal em segundos. [6]	3
2.1	HDL-64E LiDAR	5
2.2	Nuvem de pontos adquirida pelo HDL-64E LiDAR num cruzamento	6
2.3	Exemplos de sensores que usam técnica de ToF. a) Panasonic D-Imager [10]; b) SwissRanger SR4000 [11].	6
2.4	Laser 3D. a) LRF - Laser <i>Sick LMS</i> 200; b) sistema mecânico do laser rotativo [3].	7
2.5	Sensor Laser Hokuvo. [2]	7
2.6	a) Dados de uma ponte gravada por um robô móvel usando um laser SICK	•
-	LMS; b) mapa de elevação obtido utilizando os dados da figura 2.6 (a). [14]	8
2.7	Representação dos pontos da figura 2.6 (b) de acordo com a sua classifi-	
	cação.	
	Tons de azul indicam zonas adquiridas acima do plano de aquisição do	
	sensor; Tons de vermelho indicam zonas verticais; Tons de verde indicam	
	zonas transponiveis; Zonas nao transponiveis nao se encontram represen-	0
28	Extensão do mapo do elovação do figuro 2.6 [14]	0
2.0 2.0	a) Mana MIS da ambiente representada na figura 2.6 (a); h) regultada da	9
2.9	classificação para o mana MLS. Tons de amarelo representam superfícies	
	navegáveis, enquanto que tons de azul representam as não navegáveis e	
	tons de vermelho representam as verticais. [13]	9
2.10	Exemplo mapa criado usando SLAM. [16]	10
3.1	Princípio de funcionamento do laser Sick LMS 200	11
3.2	Coordenadas espaciais. [3]	12
3.3	Bloco I do comando 77h para configuração do laser. [18]	13
3.4	Sistemas de coordenadas presentes no AtlasCar. [6]	15
3.5	Comunicação dos módulos do laser 3D	16
3.6	Comunicação dos módulos do laser 3D incorporando o módulo de cons- trução de nuvem de pontos.	18

3.7	a) Tráfego de mensagens em situação normal; b) tráfego de mensagens na situação em que existe irregularidade nas comunicações. Nota - Escala horizontal em segundos.	19
3.8	Ilustração de acumulação de dados e seu processamento	20
3.9	a) Imagem do ambiente; b) nuvem de pontos criada utilizando os time stamps dos ângulos ϕ	22
4.1	Nuvem de pontos filtrada da imagem 3.9 (a).	24
4.2	a) Cálculo das normais utilizando K=5; b) cálculo das normais utilizando K=20	24
4.3	a) Cálculo das normais utilizando $R=0.1$; b) cálculo das normais utilizando $R=0.4$.	25
4.4	Vizinhança N8 para uma célula	26
4.5	Mapa de elevação com representação da normal média da nuvem de pontos da figura 4.1.	28
4.6	Resultado da aplicação da interpolação de propriedades para o mapa de elevação da figura 4.5.	29
4.7	Acessibilidade das células em coordenada z, ϖ_z ; Nota - Cor azul corresponde a acessibilidade 1 e cor vermelha corresponde a acessibilidade 0.	30
4.8	a) Acessibilidade das células em α , ϖ_{α} ; b) acessibilidade das células em β , ϖ_{β} ; c) acessibilidade das células em γ , ϖ_{γ} . Nota - Cor azul corresponde a acessibilidade 1 e cor vermelha corresponde a acessibilidade 0	31
4.9	Mapa resultante do produto das acessibilidades, ϖ_z , ϖ_α , $\varpi_\beta \in \varpi_\gamma$. Nota - Cor azul corresponde a acessibilidade 1 e cor vermelha corresponde a acessibilidade 0	31
4.10	Possíveis trajetórias a executar. [25]	32
5.1	a) Imagem do ambiente; b) nuvem de pontos adquirida com o laser 3D antes das alterações	35
5.2	Nuvem de pontos adquirida com o laser 3D antes das alterações, superfície planar (3583 pontos) representada a cor preta	36
5.3	Nuvem de pontos adquirida com o sensor Velodyne, superfície planar (1960 pontos) representada a cor preta.	36
5.4	Nuvem de pontos adquirida com o laser 3D com correção do <i>scan</i> utilizando os <i>time stamps</i> de ϕ , superfície planar (3790 pontos) representada a cor preta.	37
5.5	Função densidade de probabilidade obtida para: Laser 3D antes de efetu- adas as alterações - cor vermelha; Sensor Velodyne - cor preta; Laser 3D com correção do scan utilizando os time stamps de ϕ - cor azul	37
5.6	Nuvens de pontos adquiridas com o laser 3D representando a cor preta as superfícies planares: a) S1 (3790 pontos); b) S2 (2422 pontos)	38

5.7	Função densidade de probabilidade obtida utilizando nuvens de pontos do laser 3D construídas com diferentes metodologias, para as superfície pla- nares: a) S1 ; b) S2 . Nota - Nuvem de pontos construída: Com correção	
	do scan utilizando o time stamp de ϕ - cor preta; Com correção do scan	
	utilizando o $time\ stamp$ de início do $scan$ - cor vermelha; Sem correção do	
50	scan - cor azul.	39
5.8	a) Imagem do ambiente em que existe um obstaculo negativo (buraco); b) imagem do ambiente em que existe uma subida com passeio; c) imagem do ambiente em que existe uma descida; d) imagem do ambiente em que	
	eviste um obstáculo positivo (poste de sinalização)	/11
5.9	a) Nuvem de pontos da cena A : b) representação de um polígono a cor	71
	verde de uma zona acessível, a cor vermelha de uma zona inacessível (bu-	
	raco)	42
5.10	a) Nuvem de pontos da cena B ; b) vista lateral da nuvem pontos repre-	
	sentada na figura 5.10 (a) ; c) representação de um polígono a cor verde $-$	
	de uma zona acessível, a cor vermelha de uma zona inacessível (passeio).	43
5.11	a) Nuvem de pontos da cena C ; b) representação de um polígono a cor	
	verde de uma zona acessível, a cor vermelha de uma zona inacessível (ár-	
5 19	vore)	44
5.12	a) Nuvem de pontos da cena D ; b) representação de um polígono a cor verde de uma zona acessível, a cor vermelha de uma zona inacessível (poste	
	de sinalização).	45
5.13	Ilustração da influência do uso de células de reduzidas e elevadas dimen-	
	soes, na construção do mapa de acessibilidade da nuvem de pontos da cena \mathbf{D} \mathbf{D} \mathbf{D} \mathbf{C} \mathbf{C} \mathbf{D} \mathbf{C} \mathbf{D} \mathbf{D} \mathbf{C} \mathbf{D} D	10
F 14	D . a) dimensoes de celula 0.1×0.1 m; b) dimensoes de celula 0.8×0.8 m.	49
0.14	Mapa de acessionidade da nuvem de pontos da cena \mathbf{A} ; % zona acessivel detetada -70.25 ; % zona inacessível detetada -66.66	50
5 15	Mapa de acessibilidade da nuvem de pontos da cena \mathbf{B} : % zona acessível	50
0.10	detetada - 81 51: % zona inacessível detetada - 100	51
5.16	Mapa de acessibilidade da nuvem de pontos da cena C: % zona acessível	01
	detetada - 60.22; % zona inacessível detetada - 100.	52
5.17	Mapa de acessibilidade da nuvem de pontos da cena \mathbf{D} ; % zona acessível	
	detetada - 92.35; % zona inacessível detetada - 100. $\ldots \ldots \ldots \ldots \ldots$	53
R 1	Mana de elevação da cena A	64
B.2	Mapa de elevação da cena B	65
B.3	Mapa de elevação da cena \mathbf{C}	66
B.4	Mapa de elevação da cena D	67

Capítulo 1

Introdução

Esta dissertação enquadra-se nos sistemas de perceção avançada em curso no Laboratório de Automação e Robótica, em especial nos sistemas afetos ao projeto *ATLAS* e mais em particular ao *AtlasCar*.

1.1 Projeto ATLAS

O projeto ATLAS foi criado pelo grupo de Automação e Robótica do Departamento de Engenharia Mecânica da Universidade de Aveiro para a participação em competições de robótica [1]. Ao longo dos anos, o projeto ATLAS foi colecionando prémios em competições de condução autónoma, com o desenvolvimento de pequenos rôbos - figura 1.1 - .



Figura 1.1: Robôs ATLAS para condução autónoma. [2]

Com toda a experiência adquirida no desenvolvimento destes pequenos rôbos, o grupo ATLAS decidiu iniciar um novo projeto cujo principal objetivo é desenvolver novas soluções no processamento avançado e sistemas ativos de segurança para implementação na área automóvel. Com o intuito de alcançar os seus objetivos, o projeto ATLAS desenvolveu um protótipo à escala real, o AtlasCar - figura 1.2 - . Este veículo encontra-se equipado com diversos sensores, dos quais podemos destacar, câmaras, sensores laser 2D e 3D, GPS, IMU, entre outros. [2]



Figura 1.2: AtlasCar. [1]

1.2 Apresentação do problema

A necessidade da representação e reconhecimento do ambiente que rodeia um veículo é fundamental para o apoio à condução. É por essa razão que, hoje em dia, os automóveis são equipados com diversos dispositivos que permitem uma condução cada vez mais segura.

Para aplicações de condução autónoma ou assistida , como o *AtlasCar*, possuir informação sobre o ambiente é vital para o seu funcionamento. Este trabalho, foca-se sobre o reconhecimento do ambiente com base em informação 3D, obtida através do laser 3D desenvolvido pelo Laboratório de Automação e Robótica. Este mecanismo apresenta algumas limitações, das quais se pode destacar o sincronismo entre os dados 2D e a medição do ângulo externo [3].

O trabalho a desenvolver vai no sentido de incorporar os dados adquiridos pelo laser 3D na arquitetura implementada no *AtlasCar* e posteriormente utilizar a nuvem de pontos obtida por esse sensor para fazer o reconhecimento do espaço navegável.

A deteção do espaço navegável consiste em identificar as zonas onde o veículo pode circular, bem como aquelas onde o veículo não o pode fazer. Existe um grau de incerteza associado a qualquer das situações e ainda zonas para as quais nada se pode concluir. Limites físicos do espaço navegável:

- Identificação da estrada (plana, subida, descida);
- Identificação de passeios;
- Identificação de obstáculos positivos ou negativos;
- Identificação de barreiras.

1.3 Objetivos

Numa primeira etapa, pretende-se refinar o *hardware* existente para o sistema de aquisição laser 3D, para melhorar o desempenho no que toca ao sincronismo entre os dados 2D e a medição do ângulo externo. Posteriormente, irá ser realizado um módulo de *software* que permita gerar nuvens de pontos com base nos dados do sensor e fazer a integração do sistema na arquitetura do AtlasCar.

Numa segunda fase, pretende-se desenvolver um módulo de *software* para detetar os limites físicos do espaço navegável (estrada, passeio, bermas, etc) e de obstáculos com base em informação 3D.

No final, deve-se alcançar uma melhoria na qualidade dos dados do laser 3D, e uma aplicação capaz de fazer o reconhecimento do espaço navegável utilizando informação 3D.

1.4 Robot Operating System

O projeto AtlasCar está desenvolvido sobre a plataforma ROS, portanto para que o trabalho desenvolvido nesta dissertação seja compatível, e facilmente integrável no projeto AtlasCar, deve ser também desenvolvido nessa mesma plataforma.

O ROS é uma plataforma de desenvolvimento com inúmeras bibliotecas para desenvolvimento de *software* e para comunicação com *hardware* [4]. Esta plataforma foi desenvolvida especialmente para projetos de grandes dimensões, e a sua arquitetura modular permite que um projeto bastante complexo possa estar implementado em pequenos módulos, cada um com uma aplicação específica [5].

A comunicação entre módulos, ou nodos, é realizada através de mensagens publicadas num tópico; esse tópico pode depois ser subscrito por outro módulo e desta forma aceder à informação contida na mensagem, na figura 1.3 é possível visualizar um exemplo de troca de mensagens entre nodos.

Além disso o ROS tem a particularidade de poder gravar as mensagens publicadas em disco, para que mais tarde possam voltar a ser usados, sendo bastante útil para se conseguir obter dados experimentais e mais tarde trabalhar em laboratório.



Figura 1.3: Exemplo de troca de mensagens entre três nodos. a) troca de mensagens, nodos sinalizados como elipses e tópicos como quadrados; b) tráfego das mensagens para cada tópico, escala horizontal em segundos. [6]

Outra particularidade do ROS é a cadeia cinemática estabelecida entre referenciais de vários sensores. Existe um módulo que permite detetar as transformações entre os referenciais ao longo do tempo, e é portanto possível transformar os dados adquiridos por um determinado sensor para um outro referencial sem ser necessário o utilizador definir matrizes de transformação.

Capítulo 2

Estado de arte

Os veículos autónomos recorrem a várias tecnologias para obter uma perceção do ambiente que os rodeiam, como por exemplo, radar, visão artificial, LiDAR e GPS.

2.1 Sistemas de aquisição 3D

Hoje em dia a reconstrução do ambiente com base em informação 3D tem sido um foco de desenvolvimento, sendo aplicada a diversas áreas de investigação como navegação, mapeamento, localização, entre outros.

Existem várias tecnologias capazes de reconstruir o ambiente, no entanto existem algumas que melhor se adaptam às necessidades de um veículo autónomo. Por exemplo, câmaras 3D como a Kinect que usam a projecção de uma luz estruturada, não irão ter um bom desempenho no exterior devido à luz solar [7].

2.1.1 Velodyne HDL-64E LiDAR

O Velodyne HDL-64E LiDAR - figura 2.1 - é o sensor que o carro da Google utiliza para gerar informação a três dimensões do ambiente, é também muito utilizado nos veículos que participam na competição "Grand Challenge" [8]. É capaz de gerar dados a uma taxa de 1.3 milhões de pontos por segundo, possui 64 emissores laser e tem uma amplitude de deteção horizontal de 360 ° e vertical de 26,8 °. O seu custo de mercado é de cerca de 75 mil dólares. [9]



Figura 2.1: HDL-64E LiDAR.

Na figura 2.1 é possível visualizar uma nuvem de pontos adquirida por este sensor.



Figura 2.2: Nuvem de pontos adquirida pelo HDL-64E LiDAR num cruzamento.

2.1.2 Câmaras 3D

As câmaras 3D podem utilizar dois métodos diferentes para representar a terceira dimensão numa imagem a duas dimensões. Um já foi mencionado , que é utilizando uma fonte de luz estruturada (é o caso da Kinect). Outro método que é usado, é técnica de tempo de voo (ToF - time of flight).

Na figura 2.3 é possível visualizar dois sensores que utilizam a técnica de ToF.



Figura 2.3: Exemplos de sensores que usam técnica de ToF. a) Panasonic D-Imager [10]; b) SwissRanger SR4000 [11].

Este tipo de sensores têm alcance máximo de cerca de 10 metros e um custo de cerca de 5 mil dólares.

2.1.3 Câmaras estéreo

A visão estéreo consiste na interpretação de dados de duas ou mais imagens obtidas do mesmo cenário de pontos de vista diferentes. Sabendo a diferença de posição entre duas câmaras (baseline) e as suas propriedades óticas, é possível estimar a localização de objetos em profundidade num determinado cenário. [12]

2.2 Sensores LRF

Devido a alguns fatores limitativos dos sensores 3D atualmente disponíveis, em especial o seu custo, uma outra solução tem vindo a ser utilizada para gerar uma representação a três dimensões do ambiente. Essa solução tenta aproveitar as vantagens dos sensores que efetuam scans 2D - figura 2.4 (a) - e conferir mais um grau de liberdade ao sistema, através de um sistema mecânico controlado por um servo motor. Esta solução foi implementada pelo Laboratório de Automação e Robótica (LAR) e depois foi incorporada no AtlasCar, o mecanismo pode ser visualizado na figura 2.4 (b).



Figura 2.4: Laser 3D. a) LRF - Laser Sick LMS 200; b) sistema mecânico do laser rotativo [3].

Também recorrendo a um laser 2D - figura 2.5 - , o LAR implementou um método de reconstrução dinâmica de um mapa local, na vizinhança de um veículo em movimento (AtlasCar). O mapa local consiste numa nuvem de pontos 3D que é obtida através da acumulação de diversos varrimentos laser à medida que o AtlasCar navega. [2]



Figura 2.5: Sensor Laser Hokuyo. [2]

2.3 Mapas para navegação

Para efetuar navegação em ambientes exteriores ou em superfícies não planares, é necessário uma estrutura de dados que represente de forma compacta o ambiente e ao mesmo tempo que permita realizar tarefas de planeamento e localização. [13]

Os mapas de elevação representam de forma compacta informação 3D: é guardada em cada célula de uma grelha o valor da altura correspondente a essa região. A desvantagem da utilização desta representação é o facto de fazer uma caraterização discreta do ambiente, ou seja, para efeitos de navegação, zonas como túneis ou passagens sob pontes, seriam representados erroneamente como obstáculo - figura 2.6 (b) - .



Figura 2.6: a) Dados de uma ponte gravada por um robô móvel usando um laser SICK LMS; b) mapa de elevação obtido utilizando os dados da figura 2.6 (a). [14]

Para tentar ultrapassar algumas das restrições dos mapas de elevação, além do valor da altura, cada célula da grelha é classificada em quatro classes: 1) contendo dados adquiridos acima do plano de aquisição do sensor, 2) lacunas verticais, 3) estruturas verticais ou 4) como células transponíveis. Na figura 2.7 é possível visualizar a aplicação destes classificadores. [14]





Tons de azul indicam zonas adquiridas acima do plano de aquisição do sensor; Tons de vermelho indicam zonas verticais; Tons de verde indicam zonas transponíveis; Zonas não transponíveis não se encontram representadas. [14]

De notar que na figura 2.7, zonas com inclinação como as subidas laterais são classificadas como não transponíveis ou não navegáveis o que poderá não ser totalmente verdade. Com base na classificação representada na figura 2.7 é possível criar uma extensão do mapa de elevação ilustrado na figura 2.6, o resultado pode ser visualizado na figura 2.8.



Figura 2.8: Extensão do mapa de elevação da figura 2.6 .[14]

Outro algoritmo de construção de mapas é o MLS (*Multi Level Surface Maps*), consiste numa grelha 2D de dimensões variáveis em que cada célula guarda uma lista de superfícies que são utilizadas para representar o ambiente - figura 2.9 (a) - . Este algoritmo também prevê a classificação das células por classes, células horizontais ou células verticais. Após essa classificação, para cada célula horizontal é verificado se esta tem entre 5 a 8 células vizinhas, caso seja verificada esta condição, e a diferença em altura entre a célula e as células vizinhas for inferior a 10 cm, a célula é considerada navegável [13]. A figura 2.9 (b) ilustra o resultado obtido da aplicação desta classificação.



(b)

Figura 2.9: a) Mapa MLS do ambiente representado na figura 2.6 (a); b) resultado da classificação para o mapa MLS. Tons de amarelo representam superfícies navegáveis, enquanto que tons de azul representam as não navegáveis e tons de vermelho representam as verticais. [13]

Nos últimos anos, tem existido uma especial atenção em algoritmos de SLAM (*Simultaneous Localization and Mapping*) por parte da comunidade robótica. SLAM é o processo de construção de um mapa de um ambiente desconhecido através de uma sequência de medições adquiridas por sensores colocados num robô móvel e inclui a localização do robô nesse mapa. Este tipo de mapas é excelente para navegação local e para evitar obstáculos [15].



Figura 2.10: Exemplo mapa criado usando SLAM. [16]

Este algoritmo cria então uma mapa global no qual se pode efetuar tarefas de planeamento. Para uma aplicação como a do *AtlasCar* um mapa global não é interessante. Interessante é ter informação na proximidade do veículo no instante em que ele se encontra e efetuar tarefas de planeamento nesse mesmo instante.

Capítulo 3

Sistema de aquisição 3D

3.1 Descrição do sensor

Neste trabalho pretende-se utilizar um laser 2D - figura 2.4 (a) - em rotação para gerar informação 3D. O laser 2D têm um espelho interno rotativo que reflete um feixe laser em diferentes ângulos, sendo possível ter uma perceção bidimensional do espaço com base no tempo de voo de cada feixe, ou seja, desde que o feixe foi emitido e refletiu num objeto, até que foi recebido de volta. A distância de um objeto ao sensor é proporcional ao tempo de voo do feixe.



Figura 3.1: Princípio de funcionamento do laser Sick LMS 200.

Para se obter uma perceção tridimensional do espaço é conferido ao sistema mais uma grau de liberdade através da rotação de um veio perpendicular ao laser 2D - figura 2.4 (b) - . A rotação do veio é conferida por um motor passo-a-passo controlado por um microcontrolador, e para evitar constrangimentos criados pela cablagem são utilizados anéis de escorregamento para transferir sinais e potência entre o sensor e a plataforma fixa ao veículo. A posição do veio é determinada por um codificador incremental. [17]

O laser 2D está configurado para fazer aquisições com uma amplitude de deteção horizontal de 180° e $0, 5^{\circ}$ de resolução, o que equivale a 361 medições com um tempo de aquisição 26 milissegundos (cerca de 37.5 Hertz); no entanto, pode ser configurado para outras amplitudes e resoluções. [18]

3.2 Propriedades da configuração

Com vista a obter uma representação tridimensional do ambiente, é necessário fazer uma correta associação entre o ângulo do *scan* 2D, θ , e a posição angular externa em que o *scan* foi efetuado, ϕ . O vetor de posição em coordenas esféricas, de acordo com a figura 3.2, é definido por:

$$\vec{r} = r_i \cdot \left[\cos\theta_i \quad \sin\theta_i \cos\phi_i \quad \sin\theta_i \sin\phi_i\right] \tag{3.1}$$

onde r_i é a distância medida pelo laser para um determinado ângulo interno do espelho rotativo, θ_i , e um ângulo externo associado à rotação do veio, ϕ_i .



Figura 3.2: Coordenadas espaciais. [3]

Inicialmente não existia uma correta associação entres os ângulos $\theta \in \phi$, o microcontrolador enviava por comunicação RS232 a uma taxa elevada, através de um temporizador, a posição lida pelo codificador. Assim sendo, para cada *scan* era assumido que o valor mais recente de ϕ , enviado pelo microcontrolador, correspondia ao *scan* recentemente adquirido [3]. O vetor de posição em coordenas esféricas nessa situação é definido por:

$$\vec{r} = r_i \cdot \left[\cos\theta_i \quad \sin\theta_i \cos\phi \quad \sin\theta_i \sin\phi\right] \tag{3.2}$$

Sendo assim, é assumido que para um *scan* completo corresponde um único ângulo ϕ . Para um caso particular, em que a velocidade de rotação do veio seja baixa, pode darse o caso em que seja efetuado um *scan* completo sem que a posição do veio altere significativamente. No entanto, nesta implementação não é garantido que a posição externa do laser enviada pelo microcontrolador corresponda ao *scan* recebido, isto é, não existe um sincronismo entre os ângulos $\theta \in \phi$.

3.3 Sincronismo do sistema

Caso seja necessário operar dois lasers *scanners* em proximidade um do outro, poderão ocorrer algumas interferências entre os sensores. Para prevenir esta situação, existe a possibilidade de sincronizar a rotação dos espelhos rotativos dos dois sensores, ficando

estes desfasados de 180° no que respeita à emissão do feixe. Nessa situação, um deles é configurado como *master* e o outro como *slave* [19]. O *master* envia um pulso de sincronismo de 24V cujo flanco descendente do pulso coincide com a posição 0° do espelho rotativo [20].

Com o objetivo de sincronizar os ângulos $\theta \in \phi$, pretende-se utilizar o pulso de sincronismo, que indica o início de um *scan*, para que nesse instante o microcontrolador envie a posição angular do veio. Desta forma, pode-se associar com confiança o início do *scan* a um ângulo ϕ .

3.3.1 Configuração do laser

Para utilizar o pulso de sincronismo é necessário configurar o laser como *master*, para tal recorreu-se ao uso de uma *toolbox* [21] que permite comunicar e configurar o laser Sick LMS 200.

De acordo com a figura 3.3, é necessário escrever no bloco I da posição de memória 77h da *EEPROM* a seguinte mensagem 0x42.

Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 0 1 0 0 0 0 1 0

1	Data class	BYTE
Restart	Content	00h: Restart when restart button is actuated
		01h: Restart after a set time
		02h: No restart block (default setting)
		03h: Restart button switches field set, restart after a set time
		04h: Restart button switches field set, no restart block
		05h: LMS 2xx operates as a slave, restart after a set time
		06h: LMS 2xx operates as a slave, immediate restart
		Bit 5:
		0: No motor flap (default setting)
		 Use motor flap (output A controls the motor flap, output A cannot be used for
		evaluating the field)
		Bit 6:
		0: No master (default setting)
		1: Master (output C outputs the synchronisation cycle)
		Bit 7:
		0: Time basis for restart 1 s (default setting)
		1: Time basis for restart 1/10 s
	Hex. value	xxh

Figura 3.3: Bloco I do comando 77h para configuração do laser. [18]

3.3.2 Implementação do pulso de sincronismo

Para que no instante em que é iniciado um *scan* seja enviado por RS232 a posição do veio que lhe corresponde, é necessário utilizar o pulso de sincronismo para gerar um *interrupt* no microcontrolador. É então necessário reprogramar o microcontrolador, deixando-se de se utilizar um *interrupt* gerado por um temporizador para se usar um *interrupt* gerado por um evento externo. O microcontrolador utilizado é PIC18F2431, que prevê que *interrupts* gerados por eventos externos devem ser configurados nos pinos INTx.

Consultando o modo de configuração dos registos INTCON2 e INTCON3, informação disponível no *datasheet* do microcontrolador [22], é alterada a rotina de configuração inicial do microcontrolador para contemplar as seguintes configurações.

```
//define interrupt in pin INT1
TRISCbits.TRISC4=1; //define pin INT1 as input
INTCON3bits.INT1E=1; //enable interrupt
INTCON3bits.INT1IF=0; //clear interrupt flag
INTCON3bits.INT1IP=0; //set as low priority interrupt
INTCON2bits.INTEDG1=0; //interrupt at the end of the pusle
(falling edge)
```

Efetuada a configuração do pino INT1 como *interrupt*, é necessário implementar uma rotina de interrupção para o evento gerado. Essa rotina deve ser capaz de efetuar a leitura da posição, guardada em memória por um módulo próprio que o microcontrolador possui para efetuar leituras de posição de um codificador, e enviar essa informação.

O pulso de sincronismo tem uma frequência de 75 Hertz (cerca de 13 ms), e é necessário verificar que o microcontrolador tem tempo suficiente para enviar a informação antes de ser gerado outro *interrupt*. A mensagem enviada por RS232 contem no máximo 13 caracteres, "P XXXX V XXX\n", a posição do codificador varia entre 0 e 2047 pulsos por rotação, e a velocidade entre 25 e 300 rpm. Para cada caractere é utilizada a seguinte estrutura do protocolo de comunicação RS232; 1 *start bit*, 8 *bits* de dados e 1 *stop bit*; portanto 10 bits por caractere o que para 13 caracteres significa que são enviados no máximo 130 bits. Portanto efetuando a comunicação a uma taxa de 115 Kbaud, o tempo de envio de uma mensagem completa é de cerca de 1,1 ms.

Além das alterações efetuadas na programação do microcontrolador foi necessário refazer o circuito do microcontrolador, apêndice A. De notar que o pulso de sincronismo antes de poder ser interpretado pelo microcontrolador, deve passar por um divisor resistivo para baixar a tensão para cerca de 5V.

Observando as frequências de aquisição de dados do laser e do pulso de sincronismo, verifica-se que a frequência do pulso de sincronismo é duas vezes superior à da aquisição de dados. Isto deve-se ao facto do pulso ser emitido no instante em que o espelho rotativo do laser passa na posição de 0°, ou seja, para uma resolução angular de 0.5° são efetuados dois *scans* com resolução de um grau e desfasados de meio grau, logo o espelho passa duas vezes pela posição de 0°, sendo enviados por isso dois pulsos. Sendo assim, o vetor de posição representado na equação (3.1), corresponde a um *scan* parcial.

3.4 Reconstrução tridimensional

3.4.1 Sistemas de coordenadas do AtlasCar

Em projetos como o AtlasCar que utilizam diversos sensores para fazer um reconhecimento do ambiente, é necessário definir vários sistemas de coordenadas que permitam relacionar os sensores entre si e para determinar a posição do veículo em relação ao ambiente. Em ROS, essa informação é disponibilizada por ficheiros do tipo Unified Robot Description Format (URDF), que permitem descrever um determinado robô. A informação disponibilizada por este tipo de ficheiro permite definir a posição e orientação dos sensores relativamente a um referencial à escolha. [2]

Na figura 3.4 é possível visualizar os diversos sistemas de coordenadas implementados no AtlasCar. A gestão dos sistemas de coordenadas é efetuada por um módulo do ROS, $robot_state_publisher$, que subscreve os valores dos ângulos de juntas, publicadas no tópico /trf/joints, e publica transformações no tópico /trf/frames usando a cinemática descrita no ficheiro URDF. [6]



Figura 3.4: Sistemas de coordenadas presentes no AtlasCar. [6]

3.4.2 Integração com o ROS

Para usar os dados do laser 3D é necessário aceder à informação proveniente do laser Sick LMS 200 e do microcontrolador. O ROS já tem um módulo que comunica via RS422 com o laser Sick LMS 200. Para comunicar com o microcontrolador existe um módulo, laser_rotate3D, desenvolvido no LAR.

Como já foi referido, o módulo responsável pela publicação das transformações de referenciais é o robot_state_publisher. Portanto, o módulo laser_rotate3D publica no tópico /trf/joints o valor de ϕ e só mais tarde será publicada a transformação. Desta forma, perde-se o sincronismo entre ângulo ϕ e a transformação que lhe corresponde. É então necessário alterar o ficheiro URDF, para que o módulo robot_state_publisher não publique as transformações associadas ao laser 3D, e alterar o módulo laser_rotate3D para que além de publicar o ângulo ϕ , publicar também a transformação que lhe corresponde, mantendo assim o sincronismo.

O laser 3D está associado ao referencial fixo $/atc/laser/roof_rotating/base$ - figura 3.4 - . A figura 3.5 ilustra a estrutura de comunicação dos módulos do laser 3D. O nodo /snr/las/0/node publica uma mensagem LaserScan no tópico /snr/las/0/scan, essa mensagem, contendo o scan lido, está associada ao referencial rotativo $/atc/laser/roof_rotating$,

este referencial roda em torno do referencial $/atc/laser/roof_rotating/base$. O nodo /snr/las3d/node publica o ângulo ϕ , no tópico /trf/joints, e a transformação associada a essa rotação entre os referenciais $/atc/laser/roof_rotating/base$ e $/atc/laser/roof_rotating$ no tópico /trf/frames.



Figura 3.5: Comunicação dos módulos do laser 3D.

3.4.3 Construção da nuvem de pontos

Com o objetivo de construir uma nuvem de pontos com base na informação do laser 3D, foi desenvolvido um módulo em ROS, o $laser3D_pointcloud$. Este módulo será responsável por subscrever o tópico contendo o scan do laser 2D e o tópico com o ângulo ϕ , e fazer a acumulação dos vários scans para o referencial fixo do laser 3D, construindo assim uma nuvem de pontos que representa tridimensionalmente o ambiente. Para realizar operações sobre nuvens de pontos foi utilizada um biblioteca já existente, a *Point Cloud Library* (PCL) [23], que está disponível em ambiente ROS.

Para se utilizar um *scan* para construir uma nuvem de pontos, é necessário converter os dados do laser para coordenadas cartesianas, segundo o sistema de equações (3.3).

$$\begin{cases} x_i = r_i \cos \theta_i \\ y_i = r_i \sin \theta_i \\ z_i = 0 \end{cases}$$
(3.3)

Em seguida cada ponto do *scan*, $\vec{p_i}$, é transformado para o referencial fixo do laser 3D usando a matriz de transformação definida entre o referencial $/atc/laser/roof_rotating$, vamos designar r2, e o referencial $/atc/laser/roof_rotating$, vamos designar r1. Para transformar cada pondo individualmente é necessário aceder à transformação $r^2\mathbf{T}_{r1}$ em instantes de tempo diferentes, t.

$$\vec{\mathbf{p}}_i' = {}^{r_2} \mathbf{T}_{r_1}^t \cdot \vec{\mathbf{p}}_i \tag{3.4}$$

em que para o instante $t = t_0$, início do *scan*, irá corresponder $\theta_i = \theta_0$, \vec{p}_0 . Para o instante $t = t_n$, fim do *scan*, corresponderá $\theta_i = \theta_n$, \vec{p}_n . As transformações ${}^{r_2}\mathbf{T}_{r_1}^{t_0}$ e ${}^{r_2}\mathbf{T}_{r_1}^{t_n}$,
correspondem a transformações publicadas pelo nodo /snr/las3d/node no instante em que são iniciados os scans, sendo que o instante $t = t_n$ do $scan_k$ corresponde ao instante $t = t_0$ do $scan_{k+1}$. A formulação apresentada na equação (3.4), permite que futuramente possam ser incorporadas mais transformações, por exemplo, usando o egomotion do carro, seria possível transformar as medições individuais de um scan com base no movimento do veículo.

Para calcular as transformações nos instantes de tempo intermédios o ROS utiliza técnicas de interpolação em quatérniões. As matrizes de transformação geométricas ou na forma de quatérnios são possíveis de converter bidirecionalmente de uma para outra. Usando a técnica de interpolação SLERP (*The Spherical Linear Interpolation*) é possível calcular um quatérnio entre dois quatérnios em instantes de tempo diferentes. A função **slerp** é definida por:

$$\mathbf{slerp}(^{i}\mathbf{q}_{j}^{a}, ^{i}\mathbf{q}_{j}^{b}, \alpha) = ^{i}\mathbf{q}_{j}^{a}((^{i}\mathbf{q}_{j}^{a})^{-1} \cdot ^{i}\mathbf{q}_{j}^{b})^{\alpha}$$
(3.5)

onde α é o parâmetro de interpolação, definido entre 0 e 1. Se $\alpha = 0$ ou $\alpha = 1$, os quatérnios que a função retorna são ${}^{i}\mathbf{q}_{j}^{a}$ ou ${}^{i}\mathbf{q}_{j}^{b}$, respetivamente. Esta ferramenta está implementada em ROS, permitindo assim determinar a transformação entre dois referenciais para um instante de tempo específico, desde que a transformação esteja definida antes e depois do tempo requerido. [6]

É possível então adaptar a equação (3.4) para a seguinte formulação:

$$\vec{\mathbf{p}}_{i}^{\prime} = \mathbf{slerp}(^{r2}\mathbf{q}_{r1}^{t_{0}}, ^{r2}\mathbf{q}_{r1}^{t_{n}}, \alpha) \cdot \vec{\mathbf{p}}_{i}$$
(3.6)

onde é assumido que o scan é adquirido pelo las er 3D a uma velocidade constante, portanto o valor de α é definido por:

$$\alpha = i \cdot \frac{1}{n-1} \tag{3.7}$$

em que i representa o índice do ponto do scan, e n o número total de pontos do scan. Em seguida é ilustrado o processo de construção de nuvem de pontos.

Construção de nuvem de pontos

Inputs:

scan: Sequência de n pontos medidos;

- r1: Referencial de acumulação do scan;
- t_0 : Instante inicial em que foi adquirido o scan;
- t_n : Instante final em que foi adquirido o scan;

Outputs:

PointCloud: Nuvem de pontos obtida por transformação do *scan*;

- 1: n: Número de pontos presentes no scan;
- 2: for (i=0; i< n; i++) do
- 3: Transformar os pontos do *scan* para coordenadas cartesianas; (3.3)
- 4: Calcular o valor de α ; (3.7)
- 5: Transformar os pontos do *scan* para o referencial r1; (3.6)

6: end for

7: $PointCloud \leftarrow scan$; Acumular a nuvem de pontos com base no scan transformado;

Como já foi referido, este processo de construção de nuvem de pontos é gerido pelo o módulo laser3D_pointcloud que subscreve as mensagens do laser 3D, scan e ϕ - figura 3.6 - .



Figura 3.6: Comunicação dos módulos do laser 3D incorporando o módulo de construção de nuvem de pontos.

Em ROS, a cada mensagem está associado o instante em que esta foi criada (time stamp), assim, é necessário decidir quais os time stamps que serão utilizados para definir os instantes $t_0 e t_n$. Existem duas possibilidades para definir os instantes de tempo. Pode ser usado o time stamp do scan para definir o instante t_0 e usar o tempo de duração do scan para definir o instante t_n .

$$\begin{cases} t_0 = scan_{time_stamp} \\ t_n = scan_{time_stamp} + (scan_{size} \cdot scan_{time_increment}) \end{cases}$$
(3.8)

Esta formulação não é a mais precisa, já que o instante em que o *scan* foi publicado pelo nodo /snr/las/0/node não corresponde exatamente ao instante de tempo em que foi publicada a transformação $r^2 \mathbf{T}_{r1}$. No entanto, tem a vantagem de não ser necessário atrasar o processamento dos *scans* recebidos para se ter acesso aos instantes $t_0 \in t_n$. A outra possibilidade é usar os *time stamps* dos ângulos ϕ .

$$\begin{cases} t_0 = \phi_{t0} \\ t_n = \phi_{tn} \end{cases}$$
(3.9)

Esta formulação é mais precisa, pois a publicação dos ângulos ϕ e das transformações $r^2 \mathbf{T}_{r1}$ são efetuadas no mesmo instante de tempo e pelo mesmo nodo, /snr/las3d/node. No entanto, é necessário atrasar o processamento dos *scans* recebidos, já que o *scank* só poderá ser processado quando for recebida informação do ϕ_{t0} e do ϕ_{tn} , em que ϕ_{tn} corresponde ao ϕ_{t0} do $scan_{k+1}$. Além disso, pode acontecer que existam atrasos nas comunicações e que os dados não sejam recebidos ao nível de *software* como foram gerados ao nível de *hardware* - figura 3.7 - .



Figura 3.7: a) Tráfego de mensagens em situação normal; b) tráfego de mensagens na situação em que existe irregularidade nas comunicações. **Nota** - Escala horizontal em segundos.

Para lidar com o facto de ser necessário atrasar o processamento dos *scans* recebidos, e com situações de irregularidade de comunicações ou até mesmo perda de dados, foi implementado um algoritmo que vai guardando em memória os *scans* e os ϕ , que são processados sempre que possível e são descartados após o seu processamento.

Algoritmo para associação de scans ao seu ϕ inicial e final

Variáveis:

 ϕ_vect : Mensagens de ϕ guardadas num vetor; $scan_vect$: Mensagens de scan guardadas num vetor t_0 : Instante inicial em que foi adquirido o scan; t_n : Instante final em que foi adquirido o scan; $working_scan$: Scan que está ser processado; $need_data$: Nova informação requerida;

```
1: if (\phi\_vect_{size} < 2) then
```

- 2: Não existe ϕ suficientes para associar a um *scan*
- 3: return

```
4: end if
```

5: while $(\phi_{vect_{size}} \ge 2 \text{ and } scan_{vect_{size}} \ge 0)$ do

- 6: $working_scan \leftarrow scan_vect$; Processar scan mais antigo no vetor de scans
- 7: Eliminar valor mais antigo no vetor de *scans*
- 8: $t_0 \leftarrow \phi_vect_{stamp}$; Associar a t_0 o valor mais antigo no vetor de ϕ
- 9: Eliminar valor mais antigo no vetor de ϕ

```
10: t_n \leftarrow \phi\_vect_{stamp}; Associar a t_n o valor mais antigo no vetor de \phi
```

```
11: while (working\_scan_{stamp} < t_0) do
```

```
12: if scan\_vect_{size} > 0) then
```

```
13: working\_scan \leftarrow scan\_vect; Processar scan mais antigo no vetor de
```

	scans
14:	else
15:	$need_data = 1$
16:	break
17:	end if
18:	Eliminar <i>scan</i> mais antigo
19:	end while
20:	while $(working_scan_{stamp} > t_n)$ do
21:	Eliminar valor mais antigo de ϕ
22:	if $(phi_vect_{size} > 0)$ then
23:	$t_n \leftarrow \phi_vect_{stamp}$; Associar a t_n o valor mais antigo no vetor de ϕ
24:	else
25:	$need_data = 1$
26:	break
27:	end if
28:	end while
29:	if $(need_data == 1)$ then
30:	break
31:	end if
32:	end while
33:	$working_scan$ pode ser processado com instante inicial = t_0 e instante final = t_n

A figura 3.8 ilustra de forma mais simplificada o algoritmo.



Figura 3.8: Ilustração de acumulação de dados e seu processamento

Na situação em que é necessário atrasar o processamento dos *scans*, foi observado experimentalmente que os tamanhos dos vetores contendo as mensagens *scan* e de ϕ não ultrapassam o valor de cinco mensagens acumuladas, sendo dois o valor médio de mensagens acumuladas. Já a situação de perdas de dados não é muito comum, o efeito é mais notório quando é feito um pós processamento em laboratório, o que indica que terá existido falhas de gravação de mensagens para o *bag* de dados. No entanto, o módulo fica preparado para lidar com estas situações.

Para que se possa utilizar o módulo *laser3D_pointcloud* implementando diferentes metodologias para associação dos instantes de tempo aos *scans*, foi criado um ficheiro *launch* com parâmetros configuráveis que irá iniciar o módulo *laser3D_pointcloud*.

```
<launch>
       <proup ns="/snr/las3d/fus/pointcloud">
              <!-- Remmaping topics-->
              <remap from="/joint_state" to="/trf/joints"/>
              <remap from="/laserscan0" to="/snr/las/0/scan"/>
              <remap from="/pc_out" to="/snr/las3d/fus/pointcloud"/>
              <remap from="/tf" to="/trf/frames"/>
              <!-- Set the accumulation frame_id-->
              <remap from="/ac_frame" to="/atc/laser/roof_rotating/base"/>
              <node name="node" pkg="laser3D_pointcloud" type="las3D_pc">
                      <!-- Set the accumulation_mode;
                      1 - Use slerp projection, with time t0 and tn based on
                          the phi stamp;
                      2 - No correction (project the laser scan in only one
                          phi), using t0 as scan start time
                      3 - Use slerp projection, with time t0 and tn based on
                          the scan time; -->
                      <param name="accumulation_mode" type="int" value="1"/>
                      <!-- Max of scans to be accumulated -->
                      <param name="max_scans_accumulated" type="int"</pre>
                          value="500"/>
                      <!-- Set the output_frequency-->
                      <param name="output_frequency" type="double"</pre>
                          value="200.0"/>
              </node>
       </group>
</launch>
```

A figura 3.9 ilustra o resultado de construção de uma nuvem de pontos utilizando os time stamps dos ângulos ϕ .



(a)



Figura 3.9: a) Imagem do ambiente; b) nuvem de pontos criada utilizando os time stamps dos ângulos $\phi.$

Capítulo 4

Mapas de acessibilidade e de navegabilidade

Neste capítulo será descrita a metodologia implementada para a deteção do espaço navegável usando informação 3D. Foi desenvolvido um módulo em ROS, *navigability_map*, que será responsável por subscrever nuvens de pontos e criar um mapa de acessibilidade. Esse mapa de acessibilidade será calculado com base em algumas propriedades da nuvem de pontos, entre elas a coordenada z dos pontos, para tentar detetar descontinuidades no terreno, e as normais para tentar detetar superfícies mais irregulares, como buracos na estrada. Neste trabalho, foram utilizadas as nuvens de pontos construídas com base na informação do laser 3D. Será descrito a metodologia para a construção do mapa de acessibilidade e mais à frente será descrito o conceito de navegabilidade.

4.1 Pré-processamento da nuvem de pontos

Foi aplicado à nuvem de pontos um filtro para descartar pontos acima dos 2 metros de altura e a mais de 25 metros de distância. Acima de 2 metros os obstáculos não interferem com a navegabilidade do veículo, e a 25 metros de distância a qualidade de dados do sensor não justifica a sua utilização. A nuvem de pontos representada na figura 3.9 (b) contém 98077 pontos, aplicando o filtro reduz-se para 74032 pontos e obtém-se a nuvem de pontos representada na figura 4.1.

Além da aplicação do filtro são estimadas as normais da nuvem de pontos. Existem duas possibilidades para estimar as normais associadas a superfícies inicialmente representadas por uma nuvem de pontos. A reconstrução da superfície, e.g. por ajuste de um modelo de ordem superior, e posterior cálculo da normal, é um procedimento com custo computacional incompatível com a aplicação presente. Em alternativa, é possível estimar a normal "num ponto" aplicando o produto externo aos vetores definidos entre um ponto e os seus vizinhos. Neste trabalho foi utilizada a segunda abordagem, que está implementada na biblioteca PCL; o cálculo das normais é reduzido a uma análise de valores próprios (PCA - *Principal Component Analysis*) [24]. A biblioteca PCL utiliza a estrutura Kd-Tree para representar a nuvem de pontos numa forma estruturada, tornando mais rápida a análise de vizinhos numa nuvem de pontos.

Para realizar uma análise de vizinhos pode ser utilizado o método dos K vizinhos mais próximos; os pontos são organizados de acordo com a sua distância em relação ao ponto



Figura 4.1: Nuvem de pontos filtrada da imagem 3.9 (a).

que está ser processado, e são selecionados os K que estão mais próximos para definir uma superfície e calcular a normal que lhe está associada. Outro método possível é usar os vizinhos contidos numa esfera de raio R centrada no ponto que está a ser processado. Nas figuras 4.2 e 4.3 é possível visualizar a aplicação destas metodologias; em ambas as figuras é possível perceber que para valores mais baixos de K e R as normais são mais irregulares.



Figura 4.2: a) Cálculo das normais utilizando K=5; b) cálculo das normais utilizando K=20.

24



Figura 4.3: a) Cálculo das normais utilizando R=0.1; b) cálculo das normais utilizando R=0.4.

Neste trabalho foi utilizada a vizinhança geométrica, R, para evitar que pontos *outliers* influenciem o cálculo da normal de "um ponto"; situação que é suscetível de acontecer numa vizinhança de conectividade, K, pois é definido um número fixo de vizinhos para o cálculo da normal correndo-se por isso o risco de utilizar um ponto que geometricamente não deve ser considerado como vizinho.

4.2 Grelha de ocupação

Para representar a informação 3D foi construída uma grelha de ocupação, a cada célula da grelha está associado um conjunto de pontos da nuvem e suas propriedades.

4.2.1 Inicialização da grelha

A grelha é criada com um número de linhas relacionado com o máximo valor da coordenada x e com um número de colunas relacionado com o máximo valor absoluto da coordenada y.

```
/*********** Get grid parameter ********/
11_
//|
//| Save Total Row
                    //|____
                ____ |
total_row=(int)roundl((max_Xval/Sx)+0.5); //+0.5 to round up
//__
//|
//|
     Get Center Col
                    1
              _____I
//|__
int auxCenterCol=(int)roundl((max_Yval/Sy)+0.5); //+0.5 to round up
int center_col;
if (auxCenterCol % 2) //auxCenterCol is odd
       center_col=auxCenterCol;
else
```

```
center_col=auxCenterCol+1;
//______
//| | |
//| Save Total Col |
//|_____|
total_col=center_col*2+1;
//______
//| | |
//| Save Center Col |
//|_____|
CARaxis_col=center_col; //Grid center is coincident with the axis
```

A grelha é então criada com um total de linhas igual ao valor de $total_row$ e um total de colunas igual ao valor de $total_col$; estes valores foram calculados com base nas variáveis $Sx \in Sy$ que representam as dimensões das células. O valor central da grelha $(CARaxis_col)$ é utilizado para a representar em torno da origem do referencial da nuvem de pontos - figura 4.4 - . Após a criação da grelha são processados os pontos da nuvem e é armazenada a informação para a célula que lhes corresponde; a equação (4.1) ilustra como é calculada a célula C(i, j) a qual um ponto, $\vec{\mathbf{p}}_n$, pertence .

$$\vec{\mathbf{p}}_n = (p_x, p_y) \hookrightarrow C(i, j) : \begin{cases} i = round\left(\frac{p_x}{S_x}\right) \\ j = CARaxis_col + round\left(\frac{p_y}{S_y} + k\right) \end{cases}$$
(4.1)

onde p_x e p_y representam as coordenas $x \in y$ do ponto $\vec{\mathbf{p}}_n$. O cálculo da coluna a que um ponto pertence é feito incluindo a posição central da grelha para que esta fique definida em torno da origem do referencial da nuvem de pontos, em que k toma os valores de +0.5 ou -0.5 caso p_y seja positivo ou negativo.



Figura 4.4: Vizinhança N8 para uma célula.

Atribuição dos pontos da nuvem de pontos às células das grelha

Inputs:

PointCloud: Nuvem de pontos;

Normals: Normais da nuvem de pontos;

1: m: Número de pontos presentes na PointCloud;

2: *i*: Linha da grelha correspondente ao ponto a ser processado;

- 3: *j*: Coluna da grelha correspondente ao ponto a ser processado;
- 4: for (n=0; n<m; n++) do

5: Calcular i; (4.1)

- 6: Calcular j; (4.1)
- 7: Célula $(i, j) \leftarrow$ coordenadas do ponto $\vec{\mathbf{p}}_n$ e ângulos da normal que lhe está associado.
- 8: end for

4.2.2 Cálculo de propriedades das células

Utilizando a informação armazenada nas células da grelha, foi calculada a média aritmética da coordenada z, μ_z , e a média aritmética dos ângulos das normais associadas aos pontos de uma célula , μ_{α} , $\mu_{\beta} \in \mu_{\gamma}$. Para representar as normais não foram utilizados os cossenos diretores devido ao seu comportamento não linear, mas sim os ângulos.

É possível visualizar na figura 4.5 um mapa de elevação incluindo também a representação da normal média, que é calculada com base na média dos ângulos das normais contidas na célula. Além disso também foi associado a cada célula um valor de confiança, ν , para os valores μ_z , μ_α , $\mu_\beta \in \mu_\gamma$. Esse valor de confiança não tenta descrever a confiança no valor da média mas sim o grau da sua representatividade dos valores contidos na célula, para posteriormente ser utilizado no cálculo de acessibilidade.

As equações (4.2), (4.3), (4.4) e (4.5) ilustram como são calculados os valores de confiança de uma célula.

$$\boldsymbol{\nu}_{z} = 1 - \frac{\boldsymbol{\sigma}_{z}}{\boldsymbol{\sigma}_{0z}} \qquad (4.2) \qquad \boldsymbol{\nu}_{\alpha} = 1 - \frac{\boldsymbol{\sigma}_{\alpha}}{\boldsymbol{\sigma}_{0\alpha}} \qquad (4.3)$$

$$\boldsymbol{\nu}_{\beta} = 1 - \frac{\boldsymbol{\sigma}_{\beta}}{\boldsymbol{\sigma}_{0\beta}} \qquad (4.4) \qquad \boldsymbol{\nu}_{\gamma} = 1 - \frac{\boldsymbol{\sigma}_{\gamma}}{\boldsymbol{\sigma}_{0\gamma}} \qquad (4.5)$$

em que ν representa a confiança, que é calculada com base nos desvio padrão corrigido, σ , e de uma constante σ_0 para normalizar o valor de ν entre 0 e 1, no entanto é necessário saturar a confiança a 0 para casos em que o valor de σ seja superior a σ_0 . O valor da constante σ_0 é definido experimentalmente. Este cálculo só é valido para células com mais de um ponto; para células com apenas um ponto é atribuída uma confiança de 0.5. Não foi utilizado o valor da média como constante normalizadora, porque nessa situação a confiança calculada descreveria a confiança no valor da média. Imagine-se a

situação em que duas células apresentam o mesmo valor de σ_z , mas uma delas é uma célula contida no plano da estrada, denomine-se por Ca, e outra é uma célula contida num veículo, Cb. Nesta situação, a célula Ca deveria ter uma confiança na representatividade dos seus valores igual à célula Cb, no entanto se a constante normalizadora fosse o valor da média isso não se verificaria, já que μ_{bz} seria superior μ_{az} .



Figura 4.5: Mapa de elevação com representação da normal média da nuvem de pontos da figura 4.1.

4.2.3 Interpolação de propriedades das células

É possível observar na figura 4.5 que existem algumas células que não contêm informação. Isto deve-se ao facto da nuvem de pontos ser esparsa e também está relacionado com a dimensão das células, que neste caso é de 0.25×0.25 m. Para células vazias que têm pelo menos quatro células vizinhas, na vizinhança N8 com informação completa, é atribuída à célula o valor da mediana dos vizinhos para a coordenada z, para os ângulos das normais, e para a confiança da célula; entenda-se por informação completa células com pelo menos um ponto e com normal média definida. Também existe a possibilidade de uma célula ser ocupada e não ter uma normal definida, para esse caso também é interpolado o valor para os ângulos da normal. Foi utilizada a mediana em vez da média para interpolar informação para a célula, na tentativa de reduzir a influência de *outliers*. Na figura 4.6 é possível visualizar o resultado da interpolação de propriedades das células para o mapa de elevação representado na figura 4.5.



Figura 4.6: Resultado da aplicação da interpolação de propriedades para o mapa de elevação da figura 4.5.

4.2.4 Cálculo da disparidade das células

Nesta fase são associadas a uma célula 8 variáveis que a caraterizam, μ_z , μ_α , μ_β , μ_γ , ν_z , ν_α , ν_β ; torna-se então difícil de atribuir a uma célula um único valor que caracteriza a sua navegabilidade. Portanto, para atribuir um único valor que permita depois calcular um mapa de navegabilidade, é calculada uma propriedade intermédia, a acessibilidade. A acessibilidade representa, numa escala de 0 a 1, o quanto uma célula é ou não transponível de acordo com a sua vizinhança, ou seja, está relacionada com a diferença que existe entre a célula e a sua vizinhança. Para este cálculo só são considerados os vizinhos de uma célula que contêm informação, em que k representa o número de vizinhos de uma célula.

As equações (4.6), (4.7), (4.8) e (4.9) ilustram como são calculados os valores de disparidade, ϑ , de uma célula.

$$\boldsymbol{\vartheta}_{z}^{n} = \frac{\sum_{i=1}^{k} \frac{|\boldsymbol{\mu}_{z}^{i} - \boldsymbol{\mu}_{z}^{n}|}{\sqrt{\boldsymbol{\nu}_{z}^{i} \cdot \boldsymbol{\nu}_{z}^{n}}}{k} \qquad (4.6) \qquad \boldsymbol{\vartheta}_{\alpha}^{n} = \frac{\sum_{i=1}^{k} \frac{|\boldsymbol{\mu}_{\alpha}^{i} - \boldsymbol{\mu}_{\alpha}^{n}|}{\sqrt{\boldsymbol{\nu}_{\alpha}^{i} \cdot \boldsymbol{\nu}_{\alpha}^{n}}}{k} \qquad (4.7)$$
$$\boldsymbol{\vartheta}_{\beta}^{n} = \frac{\sum_{i=1}^{k} \frac{|\boldsymbol{\mu}_{\beta}^{i} - \boldsymbol{\mu}_{\beta}^{n}|}{\sqrt{\boldsymbol{\nu}_{\beta}^{i} \cdot \boldsymbol{\nu}_{\beta}^{n}}}{k} \qquad (4.8) \qquad \boldsymbol{\vartheta}_{\gamma}^{n} = \frac{\sum_{i=1}^{k} \frac{|\boldsymbol{\mu}_{\alpha}^{i} - \boldsymbol{\mu}_{\alpha}^{n}|}{\sqrt{\boldsymbol{\nu}_{\alpha}^{i} \cdot \boldsymbol{\nu}_{\alpha}^{n}}}{k} \qquad (4.7)$$

o valor de disparidade para cada uma das propriedades tenta traduzir o quanto uma célula se difere das suas células vizinhas. No cálculo da disparidade de uma célula foram utilizados os valores de confiança para influenciar a diferença entre ela e as suas vizinhas, recorrendo-se à média geométrica das confianças para dar importância a células com

baixas confianças. Desta forma, uma célula com confiança muito baixa mas com células vizinhas com confiança elevada irá apresentar um valor de disparidade substancialmente elevado quando comparado com uma situação em que se utilizasse a média aritmética, que tornaria menos notória a influência da célula de confiança muito baixa. No entanto, pode dar-se o caso em que a célula que está ser avaliada e as suas células vizinhas apresentem valores de confiança elevados, mas existir uma célula vizinha com confiança zero; nesse caso o valor de disparidade da célula que está ser avaliada tende para infinito, é então necessário limitar a diferença entre a célula e as células vizinhas para cada uma das propriedades.

$$\boldsymbol{\vartheta}^{n} = \frac{\sum_{i=1}^{k} \min\left(\frac{|\boldsymbol{\mu}^{i} - \boldsymbol{\mu}^{n}|}{\sqrt{\boldsymbol{\nu}^{i} \cdot \boldsymbol{\nu}^{n}}}, \mathbf{Th}\right)}{k}$$
(4.10)

4.3 Mapas de acessibilidade

Usando a informação da disparidade das células é possível comparar com um valor limite, Th, determinando a acessibilidade das células, ϖ .

$$\boldsymbol{\varpi}_{z}^{n} = 1 - \frac{\boldsymbol{\vartheta}_{z}^{n}}{\mathbf{Th}_{z}} \qquad (4.11) \qquad \boldsymbol{\varpi}_{\alpha}^{n} = 1 - \frac{\boldsymbol{\vartheta}_{\alpha}^{n}}{\mathbf{Th}_{\alpha}} \qquad (4.12)$$
$$\boldsymbol{\varpi}_{\beta}^{n} = 1 - \frac{\boldsymbol{\vartheta}_{\beta}^{n}}{\mathbf{Th}_{\beta}} \qquad (4.13) \qquad \boldsymbol{\varpi}_{\gamma}^{n} = 1 - \frac{\boldsymbol{\vartheta}_{\gamma}^{n}}{\mathbf{Th}_{\gamma}} \qquad (4.14)$$

Nas figuras 4.7, 4.8 e 4.9 é possível visualizar o resultado da aplicação da metodologia em cima descrita, foi utilizado um valor de Th_z de 10 cm e 0.2 radianos para os valores de Th_{α} , Th_{β} e Th_{γ} . Para o cálculo da confiança foram utilizados valores de $\sigma_0 = 2 \cdot Th$.



Figura 4.7: Acessibilidade das células em coordenada z, ϖ_z ; Nota - Cor azul corresponde a acessibilidade 1 e cor vermelha corresponde a acessibilidade 0.





Nota - Cor azul corresponde a acessibilidade 1 e cor vermelha corresponde a acessibilidade 0.



Figura 4.9: Mapa resultante do produto das acessibilidades, ϖ_z , ϖ_α , $\varpi_\beta \in \varpi_\gamma$. **Nota** - Cor azul corresponde a acessibilidade 1 e cor vermelha corresponde a acessibilidade 0.

4.4 Mapas de navegabilidade

A definição de navegabilidade é um pouco mais complexa que a definição de acessibilidade, já que a navegabilidade não é uma propriedade exclusivamente intrínseca à nuvem de pontos. A navegabilidade pode definir-se como uma zona onde o veículo pode circular, portanto tanto o plano da estrada é navegável como o plano passeio, caso o veículo já se encontre nesse plano. Sendo assim, a navegabilidade depende tanto da trajetória que se quer seguir como da zona que se pretende alcançar. Portanto, não é possível construir um único mapa de navegabilidade.

Para calcular a navegabilidade de uma qualquer trajetória pode recorrer-se ao mapa de acessibilidade, sobrepondo as trajetórias sobre as células e determinando qual a trajetória possível de realizar para alcançar a zona que se pretende. A manobra de estacionamento é um exemplo da aplicação deste método; foi desenvolvido no LAR um módulo que efetua uma procura de um volume vazio onde se possa estacionar o veículo e são calculadas diversas trajetórias possíveis como ilustrado na figura 4.10 [25].



Figura 4.10: Possíveis trajetórias a executar. [25]

A avaliação de qual trajetória a executar proposta em [25] não tem em consideração o terreno sobre qual a trajetória irá ser executada. Utilizando o mapa de acessibilidade pode ser considerado o estado do terreno em que a trajetória seria executada, podendo até mesmo levar à invalidação da trajetória.

Capítulo 5 Resultados e discussão

Neste capítulo serão apresentados os resultados obtidos com as alterações propostas no capítulo 3 para o refinamento do sistema de aquisição 3D; avaliando-se a qualidade da nuvem de pontos do laser 3D antes e depois das alterações propostas neste trabalho, comparado também a qualidade da nuvem de pontos do laser 3D com uma nuvem de pontos adquirida com o sensor Velodyne [26]. Foi também avaliada a qualidade da nuvem de pontos utilizando as diferentes metodologias apresentadas no capítulo 3 para a construção da nuvem de pontos.

Também serão analisados os resultados obtidos utilizando a metodologia descrita no capítulo 4, na deteção do espaço navegável; avaliando-se a influência dos parâmetros dos algoritmos no cálculo dos mapas de acessibilidade.

5.1 Resultado das alterações no sistema de aquisição 3D

No capítulo 3 foram apresentadas algumas alterações no sistema de aquisição 3D. Para avaliar o resultado das alterações, foram consideradas superfícies de geometria simples e avaliada a dispersão dos pontos detetados pelo *scanner* em relação a essa superfície. Neste caso avaliou-se a dispersão em torno de superfícies planares horizontais, sendo este resultado apresentado na forma de função densidade de probabilidade empírica. Recorreu-se a uma função disponível no MATLAB [27] que utiliza o método *kernel density estimation* (KDE) para estimar a função densidade de probabilidade de uma amostra.

Foi obtida a função densidade de probabilidade para superfícies planares horizontais, representadas a preto nas nuvens de pontos das figuras 5.2, 5.3, 5.4, 5.6 (a) e 5.6 (b). Para cada superfície, foi efetuada uma análise com base na função densidade de probabilidade da probabilidade, calculando a probabilidade de ocorrência de pontos num intervalo de 5 cm, que representa no máximo uma diferença de 10 cm que foi o limite máximo aceitável na cota z para a navegabilidade do veículo. Esta análise representa de certa forma a qualidade da nuvem de pontos.

Na figura 5.5 estão representadas as funções densidade de probabilidade para as nuvens de pontos adquiridas utilizando o laser 3D, antes e depois das alterações efetuadas neste trabalho, e de uma nuvem de pontos adquirida com o sensor Velodyne. Na tabela 5.1 estão sumariados os resultados obtidos comparando o erro máximo, desvio padrão e probabilidade de ocorrência de pontos para as funções densidade de probabilidade representadas na figura 5.5.

Na figura 5.7 estão representadas as funções densidade de probabilidade para as superfícies planares ilustradas nas figuras 5.6 (a) e 5.6 (b), utilizando as diferentes metodologias apresentadas no capítulo 3 para a construção da nuvem de pontos. Na tabela 5.2 estão sumariados os resultados obtidos comparando o erro máximo, desvio padrão e probabilidade de ocorrência de pontos para funções densidade de probabilidade representadas na figura 5.7.

Analisando a tabela 5.1, verifica-se que existiu uma melhoria nos dados do laser 3D após as implementações descritas no capítulo 3, reduzindo consideravelmente o erro máximo das medições em torno de uma superfície plana. Também é possível verificar que a probabilidade de ocorrência de pontos num intervalo de 5 cm, para o laser 3D com correção do *scan* utilizando os *time stamps* de ϕ , é superior à do laser 3D sem alterações e à do laser Velodyne.

Por análise da tabela 5.2, é possível concluir que o método de correção das medições individuais de um *scan* utilizando os *time stamps* de ϕ , revelou-se mais preciso e robusto na construção da nuvem de pontos, quando comparado com os outros dois métodos.

Tabela 5.1: Resultados de análise de uma superfície planar horizontal.
Nota - Sensor L1, Laser 3D sem alterações; Sensor L2, Laser Velodyne; Sensor L3, Laser 3D com correção do scan utilizando os time stamps de φ.

	Sensor L1	Sensor $L2$	Sensor L3
Erro máximo [m]	0.6	0.2	0.15
Desvio padrão [m]	0.11	0.06	0.04
Probabilidade de pontos num intervalo de 5 cm	0.51	0.52	0.84

Tabela 5.2: Resultados de análise de superfície planares horizontais, utilizando diferentes metodologias de construção da nuvem de pontos utilizando os dados do laser

3D.

Nota -

Superfície S1 representada a preto na figura 5.6 (a);

Superfície **S2** representada a preto na figura 5.6 (b);

Metodologia **A**, Laser 3D com correção do scan utilizando os time stamps de ϕ ; Metodologia **B**, Com correção do scan utilizando o time stamp de inicio do scan; Metodologia **C**, Sem correção do scan.

		Metodologia ${\bf A}$	Metodologia ${\bf B}$	Metodologia ${\bf C}$
	Erro máximo [m]	0.14	0.18	0.2
Superfície S1	Desvio padrão [m]	0.04	0.04	0.05
	Probabilidade de pontos	0.84	0.81	0.68
	num intervalo de 5 cm	0.04	0.01	0.00
	Erro máximo [m]	0.12	0.35	0.3
Superfície $S2$	Desvio padrão [m]	0.03	0.07	0.06
	Probabilidade de pontos num intervalo de 5 cm	0.89	0.42	0.64





(b)

Figura 5.1: a) Imagem do ambiente; b) nuvem de pontos adquirida com o laser 3D antes das alterações.



Figura 5.2: Nuvem de pontos adquirida com o laser 3D antes das alterações, superfície planar (3583 pontos) representada a cor preta.



Figura 5.3: Nuvem de pontos adquirida com o sensor Velodyne, superfície planar (1960 pontos) representada a cor preta.



Figura 5.4: Nuvem de pontos adquirida com o laser 3D com correção do scan utilizando os time stamps de ϕ , superfície planar (3790 pontos) representada a cor preta.



Figura 5.5: Função densidade de probabilidade obtida para: Laser 3D antes de efetuadas as alterações - cor vermelha; Sensor Velodyne - cor preta; Laser 3D com correção do scan utilizando os time stamps de ϕ - cor azul.



(a)



Figura 5.6: Nuvens de pontos adquiridas com o laser 3D representando a cor preta as superfícies planares: a) **S1** (3790 pontos); b) **S2** (2422 pontos).



Figura 5.7: Função densidade de probabilidade obtida utilizando nuvens de pontos do laser 3D construídas com diferentes metodologias, para as superfície planares: a) S1; b) S2.

Nota - Nuvem de pontos construída: Com correção do scan utilizando o time stamp de ϕ - cor preta; Com correção do scan utilizando o time stamp de início do scan - cor vermelha; Sem correção do scan - cor azul.

Diogo Artur Fonseca Matos

5.2 Resultados na criação dos mapas de acessibilidade

No capítulo 4 foi apresentada a metodologia implementada para a deteção do espaço navegável recorrendo a informação 3D. Como já foi referido na secção 4.4, o espaço navegável depende do objetivo do planeador de trajetórias, mas pode ser calculado recorrendo aos mapas de acessibilidade.

O cálculo da acessibilidade depende de alguns parâmetros referidos no capítulo 4; o raio, R, de procura da vizinhança para o cálculo das normais da nuvem de pontos; a constante normalizadora, σ_0 , do valor de confiança, ν , de cada célula; e das dimensões das células, Sx e Sy. Para avaliar a influência destes parâmetros no cálculo da acessibilidade, foram consideradas quatro nuvens de pontos diferentes e identificadas com um polígono uma zona acessível e outra inacessível. As zonas foram consideradas como acessíveis e não acessíveis com base na informação visual de uma imagem captada por uma câmara presente no AtlasCar. A eficiência do algoritmo para a zona acessível foi avaliada segundo o quociente entre o número total de células acessíveis contidas no polígono e o número total de células contidas no polígono; já para a zona inacessível, foi calculado o quociente entre número total de células inacessíveis contidas no polígono e o número total de células contidas no polígono. Foram consideradas células acessíveis, células com valor de acessibilidade superior a 0.25; e células inacessíveis, células com valor de acessibilidade igual ou inferior a 0.25.

A figura 5.8 ilustra as quatro situações estudadas, para cada uma das situações foram construídas nuvens de pontos representadas nas figuras 5.9, 5.10, 5.11 e 5.12. Para cada nuvem de pontos foi avaliada influência dos parâmetros do algoritmo na sua eficiência na deteção de zona acessível e inacessível, os resultados estão sumariados nas tabelas 5.3, 5.4, 5.5 e 5.6.

Analisando a tabela 5.3 que representa a variação do tamanho de células quadradas, mantendo constante R=0.4 m e $\sigma_0 = 2 \cdot Th$, é possível verificar que células de dimensões reduzidas, 10 a 20 cm, e células de maiores dimensões, superiores a 50 cm, apresentam menor eficiência na deteção de zonas acessíveis e inacessíveis. Além disso, células de dimensões reduzidas requerem um maior capacidade computacional e representam de forma mais irregular as zonas em estudo, como se pode visualizar na figura 5.13 (a); analisando a figura 5.13 (b) é possível visualizar que células de grandes dimensões reduzem significativamente o espaço navegável, podendo inclusive em algumas situações tornar intransponível a passagem do veículo devido à influência de obstáculos.

Por análise da tabela 5.4 que representa a variação do tamanho de células não quadradas, mantendo constante R=0.4 m e $\sigma_0 = 2 \cdot Th$, verifica-se que não existe uma melhoria por se utilizar células não quadradas, existindo em algumas situações menor eficiência do algoritmo na deteção de zonas acessíveis e inacessíveis.

Analisando a tabela 5.5 que representa a variação do o raio, R, de procura da vizinhança para o cálculo das normais da nuvem de pontos, mantendo constante dimensões da célula 0.35×0.35 m e $\sigma_0 = 2 \cdot Th$, verifica-se que aumentando o raio aumenta a eficiência do algoritmo na deteção de zona acessível, esta relação justifica-se pelo facto da nuvem de pontos ser esparsa e com o aumento do raio as normais ficam mais regulares, aumentado por isso a sua acessibilidade. No entanto, quando se pretende detetar zonas irregulares no terro, como buracos, a utilização de um raio elevado leva a uniformização das normais nessa zona diminuindo por isso a eficiência do algoritmo na deteção desse tipo de zonas. Por análise da tabela 5.6 que representa a variação da constante normalizadora, σ_0 , mantendo constante dimensões da célula 0.35×0.35 m e R=0.4 m, verifica-se que aumentando o valor de σ_0 existe uma ligeira melhoria na deteção de zonas acessíveis. No entanto, para valores superiores a $6 \cdot Th$ existe uma diminuição da eficiência do algoritmo na deteção de zonas inacessíveis.

Com base nos resultados descritos, foram selecionados os parâmetros que de um modo geral apresentaram uma melhor eficiência do algoritmo; R=0.4 m, $\sigma_0 = 4 \cdot Th$ e dimensões de célula de 0.35×0.35 m. Utilizando esses parâmetros foram criados os mapas de acessibilidade representados nas figuras 5.14, 5.15, 5.16 e 5.17. No apêndice B estão representados os mapas de elevação para as quatro nuvens de pontos estudadas.

Além disso for avaliado o tempo de processamento do algoritmo na criação dos mapas de acessibilidade, analisando a tabela 5.7 verifica-se que o cálculo das normais da nuvem de pontos demora cerca de 99% do tempo total de processamento do algoritmo.



(a) Cena A

(b) Cena ${\bf B}$



(c) Cena ${\bf C}$

(d) Cena ${\bf D}$

Figura 5.8: a) Imagem do ambiente em que existe um obstáculo negativo (buraco); b) imagem do ambiente em que existe uma subida com passeio; c) imagem do ambiente em que existe uma descida; d) imagem do ambiente em que existe um obstáculo positivo (poste de sinalização).



Figura 5.9: a) Nuvem de pontos da cena **A**; b) representação de um polígono a cor verde de uma zona acessível, a cor vermelha de uma zona inacessível (buraco).



Figura 5.10: a) Nuvem de pontos da cena **B**; b) vista lateral da nuvem pontos representada na figura 5.10 (a) ; c) representação de um polígono a cor verde de uma zona acessível, a cor vermelha de uma zona inacessível (passeio).

43



Figura 5.11: a) Nuvem de pontos da cena C; b) representação de um polígono a cor verde de uma zona acessível, a cor vermelha de uma zona inacessível (árvore).



Figura 5.12: a) Nuvem de pontos da cena **D**; b) representação de um polígono a cor verde de uma zona acessível, a cor vermelha de uma zona inacessível (poste de sinalização).

			Dimensã	o de célul	as quad	radas [m]	
		0.1	0.2	0.35	0.5	0.65	0.8
Cono A	% zona acessível detetada	64.41	71.17	76.63	77.54	78.14	72.19
Cella A	% zona inacessível detetada	0	10	66.66	25	50	0
Cone B	% zona acessível detetada	75.42	75.92	78.25	76.52	64.61	51.61
	% zona inacessível detetada	67.68	86.52	100	100	100	100
Cono C	% zona acessível detetada	67.81	58.11	56.91	58.06	57.35	53.79
Cella C	% zona inacessível detetada	75.93	84.84	100	100	100	100
Cono D	%zona acessível detetada	90.17	93.85	91.46	89.59	80.91	81.39
	% zona inacessível detetada	100	100	100	100	-	-

Tabela 5.3: Resultado da eficiência do algoritmo variando as dimensões de células quadradas, Sx = Sy.

Tabela 5.4: Resultado da eficiência do algoritmo variando as dimensões de células não quadradas, $Sx \neq Sy$.

		Dimens	são de células n	ião quadradas [m× m]
		0.2 imes 0.35	0.35 imes 0.2	0.35 imes 0.5	0.5 imes 0.35
Cone A	%zona acessível detetada	75.85	73.21	76.62	78.07
Cella A	%zona inacessível detetada	37.51	26.66	50	66.66
Cono D	% zona acessível detetada	78.03	77.64	77.75	75.37
	% zona inacessível detetada	97.59	90.58	96.96	100
Cone C	%zona acessível detetada	55.79	56.31	57.43	57.56
Cella C	% zona inacessível detetada	94.87	97.37	100	100
Cone D	%zona acessível detetada	93.82	91.08	91.08	88.11
Cena D	% zona inacessível detetada	100	100	-	100

Nota - Raio, R, em metros					
		R = 0.3	R = 0.4	R = 0.5	
Cono A	%zona acessível detetada	64.35	76.63	83.40	
Cena A	%zona inacessível detetada	66.66	66.66	16.66	l
Cone B	%zona acessível detetada	68.49	78.25	85.65	
	%zona inacessível detetada	100	100	93.88	
Cono C	%zona acessível detetada	44.71	56.91	66.71	
Cella C	%zona inacessível detetada	100	100	100	
Cone D	%zona acessível detetada	87.64	91.46	92.58	
\bigcirc \mathbf{D}	% zona inacessível detetada	100	100	100	

Tabela 5.5: Resultado da eficiência do algoritmo variando o raio, R, de procura da vizinhança para o cálculo das normais da nuvem de

pontos. **Nota** - Raio, R, em metros

Tabela 5.6: Resultado da eficiência do algoritmo variando a constante normalizadora, σ_0 , do valor de confiança, ν .

		$\sigma_0 = 2 \cdot Th$	$\sigma_0 = 4 \cdot Th$	$\sigma_0 = 6 \cdot Th$	$\nu = 1$
Cono A	% zona acessível detetada	76.63	79.26	80.27	82.05
Cena A	% zona inacessível detetada	66.66	66.66	50	16.67
Cono P	% zona acessível detetada	78.25	81.51	82.25	84.03
Cena D	% zona inacessível detetada	100	100	97.95	95.92
Cono C	% zona acessível detetada	56.91	59.25	61.46	62.57
Cena C	% zona inacessível detetada	100	100	95.24	95.24
Cono D	% zona acessível detetada	91.46	92.36	92.58	93.93
	% zona inacessível detetada	100	100	100	100

R = 0.6
87.81
0
90.38
71.43
71.69
100
91.15
100

	Tempo de cálculo das normais [s]	Tempo de total de processamento do algoritmo [s]
Cena ${\bf A}$	14.441	14.606
Cena ${\bf B}$	9.027	9.101
Cena ${\bf C}$	11.383	11.587
Cena ${\bf D}$	8.423	8.514

Tabela 5.7: Tempos de processamento do algoritmo na construção dos mapas de acessibilidade



Figura 5.13: Ilustração da influência do uso de células de reduzidas e elevadas dimensões, na construção do mapa de acessibilidade da nuvem de pontos da cena \mathbf{D} . a) dimensões de célula 0.1×0.1 m; b) dimensões de célula 0.8×0.8 m.

49



Figura 5.14: Mapa de acessibilidade da nuvem de pontos da cen
a ${\bf A};$ % zona acessível detetada - 79.25; % zona inacessível detetada - 66.66.



Figura 5.15: Mapa de acessibilidade da nuvem de pontos da cen
a ${\bf B};$ % zona acessível detetada - 81.51; % zona inacessível detetada - 100.



Figura 5.16: Mapa de acessibilidade da nuvem de pontos da cen
a ${\bf C};$ % zona acessível detetada - 60.22;% zona inacessível detetada - 100.


Figura 5.17: Mapa de acessibilidade da nuvem de pontos da cen
a ${\bf D};$ % zona acessível detetada - 92.35; % zona inacessível detetada - 100.

53

Capítulo 6 Conclusões e trabalho futuro

6.1 Sistema de aquisição 3D

O trabalho desenvolvido no sistema de aquisição 3D permitiu efetuar com sucesso o sincronismo entre as medições efetuadas pelo sensor 2D e sua posição angular externa. Esta alteração revelou-se um fator importante na melhoria da qualidade da nuvem de pontos que representa o ambiente. Foram apresentadas diferentes metodologias para a construção da nuvem de pontos, sendo a correção individual das medições do *scan*, utilizando as transformações calculadas nos instantes de tempo em que são publicados os ângulos ϕ , a que melhores resultados revelou na qualidade da nuvem de pontos.

Apesar das melhorias na qualidade dos dados do laser 3D, apresentando em algumas zonas qualidade semelhante ou superior aos dados do sensor Velodyne, o laser 3D ainda apresenta algumas limitações. Para construir uma representação tridimensional do ambiente com a densidade de pontos ilustrada na figura 3.9 (b), são necessários cerca de cinco segundos, este número é significativamente inferior quando comparado com a taxa de 1,3 milhões de pontos por segundo do sensor Velodyne. Ainda assim, não fica excluída a possibilidade da utilização do laser 3D numa aplicação dinâmica, podendo ser incorporada a transformação do *scan* utilizando a localização do veículo em cada instante em relação ao ambiente.

6.2 Deteção do espaço navegável

Neste trabalho foi implementado um método para deteção do espaço navegável usando informação 3D, criando um mapa de acessibilidade que pode ser usado para calcular a navegabilidade de uma qualquer trajetória.

O mapa de acessibilidade é calculado recorrendo a uma nuvem de pontos e suas propriedades, cota em z e normais. É associado a cada célula de um grelha bidimensional propriedades que permitem calcular a sua acessibilidade. O algoritmo para o cálculo da acessibilidade apresenta alguns parâmetros que influenciam a sua eficiência, verificandose que os parâmetros que têm maior influência na eficiência do algoritmo são as dimensões das células e o raio de procura de vizinhança para estimar as normais da nuvem de pontos. É possível concluir com base nos resultados apresentados na secção 5.2 que existe um gama de valores preferenciais para estes parâmetros, sendo cerca de 40 cm para o raio e cerca de 35 cm para a dimensão das células. Nos mapas de acessibilidade foram identificadas com taxas de sucesso superior a 60% as zonas em que o veículo pode circular (estrada plana, subida), bem como aquelas em que não pode fazer (obstáculos positivos e negativos, bermas, passeios); sendo de 78.33% e 91.67%, a taxa média de sucesso na identificação de zonas em que o veículo pode ou não circular, para as quatro situações estudadas neste trabalho

6.3 Trabalho futuro

Com a realização deste trabalho foi melhorado o laser 3D desenvolvido no LAR e foi integrado em ambiente ROS. Como foi mencionado na secção 3.4 pode existir alguns atrasos nas comunicações, e seria portanto interessante no futuro desenvolver um sistema que fundisse a informação do laser 2D e do codificador externo e só depois essa informação ser incorporada em ambiente ROS.

Seria também interessante tentar utilizar o laser 3D para ambientes dinâmicos. No entanto seria complicado obter perceção do ambiente dinâmico apenas recorrendo ao laser 3D, já que é necessário fazer uma acumulação dos dados do sensor, o que levaria em algumas situações a uma representação errada do ambiente. Mas pode-se colocar a hipótese de utilização do laser 3D para efetuar uma primeira reconstrução do ambiente e utilizar outros sensores mais rápidos na aquisição de dados, mas mais limitativos em termos de representação do ambiente, para focar em zonas de potencial interesse com base na informação do laser 3D e obter informação mais detalhada sobre essa mesma zona.

Em relação aos mapas de acessibilidade apresentados neste trabalho, seria interessante estudar a influência da utilização de uma grelha não regular, tornando assim mais intuitiva a avaliação da navegabilidade de trajetórias, já que o *AtlasCar* é um veículo não holonómico. Outra particularidade que não foi aprofundada neste trabalho foi o cálculo das normais da nuvem de pontos, para a qual foi utilizada uma função já disponível na biblioteca PCL; esse cálculo é a fase que maior recursos consome na aplicação do método apresentado neste trabalho. Uma possível solução seria criar uma função responsável por calcular a normal de cada célula utilizando apenas os pontos presentes nessa célula, ou caso se revele necessário utilizando também pontos em células vizinhas; em qualquer uma das situações o tempo de procura de vizinhos poderia diminuir, reduzindo assim o custo computacional.

Referências

- Atlas. Atlas project. http://atlas.web.ua.pt/, 2013. [Consultado em: Maio de 2013].
- [2] Pedro Salvado. *Reconstrução dinâmica de mapa local para o AtlasCar*. Tese de mestrado, Universidade de Aveiro, 2012.
- [3] R. Pascoal and V.M. Santos. Compensation of azimuthal distortions on a free spinning 2d laser range finder for 3d data set generation. In Atas do Encontro Científico do Festival Nacional de Robótica, 2010.
- [4] ROS. Ros. http://www.ros.org/wiki/, 2013. [Consultado em: Maio de 2013].
- [5] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.
- [6] Miguel Riem de Oliveira. Automatic Information and Safety Systems for Driving Assistance. Phd thesis, University of Aveiro, 2013.
- [7] Matthew S Roscoe, Kenneth B. Kent, Rainer Herpers, and Paul G Plöger. Determining the viability of the kinect as a research tool. http://www.cs.unb.ca/research-expo/2012/submissions/Matthew%20Roscoe_ mat.roscoe@unb.ca_Poster.pdf, 2013. [Consultado em: Maio de 2013].
- [8] Craig Glennie and Derek D. Lichti. Temporal stability of the velodyne HDL-64E s2 scanner for high accuracy scanning applications. *Remote Sensing*, 3:539–553, 2011.
- [9] Velodyne. Velodyne LiDAR. http://velodynelidar.com/lidar/lidar.aspx, 2013. [Consultado em: Maio de 2013].
- [10] Vladimir Koifman. Image Sensors World. http://image-sensors-world. blogspot.pt/2010/06/panasonic-announces-d-imager-3d-imager.html, 2013.
 [Consultado em: Maio de 2013].
- [11] MESA. MESA imaging. http://www.mesa-imaging.ch/prodview4k.php. [Consultado em: Maio de 2013].
- [12] Tiago Talhada. Perceção 3D utilizando uma câmara estéreo. Tese de mestrado, Universidade de Aveiro, 2012.

- [13] R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.
- [14] Patrick Pfaff, Rudolph Triebel, and Wolfram Burgard. An efficient extension of elevation maps for outdoor terrain mapping. In In Proc. of the Int. Conf. on Field and Service Robotics (FSR), page 165–176, 2005.
- [15] G. Tuna, K. Gulez, V.C. Gungor, and T. Veli Mumcu. Evaluations of different simultaneous localization and mapping (SLAM) algorithms. In *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, pages 2693–2698, 2012.
- [16] S. Zaman, W. Slany, and G. Steinbauer. ROS-based mapping, localization and autonomous navigation using a pioneer 3-DX robot and their relevant issues. In *Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International*, pages 1–5, 2011.
- [17] João Dias. Distanciómetro 3D baseado numa unidade laser 2D em movimento contínuo. Tese de mestrado, Universidade de Aveiro, 2009.
- [18] SICK. Telegrams for Operating/Configuring the LMS 2xx. http://www. sick-automation.ru/images/File/pdf/LMS%20Telegram%20Listing.pdf. Datasheet do sick lms200.
- [19] SICK. Technical description. http://sicktoolbox.sourceforge.net/docs/ sick-lms-technical-description.pdf, 2013. [Consultado em: Maio de 2013].
- [20] A. Harrison and P. Newman. High quality 3D laser ranging under general vehicle motion. In *IEEE International Conference on Robotics and Automation*, 2008. ICRA 2008, pages 7–12, 2008.
- [21] P. Derenick and T. Miller. The sick lidar matlab/c++ toolbox. http:// sicktoolbox.sourceforge.net, 2008. [Consultado em: Maio de 2013].
- [22] Microchip. Pic18f2331/2431/4331/4431 datasheet. http://ww1.microchip.com/ downloads/en/DeviceDoc/39616d.pdf. Datasheet para PICs da Microchip.
- [23] R.B. Rusu and S. Cousins. 3D is here: Point cloud library (PCL). In 2011 IEEE International Conference on Robotics and Automation (ICRA), pages 1–4, 2011.
- [24] Radu Bogdan Rusu. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. Phd thesis, Institut f
 ür Informatik der Technischen Universit
 ät M
 ünchen, 2009.
- [25] Joel Pereira. Autonomous parking using 3D perception. Master's thesis, University of Aveiro, 2012.
- [26] Kevin Lai and Dieter Fox. Object recognition in 3D point clouds using web data and domain adaptation. International Journal of Robotics Research, 29(8):1019–1037, 2010.
- [27] MathWorks. MathWorks. http://www.mathworks.com/help/stats/ksdensity. html, 2013. [Consultado em: Maio de 2013].

Apêndice A Circuito microcontrolador

Esquema elétrico do circuito do microcontrolador com incorporação do pulso de sincronismo proveniente do laser planar 2D.



A.Circuito microcontrolador

Apêndice B Mapas de elevação

Representação dos mapas de elevação para as cenas estudadas no capítulo 5.



Figura B.1: Mapa de elevação da cen
a ${\bf A}$



65





Figura B.4: Mapa de elevação da cen
a ${\bf D}$

Apêndice C

Launch file módulo navigability_map

Launch file para iniciar o módulo navigability_map. Deve ser definido o tópico para subscrever uma nuvem de pontos; e alterar, caso se pretenda, os valores definidos por defeito para os parâmetros na criação dos mapas de acessibilidade.

```
<launch>
```

```
<group ns="/navigability_map">
       <!-- Remmaping topics-->
       <remap from="/pointcloud0" to="/snr/las3d/fus/pointcloud"/>
       <node name="nav_map" pkg="navigability_map" type="nav_map">
               <!--
                      PARAM
                                      -->
               <param name="output_frequency" value="200.0"/>
               <!--
                      PointCloud distance filters
                                                    -->
               <param name="PointCLoud_Xmin_filter" type="double"</pre>
                  value="0"/>
               <param name="PointCLoud_Xmax_filter" type="double"</pre>
                   value="25.0"/>
               <param name="PointCLoud_Ymax_filter" type="double"</pre>
                   value="25.0"/>
               <param name="PointCLoud_Ymin_filter" type="double"</pre>
                   value="-25.0"/>
               <param name="PointCLoud_Zmax_filter" type="double"</pre>
                   value="2"/>
               <param name="PointCLoud_Zmin_filter" type="double"</pre>
                   value="-10.0"/>
               <!-- Cells dimension
                                          -->
               <param name="Grid_Sx" type="double" value="0.35"/>
               <param name="Grid_Sy" type="double" value="0.35"/>
               <!--
                      Normal Estimation
                                              -->
               <param name="Radius_neighbors" type="double" value="0.4"/>
               <param name="K_neighbors" type="int" value="20"/>
               <!-- 1 - Use radius search method;
                   0 - Use neighbours search method -->
               <param name="Use_Radius_Search" type="int" value="1"/>
               <!--
                      Acessibility thresholds
                                                       -->
               <param name="Zmax_heigh_difference" type="double"</pre>
                   value="0.1"/>
               <param name="angleX_max_difference" type="double"</pre>
                   value="0.2"/>
               <param name="angleY_max_difference" type="double"</pre>
                   value="0.2"/>
               <param name="angleZ_max_difference" type="double"</pre>
                  value="0.2"/>
               <!-- Confidence value for a cell with 1 point
                                                                     -->
               <param name="default_confidence" type="double"</pre>
                   value="0.5"/>
               <!--
                      Normalization constants for the confidence -->
               <param name="Standard_Deviation_max_confidence"</pre>
```

```
type="double" value="0.4"/>
<param name="Standard_Deviation_anglex_max_confidence"
   type="double" value="0.8"/>
<param name="Standard_Deviation_angley_max_confidence"
   type="double" value="0.8"/>
<param name="Standard_Deviation_anglez_max_confidence"
   type="double" value="0.8"/>
<!-- 1 - Z accessibility;
   2 - Angle X accessibility
   3 - Angle Y accessibility
   4 - Angle Z accessibility
   5 - Total accessibility -->
<param name="debug_accessibility" type="int" value="5"/>
```

</node>

</group>

</launch>