



Instituto Universitário de Lisboa

Departamento de Ciências e Tecnologias da Informação

Deteção e Seguimento de Trilhos em Ambientes Florestais com Recurso a Um Sistema de Visão Monocular

André Filipe Gomes da Silva

Dissertação submetida como requisito parcial para obtenção do grau
de Mestre em Engenharia de Telecomunicações e Informática

Supervisor

Prof. Doutor Pedro Figueiredo Santana, Professor Auxiliar
ISCTE - Instituto Universitário de Lisboa

Outubro, 2017

"Standing on the shoulders of giants."

Bernard of Chartres

Resumo

Esta dissertação propõe um sistema de apoio à navegação baseado em visão monocular passível de ser instalado em veículos aéreos não tripulados de pequena e média dimensão, concebidos para realizar missões em ambiente florestal (e.g., busca e salvamento). A navegação autónoma em ambiente florestal apresenta um conjunto de desafios que vão desde a complexidade dos obstáculos presentes (e.g., árvores) até ao ruído provocado pela vegetação nos sinais de radiofrequência tipicamente usados para controlo remoto ou auto-localização (GPS). Neste contexto, o trabalho proposto nesta dissertação aborda a questão da utilização de trilhos para navegar de forma segura neste tipo de ambientes, explorando o facto dos trilhos poderem ser caracterizados como sendo vias de comunicação terrestre, que se apresentam por norma livres de obstáculos. Para tal, é estendida a técnica de deteção e seguimento de trilhos proposta por Santana et al. [68], à qual é adicionada informação tridimensional do ambiente para aumentar a sua robustez na presença de trilhos mais desafiantes. Concretamente, a informação tridimensional do ambiente obtida através de sequências de imagens capturadas por uma câmara monocular equipada no veículo aéreo é utilizada para modular essa mesma técnica de deteção de trilhos. Os resultados experimentais, obtidos a partir de um conjunto de 12 vídeos filmados com recurso a uma câmara montada num veículo aéreo não tripulado de pequena dimensão, tele-operado, mostram a capacidade do sistema proposto na identificação da região hipótese do trilho, tendo sido obtida uma taxa de sucesso de 97,78%, sendo superior aos 90.30% obtidos pelo sistema que serviu de base a este trabalho.

Palavras-chave: Reconstrução Tridimensional, Deteção de Obstáculos, UAV, Deteção e Seguimento de Trilhos, PCL, SLAM, Trilhos;

Abstract

This dissertation proposes a support system to navigation based on monocular vision system susceptible of being installed in unmanned small and medium-sized aerial vehicles, built to perform missions in forest environments (e.g., search and rescue). Autonomous navigation in forest environments presents a set of challenges from the complexity of existing obstacles (e.g., trees) to the noise caused by vegetation in the radiofrequency signals typically used for remote control or auto-location (GPS). In this context, this dissertation addresses the use of trails to navigate safely in this kind of environments, exploring the fact that trails can be characterized as means of terrestrial communication, normally free of obstacles. For that, the detection and trail following technique proposed by Santana et al. [68] is used, to which tridimensional environment information is added, to increase its sturdiness upon more challenging trails. More specifically, the environment's tridimensional information obtained through sequences of images captured by a monocular camera equipped in the aerial vehicle is used to adjust that trail detection technique. The experimental results, obtained via a set of 12 videos recorded with a camera installed in a tele-operated, unmanned small-sized aerial vehicle, show the capacity of the proposed system in identifying the trail path, achieving a success rate of 97,78%, meaning a higher success rate than the 90.30% achieved with the system used as basis for the work developed in this dissertation.

Keywords: 3D Reconstruction, Obstacle Detection, UAV, Path Detection and Following, PCL, SLAM, Trails.

Agradecimentos

Em primeiro lugar gostaria de agradecer a minha família, pelo suporte e motivação que me foram transmitindo ao longo deste percurso e pela paciência que tiveram para me ajudar a ultrapassar as dificuldades encontradas ao longo desta fase da minha vida. De seguida gostaria de agradecer à Introsys, pela disponibilidade demonstrada e por ter fornecido todas as ferramentas necessárias para a realização do trabalho proposto nesta dissertação. Em especial gostaria de agradecer a toda a equipa do departamento de Investigação, Desenvolvimento e Inovação, em particular à Raquel Caldeira e ao Magno Guedes pelo apoio, motivação e orientação. Gostaria igualmente de agradecer ao Prof. Doutor Pedro Santana pela orientação e motivação, mas também pelos conselhos e auxílio na resolução de problemas que foram surgindo no decorrer do trabalho desenvolvido nesta dissertação. Por fim gostaria de agradecer a todos aqueles que de forma direta ou indireta, ajudaram-me a concluir este percurso.

Conteúdo

Resumo	v
Abstract	vii
Agradecimentos	ix
Lista de Figuras	xiii
Acrónimos	xv
Símbolos	xvii
1 Introdução	1
1.1 Motivação	1
1.2 Enquadramento	2
1.3 Objetivos e Solução Proposta	4
1.4 Questão de Investigação	5
1.5 Método de Investigação	6
1.6 Organização do documento	7
2 Revisão da Literatura	9
2.1 Métodos para deteção de estradas	9
2.2 Métodos para deteção de trilhos	15
2.2.1 Métodos baseados em técnicas de segmentação	15
2.2.2 Métodos baseados em aprendizagem	19
2.2.3 Método baseados em agentes	21
2.2.4 Métodos baseados em reconstrução tridimensional	25
2.3 Considerações	26
3 Sistema Proposto	29
3.1 Visão geral do sistema	30
3.2 Sistema de aquisição de informação tridimensional	33
3.3 Processamento da informação tridimensional	36
3.3.1 Segmentação do plano do chão e obstáculos	40
3.3.2 Projecção dos obstáculos no plano da câmara	44
3.4 Integração da informação tridimensional no detetor de trilhos	45

3.5	Algoritmo para validação da presença de trilho	49
4	Resultados Experimentais	53
4.1	Instalação experimental	53
4.1.1	Calibração da câmara monocular	55
4.1.2	Parametrização do sistema	56
4.2	Conjunto de dados	58
4.3	Análise de Resultados	58
4.3.1	Avaliação da primeira hipótese (R1) em estudo	60
4.3.2	Avaliação da segunda hipótese (R2) em estudo	61
4.3.3	Avaliação da terceira hipótese (R3) em estudo	61
4.3.4	Avaliação da quarta hipótese (R4) em estudo	65
4.3.5	Avaliação da quinta hipótese (R5) em estudo	69
4.3.6	Casos de falha	71
5	Conclusões e Trabalho Futuro	75
5.1	Conclusões	75
5.2	Trabalho Futuro	77

Lista de Figuras

1	Aparências de trilhos em diferentes ambientes.	3
2	Segmentação da estrada com base na técnica de <i>Region Growing</i> aplicada a imagens de cor.	12
3	Identificação do ponto de fuga de acordo com Rasmussen et al. [55].	13
4	Método bio-inspirado para detecção e seguimento de estradas proposto por Broggi et al. [11].	14
5	Descrição visual do funcionamento do algoritmo proposto por Bartel et al. [3].	17
6	Segmentação do trilho segundo o método descrito por Rasmussen e Scott et al. [60].	18
7	Os diferentes passos no processo de segmentação do trilho nos métodos propostos por Rasmussen et al. [56] e Blas et al. [5].	19
8	Imagem retirada do trabalho proposto por Giusti et al. [32].	21
9	Visão geral do sistema proposto por Santana et al. [68]	24
10	Possível trajetória percorrida por um agente virtual.	24
11	Resultado do sistema proposto por Santana et al. [68].	25
12	Resultado do sistema proposto por Rasmussen et al. [61].	26
13	Visão geral do Sistema.	31
14	Exemplos da reconstrução tridimensional obtida por ambos os sistemas em ambiente exterior.	34
15	Nuvens de pontos obtidas sem aplicação de filtros.	36
16	Visão geral sobre os processos utilizados para processamento da informação tridimensional.	37
17	Referencial da câmara em relação ao referencial mundo.	37
18	Cálculo da diferença entre normais.	40
19	Segmentação do plano do chão.	43
20	Exemplos de nuvens de pontos segmentadas.	43
21	Modelo <i>pinhole</i> da câmara utilizado para projeção dos pontos no plano da câmara.	44
22	Projeção dos pontos 3D no plano da câmara.	45
23	Processo de influência no comportamento das agentes virtuais de forma a prevenir a seleção de regiões identificados como obstáculos.	48
24	Ângulo de inclinação, Θ , da câmara em relação ao plano do chão. .	49

25	Possível integração do sistema proposto como auxílio à navegação de um UAV.	50
26	Interceção entre a região hipótese do trilho e as regiões identificadas como obstáculos.	51
27	Descrição dos nós ROS utilizados pelo sistema proposto.	55
28	Sub-conjunto de imagens utilizadas para calibrar a câmara utilizada para adquirir o conjunto vídeos utilizado para validar o sistema proposto.	56
29	Plataforma UAV utilizada na aquisição dos vídeos de teste usados para validar o sistema proposto, e exemplos dos percursos realizados pelo UAV.	59
30	Exemplo de reconstrução tridimensional do LSD-SLAM.	61
31	Sub-conjunto de resultados que demonstra a reconstrução tridimensional do ambiente.	62
32	Sub-conjunto de resultados típicos que demonstra a correta identificação dos obstáculos próximos do robô.	63
33	Dados obtidos referentes ao ângulo de inclinação da câmara em relação ao solo.	64
34	Imagens retiradas de testes realizados no Vídeo 6.	67
35	Imagens do resultado obtido pelo detetor original numa situação de falha.	67
36	Caso de falha do detetor original na sequencia de imagens do Vídeo 6.	68
37	Resultado obtido na identificação da região hipótese do trilho por vídeo.	69
38	Sub-conjunto de resultados típicos que demonstram o funcionamento do módulo Validador.	71
39	Casos de falha identificados durante a realização dos testes.	73

Acrónimos

UAV	Unmanned Aerial Vehicle
GPS	Global Positioning System
DARPA	Defense Advanced Research Projects Agency
IMU	Inertial Measurement Unit
RADAR	Radio Detection And Ranging
SCARF	Supervised Clustering Applied to Road Following
UNSCARF	Unsupervised Clustering Applied to Road Following
LADAR	Laser Detection And Ranging
(RGB)	Red, Green, Blue
(DDN)	Deep Neural Network
GS	Go Straight
TR	Turn Right
TL	Turn Left
SGBM	Semi-Global Block-Matching Algorithm
UGV	Unmanned Ground Vehicle
SLAM	Simultaneous Localization and Mapping
LSD-SLAM	Large-Scale Direct Monocular SLAM
CPU	Central Processing Unit
ROS	Robot Operating System
ORB-SLAM	Oriented FAST and Rotated BRIEF - SLAM
REMODE	Regularized Monocular Depth Estimation
GPU	Graphics Processing Unit
DoN	Difference of Normals
SVD	Singular-Value Decomposition

RANSAC	Random Sample Consensus
NF	Neural Field
RAM	Random-Access Memory
PCL	P oint Cloud Library
OpenCV	Open Source Cumputer Vision Library
BSD	Berkeley Software Distribution
SOC	Sytems-on-a-Chip

Símbolos

k_{saltos}	Número de saltos de um agente virtual (pixeis visitados)
h_{max}	Altura máxima
$f_i[k]$	Função de cálculo do depósito de feromona virtual num determinado salto
NF	Filtro temporal
$\mathbf{p}$	Um determinado ponto numa nuvem de pontos
$\mathbf{S}$	Transformação composta por translação, rotação e escala
$\mathbf{S}^{-1}$	Transformação inversa de $\mathbf{S}$
$\mathbf{R}$	Matriz de rotação
$\mathbf{t}$	Vetor de translação
SO(3)	Grupo de rotações tridimensionais
s	Escalar de escala
c_x	Coordenada x do centro do sensor ótico da câmara
c_y	Coordenada y do centro do sensor ótico da câmara
f_x	Distância focal em x do sensor ótico da câmara
f_y	Distância focal em y do sensor ótico da câmara
$\mathbf{p}$	Ponto $\mathbf{p}$ pertencente a uma nuvem de pontos
$\mathbf{p}'$	Ponto p transformado
C_i	Referencial da câmara
W	Referencial mundo
p_{D_i}	Distância inversa ao ponto $\mathbf{p}$
p_x	Coordenada x do ponto $\mathbf{p}$
p_y	Coordenada y do ponto $\mathbf{p}$
p_r	Informação de cor vermelha no ponto $\mathbf{p}$

p_g	Informação de cor verde no ponto $\mathbf{p}$
p_b	Informação de cor azul no ponto $\mathbf{p}$
P_j	Nuvem de pontos instantânea
μ	Média
σ	Desvio padrão
k	Valor escalar de grandeza
d_p	Distância média ao ponto p
P	Nuvem de pontos
P_A	Nuvem de pontos acumulada
P'_A	Nuvem de pontos P_A após ser aplicado um filtro
P''_A	Nuvem de pontos P'_A após ser aplicado um filtro
τ	Conjunto de nuvem de pontos instantâneas P_j acumuladas que compõem P_A
α	Valor escalar
$\mathbf{n}$	Vetor normal
P_k	Nuvem de pontos composta pelos k vizinhos de um determinado ponto $\mathbf{p}$
$\bar{\mathbf{p}}$	representa a centróide dos k vizinhos mais próximos de $\mathbf{p}$
$\mathbf{v_p}$	Ponto de vista
$\hat{\mathbf{n}}$	Vetor unitário
r	Raio de uma esfera em torno de um ponto $\mathbf{p}$
r_s	Raio pequeno utilizado no cálculo das DoN
r_l	Raio grande utilizado no cálculo das DoN
h	Plano hipótese
H	Conjunto de planos hipótese h
P_h	Nuvem de pontos que pertencentes ao plano h
d_{plano}	Limiar da distância ao plano
$d(\mathbf{p}, h)$	Função que permite obter a distância de um ponto $\mathbf{p}$ a um plano h
$score(h)$	Função que permite obter a pontuação de um determinado plano h
b	Plano h com maior pontuação
b'	Versão refinada de b

P_b	Conjunto de pontos <i>inliers</i> de b
P'_b	Conjunto de pontos <i>inliers</i> de b'
(a, b, c, d)	Coefficientes da equação do plano
P_I	Nuvem de pontos que contem os pontos <i>inliers</i> a um plano
P_c	Nuvem de pontos que contem os pontos caracterizados como chão
P_{obs}	Nuvem pontos que contem os pontos caracterizados como obstáculo
π	Plano do chão dado definido por (a, b, c, d)
$d_m(P_I, \pi)$	Função que calcula a distância média de todos os pontos contido em P_I em relação ao plano do chão π
t_d	Limiar utilizado para criar as nuvens de pontos P_c e P_{obs}
v	Escalar definido empiricamente
(u, v)	Par de coordenadas de um pixel no referencial da câmara
(x, y)	Par de coordenadas num determinado referencial
$mobs$	Máscara de obstáculos
mht	Máscara da região hipótese do trilho
p_o	Número de pixeis descritos como obstáculo
i	Índice
j	Índice
$\varepsilon(x, y)$	Função que permite avaliar se um determinado pixel, nas coordenadas (x, y) , se encontra descrito em simultâneo em NF e m_{obs}
$NF(x, y)$	Função que permite obter o nível de atividade dos agentes virtuais num determinado pixel no par de coordenadas (x, y)
$m_{obs}(x, y)$	Função que permite determinar se um pixel nas suas coordenadas (x, y) é descrito como obstáculo
Θ	Ângulo de inclinação da câmara
h_{ini}	Valor inicial definido para a altura máxima
β	Valor escalar definido empiricamente
Q	Representa o conjunto de pixeis visitados pelos agentes virtuais
$\mathbf{q}$	Representa um pixel no par de coordenadas (x, y)
$\phi(\mathbf{q})$	Função que permite avaliar se um determinado pixel, nas coordenadas (x, y) , se encontra descrito em simultâneo em m_{ht} e m_{obs}

q_t	Limiar definido empiricamente
-------	-------------------------------

Capítulo 1

Introdução

1.1 Motivação

Nos últimos anos foi possível observar um enorme avanço no desenvolvimento de tecnologias para plataformas aéreas não tripuladas, denominadas de UAV (*Unmanned Aerial Vehicle*) para uso militar e comercial [48, 18]. Mais recentemente, com a generalização e aparecimento de UAV de pequena e média dimensão, que ao contrário dos seus antecessores concebidos e otimizados para operar a grandes altitudes e percorrer longas distâncias, estão preparados para operar próximo do solo onde é mais propício encontrar espaços complexos, sejam eles interiores ou exteriores [75, 50, 24, 33, 78, 76, 16]. Este tipo de plataformas torna-se assim ideal para utilização em ambientes onde plataformas terrestres apresentam uma maior dificuldade de acesso, como florestas. Um UAV ao ser capaz de seguir trilhos numa floresta, permite que sejam exploradas grandes áreas a baixa altitude de forma segura, partindo do pressuposto que tipicamente os trilhos apresentam uma densidade de vegetação reduzida. A utilização de técnicas para navegação baseadas em visão, como a proposta por Santana et al. [68] para deteção e seguimento de trilhos em robôs terrestres, estendida e aplicada ao caso dos UAV nesta dissertação, vem dar resposta às necessidades de navegação segura em ambientes florestais onde, na maioria dos casos, a navegação por GPS é insuficiente [80]. Os

sistemas baseados em visão permitem também evitar o elevado consumo energético de métodos sensoriais alternativos (e.g., sensores laser), aumentando assim a autonomia energética do veículo. Por outro lado, permite igualmente otimizar recursos como, por exemplo, os custos computacionais associados ao planeamento de trajetórias, uma vez que existem menos obstáculos (e.g., árvores) no caminho do UAV, não sendo necessário um planeamento recorrente de novas rotas. É ainda possível uma recolha mais detalhada de informação visual, que poderia ser ocultada pela copa das árvores ou de difícil perceção quando adquirida a grandes altitudes, como tipicamente necessário quando se recorre aos meios mais tradicionais como imagens aéreas ou de satélite. Esta solução pode então ser utilizada em aplicações que podem ir desde a busca e salvamento [33], ao inventario florestal [16], à monitorização ambiental [24] ou ao mapeamento de trilhos/caminhos [50].

1.2 Enquadramento

A identificação de caminhos como áreas propícias para navegação de veículos autónomos, sejam eles terrestres ou aéreos, teve um grande impulso quando em 2004 a Agência Norte Americana de Projetos de Pesquisa Avançada de Defesa (na sigla em inglês DARPA), lançou o primeiro desafio, o *DARPA Grand Challenge*, que tinha como principal premissa dotar um carro com a capacidade de navegar de forma autónoma em terreno irregular. O objetivo principal seria percorrer um trajeto pré-delineado e onde a única informação disponibilizada às equipas eram pontos de controlo, através de coordenadas GPS, por onde os carros tinham que passar. Do trabalho realizado pelas equipas de investigação presentes no desafio surgiram alguns trabalhos [1, 13, 21] que viriam mais tarde a servir de referência para o seguimento de estradas em ambientes rurais e mais tarde de trilhos.

Os trilhos são por definição considerados vias de comunicação terrestre que se apresentam por norma livres de obstáculos e permitem ligar dois ou mais pontos geográficos proporcionando a movimentação de pessoas e veículos de forma mais eficiente. Por conseguinte, podem ser considerados como casos particulares de

caminhos ou estradas. No entanto, toda a estrutura que demarca claramente uma estrada do seu meio envolvente pode não ser observada no caso de um trilho, nomeadamente a presença de pavimento e de marcas rodoviárias, o que dificulta a identificação dos seus limites ou aparência. Na Figura 1 são apresentados exemplos de trilhos em diferentes tipos de ambiente, onde se pode verificar a dificuldade inerente em identificar os limites ou forma dos mesmos devido à sua semelhança com a área envolvente.



(A) Trilho em deserto.



(B) Trilho em ambiente florestal.

FIGURA 1: Aparências de trilhos em diferentes ambientes.

A detecção de trilhos naturais surge assim como um tópico de investigação, que ao longo dos últimos anos tem vindo a ser desenvolvido [55, 3, 59, 66, 65, 68, 61], com o objetivo de dotar robôs terrestres todo-o-terreno com capacidades de navegação autónoma em ambientes exteriores complexos. A presença de trilhos naturais neste tipo de ambientes motivou a sua utilização como alternativa para auxílio à navegação, já que geralmente assumem uma estrutura saliente do meio envolvente o que permite que sejam utilizados como referência. Contudo, tipos de solo exigentes, declives acentuados ou a própria dimensão dos trilhos são fatores que podem limitar a utilização de robôs terrestres todo-o-terreno em ambientes florestais, surgindo assim a possibilidade de utilização de UAV como alternativa. Os UAV de pequena e média dimensão apresentam-se como plataformas leves, robustas e de baixo custo o que os torna uma boa alternativa para a realização de missões rápidas em ambientes florestais. A desvantagem deste tipo de plataformas é, sobretudo, a quantidade limitada de equipamentos que podem transportar devido às restrições de peso, capacidade energética, dimensão, além das limitações impostas ao sinal GPS neste tipo de ambiente, quando se deslocam abaixo do nível

das copas das árvores, o que faz da utilização de técnicas de visão por computador uma alternativa à navegação baseada em GPS.

Com vista a reduzir as limitações impostas à navegação baseada em GPS em UAV, um conjunto de técnicas alternativas têm vindo a ser desenvolvidas, como por exemplo técnicas de localização com recurso a telemetria laser [20], técnicas de visão por computador baseadas no cálculo de deslocamento através da análise de fluxo ótico [37] juntamente com aprendizagem supervisionada [62], técnicas baseadas somente em aprendizagem supervisionada [32], técnicas de localização com recurso a uma câmara monocular [43] e técnicas de reconstrução tridimensional para deteção de obstáculos [10]. O trabalho apresentado nesta dissertação pretende, assim, explorar a presença de trilhos naturais nas florestas de forma a permitir navegação segura de UAV.

1.3 Objetivos e Solução Proposta

O principal objetivo desta dissertação é o desenho e implementação de um sistema de apoio à navegação baseado em visão, passível de ser instalado em UAV de pequena e média dimensão, para que estes possam realizar missões em ambiente florestal. Para tal, esta dissertação estende a técnica para deteção de trilhos proposta por Santana et al. [68] de forma a aumentar a sua robustez na presença de trilhos mais desafiantes. Em particular, o objetivo principal é modelar essa mesma técnica de deteção de trilhos recorrendo a informação tridimensional do ambiente obtida através de sequências de imagens adquiridas com recurso a uma câmara monocular que equipa o UAV. Essa informação tridimensional permite, assim, detetar a presença de obstáculos no ambiente que se encontram descritos no campo visão do UAV e estimar a orientação da câmara em relação ao solo que é seguidamente utilizado para modular a técnica de deteção de trilhos [68] de forma a que esta não seja influenciada por esses mesmos obstáculos. O sistema proposto foi validado num conjunto de dados de teste, recolhidos previamente em contexto

real por um UAV de pequena dimensão, processados através de uma Unidade de Processamento Central (CPU¹).

1.4 Questão de Investigação

Tendo em conta o caso específico de aplicação do sistema proposto nesta dissertação a UAV de baixo custo, com recursos limitados, para navegação autónoma, perto do solo e em ambiente florestal, foi decidido usar como única referência para a deteção e seguimento de trilhos, uma câmara monocular. Com o intuito de validar a aplicabilidade desta solução, podem ser definidas cinco questões de investigação às quais o trabalho desenvolvido deve responder:

1. É possível, recorrendo a uma câmara monocular montada num UAV obter uma correta reconstrução tridimensional de ambientes não estruturados, onde poderão ser encontrados trilhos?
2. É possível identificar objetos caracterizados como obstáculos a partir da reconstrução tridimensional obtida pelo UAV que possam influenciar negativamente o seguimento de trilhos?
3. É possível obter uma estimativa quanto à orientação da câmara em relação ao solo tendo por base a reconstrução tridimensional obtida pelo UAV?
4. É possível obter uma solução integrada através da inclusão de informação volumétrica do ambiente que permita melhorar a deteção e seguimento de trilhos em ambiente florestal?
5. É possível obter um grau de certeza quanto à correta identificação da região hipótese do trilho?

¹CPU, que significa no seu acrónimo em inglês *Computer Process unit*.

1.5 Método de Investigação

O modelo de investigação seguido na realização da presente dissertação seguiu o modelo *Design Science Research* proposto por Peffers et al. [52], onde foi seguida uma abordagem centrada no desenho e desenvolvimento. Com esta abordagem pretende-se dar continuidade a algo que já existe mas que ainda não se encontra completamente desenvolvido, com o principal objetivo de contribuir para o avanço das técnicas de deteção de trilhos e dotar assim UAV de pequena e média dimensão, e não só, de capacidades de navegação segura em ambientes exteriores e que por sua vez podem ser utilizados para auxiliar seres humanos nas suas funções (Secção 1.1). Esta abordagem pode então ser dividida nas seguintes atividades:

1. Desenho e desenvolvimento - determinar quais as funcionalidades pretendidas e desenhar arquitetura do sistema. Para isso será necessário adquirir conhecimento na área de atuação, neste caso específico visão por computador, para de seguida poder contribuir com novas técnicas ou melhorar as já existentes (Capítulo 2 e Capítulo 3);
2. Demonstração - esta atividade tem como principal objetivo validar através de testes em ambiente real ou simulado o correto funcionamento da solução alcançada (Capítulo 4);
3. Avaliação - pretende validar os resultados obtidos e comparar com os de soluções anteriores e/ou com os requisitos definidos inicialmente. Estes resultados poderão ser obtidos estatisticamente validando o número de casos de sucesso e insucesso, neste caso em particular da correta deteção dos trilhos (Capítulo 4 e Capítulo 5);
4. Comunicação - Por fim pretende-se disseminar o novo conhecimento gerado através da investigação, contribuindo desta forma para o avanço do estado-de-arte e para a base de dados de conhecimento da sociedade.

1.6 Organização do documento

Esta dissertação, em adição ao presente capítulo, é composta por cinco capítulos, organizados da seguinte forma:

Capítulo 2 apresenta uma breve revisão da literatura sobre as diferentes técnicas utilizadas para detecção e seguimento de estradas e trilhos;

Capítulo 3 descreve o sistema proposto que estende o trabalho desenvolvido por Santana et al. [68], ao qual é adicionada a capacidade de utilizar informação volumétrica sobre o ambiente;

Capítulo 4 apresenta a configuração experimental utilizada e os resultados obtidos a partir de um conjunto de vídeos utilizados para validar o sistema proposto;

Capítulo 5 apresenta um conjunto de conclusões ao trabalho desenvolvido no decorrer desta dissertação e as oportunidades de melhoria para trabalho futuro.

Capítulo 2

Revisão da Literatura

Este capítulo apresenta uma análise às tecnologias e métodos existentes utilizados para a deteção e seguimento de trilhos. Ao longo dos últimos dez anos verificou-se a adoção e criação de diferentes técnicas de identificação de trilhos com recurso a dados sensoriais. Apesar do vasto desenvolvimento e aperfeiçoamento destas técnicas, verifica-se que atualmente a solução para o problema em causa não se encontra ainda completamente resolvida. Estas técnicas têm como base o trabalho desenvolvido na deteção e segmentação de estradas [2], utilizado sobretudo nos sistemas de navegação autónoma e de auxílio à condução, que mais recentemente têm vindo a ser adotados e adaptados pelos fabricantes automóveis para integração nos seus produtos [30, 31, 47, 51, 6].

2.1 Métodos para deteção de estradas

A existência de diferentes tipos de estradas, pavimentadas ou de terra batida, faz com que a tarefa de identificar e extrair informação não seja igual para os diferentes casos. No caso das estradas pavimentadas, a sua aparência, a presença de marcas rodoviárias como linhas, setas ou guias e o facto de serem estruturas monótonas facilita o processo de identificação e seguimento das mesmas [64, 2, 41]. Por sua vez, estradas sem pavimento, como estradas de terra batida, apresentam

um conjunto diferente de desafios pois carecem de uma estrutura bem definida e da existência de pistas visuais de fácil identificação e extração. Stanley [74], o robô que ganhou o *DARPA Grand Challenge*, recorrendo a um complexo sistema de sensores composto por lasers, câmaras monoculares e binoculares, RADAR, GPS e uma unidade de medição inercial (IMU), foi capaz de navegar por um percurso acidentado sem qualquer intervenção humana cumprindo os 281 Km em menos de 10 horas. Este sistema conseguiu assim com sucesso identificar a estrada e utilizar os dados recolhidos para navegar, e de forma segura, executar a missão. Dahlkamp et al. [21] descreve o sistema de visão utilizado pelo Stanley. O sistema utiliza um laser para mapear e criar um modelo volumétrico (3D) da área imediatamente à frente do veículo, até uma distância máxima de aproximadamente 20 metros, que é utilizada para identificar se a superfície é plana e suscetível de ser percorrida pelo veículo, que juntamente com as imagens recolhidas pela câmara monocular segmenta e classifica a região que corresponde à estrada. A região da imagem classificada como estrada é então utilizada por um algoritmo de aprendizagem *on-line* para treinar um classificador, isto é, o sistema é treinado para classificar as regiões da imagem como "estrada"/"não-estrada" de forma autónoma através da informação recolhida pelo laser que é utilizada para refinar este classificador ao longo do tempo. O classificador treinado pelo sistema é depois aplicado à totalidade da imagem adquirida pelo sistema de visão, o que permite obter uma análise do caminho a percorrer até uma distância de 70 metros. Assim, para obter uma análise morfológica da estrada é necessário recorrer à fusão de informação sensorial de vários equipamentos o que torna o sistema pouco flexível e passível de ser utilizada em todo o tipo de veículos, como por exemplo num veículo aéreo não tripulado devido aos limites de carga ou ao elevado custo computacional associado.

Alon et al. [1] propõem um sistema semelhante, classificando regiões da imagem como "estrada" e "não-estrada" em conjugação com a segmentação dos limites da estrada. A classificação é feita através de um sistema de aprendizagem *off-line* que utiliza um conjunto pré-definido de imagens, onde as regiões já se encontram previamente classificadas como "estrada"/"não-estrada", para inferir através da segmentação das imagens capturadas pela câmara se o veículo está na estrada

ou não. A segmentação é feita recorrendo a filtros de textura à semelhança dos utilizados por Malik et al. [45]. Assim, o modelo criado pelo sistema parte do pressuposto que o aspeto da estrada é conhecido e que um conjunto de imagens segmentadas onde as regiões já se encontram classificadas é fornecido ao sistema.

As abordagens anteriormente apresentadas têm por base a aprendizagem de modelos da estrada que são depois utilizados pelos classificadores para identificar as regiões que pertencem à estrada e as que pertencem à sua envolvente. Isto é realizado através da verificação de quão bem estes modelos se correlacionam com os dados observados através dos sistemas de visão que equipam os robôs. Existem contudo casos onde não é possível aprender de forma robusta modelos das estradas, o que torna difícil a utilização dessas abordagens. Uma alternativa a estas abordagens pode passar pela utilização de técnicas como *Region Growing* para segmentar as regiões salientes da imagem, que podem ser classificadas como estrada. Esta técnica agrupa os píxeis em regiões homogéneas com base nas suas características (e.g., cor ou intensidade). Inicialmente é selecionada uma região da imagem e todos os píxeis vizinhos com características semelhantes (i.e., homogéneos) são agrupados, sendo que este processo é repetido até que todos os píxeis da imagem estejam classificados. Esta técnica funciona partindo dos pressupostos de que: a) a estrada apresenta uma diferença de cor em relação a sua envolvente e que esta é homogénea em toda a sua extensão, b) a estrada se afasta da câmara e assim é apresentado na imagem de baixo para cima, e c) o veículo encontra-se no centro da estrada e, portanto, a área da imagem localizada na parte inferior-centro contém uma fração da estrada. Consequentemente, um modelo da estrada pode ser criado aplicando a técnica de *Region Growing* nesta área da imagem. Fernandez et al. [27] e Chang et al. [12] recorrem a esta abordagem como base para os seus trabalhos. Contudo, a abordagem baseada em *Region Growing* tende a falhar sempre que a estrada e a sua envolvente apresentem características semelhantes (i.e., gradientes de cor semelhante) ou a não existência de estrada na região indicada a partir da qual é aplicada a técnica de *Region Growing*, pois é assumido que a estrada se apresenta inicialmente na parte inferior-centro da imagem.

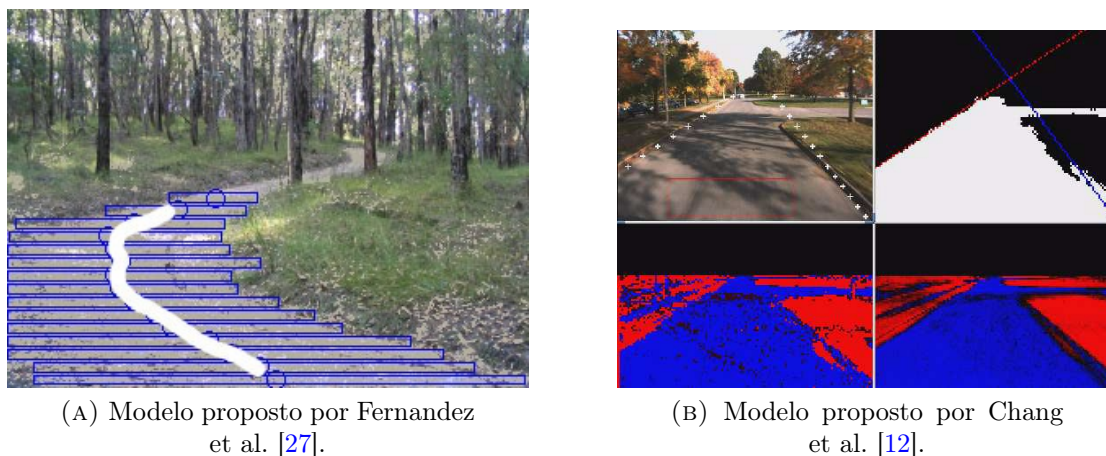


FIGURA 2: Segmentação da estrada com base na técnica de *Region Growing* aplicada a imagens de cor. (A) Os retângulos azuis representam os diferentes segmentos identificados como pertencentes à estrada e a linha branca representa a trajetória calculada com base nos segmentos; (B) O retângulo vermelho na primeira imagem define a região que contém parte da estrada, as linhas verde e vermelha na segunda imagem definem os limites da estrada e na terceira e quarta imagens a estrada é segmentada a azul e a área envolvente a vermelho.

Como solução alternativa, outros sistemas, como o *SCARF* proposto por Thorpe et al. [72] e o *UNSCARF* proposto por Crisman et al. [19], tentam obter o modelo da estrada através do reconhecimento de padrões na imagem. Para o conseguir, os sistemas agrupam os píxeis com gradientes de cor semelhante em regiões, para cada imagem capturada pela câmara. Após as regiões terem sido identificadas, os algoritmos escolhem o conjunto de regiões que de forma combinada apresentam a forma esperada para um caminho. Desta forma, esta abordagem evita que seja necessário manter um histórico sobre a informação da cor da estrada de imagem para imagem. Contudo, estes sistemas tornam difícil desambiguar situações onde mais do que uma região da imagem apresenta uma forma semelhante (i.e., ambas as regiões terem uma representação geométrica idêntica), o que pode acontecer em ambientes onde a estrada e a sua envolvente se fundem.

Como solução para este problema, Rasmussen et al. [55] propõem um método para identificar o ponto de fuga, isto é, o ponto no horizonte onde os limites laterais da estrada se intercetam, como forma de caracterizar a estrada. Para tal, a imagem é dividida em regiões mais pequenas onde são aplicados filtros que respondem a variações na textura. Estas variações permitem estimar uma

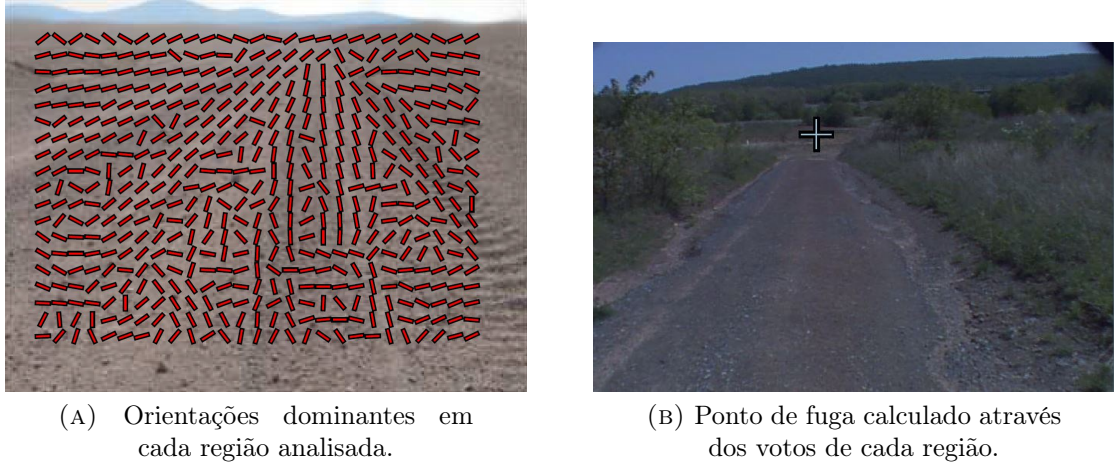


FIGURA 3: Identificação do ponto de fuga de acordo com Rasmussen et al. [55].

aproximação à direção dominante de cada região, que é mais tarde utilizada por um sistema de votos para estimar o ponto de fuga. O sistema permite assim identificar a direção que o veículo deve seguir de forma a percorrer a estrada que se apresenta à sua frente. Contudo, este sistema apresenta algumas desvantagens, como não ser possível identificar os limites da estrada.

Uma solução interessante para a segmentação de estradas é proposto por Broggi et al. [11], que recorre a um método bio-inspirado (i.e., inspirado na natureza) para identificar os limites do caminho e assim obter a sua orientação. O método utiliza como inspiração o comportamento das formigas [46], que à medida que avançam deixam feromonas para que outras formigas as sigam. Este comportamento é explorado por Broggi et al. [11] através da utilização de agentes [38] que reproduzem o comportamento das formigas. Para tal os agentes são organizados em duas colónias, responsáveis por identificar os limites laterais do caminho, à esquerda e à direita. Inicialmente é necessário identificar e definir as regiões da imagem onde serão depositados os agentes e quais os limites superior e inferior que restringem o movimento dos mesmos (ver Figura 4b). Assim, é feita uma segmentação à sequência inicial de imagens de forma a obter uma imagem monocromática com os limites do caminho. Estes últimos são usados para definir uma função heurística onde: (a) o nível de luminosidade dos píxeis pertencentes aos limites é diretamente proporcional à atração dos píxeis correspondentes na imagem original e (b) o custo de deslocamento em direção a um píxel na imagem original

é inversamente proporcional ao nível de luminosidade dos píxeis correspondentes pertencentes aos limites, isto é, influencia sobre o comportamento dos agentes. De seguida é definido um ponto de atração usado para influenciar o movimento dos agentes. De notar que esse ponto de atração é atualizado de imagem para imagem com base nos limites obtidos anteriormente. Após a configuração inicial, os agen-

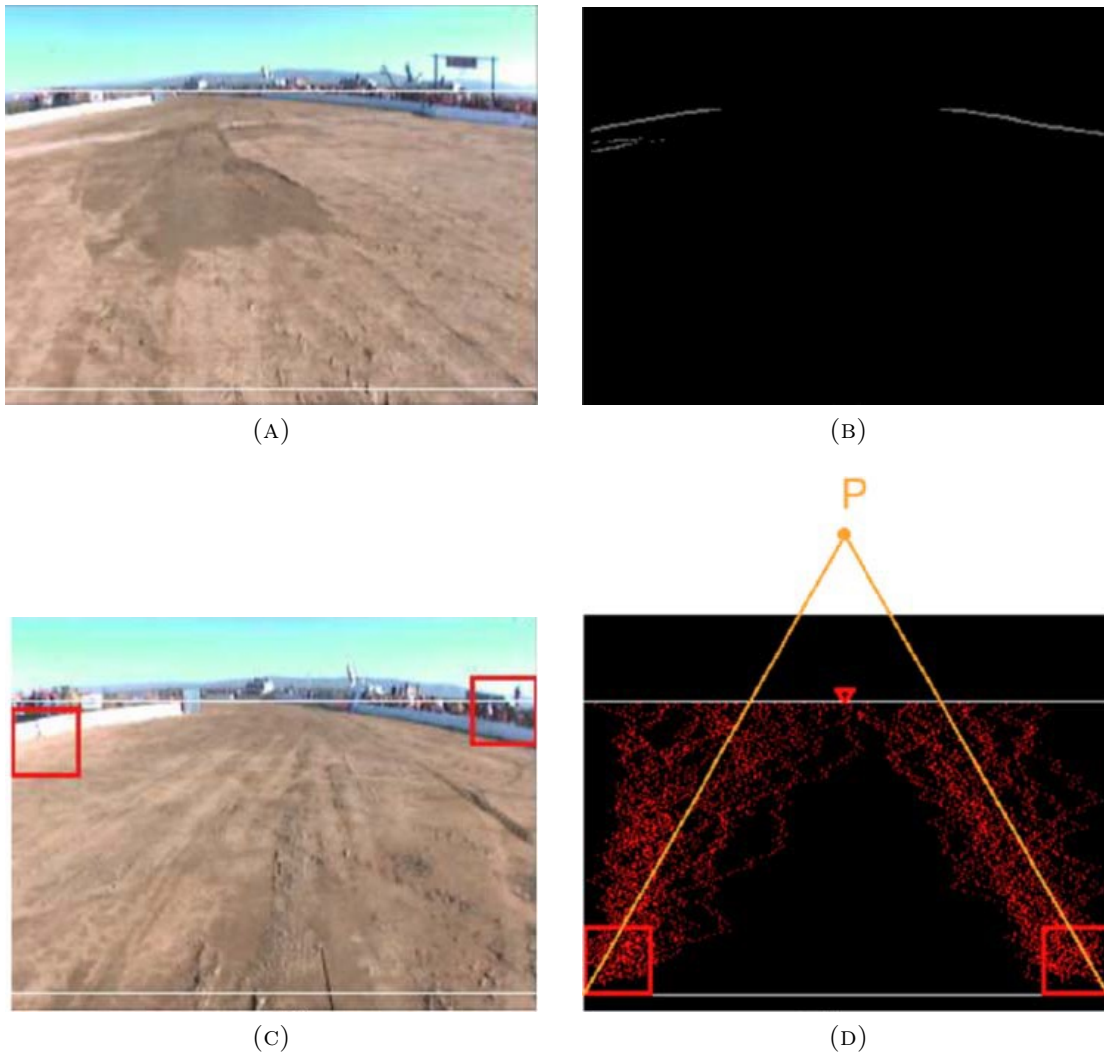


FIGURA 4: Método bio-inspirado para deteção e seguimento de estradas proposto por Broggi et al. [11] onde: (A) Imagem inicial com os limites superior e inferior; (B) Imagem monocromática com os limites obtidos através de técnicas de segmentação; (C) Os retângulos vermelhos representam as regiões onde são inicialmente depositados os agentes; (D) As manchas vermelhas representam os agentes das duas colónias e o ponto P indica o ponto de atração;

tes são depositados nas regiões definidas e divididos em sub-grupos que assumem diferentes comportamentos à medida que o sistema vai executando. Durante as primeiras iterações do sistema, o comportamento dos agentes é modelado pelas

heurísticas, mas à medida que o sistema vai executando o comportamento dos agentes passa a ser mais sensível às feromonas em detrimento das heurísticas. No final de cada iteração do sistema um agente é criado por colônia, movendo-se pixel a pixel de acordo com um comportamento modelado somente pelas feromonas, até chegar ao limite superior definido previamente. Assim o trajeto percorrido pelos dois agentes determina os limites laterais da estrada, que são usados na próxima iteração do sistema. Esta abordagem depende muito de uma segmentação bem sucedida dos limites da estrada para uma correta configuração do sistema e, como já referido anteriormente, em muitos casos isto não é possível devido à semelhança da estrada com as regiões envolventes.

2.2 Métodos para detecção de trilhos

Esta secção apresenta de modo mais detalhado as técnicas desenvolvidas especificamente para o problema de identificação de trilhos em sequências de imagens. Como já referido anteriormente, os trilhos carecem de estruturas bem definidas ou de elementos visuais de fácil identificação, como linhas, e ao contrario das estradas de terra batida a sua morfologia não é monótona o que apresenta um novo conjunto de desafios.

2.2.1 Métodos baseados em técnicas de segmentação

Os trilhos na sua maioria apresentam características que podem ser exploradas, com o objetivo de os identificar e destacar da área envolvente. À imagem das abordagens apresentadas anteriormente isto pode ser conseguido através de diversas técnicas de segmentação [44, 45, 15], que podem ser utilizadas separadamente ou em conjunto. Mesmo trabalhos que exploram outros métodos, como métodos baseados em aprendizagem ou recorrendo à utilização de agentes, utilizam como base técnicas de segmentação. Bartel et al. [3] propõem que a segmentação seja feita recorrendo à identificação dos limites do trilho, isto é, segmentação com base

na detecção de arestas. Para isso, é considerado que as regiões envolventes ao trilho são compostas, na sua maioria, por vegetação e, portanto, apresentam uma coloração verde. Com base nesta premissa todos os píxeis da imagem de cor verde (ver Figura 5a) são substituídos por píxeis de cor preta (ver Figura 5b) com o objetivo de aumentar o contraste do trilho em relação as restantes regiões da imagem. De seguida são aplicados três filtros, primeiro é aplicado um filtro *Gaussian Blur* para remover a presença de pequenas texturas (ver Figura 5c), o segundo filtro aumenta o contraste da imagem (ver Figura 5d) permitindo assim identificar o trilho como a região mais clara da imagem e, por último, é aplicado um filtro *Threshold* que converte todos os píxeis acima de um determinado valor de intensidade em píxeis de cor branca e abaixo desse valor em píxeis de cor preta (ver Figura 5e). Após serem obtidas as duas regiões trilho (branco) e não-trilho (preto) é aplicado um filtro *Gradient* que permite identificar os seus limites, isto é, as suas arestas (ver Figura 5f). Como forma de evitar a identificação de falsos positivos, isto é, sombras ou outro tipo de artefactos dentro da região delimitada pelo trilho, é aplicado um algoritmo que permite extrair o maior limite de um conjunto de limites dentro de uma determinada região. Contudo, esta abordagem tende a falhar sempre que a principal premissa, de que as regiões envolventes ao trilho são vegetação e que esta assume na maioria dos casos cor verde, não se verifica.

Alternativas ao problema da segmentação de trilhos sugerem a identificação de modelos do trilho com base na sua aparência, ou seja, através da cor, textura, contraste e forma. Neste contexto, Rasmussen et al. [60] propõe que o modelo do trilho seja extraído com base na informação de cor de cada pixel. Para isso o método utilizado por Rasmussen et al. [60] classifica inicialmente o terreno como plano, denso ou florestal com base em informação recolhida por um sistema *LADAR*¹. Após a classificação inicial, caso o terreno tenha sido classificado como denso ou florestal, os dados recolhidos pelo sistema *LADAR* são utilizados de forma exclusiva para identificar as zonas onde não existe vegetação ou obstáculos, que são assim consideradas como trilho. Por outro lado, se o terreno for classificado como plano, os dados recolhidos pelo sistema *LADAR* não são suficientes

¹LADAR, que significa no seu acrónimo em inglês LAsER Detection And Ranging, é um sistema ótico que permite medir distâncias através da projecção de feixes de luz.

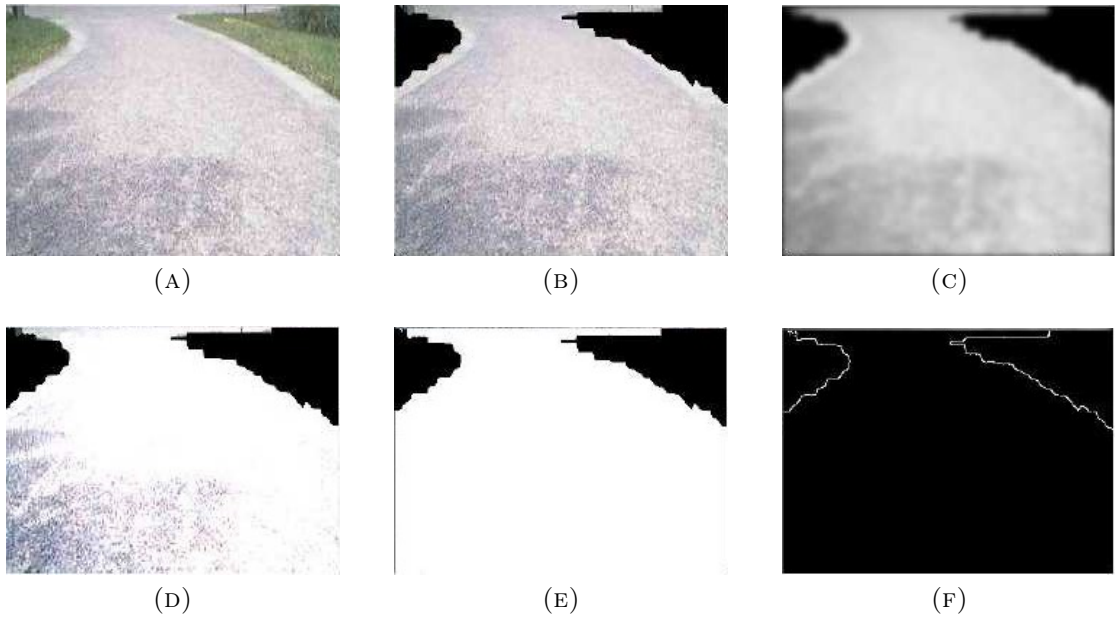


FIGURA 5: Descrição visual do funcionamento do algoritmo proposto por Bartel et al. [3] onde: (A) Imagem original do caminho; (B) Substituição dos píxeis verdes por píxeis pretos; (C) Aplicação do filtro *Gaussian Blur*; (D) Aplicação do filtro para aumento do contraste; (E) Aplicação do filtro *Threshold*; (F) Segmentação dos limites.

para determinar a presença de um trilho, dado que estes podem apresentar uma altura semelhante à sua envolvente quando não se encontra limitado por vegetação alta como arbustos ou árvores. Neste caso são utilizadas técnicas de segmentação com base na informação de cor de cada pixel. O processo de segmentação assume dois pressupostos: (a) o trilho encontra-se localizado no centro da imagem e (b) as áreas adjacentes à esquerda e à direita do trilho representam as regiões não-trilho. Partindo destes pressupostos o sistema aprende um modelo de cor das regiões não-trilho através de um histograma de cor (RGB), que é depois aplicado à totalidade da imagem de forma a classificar cada pixel como trilho/não-trilho de acordo com a semelhança das várias regiões da imagem relativamente ao modelo aprendido.

Os trilhos, sendo estruturas disruptivas no ambiente onde se encontram, podem ser analisados através das suas características geométricas. Para tal são utilizadas um conjunto de técnicas que tentam validar os modelos aprendidos por meio da segmentação das imagens recorrendo a modelos teóricos que descrevem a geometria expectável do trilho. Alguns trabalhos, como os desenvolvidos por

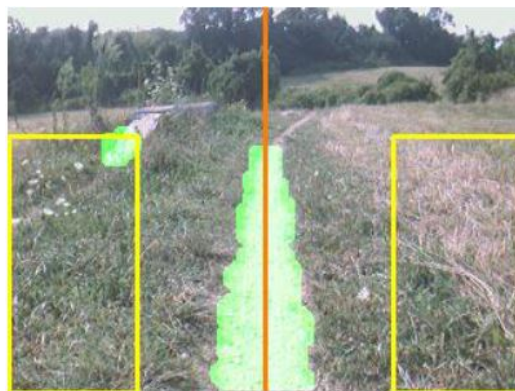


FIGURA 6: Segmentação do trilho segundo o método descrito por Rasmussen e Scott et al. [60]. As regiões delimitadas pelos retângulos amarelos representam as regiões não-trilho utilizadas como referencia para calcular o modelo de cor. As regiões pintadas de cor verde representam as regiões identificadas como trilho.

Rasmussen et al. [59, 56, 57] e Blas et al. [5] seguem esta linha de investigação. A ideia de que o trilho assume uma forma triangular quando visto em perspectiva é utilizada por Rasmussen et al. [56] para criar uma região hipótese que pode conter o trilho. Juntamente com esta região são definidas igualmente as regiões envolventes à esquerda e à direita do trilho, que idealmente apresentam características semelhantes, isto é, vegetação idêntica. Para segmentar o trilho da imagem é utilizado um filtro de partículas, onde cada partícula representa um conjunto de triângulos correspondentes à região pertencente ao trilho e às suas envolventes. Assim cada partícula tem a si associada um peso calculado através de uma função que avalia a sua semelhança a um modelo que caracteriza o trilho. O método procede de forma a afinar o conjunto de partículas para que o peso destas cresça progressivamente. A função para determinar quão adequado é o triângulo representado por uma dada partícula começa por catalogar cada pixel da imagem recorrendo a um conjunto de *textons* [81] que descrevem as características da textura da imagem em torno do pixel no espaço de cor *CIE-Lab*. Para isso é utilizada uma técnica de histograma baseada em *K-Mean Clustering* que é aplicada a todos os píxeis que não apresentem cores saturadas, de forma a obter um conjunto inicial de *textons* correspondentes a um pequeno grupo de cores presentes na imagem. Seguidamente, a mesma técnica é replicada mas desta vez para cores saturadas, obtendo assim o conjunto final de *textons*. Por fim é criado um modelo da distribuição de cor para cada região (ver Figura 7a), através de um histograma com

base nos *textons* encontrados anteriormente, que é usado para comparar com os histogramas das áreas associadas à partícula em questão e desta forma avaliar as semelhanças recorrendo à métrica *chi-square*. O resultado é obtido através de uma percentagem que indica o grau de certeza da presença de um determinado pixel na região referente ao trilho (T) ou à sua envolvente (Hl ou Hr).

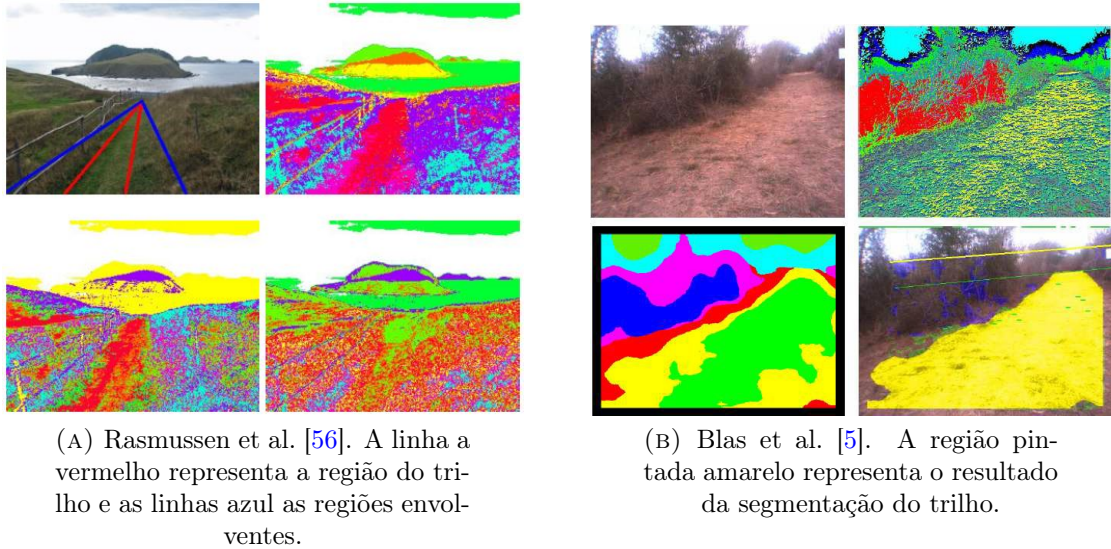


FIGURA 7: Os diferentes passos no processo de segmentação do trilho nos métodos propostos por Rasmussen et al. [56] e Blas et al. [5].

As principais limitações desta abordagem prendem-se com o facto de falharem a correta identificação do trilho em caso de interrupção visível no mesmo. Ou seja, situações onde o modelo inicial do trilho não é bem definido, isto é, a região indicada como hipótese, pode induzir em erro a função utilizada para avaliar a semelhança entre o modelo e as regiões segmentadas devido ao facto de que, nem sempre os trilhos podem ser abstraídos como formas triangulares.

2.2.2 Métodos baseados em aprendizagem

Os métodos que recorrem a técnicas de aprendizagem *of-line* caracterizam-se pela utilização de informação recolhida a cada iteração do sistema para refinar os modelos. Nos casos de estudo analisados, os modelos descrevem as características do trilho. Algumas das abordagens apresentadas anteriormente utilizam esta técnica,

tais como Rasmussen et al. [56] e Blas et al. [5], de forma a melhorar o desempenho dos seus algoritmos na segmentação e detecção do trilho face à restante área envolvente. É de referir também, que o trabalho proposto por Santana et al. [68] aplica técnicas de aprendizagem *on-line*, as quais são abordadas na Secção 2.2.3.

Por outro lado, as técnicas de aprendizagem *off-line* recorrem a conjuntos de dados previamente recolhidos e manualmente catalogados para treinar os seus modelos. Após os modelos terem passado pela fase de treino, esses modelos são utilizados pelos sistemas para reconhecerem um conjunto pré-definido de características e assim inferir ações a tomar como, por exemplo, quais as regiões da imagem que devem ser segmentadas ou quais as regiões da imagem que apresentam características semelhantes às do trilho. Recentemente, Giusti et al. [32] propôs um método capaz de avaliar a orientação do trilho através de imagens capturadas por uma câmara monocular. Para tal, Giusti et al. [32] treinou um classificador através de uma técnica de aprendizagem supervisionada, baseada em *Deep Neural Networks (DDNs)* [17], capaz de catalogar uma sequência de imagens em três classes, de acordo com a orientação aparente do trilho: GS - frente; TR - esquerda; e TL - direita (ver Figura 8). O conjunto de imagens utilizado para treinar o classificador foi obtido por uma pessoa que percorreu um total de sete quilómetros durante os quais foram gravados aproximadamente oito horas de vídeo com recurso a três câmaras dispostas em diferentes ângulos, uma delas na direção do movimento e as outras duas com um azimute de aproximadamente 30° e -30°, respetivamente, em relação à orientação da câmara principal. Desta forma as imagens obtidas em cada câmara foram catalogadas de acordo com a sua orientação (frente, esquerda ou direita), onde as imagens capturadas pela câmara frontal, na direção do movimento, caracterizam a aparência do trilho (GS), enquanto que as câmaras laterais caracterizam as regiões envolventes ao trilho (TR e TL). O resultado do classificador é interpretado então como uma probabilidade de a imagem obtida durante a execução do sistema pertencer a uma das três classes, e desta forma corrigir o movimento do veículo de forma a este se manter centrado no trilho.

Este método pode mostrar-se eficaz na orientação de um veículo ao longo do trilho, mas necessita que o classificador seja treinado através de um vasto conjunto

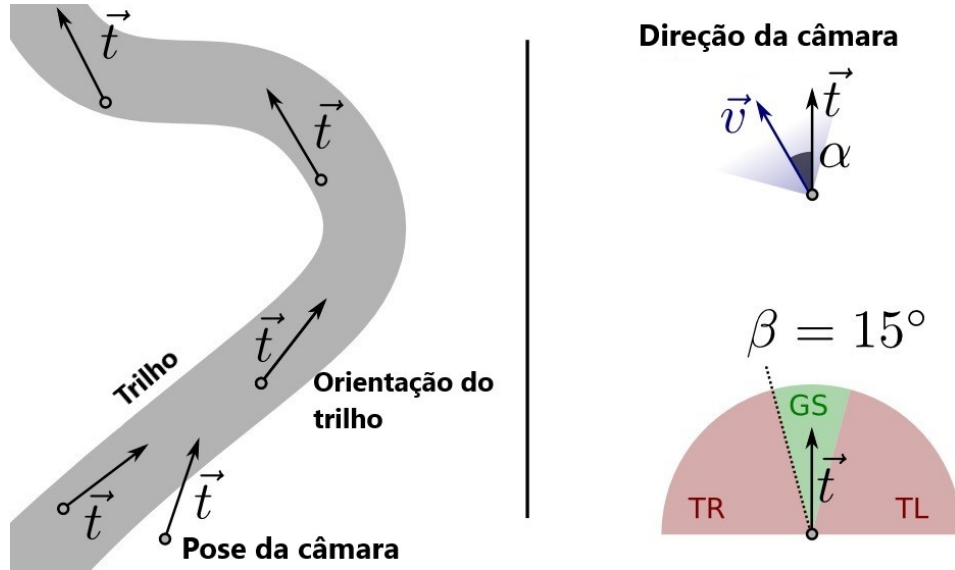


FIGURA 8: Imagem retirada do trabalho proposto por Giusti et al. [32]. À esquerda uma representação dos vetores direção $\vec{t}$ que indicam a orientação do trilho a seguir pelo robô. À direita uma representação das regiões GS, TR e TL que representam as classes treinadas pelo *DDN* que correspondem a ações dependendo do ângulo α entre $\vec{v}$ (pose atual do robô) e $\vec{t}$ (pose do trilho).

de imagens de forma a ser capaz de distinguir as diferentes aparências que um trilho pode assumir. Consequentemente, só pode ser usado em ambientes semelhantes aos cobertos pelo conjunto de treino. Além disso, o classificador treinado através do *DDN* requer *hardware* especializado e tempo de treino, que no caso apresentado durou aproximadamente três dias recorrendo a uma máquina equipada com uma placa de aceleração gráfico *Nvidia GTX 580*, o que torna também esta solução computacionalmente exigente. De notar também, que este método não segmenta o trilho da sua envolvente, não sendo por isso possível efetuar um planeamento fino do movimento do robô nem integração com sistemas de deteção de obstáculos.

2.2.3 Método baseados em agentes

Como já referido, os trilhos surgem como estruturas disruptivas no ambiente em que se encontram, o que permite explorar a sua saliência nas imagens adquiridas pelo robô. Utilizando esta característica, Santana et al. [67, 68] propõem um método que explora modelos de Saliência Visual [8] como apoio a uma técnica de segmentação bio-inspirada no comportamento de insetos sociais [7].

No seguimento do modelo proposto por Broggi et al. [11], aplicado à segmentação de estradas na identificação dos seus limites, Santana et al. [68] propõe que um grupo de agentes virtuais, modelados por comportamentos semelhantes aqueles observados nas colónias de formigas [23], explorem as regiões da imagem que apresentam uma maior conspicuidade, ou seja, nas regiões que apresentam um maior contraste em relação as regiões envolventes. A Figura 9 ilustra a visão geral do modelo proposto por Santana et al. [68], que pode ver o seu funcionamento descrito da seguinte forma. Primeiro é aplicada uma técnica, baseada em modelos de Saliência Visual, que permite ao sistema identificar as regiões da imagem que apresentam uma maior conspicuidade (i.e., são mais salientes), e que por sua vez estão associadas a uma maior probabilidade de pertencerem ao trilho, como mostrado por Santana et al. [66]. Estas regiões são então representadas através de dois mapas de conspicuidade, que analisam a cor e a intensidade da imagem. De seguida, é aplicada uma técnica de segmentação bio-inspirada no comportamento das formigas a cada um dos mapas. Para cada mapa é depositado um grupo de agentes virtuais que ao explorarem as regiões mais salientes, saltando de pixel em pixel através de comportamentos modelados por um conjunto de regras pré-definido, dão origem a dois mapas de feromonas virtuais referentes aos respetivos mapas de conspicuidade (i.e., cor e intensidade), que descrevem as trajetórias realizadas por cada agente virtual. Uma trajetória é definida por um conjunto de saltos, k_{saltos} (ver 10). O agente pode não atingir o número máximo de saltos se entretanto atingir uma linha limite na imagem, definida por h_{max} . Esta linha define a altura máxima na imagem em que é expectável encontrar o "ponto de fuga" do trilho.

Para cada pixel analisado, ou seja, para o k -ésimo pixel visitado pelo agente virtual, é calculada a quantidade de feromona virtual a ser depositada nesse mesmo pixel através de uma função de avaliação, $f_i[k]$, que tem em conta a informação de conspicuidade e de feromonas virtuais já depositadas. O depósito de feromonas tem como principal objetivo permitir aos agentes virtuais interagirem entre si através de um mecanismo denominada de *stigmergy*, inicialmente observado por Grassé et al. [34], e que apresenta duas características principais: (a) é uma forma

de comunicação indireta, ou seja, os agentes virtuais não comunicam diretamente entre si mas através do depósito de feromonas virtuais nos respectivos mapas e (b) a informação é local, ou seja, apenas pode ser acedida por agentes virtuais que visitem o mesmo pixel, ou as suas imediações. De referir que, as feromonas virtuais de um determinado mapa podem influenciar o comportamento dos agentes virtuais que operam no outro mapa.

Os dois mapas de feromonas virtuais são então combinados de forma a obter um único mapa que representa a região hipótese do trilho (i.e., mais saliente). Esta hipótese é utilizada para atualizar um filtro temporal implementado como um campo neuronal onde é acumulada informação de histórico, entre imagens através de uma técnica de fluxo ótico, sobre a região hipótese do trilho. A informação de histórico acumulada no filtro temporal pode ser vista como a atividade dos agentes virtuais ao explorarem as regiões dos mapas de conspicuidade que se apresentam mais salientes. A informação de histórico, armazenada no filtro temporal, pode ser consultada através a função $NF(\mathbf{p})$, com $\mathbf{p} = (p_x, p_y)$, que retorna o valor de intensidade referente à atividade dos agentes virtuais num dado pixel $\mathbf{p}$. Assim, as regiões que sejam visitas por mais agentes virtuais apresentam uma maior probabilidade de pertencerem à região hipótese do trilho. Dado que o campo neuronal contém dinâmica, ou seja, não é calculado tendo em conta apenas a imagem atual, precisa de ser compensado sempre que a câmara se desloca. Este processo é realizado aplicando uma transformação ao campo neuronal de forma a compensar o movimento estimado através do cálculo do fluxo ótico entre imagens consecutivas.

Por último, um mecanismo de aprendizagem *on-line* é utilizado para desambiguar situações onde a informação de conspicuidade não é suficiente para identificar a região hipótese do trilho. Este mecanismo utiliza a informação recolhida e analisada durante a execução do sistema (i.e., modelos *bottom-up*) referente ao contraste e aparência do trilho para criar modelos *top-down* que são utilizados para melhorar o processo de segmentação. Uma descrição mais detalhada sobre o método aqui descrito, pode ser consultada através dos trabalhos [67, 68].

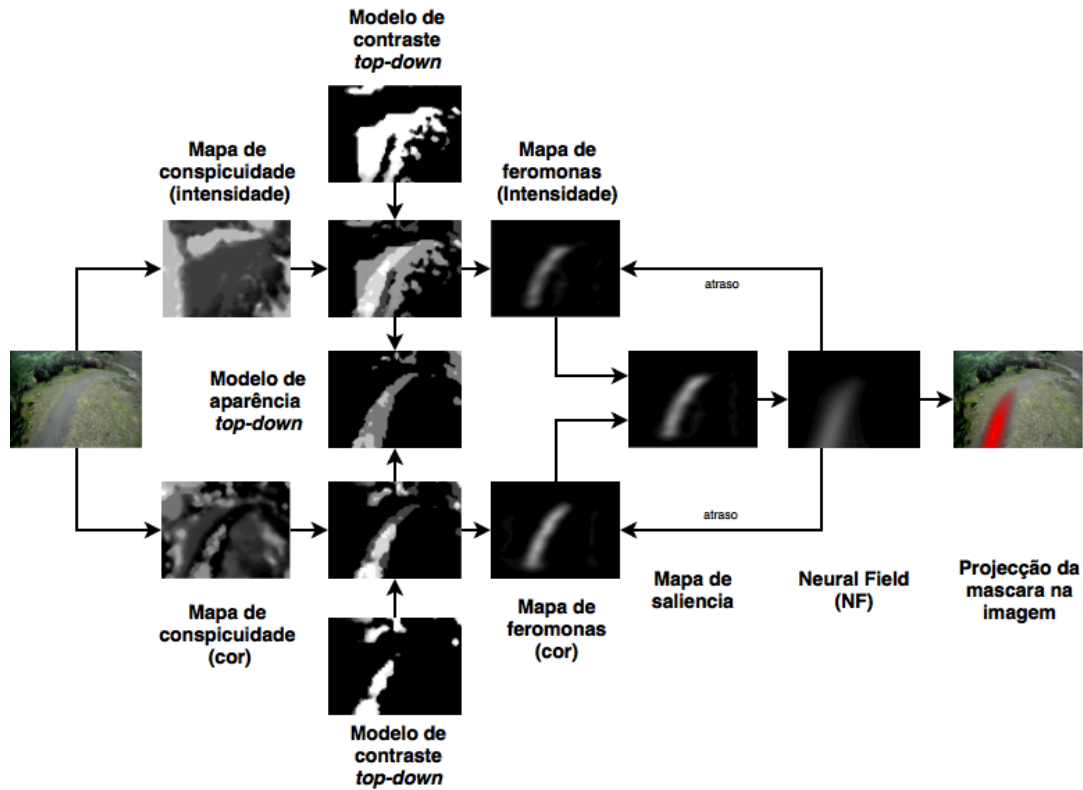


FIGURA 9: Visão geral do sistema proposto por Santana et al. [68]. Imagem adaptada de [68].

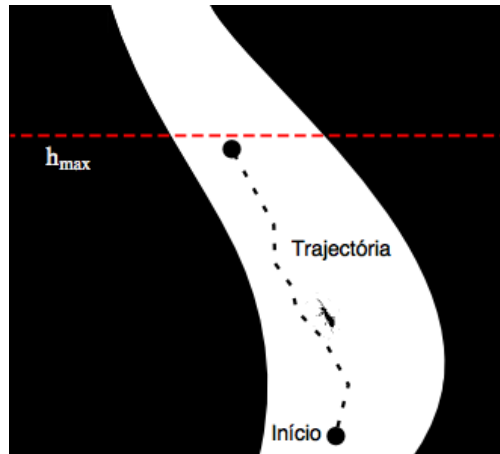


FIGURA 10: Possível trajetória percorrida por um agente virtual até um limiar, h , definido empiricamente e representado a tracejado vermelho. A trajetória do agente virtual, que representa os píxeis visitados pelo mesmo encontra-se descrito a tracejado branco.

Esta abordagem mostrou-se robusta e capaz de identificar uma grande variedade de trilhos em diferentes condições, como demonstrado em [68], mas ao assentar primariamente num modelo de Saliência Visual, onde a principal característica é a saliência do trilho, tende a falhar na presença de distratores, isto é,

elementos que sejam eles próprios salientes no ambiente como sombras ou árvores. A Figura 11 ilustra uma situação onde o detetor de trilhos ao ser influenciado por distratores no passado aprendendo a aparência desses mesmos distratores como sendo a do trilho e onde ao propagar este conhecimento, ao longo do tempo, leva à identificação regiões potencialmente perigosas para o robô.



FIGURA 11: Resultado do sistema proposto por Santana et al. [68] onde: (A) Segmentação do trilho com sucesso, região hipótese (a vermelho) projetada sobre o trilho; (B) Segmentação do trilho sem sucesso, região hipótese projetada na envolvente do trilho.

2.2.4 Métodos baseados em reconstrução tridimensional

A utilização de informação tridimensional para melhorar os métodos de segmentação para deteção de trilhos foi explorada inicialmente por Rasmussen et al. [58, 61] no seguimento dos trabalhos apresentados anteriormente. Mais concretamente, Rasmussen et al. [61] recorre a um sistema de visão binocular composto por duas câmaras omnidireccionais e um sistema *LADAR* composto por um laser *SICK LMS 291* para mapear a área envolvente ao veículo. De seguida, um mapa de profundidade é obtido aplicando o algoritmo *SGBM* [35] às imagens capturadas pelo sistema de visão binocular. O mapa de profundidade é então usado em combinação com o trabalho desenvolvido por Rasmussen et al. [57] com o objetivo de filtrar regiões que apresentem diferenças de altura em relação ao plano do chão. Por outro lado, o sistema *LADAR* é usado apenas para configurar o sistema de

visão sempre que este não seja mais capaz de identificar o trilho. Assim o método proposto por Rasmussen et al. [61] também se apresenta como uma solução sólida para a segmentação de trilhos mas necessita de um conjunto de sensores que torna o sistema pouco flexível e assim difícil de ser adaptado rapidamente em diferentes tipos de veículos (e.g., como veículos aéreos não tripulados). De referir, igualmente, que em simultâneo com o trabalho realizado nesta dissertação Fleischmann et al. [29] propôs um sistema baseado em visão binocular para deteção e seguimento de trilhos, em ambiente florestal, por meio de um Veículo Autónomo Terrestre (UGV²). O método proposto por Fleischmann et al. [29] recorre à informação tridimensional obtida através de uma câmara binocular para estimar o plano do chão que é seguidamente utilizado para, através de um conjunto de filtros, segmentar a região da imagem onde o trilho se encontra descrito. Contudo, o sistema apresenta como principal limitação a necessidade de um sistema de visão binocular, não respondendo assim aos requisitos do sistema proposto nesta dissertação.

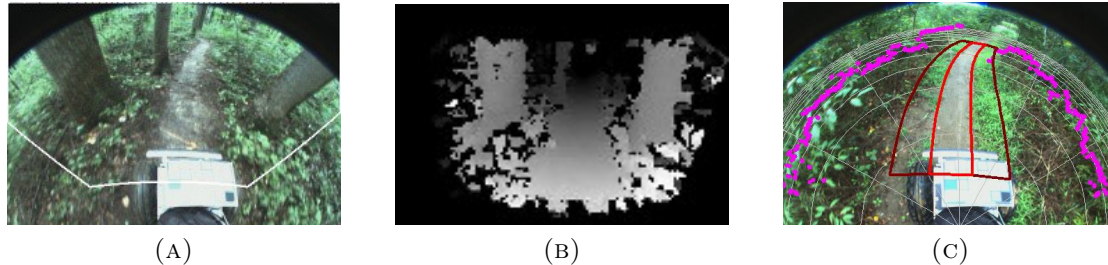


FIGURA 12: Resultado do sistema proposto por Rasmussen et al. [61] onde: (A) Imagem original do trilho obtida pela câmara binocular; (B) Mapa de disparidade obtido através do algoritmo *SGBM*; (C) Informação 3D obtida pelo laser e projetada na imagem original.

2.3 Considerações

Daquilo que pode ser observado no estado da arte atual e tendo em conta os requisitos propostos no problema de tese aqui abordado, conclui-se que os sistemas propostos por Rasmussen et al. [61] e Fleischmann et al. [29] demonstram que a

²UGV, que significa no seu acrónimo em inglês *Unmanned Ground Vehicle*

inclusão de informação tridimensional permite melhorar as técnicas de segmentação de trilhos, inspirando assim a sua inclusão no sistema proposto por Santana et al. [68]. O último, mostrou-se mais robusto e adequado aos requisitos do problema proposto, constituindo desta forma o trabalho que serviu de base a esta dissertação, onde a principal contribuição assenta na inclusão de informação tridimensional, sem recorrer a sensores adicionais para além da câmara monocular que alimenta o detetor de trilhos. Essa informação é obtida através de uma técnica de *SLAM*³ Monocular e é utilizada como forma de modular o comportamento dos agentes virtuais utilizados pelo detetor original [68], para que estes evitem regiões do espaço cuja volumetria indique como não pertencendo ao trilho, ou seja, potenciais distratores.

³O *SLAM* [73], que significa no seu acrónimo em inglês *Simultaneous Localization and Mapping*, é um problema de computação que tenta determinar a localização de um sensor, ou conjunto de sensores, no espaço ao mesmo tempo que cria um mapa que descreve a estrutura do ambiente onde se encontra.

Capítulo 3

Sistema Proposto

Este capítulo descreve o sistema proposto e desenvolvido no âmbito desta dissertação, que estende o trabalho desenvolvido por Santana et al. [68] na detecção e seguimento de trilhos. A esta metodologia é adicionada a capacidade de utilizar informação tridimensional sobre o ambiente como forma de identificar a presença de possíveis elementos distratores (e.g., vegetação alta) que possam influenciar o correto funcionamento do sistema original. De seguida na Secção 3.1 é apresentada uma visão geral do sistema onde são descritos os seus principais elementos, que por sua vez são detalhados nas secções seguintes, onde: a Secção 3.2 descreve o processo de aquisição da informação tridimensional, a Secção 3.3 descreve o processo de identificação e caracterização do plano onde se encontra o trilho e elementos distratores (i.e., possíveis obstáculos) que possam estar presentes no campo de visão do sistema. A Secção 3.4 descreve a integração da informação tridimensional no sistema proposto por Santana et al. [68]. A Secção 3.5 descreve o processo que permite avaliar se o sistema proposto ignora os elementos distratores identificados através da informação tridimensional adquirida.

3.1 Visão geral do sistema

O sistema proposto, apresentado de forma resumida nesta secção e detalhado nas seguintes, baseia-se no detetor de trilhos proposto por Santana et al. [68], denominado daqui para a frente como detetor original. Este último é capaz de explorar o facto dos trilhos serem salientes nas imagens adquiridas pela câmara do robô (i.e., destacam-se do fundo da imagem) para os detetar, aprender a aparência do trilho de forma a ajustar a detecção baseada na saliência visual e seguir o trilho ao longo do tempo através de um filtro temporal. Com estas características, o detetor original mostra-se robusto na presença de uma vasto conjunto de trilhos pouco estruturados. Contudo, a presença de elementos distratores no seu campo visual, isto é, elementos que apresentam uma maior saliência do que o próprio trilho, tal como troncos de árvore e certos tipos de vegetação densa, podem iludir o detetor original, levando-o a identificar erradamente essas regiões como pertencendo ao trilho. Para uma operação robusta, o robô tem de ser capaz de evitar estas situações e, quando isso não é possível, deve ser capaz de identificar a situação de falha para que o seu sistema de controlo reaja apropriadamente.

Com vista a reduzir o número de situações em que o detetor original falha, esta dissertação propõe a utilização de informação tridimensional, para informar o detetor sobre a presença de objetos que não podem ser considerados como pertencentes ao trilho, por exemplo, árvores e arbustos altos. Desta forma o detetor original deverá evitar considerar as regiões da imagem onde esses objetos se encontram, focando a sua operação nas restantes regiões, reduzindo assim a probabilidade do detetor original produzir resultados errados devido à presença de distratores no seu campo visual. De forma a manter o sistema compacto e flexível, isto é, manter a capacidade do sistema ser montado em robôs que tenham constrangimentos de peso, como é o caso de veículos aéreos não tripulados de pequena dimensão, foi escolhido um sistema de aquisição de informação tridimensional que utiliza apenas como recurso uma câmara de vídeo monocular. O detetor original depende de um conjunto de parâmetros que carecem de afinação empírica. O sistema aqui proposto procura mostrar que o esforço necessário a essa parametrização pode

ser reduzido ao fazer um desses parâmetros depender da pose da câmara, nomeadamente da sua orientação, sendo que a pose é estimada recorrendo ao mesmo mecanismo que permite determinar a volumetria do ambiente.

A Figura 13 apresenta um diagrama de blocos do sistema proposto nesta dissertação, organizado em quatro módulos principais. O primeiro módulo (ver Secção 3.2) é responsável pelo processo de reconstrução tridimensional do ambiente, realizada com recurso a uma técnica de localização e mapeamento simultâneos (SLAM) baseada em câmaras monoculares, nomeadamente o LSD-SLAM¹ [25]. O LSD-SLAM permite obter um conjunto de mapas de profundidade para um conjunto de imagens adquiridas pela câmara. Seguidamente, estes mapas de profundidade são utilizados para obter nuvens de pontos tridimensionais cujas coordenadas são descritas no referencial da câmara.

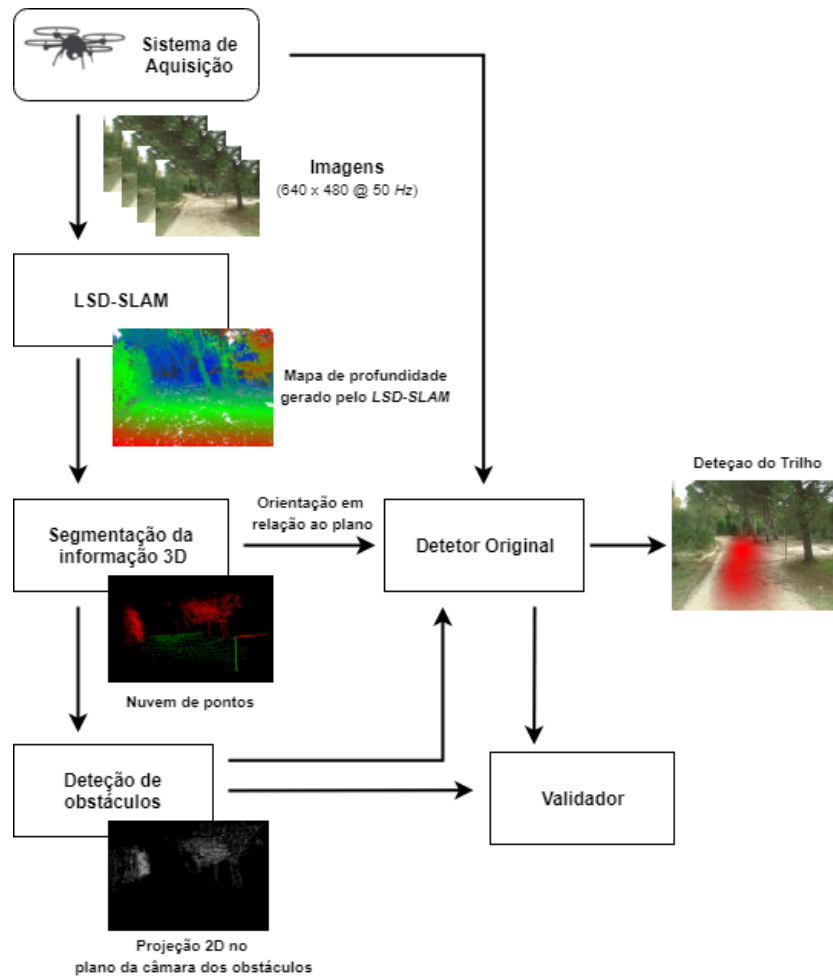


FIGURA 13: Visão geral do Sistema.

¹Large-Scale Direct Monocular SLAM

O segundo módulo (ver Secção 3.3) compreende um conjunto de processos utilizados para segmentar e analisar a informação contida nas nuvens de pontos obtidas anteriormente. Este módulo é responsável por fazer a segmentação dos elementos presentes em cada nuvem de pontos, que pode ser descrita em duas fases: na primeira, é estimado o plano do chão de forma a determinar os pontos que pertencem ao chão e que podem estar contidos na região onde o trilho se encontra descrito, e na segunda, são identificados todos os pontos que se encontram acima do plano do chão, que são classificados como possíveis obstáculos, ou seja, elementos distratores (e.g., árvores e vegetação alta). Os pontos tridimensionais correspondentes a obstáculos são seguidamente projetados na câmara de forma a produzir uma máscara bidimensional que sinaliza as regiões da imagem onde o detetor de trilhos funciona que contêm potenciais elementos distratores. Como dito anteriormente, o detetor recebe também informação relativa à inclinação da câmara, estimada a partir da inclinação relativa do plano do chão a esta.

O terceiro módulo (ver Secção 3.4), referente ao detetor de trilhos, utiliza os dados calculados no passo anterior, ou seja, a máscara bidimensional com as regiões obstáculo e o valor indicativo da inclinação do plano em relação ao plano do chão, para influenciar o comportamento do detetor original. Por último, o quarto módulo (ver Secção 3.5) descreve o processo de validação do sistema proposto, onde os resultados obtidos pelo detetor de trilhos são analisados tendo em conta as regiões obstáculo identificadas anteriormente, isto é, são verificados os casos onde existe uma interceção entre a região hipótese do trilho e as regiões caracterizadas como obstáculo. Esta informação pode então ser disponibilizada ao sistema de controlo de alto nível do robô como um grau de certeza da região hipótese do trilho ter sido corretamente identificada.

Nas secções que se seguem são apresentados de forma mais detalhada cada um dos módulos anteriormente descritos.

3.2 Sistema de aquisição de informação tridimensional

A reconstrução tridimensional do ambiente é obtida através do pacote de software LSD-SLAM [25], que implementa a técnica SLAM baseada em câmaras monoculares proposta por Engel et al. [26], e que se encontra disponível através de uma licença de código aberto². Em particular, o LSD-SLAM permite obter informação tridimensional do ambiente associada à pose da câmara, num dado instante, em tempo real com o processamento a ser efetuado num processador (CPU). Como forma de aceder a esta informação o LSD-SLAM disponibiliza um conjunto de interfaces de software, nomeadamente interfaces *ROS*³, que permite uma fácil integração com outros pacotes de software que partilhem igualmente destas interfaces, o que é o caso do detetor original [68] no qual esta dissertação se baseia. Existem outras soluções disponíveis para aquisição de informação tridimensional baseadas em técnicas SLAM, como é o caso dos sistemas ORB-SLAM [49] e REMODE [53], que partilham de características similares às do LSD-SLAM, sendo: (1) recurso a câmaras monoculares, (2) reconstrução tridimensional do ambiente, (3) pose estimada da câmara em relação ao mundo, (4) operação em tempo real e (5) interfaces *ROS*. Contudo, o sistema REMODE [53] depende da utilização de uma unidade de processamento gráfico (GPU) com suporte CUDA⁴, não sendo aplicável ao caso de estudo proposto nesta dissertação. Por outro lado, como referido por Mur-Artal et al. [49] o ORB-SLAM permite obter uma melhor estimativa da pose da câmara em relação ao ambiente em detrimento da reconstrução tridimensional da cena. A Figura 14 ilustra a reconstrução tridimensional obtida por ambos os sistemas, LSD-SLAM e ORB-SLAM, onde é possível verificar que a nuvem de pontos gerada pelo LSD-SLAM é mais densa (i.e., contém mais informação) que a nuvem de pontos gerada pelo ORB-SLAM. Tendo em conta os fatores analisados, o LSD-SLAM apresenta um conjunto de funcionalidades que vão de encontro às necessidades do

²LSD-SLAM: https://github.com/tum-vision/lsd_slam

³Robot Operating System - *Middleware* que permitem a comunicação e gestão de diferentes pacotes de software de forma distribuída

⁴CUDA - Plataforma de processamento paralelo disponibilizada pela Nvidia, para mais informações ver <https://developer.nvidia.com/cuda-zone>

sistema proposto, o que levou à sua escolha. De seguida é detalhada a forma como é obtida a nuvem de pontos a partir dos resultados obtidos através da utilização do LSD-SLAM.

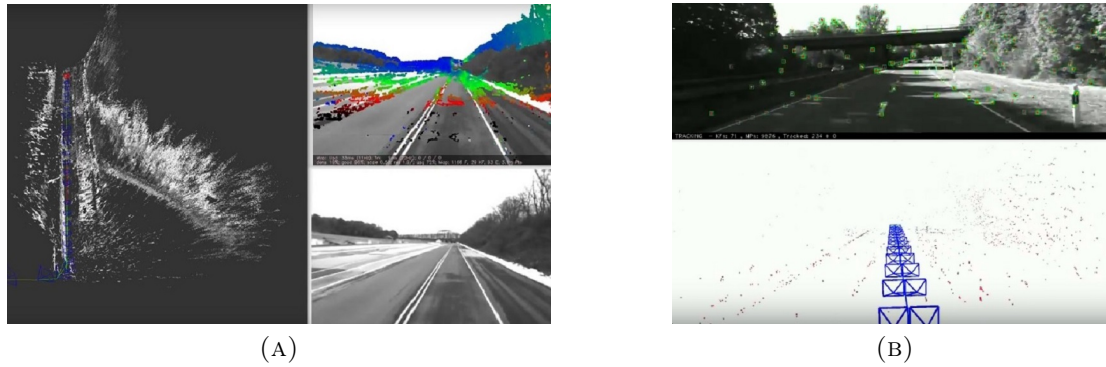


FIGURA 14: Exemplos da reconstrução tridimensional obtida por ambos os sistemas em ambiente exterior. É possível verificar que a nuvem de pontos gerada pelo LSD-SLAM [25] (A) é mais densa, e portanto contém mais informação, do que a nuvem de pontos gerada pelo ORB-SLAM [49] (B).

O LSD-SLAM é integrado como sendo um sistema de "caixa preta", isto é, o seu funcionamento interno não é tido em conta no âmbito desta dissertação, sendo apenas utilizado o resultado obtido através da sua utilização. Contudo, para que se compreendam as implicações da utilização deste sistema, apresenta-se de seguida uma breve descrição do seu modo de funcionamento. Cada imagem capturada é comparada com uma dada imagem de referencia previamente analisada e denominada de *keyframe* por forma a estimar a pose da câmara no referencial mundo. Se a imagem capturada for a primeira é considerada ela própria uma *keyframe*. As imagens analisadas são então utilizadas para criar ou atualizar mapas de profundidade associados à *keyframe* em questão. Uma nova *keyframe* é criada sempre que exista um suficiente deslocamento da câmara. Quando tal acontece a informação de profundidade da *keyframe* anterior é utilizada como base para a inicialização da nova *keyframe*. Quando uma nova *keyframe* é criada, a *keyframe* anterior é adicionada a um mapa global que mantém toda a informação capturada referente ao mundo, o que inclui a informação tridimensional de cada *keyframe* e a respetiva pose da câmara. Assim, o resultado do processamento do LSD-SLAM é composto por um conjunto de mapas de profundidade associados a cada *keyframe*

e a pose da câmara para cada uma dessas *keyframes*. O método exige que os parâmetros intrínsecos (e.g., distância focal) da câmara lhe sejam fornecidos, sendo estes tipicamente obtidos por calibração.

Um dado mapa de profundidade é composto por um conjunto de pontos dispostos no plano da *keyframe*. No fundo é como uma fotografia na qual existe informação de profundidade associada à informação de cor de cada pixel. Um dado ponto $\mathbf{p}$ é então representado por $\mathbf{p} = (p_x, p_y, p_{D_i}, p_r, p_g, p_b)$, onde p_x e p_y são as coordenadas cartesianas do ponto na imagem (coordenadas pixel), p_{D_i} representa o inverso da distância do ponto ao ambiente, ou seja, da profundidade, e p_r , p_g e p_b representam a informação de cor, no espaço de cor RGB, do pixel. Por outro lado, a pose da câmara é representada por $\mathbf{S} \in \mathbb{R}^3$ que descreve a rotação $\mathbf{R}$, escala s e a translação $\mathbf{t}$ da câmara no espaço tridimensional:

$$\mathbf{S} = \begin{pmatrix} s \cdot \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \text{ onde } \mathbf{R} \in \text{SO}(3), \mathbf{t} \in \mathbb{R}^3 \text{ e } s \in \mathbb{R}^+ \quad (3.1)$$

onde $\mathbf{R}$ é uma matriz ortogonal com $\det(\mathbf{R}) = 1$ (matriz de rotação), $s > 0$ é um número real positivo (escala) e $\mathbf{t}$ é um vetor (translação). Por último, temos os parâmetros intrínsecos da câmara onde f_x e f_y representam a distância focal e c_x e c_y representam o centro do sensor ótico da câmara. A forma de obter estes parâmetros é detalhada na Secção 4.1.1.

Estes dados são então utilizados para calcular a nuvem de pontos tridimensionais de cada mapa de profundidade. Por definição, uma nuvem de pontos P é composta por um conjunto de pontos $\mathbf{p} = (p_x, p_y, p_z)$ representados pelas suas coordenadas referenciadas num sistema de coordenadas comum no espaço tridimensional, e têm normalmente a sua origem no sensor utilizado na sua captura. Concretamente, um dado ponto $\mathbf{p} = (p_x, p_y, p_z)$ de um dado mapa de profundidade transforma-se num ponto tridimensional, $\mathbf{p}'$, da seguinte forma: $\mathbf{p}' = (\frac{p_x}{f_x} + c_x^{-1}, \frac{p_y}{f_y} + c_y^{-1}, p_{D_i}^{-1})$. De referir que as coordenadas deste ponto encontram-se descritas no referencial da câmara, C_i , aquando da aquisição da *keyframe* associada ao mapa de profundidade, que por sua vez, pode ser descrito no referencial

mundo, W , através da transformação S , definida na equação 3.1. Assim, após todos os pontos do mapa de profundidade, associado à *keyframe* em análise, terem sido transformados em pontos tridimensionais no referencial da câmara, obtêm-se uma nuvem de pontos instantânea, ou seja, P_j . Contudo, a nuvem de pontos obtida é esparsa e incompleta, não contendo informação suficiente que permita uma análise do ambiente (ver Figura 15), logo, o sistema mantém em memória um conjunto de τ nuvens de pontos (definido empiricamente) associadas às últimas *keyframes* analisadas, de forma a obter melhores resultados (i.e., uma melhor reconstrução) capazes de satisfazer as necessidades de processamento do sistema. Assim, temos que a nuvem de pontos acumulada, P_A , é uma coleção de nuvens de pontos instantâneas P_j e pode ser definida como: $P_A = P_{j-\tau} \cup \dots \cup P_j$.

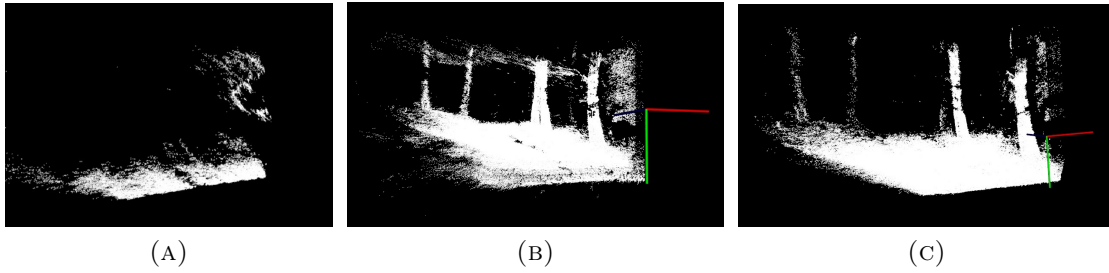


FIGURA 15: (A) Nuvens de pontos instantânea; (B e C) Nuvens de pontos acumuladas, compostas pelas nuvens de pontos referentes às últimas dez *keyframes*.

3.3 Processamento da informação tridimensional

Esta secção apresenta os processos aplicados a cada nuvem de pontos acumulada, com o objetivo de extrair um conjunto de dados que permita caracterizar os diferentes elementos presentes na mesma. Concretamente, a Figura 16 descreve os diferentes processos utilizados para processar a informação tridimensional obtida através do LSD-SLAM, onde no final é obtida uma máscara bidimensional com a representação das regiões da imagem (*keyframe*) identificadas como possíveis distratores (e.g., obstáculos) e o valor estimado da inclinação do plano do chão em relação à câmara no momento da aquisição da *keyframe*.

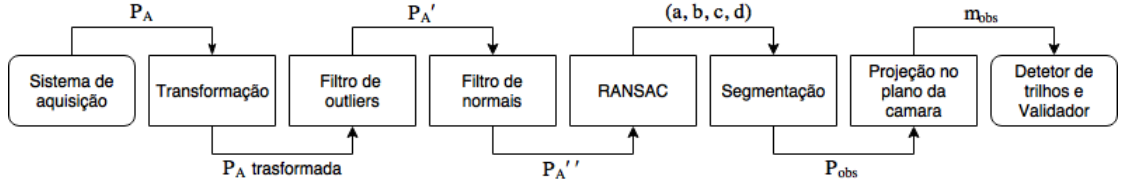


FIGURA 16: Visão geral sobre os processos utilizados para processamento da informação tridimensional.

Assim o primeiro passo após ser obtida a nuvem de pontos acumulada, P_A , associada às últimas *keyframes* analisadas, é aplicar uma transformação S^{-1} (inversa da transformação S descrita na equação 3.1), que permite que todos os pontos da nuvem de pontos P_A descritos no referencial mundo W , passem a ser descritos no referencial da câmara C_i .

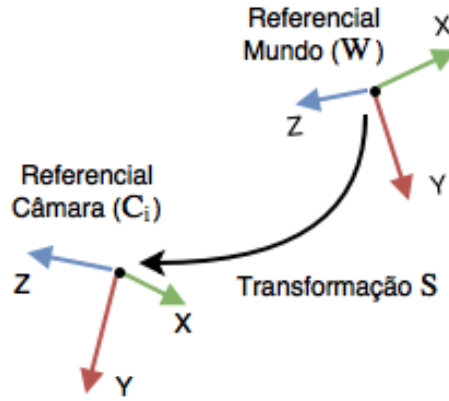


FIGURA 17: Referencial da câmara em relação ao referencial mundo.

A nuvem de pontos P_A pode ser constituída por milhares de pontos, o que torna a sua análise uma tarefa onerosa. Como forma de contornar este problema, são aplicados um conjunto de filtros que permitem diminuir a sua densidade sem perder informação de contexto (i.e., sem destruir a sua estrutura) e também remover pontos que possam ser fonte de ruído. O primeiro filtro aplicado permite a remoção destes últimos pontos, denominados de *outliers*. Assim, recorrendo a uma kd-tree [70] para aceleração do processamento, para cada ponto $\mathbf{p} \in P_A$ são escolhidos k vizinhos, sendo de seguida calculada a média, d_p , das distâncias Euclidianas entre o ponto $\mathbf{p}$ e todos os seus k vizinhos. Assumindo que as distâncias médias calculadas assumem uma distribuição Normal, estas podem ser representadas por uma média, μ , e um desvio padrão, σ , das distâncias médias calculadas

para todos os pontos $\mathbf{p} \in P_A$ e os seus k vizinhos, e pode ser definida da seguinte forma:

$$\mu = \frac{1}{|P_A|} \sum_{\mathbf{p} \in P_A} d_p \quad , \quad \sigma = \sqrt{\frac{1}{|P_A| - 1} \sum_{\mathbf{p} \in P_A} (d_p - \mu)^2} \quad (3.2)$$

onde $|P_A|$ representa o número de pontos contidos em P_A .

Assim, com base na média das distâncias médias, μ , e respetivo desvio padrão, σ , para qualquer ponto $\mathbf{p} \in P_A$ caso se verifique que a distância média entre si e os k vizinhos mais próximos é superior à da média definida por μ (i.e., apresenta uma distância anormal), o ponto é descartado, caso se verifique o oposto o ponto é adicionado a uma nova nuvem de pontos P'_A . Para verificar se a distância média d_p entre o ponto $\mathbf{p}$ e os seus k vizinhos se encontra dentro de um determinado limiar, é definido o seguinte intervalo:

$$\mu - \alpha \cdot \sigma \leq d_p \leq \mu + \alpha \cdot \sigma \quad (3.3)$$

onde α é um escalar definido empiricamente.

De seguida é aplicado um segundo filtro à nuvem de pontos P'_A obtida anteriormente, que permite remover pontos que apresentem normais⁵ com orientações invulgares (e.g., o plano do chão com normais orientadas para baixo). Assim, o primeiro passo é efetuar o cálculo das normais normalizadas, que é feito através de um técnica denominada Diferença de Normais (DoN) [39]. Esta técnica permite estimar um vetor normal normalizado com base em dois vetores normais unitários calculados no mesmo ponto. Para tal, primeiro, é necessário estimar os vetores normais ao longo das várias superfícies representadas pela numa nuvem de pontos. Existe uma grande variedade de métodos que abordam este problema, como demonstrado por Klasing et al. [42]. O método aqui utilizado, proposto por Berkman et al. [4], calcula a normal do plano tangente a uma superfície, que por sua vez pode ser estimado com recurso ao Método dos Mínimos Quadrados (utilizado para estimar um plano através de um conjunto de pontos finito como proposto por Shakarji et al. [69]). Nesse método, o cálculo da normal, $\mathbf{n}$, à superfície formada

⁵Vetor perpendicular à superfície que contém um determinado ponto.

pelo conjunto de k pontos vizinhos, P_k , mais próximos de um determinado ponto $\mathbf{p} \in P_A$ é realizado através da análise dos valores e vetores próprios da matriz de covariância $\mathbf{C} \in \mathbb{R}^{3 \times 3}$ de P_k , definida por:

$$\mathbf{C} = \frac{1}{k} \sum_{\mathbf{p} \in P_k} (\mathbf{p} - \bar{\mathbf{p}}) \cdot (\mathbf{p} - \bar{\mathbf{p}})^T \quad (3.4)$$

onde, $\mathbf{p}$ representa um dado ponto pertencente a P_k e $\bar{\mathbf{p}}$ representa a centróide⁶ dos k vizinhos mais próximos de $\mathbf{p}$. Contudo, não é possível obter uma correta orientação do vetor normal, $\mathbf{n}$, associado ao conjunto de pontos, P_k , através deste método sem que para isso seja definido o ponto de vista da câmara $\mathbf{v}_p$. De forma a orientar cada normal, $\mathbf{n}$, associada a cada ponto $\mathbf{p} \in P_A$ de forma consistente na direção da câmara, a seguinte equação deve ser satisfeita:

$$\mathbf{n} \cdot (\mathbf{v}_p - \mathbf{p}) > 0 \quad (3.5)$$

onde o ponto de vista $\mathbf{v}_p$ utilizado para o cálculo das normais é a origem, $\mathbf{v}_p = (0, 0, 0)$, visto que todas as nuvens de pontos $P_j \in P_A$ foram previamente transformadas para o referencial mundo. Como pode ser observado na equação 3.4, o cálculo das normais é feito com base nos k pontos vizinhos, que podem ser definidos através de uma região esférica de raio r em torno do ponto $\mathbf{p}$ (ver Figura 18b). Assim, torna-se extremamente importante a correta definição do raio r que define a quantidade de pontos utilizados para estimar o vetor normal $\mathbf{n}$ de cada ponto $\mathbf{p} \in P_A$. A utilização da DoN pode então ser usada como forma de contornar esta limitação. Para tal, para cada ponto $\mathbf{p} \in P_A$, são estimados dois vetores normais unitários $\hat{\mathbf{n}}(\mathbf{p}, r_l)$, $\hat{\mathbf{n}}(\mathbf{p}, r_s)$ com raios diferentes r_l e r_s , onde $r_l > r_s$. Por definição, o operador DoN é representado da seguinte forma:

$$\Delta \hat{\mathbf{n}}(\mathbf{p}, r_s, r_l) = \frac{\hat{\mathbf{n}}(\mathbf{p}, r_s) - \hat{\mathbf{n}}(\mathbf{p}, r_l)}{2}, \text{ com } r_s, r_l \in \mathbb{R} \quad (3.6)$$

⁶Centróide define o centro geométrico de um dado conjunto de pontos, obtido pela média das coordenadas de todos os pontos.

onde, $\hat{\mathbf{n}}(\mathbf{p}, r)$ é o vetor normalizado referente à normal da superfície definida no ponto $\mathbf{p}$ com raio r . Por último, é aplicado o filtro removendo assim todos os pontos que estejam fora dos intervalos definidos, empiricamente, para os resultados obtidos através do cálculo da DoN (e.g., ângulo), sendo obtida a nuvem de pontos P''_A .

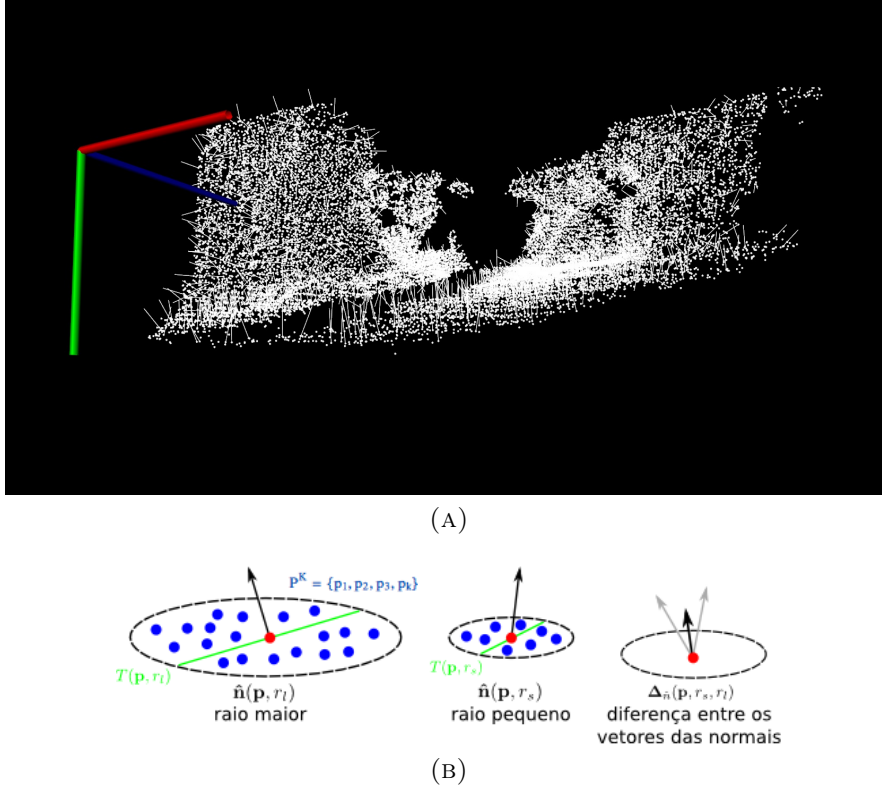


FIGURA 18: Cálculo da diferença entre normais. (A) Normais desenhadas como seguimentos de reta brancos; (B) Cálculo da diferença entre duas normais, onde o ponto vermelho corresponde ao ponto $\mathbf{p}$ à qual a normal, seta a preto, está associada.

3.3.1 Segmentação do plano do chão e obstáculos

Após aplicação dos filtros, a quantidade de pontos que compõem a nuvem de pontos P_A reduz significativamente, sem no entanto perder informação de contexto importante para a sua análise, permitindo assim um processamento mais eficiente dos algoritmos de segmentação utilizados para identificar e caracterizar os diferentes elementos presentes na nuvem de pontos. Na tarefa de identificar os obstáculos presentes no trilho, estes podem ser caracterizados como perturbações

no ambiente que se encontram ao nível do chão. Assim, torna-se necessário estimar o plano do chão descrito na nuvem de pontos P_A . Para tal, é utilizado o método Random Sample Consensus (RANSAC) [28] que gera um conjunto de hipóteses de planos de forma estocástica com a expectativa de que uma dessas aproxime suficientemente o conjunto de pontos no plano presente na nuvem de pontos. A grande vantagem deste método é a sua robustez à presença de um elevado volume de *outliers*.

A execução do método RANSAC pode ser descrita de forma genérica, para uma nuvem de pontos P , nas seguintes etapas:

1. Selecionar aleatoriamente três pontos não colineares $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ da nuvem de pontos P e gerar o respetivo plano hipótese h ;
2. A pontuação de um determinado plano hipótese é dada pela cardinalidade do conjunto de pontos *inliers*, $score(h) = |P_h|$. O ponto é considerado um *inlier* $\mathbf{p} \in P_h$, onde $\mathbf{p} = (p_x, p_y, p_z)$, caso a sua distância ao plano h , $d(\mathbf{p}, h)$, seja inferior a um determinado limiar, d_{plano} , definido empiricamente. P_h é então o conjunto de pontos *inliers* de h , ou seja, é o conjunto de pontos que se encontram dentro dos critérios definidos para pertencerem ao plano hipótese h ;
3. Repetir as etapas um e dois até obter n conjuntos de hipóteses, $h \in H$;
4. Escolher a hipótese $h \in H$ que contém a maior pontuação, $b = \operatorname{argmax}_{h \in H} score(h)$;
5. Calcular b' , que é uma versão refinada de b , através da projeção ortogonal do conjunto de pontos *inliers* de b , P_b , que estão contidos numa distribuição espacial bem definida. A projeção é feita por meio do Método dos Mínimos Quadrados Ponderados através da Decomposição em Valores Singulares (SVD). Calcular os pontos *inliers* contidos em b' , $P_{b'}$, e substituir a hipótese do plano pela atual, isto é, $b = b'$ e $P_b = P_{b'} \cdot P_{b'}$ é calculado através do processo descrito na etapa dois (2);

6. Repetir a etapa cinco (5) até $|P_b|$ se tornar constante ou o número máximo de iterações l tenha sido atingido.
7. Utilizar h como a hipótese estimada do plano do chão;

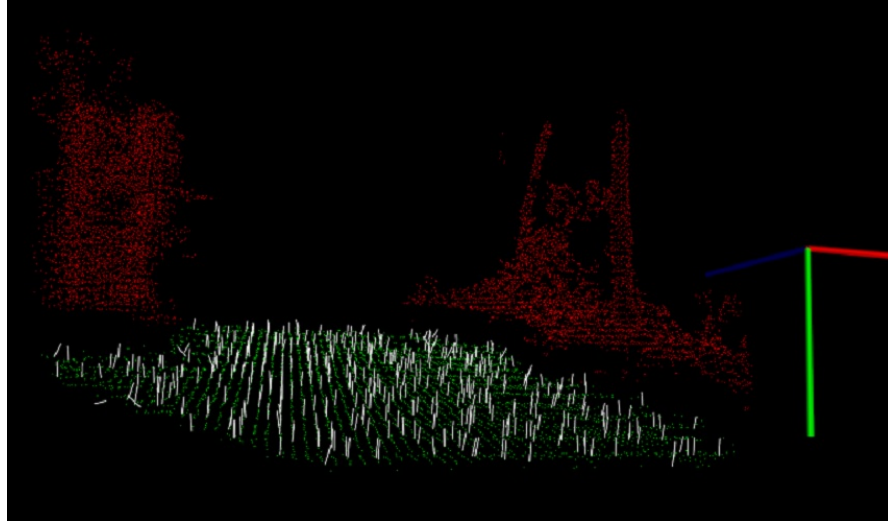
Como resultado são obtidos os coeficientes da equação do plano, (a, b, c, d) da equação do plano $ax + by + cz + d = 0$ que contém o maior número de pontos *inliers*. Assim, ao aplicar o método RANSAC à nuvem de pontos P_A'' são obtidos os coeficientes que definem a equação do plano, (a, b, c, d) , referente ao plano do chão e a respetiva nuvem de pontos segmentada, P_I , isto é, o conjunto de pontos *inliers*, $P_I \in P_A''$, definidos pelo plano do chão. No caso específico do sistema aqui proposto, o método RANSAC tem em conta a informação das normais da nuvem de pontos, P_A'' , no processo de segmentação. Para tal, na etapa (2) do método RANSAC também é tido em consideração a orientação do vetor normal, $\mathbf{n}$, do ponto $\mathbf{p}$ que deve ser perpendicular ao plano definido por h , (Figura 19), para que o ponto $\mathbf{p}$ seja considerado um *inlier*, como proposto por Holz et al. [36].

Após serem obtidos coeficientes do plano do chão, (a, b, c, d) , e a nuvem de pontos $P_I \in P_A''$, referente aos pontos *inliers* que definem o plano do chão, é calculada a média das distâncias de todos os pontos contidos em P_I em relação ao plano do chão $\pi = (a, b, c, d)$, tal que:

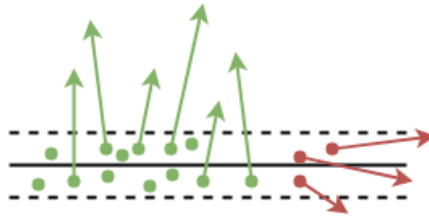
$$d_m(P_I, \pi) = \frac{1}{|P_I|} \cdot \sum_{\mathbf{p} \in P_I} \frac{(a \cdot p_x + b \cdot p_y + c \cdot p_z) - d}{\sqrt{a^2 + b^2 + c^2}} \quad (3.7)$$

onde, $|P_I|$ é o número de pontos *inliers* e $\mathbf{p} = (p_x, p_y, p_z)$ é um ponto na nuvem de pontos P_I . A distância média, d_m , é então utilizada para definir um *limiar*, t_d , que é calculado através da soma de um valor empiricamente definido, v , à distância média, d_m .

Seguidamente, com base no *limiar* calculado, são criadas duas nuvens de pontos, $P_c \in P_A''$ e $P_{obs} \in P_A''$ que definem as regiões referentes ao plano do chão e aos elementos acima do plano do chão caracterizados como obstáculos, respetivamente. Assim, para cada ponto da nuvem de pontos P_A'' é calculada a sua distância ao



(A)



(B)

FIGURA 19: Segmentação do plano do chão. (A) Normais perpendiculares ao plano do chão; (B) Ilustração do processo de segmentação com base em normais, onde, para o ponto ser selecionado a sua normal deve ser perpendicular ao plano hipótese (a verde ponto validos, a vermelho pontos inválidos).

plano do chão, definido pelos coeficientes (a, b, c, d) , e caso seja inferior ao *limiar* t_d o ponto é adicionado à nuvem de pontos P_c . Por outro lado, se a distância ao plano do chão for maior que o *limiar* t_d , o ponto é adicionado à nuvem de pontos P_{obs} . Como forma de garantir que os pontos contidos na nuvem de pontos P_{obs} se encontram descritos acima do plano, a seguinte condição têm que se verificar $a \cdot x + b \cdot y + c \cdot z + d > 0$.

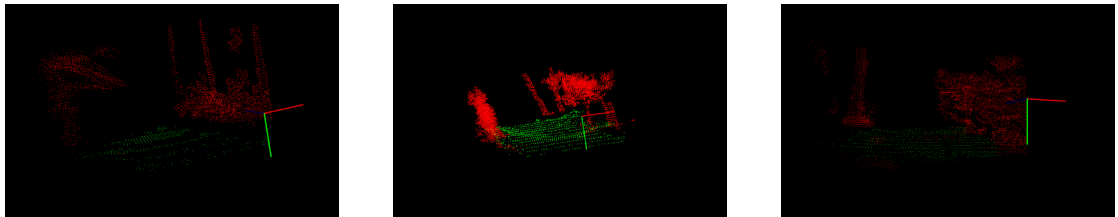


FIGURA 20: Exemplos de nuvens de pontos segmentadas. Os pontos a verde descrevem o plano do chão, e os pontos a vermelho descrevem os possíveis objetos no campo de visão da câmara.

3.3.2 Projeção dos obstáculos no plano da câmara

O detetor original executa o processo de análise do trilha sobre uma sequência de imagens bidimensionais, logo torna-se necessário transformar o resultado do processamento da informação tridimensional num formato que possa ser usado pelo detetor original, isto é, a representação dos resultados no espaço bidimensional. Para tal, é feita a projeção de todos os pontos da nuvem de pontos P_{obs} no plano da câmara, assumindo o modelo de *pinhole* [71] e recorrendo a uma matriz de projeção definida da seguinte forma:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & c_x \\ 0 & \alpha_v & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.8)$$

onde, (u, v) são as coordenadas (x, y) do pixel no referencial da câmara, (α_v, α_u) representam os valores da distância focal para (x, y) e (c_x, c_y) representam o centro do sensor ótico da câmara obtidos através do processo de calibração da câmara.

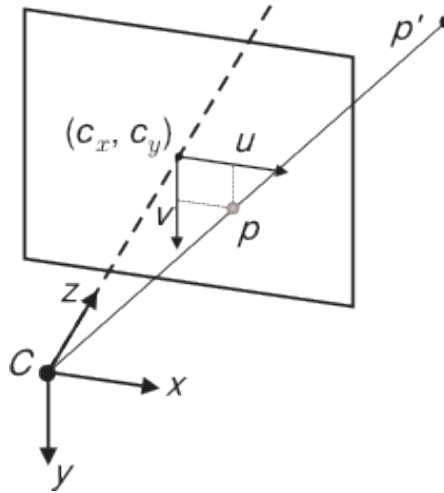


FIGURA 21: Modelo *pinhole* da câmara utilizado para projeção dos pontos no plano da câmara. Imagem adaptada de [22].

A projeção dos pontos tridimensionais no plano da câmara permite assim obter uma máscara que descreve as regiões da imagem onde existem elementos que são caracterizados como obstáculos, m_{obs} , e portanto devem ser evitados pelo detetor original. A Figura 22 ilustra os resultados obtidos pela projeção dos pontos

tridimensionais no plano da câmara (22a), onde é possível observar que os pontos desenhados na imagem original (22b) descrevem os elementos que se encontram acima do plano do chão (i.e, árvores e arbustos altos), isto é, o sistema permite obter uma máscara (22c e 22d) onde são descritos os possíveis obstáculos presentes na envolvente do trilho.



FIGURA 22: (A) Projeção dos pontos 3D no plano da câmara; (B) Pontos 3D, a vermelho, projetados na imagem original; (C) Máscara que indica as regiões caracterizadas como obstáculo em escala de cinzentos onde as regiões mais claras indicam uma maior concentração de pontos 3D; (D) Máscara binária que indica as regiões caracterizadas como obstáculo a preto.

3.4 Integração da informação tridimensional no detetor de trilhos

O detetor original utiliza como metáfora o comportamento coletivo das formigas para modelar um conjunto de agentes virtuais que percorrem o campo visual da imagem de forma a identificar, nas regiões que apresentem uma maior saliência, a presença do trilho. Para tal, é usado o conceito de feromona virtual, que ao ser depositada nas regiões visitadas e analisadas pelos agentes virtuais afeta o comportamento dos restantes agentes virtuais, isto é, a interação entre agentes virtuais é feita através do ambiente, à imagem do conceito de *stigmergy* [34]. Este processo é implementado pelo detetor original através de dois mecanismos principais: (a) depósito de feromonas virtuais nas regiões visitadas pelos agentes virtuais e (b) filtro temporal, denominado de campo neuronal (NF), que integra as feromonas virtuais ao longo de várias imagens para aumentar a robustez do sistema na presença de ruído e informação ambígua.

O mapa de obstáculos calculado pelo sistema proposto, m_{obs} , é então utilizado para condicionar o comportamento dos agentes virtuais de forma a evitar a seleção de regiões da imagem que contenham obstáculos, isto é, regiões que se apresentem elas próprias mais salientes que o trilho e portanto podem ser classificadas como região hipótese do trilho. Primeiro, o mapa m_{obs} é utilizado para modular o processo de depósito de feromonas virtuais. No detetor original para cada formiga i é calculado um valor de feromona virtual, f_i , a depositar ao longo do caminho percorrido pelos agentes virtuais. Esta feromona virtual é então utilizada pelos agentes virtuais para influenciar a escolha das regiões da imagem a visitar, isto é, os agentes virtuais tendem a escolher regiões da imagem onde existe um maior depósito de feromona virtual e estão assim associadas a uma maior probabilidade de pertencer a região hipótese do trilho. Concretamente, no sistema proposto, o nível de feromona virtual depositado por cada agente virtual, é afetado pelo número de obstáculos encontrados no trajeto percorrido, através da seguinte expressão:

$$f_i[k] = f_i[k]^* + \sqrt{1 - (p_o/k_{saltos})} \quad (3.9)$$

onde, $f_i[k]^*$ é o valor de feromona virtual depositado em cada salto k , definido pelo detetor original, p_o é o número de píxeis descritos como obstáculo no mapa m_{obs} visitados pelo agente virtual ao longo do caminho percorrido e k_{saltos} é o número de píxeis visitados pelo agente virtual na imagem original. Seguidamente, caso o nível de feromona virtual calculado para o pixel visitado pelo agente virtual seja inferior a um limiar empiricamente definido, $f_i[k] < \epsilon$, é assumida a presença de um obstáculo (i.e., distrator), logo a feromona virtual não é depositada, contudo, este limiar deve ser suficientemente baixo para acomodar o possível ruído presente no mapa m_{obs} .

Por outro lado, como forma de propagar a atuação dos agentes virtuais entre imagens analisadas é utilizado um filtro temporal definido como NF, que descreve a atividade e quantidade de feromonas virtuais depositadas pelos mesmos. Assim, como forma de limitar a atuação dos agentes virtuais, isto é, as regiões da imagem que são analisadas, a máscara de obstáculos, m_{obs} , é utilizada para diminuir a

influência das feromonas virtuais nas regiões classificadas como obstáculo. Concretamente, para cada pixel no NF verifica-se se o pixel correspondente na máscara de obstáculos, ou seja, nas mesmas coordenadas, pertence às regiões caracterizadas como obstáculo. Esta relação pode ser definida pela seguinte expressão:

$$\varepsilon(x, y) = \begin{cases} 1, & \text{se } NF(x, y) \cdot m_{obs}(x, y) = 1 \\ 0, & \text{caso contrário} \end{cases} \quad (3.10)$$

onde $\varepsilon(x, y)$ é uma função que recebe como parâmetros o par de coordenadas (x, y) comum a ambos os mapas, NF e m_{obs} , a função $NF(x, y)$ permite obter o nível de atividade dos agentes virtuais num determinado pixel no par de coordenadas (x, y) do mapa NF e a função $m_{obs}(x, y)$ permite determinar se um pixel nas suas coordenadas (x, y) é descrito como obstáculo. Assim, caso a condição se verifique, ou seja $\varepsilon(x, y) = 1$, o pixel nas coordenadas (x, y) que define a quantidade de feromonas virtuais depositadas pelos agentes virtuais no NF, numa escala com intervalo de $[0, 255]$, é alterado para metade do seu valor original, onde temos:

$$NF(x, y) \leftarrow NF(x, y) \cdot 0,5 \quad (3.11)$$

Desta forma, é possível diminuir a influência das feromonas virtuais nas regiões classificadas como obstáculos mantendo a integridade do filtro temporal, NF, ou seja, o filtro temporal não elimina por completo a informação de atividade do agente virtual o que permite recuperar e casos onde possam existir falsos positivos (e.g., pontos pertencentes ao chão identificados como obstáculos).

Outro dos parâmetros explorados nesta dissertação é a altura máxima definida para as trajetórias que os agentes virtuais podem descrever enquanto exploraram o campo visual do robô na imagem. Assim, a escolha da altura máxima afeta a área da imagem que é analisada pelos agentes virtuais. Por outro lado, a presença de regiões salientes na parte superior da imagem (e.g., céu) pode afetar o seu comportamento. O detetor original resolve este problema através de um valor máximo definido, empiricamente, para altura. No entanto, partindo da observação

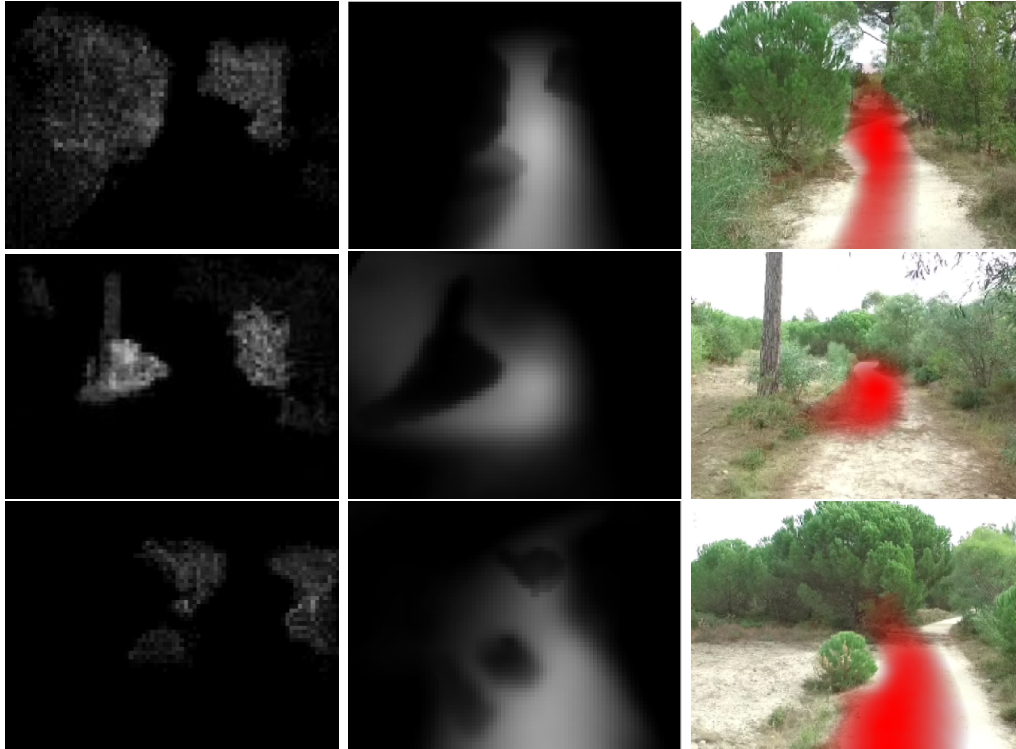


FIGURA 23: Processo de influência no comportamento das agentes virtuais de forma a prevenir a seleção de regiões identificados como obstáculos. Da esquerda para a direita, máscara de obstáculos, m_{obs} , filtro temporal, NF, e região hipótese do trilho, h_t .

que o trilho se encontra descrito no campo visual do robô abaixo da linha de horizonte, e que esta pode variar dependendo do ângulo da câmara, temos que, a linha de horizonte na imagem varia dependendo da orientação da câmara em relação ao plano do chão. Consequentemente, se a câmara estiver orientada para baixo a linha de horizonte sobe, verificando-se também o contrario, se a câmara estiver orientada para cima a linha de horizonte desce. Concretamente, o sistema proposto através do vetor normal ao plano do chão, obtido através dos coeficientes do plano (na Secção 3.3.1), calcula o ângulo que define a inclinação da câmara em relação ao plano do chão (ver Figura 24), Θ , e pode ser definido da seguinte forma:

$$\Theta = \arcsin(n_z) \quad (3.12)$$

onde $\mathbf{n} = (n_x, n_y, n_z)$ é a normal ao plano do chão obtido através do algoritmo RANSAC. O ângulo Θ é então utilizado para definir a altura máxima, h_{max} , para as trajetórias que os agentes virtuais podem descrever durante o processo de análise

do campo visual do robô, na imagem, que pode ser definido como:

$$h_{max} = h_{ini} + \Theta \cdot \beta \quad (3.13)$$

onde, h_{ini} é o valor inicial definido para a altura máxima e β é um valor, definido empiricamente, utilizado como forma mitigar a incerteza no calculo do ângulo Θ .

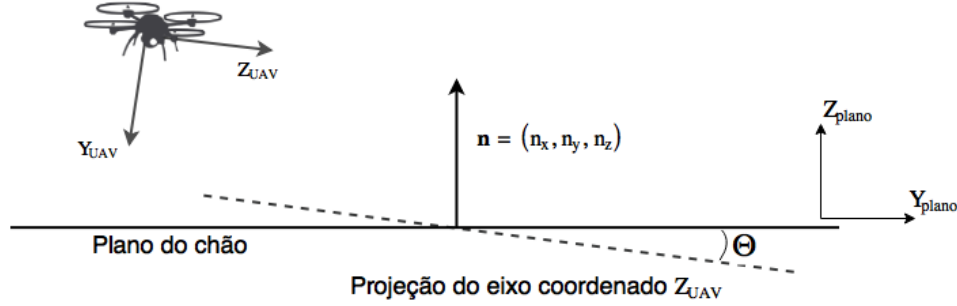


FIGURA 24: A figura ilustra o ângulo de inclinação, Θ , da câmara em relação ao plano do chão, onde temos: $\mathbf{n}$ que representa o vetor normal ao plano do chão projetado no eixo coordenado Z_{plano} do plano e a linha a tracejado representa a projeção do eixo coordenado Z_{UAV} do UAV no plano.

3.5 Algoritmo para validação da presença de trilho

Utilizando como base a informação tridimensional do ambiente e a região hipótese do trilho, o sistema proposto valida se o trilho se encontra descrito em regiões não identificadas como obstáculos. Isto permite ao sistema identificar situações potencialmente perigosas para o robô, quando existe uma má identificação da região hipótese do trilho fornecendo, por exemplo, ao sistema de controlo um maior grau de certeza da correta identificação do trilho, pois são utilizados dois mecanismos diferentes. Apesar de ambos os mecanismos estarem suscetíveis a falha, a probabilidade de ambos falharem em simultâneo é mais reduzida, o que permite ao sistema uma melhor avaliação nas ações a tomar. A Figura 25 ilustra uma possível arquitetura onde o sistema proposto é integrado com um sistema de navegação do robô. A arquitetura apresentada recorre a duas fontes de informação como forma de diminuir a incerteza do sistema de navegação aquando do planeamento das

operações do robô (e.g., comandos de locomoção), utilizando para isso informação referente à localização do trilho, no campo de visão do robô, e do grau de certeza dessa mesma localização tendo em conta a informação tridimensional do ambiente, obtida através do módulo Validador aqui descrito.

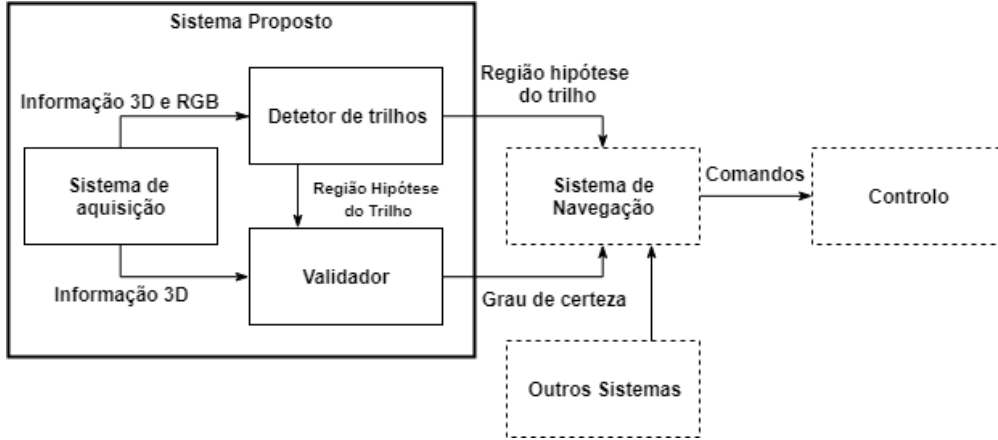


FIGURA 25: Possível integração do sistema proposto como auxílio à navegação de um UAV. Os sistemas não implementados no decorrer desta dissertação encontram-se representados por caixas a tracejado.

Para tal, são utilizadas as duas máscaras resultantes das operações anteriores, mais especificamente a máscara que caracteriza os obstáculos, m_{obs} , e a máscara que descreve a região hipótese do trilho, m_{ht} . As duas máscaras são então convertidas para imagens binárias, onde todos os píxeis que descrevam as regiões obstáculo ou a região hipótese do trilho são convertidos para píxeis de cor branca, verificando-se também o contrário, isto é, todos os píxeis que não pertençam a estas regiões são convertidos para píxeis pretos. Seguidamente, é aplicado um operador binário *AND* que identifica as regiões onde as duas máscaras se intercetam. Caso o somatório dos píxeis avaliados pela função $\phi(\mathbf{q})$ apresentem uma quantidade de píxeis acima de um determinado limiar q_t , empiricamente definido, são demarcadas através de contornos vermelhos numa imagem resultante da operação anterior e pode ser definido através da seguinte expressão:

$$\sum_{\mathbf{q} \in Q} \phi(\mathbf{q}) > q_t \quad \text{com,} \quad \phi(\mathbf{q}) = \begin{cases} 1, & \text{se } m_{ht}(q_x, q_y) \cdot m_{obs}(q_x, q_y) = 1 \\ 0, & \text{caso contrário} \end{cases} \quad (3.14)$$

onde, Q representa o conjunto de píxeis visitados pelos agentes virtuais, $\mathbf{q} = (q_x, q_y)$ representa o pixel no par de coordenadas (x, y) comum a ambas as máscaras m_{ht} e m_{obs} , a função $m_{obs}(q_x, q_y)$ permite determinar se um pixel nas suas coordenadas (x, y) é descrito como obstáculo e função $m_{ht}(q_x, q_y)$ permite determinar se um pixel nas suas coordenadas (x, y) é descrito como pertencente à região hipótese do trilho. O uso do limiar q_t tem como objetivo evitar falsos positivos. A Figura 26 ilustra a interceção entre a região hipótese do trilho e as regiões identificadas como obstáculos, onde é possível observar que a região a vermelho (Figura 26a), que descreve a região identificada como trilho, se encontra descrito sobre uma parte da vegetação presente na envolvente do trilho, que por sua vez é identificada como sendo obstáculo pelo sistema proposto (Figura 26b). Assim, o resultado obtido é dado através de um retângulo vermelho (Figura 26c) que identifica os contornos da interceção entre as duas regiões.

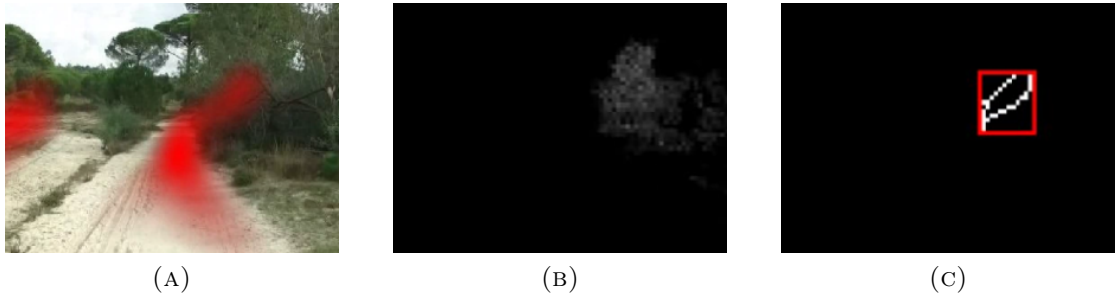


FIGURA 26: (A) Região hipótese do trilho assinalada a vermelho; (B) Máscara da região onde é identificada a presença de obstáculo detetados através da informação tridimensional; (C) Interceção entre as duas regiões, assinaladas através de contornos brancos assinalados através de um retângulo vermelho.

Capítulo 4

Resultados Experimentais

Este capítulo apresenta, na Secção 4.1, a instalação e configurações utilizadas no decorrer do desenvolvimento e dos testes experimentais realizados ao sistema proposto. Na Secção 4.2, é descrito o conjunto de dados utilizado nos testes e a forma como foram adquiridos. Por último, na Secção 4.3, é feita uma análise aos resultados obtidos onde são apresentados os casos de falha do sistema e uma discussão sobre a contribuição da presente dissertação para a deteção e seguimento de trilhos.

4.1 Instalação experimental

O sistema proposto foi desenvolvido e testado num computador portátil com uma unidade de processamento de 2.20GHz Intel Core i7-4700MQ, com 16Gb de memória RAM, correndo o sistema operativo Linux Ubuntu 14.04 LTS 64-bits (Trusty Tahr). Todo o desenvolvimento foi feito com recurso à linguagem de programação C++ e de acordo com o modelo de programação proposto pelo ROS [54], onde todos os pacotes de software, utilizados ou desenvolvidos, são compatíveis com a distribuição ROS *Indigo Igloo*¹. O ROS [54] é um *middleware* que assenta numa estrutura modular onde diferentes pacotes de software, organizados em nós (i.e.,

¹ROS Indigo Igloo: <http://wiki.ros.org/indigo>

módulos), com funções específicas (e.g., controlo, navegação, perceção) interagem entre si com o objetivo de permitir a um robô executar um determinado conjunto de tarefas. A comunicação entre nós é feita através da troca de mensagens que podem ser assíncronas, denominadas de tópicos, ou síncronas, denominadas de serviços. É ainda disponibilizado um espaço de memória comum, denominado de servidor de parâmetros, onde podem ser armazenadas um conjunto de variáveis permitindo assim que possam ser acedidas por todos os nós que compõem o sistema. Por outro lado, as bibliotecas *Point Cloud Library* (PCL) [63], na sua versão 1.7², e *Open Source Computer Vision Library* (OpenCV) [40], na sua versão 2.4³, foram utilizadas, respetivamente, para processar as nuvens de pontos tridimensionais e executar rotinas de tratamento de imagem de baixo nível. O ROS e as bibliotecas PCL e OpenCV são disponibilizados através de uma licença de código aberto BSD (*Berkeley Software Distribution*), de acesso livre para fins comerciais e de investigação.

A Figura 27 descreve os diferentes nós ROS que são executados pelo sistema. As trocas de mensagens entre cada nó são feitas através de mensagens ROS que encapsulam os dados necessários para a execução de cada nó. De uma forma resumida, o nó de *Aquisição de Vídeo* é responsável por transformar os vídeos, armazenados em ficheiros com compressão h264 e com uma resolução de 1280×720 píxeis, em imagens RGB de tamanho 640×480 píxeis. O nó *LSD-SLAM Core* é responsável por identificar as *keyframes* e gerar os respetivos mapas de profundidade. O nó *LSD-SLAM Viewer* é responsável por adquirir e tratar a informação tridimensional de forma a criar as máscaras (i.e., imagens) com as regiões identificadas como obstáculos (ver Secções 3.2 e 3.3). O nó detetor de trilhos é responsável por identificar a região hipótese do trilho evitando as regiões identificadas como obstáculos (ver Secção 3.4). O nó *Validador* é responsável por validar se o trilho não se encontra descrito nas regiões identificadas como obstáculos (ver Secção 3.5). Por último, a *Interface Gráfica* utilizada é disponibilizada pelo ROS, denominada de RQT, e é responsável por mostrar um conjunto de dados que permite validar

²PCL: <http://pointclouds.org/>

³OpenCV: <http://opencv.org/>

a correta identificação da região hipótese do trilho (e.g., máscara de obstáculos e filtro temporal).

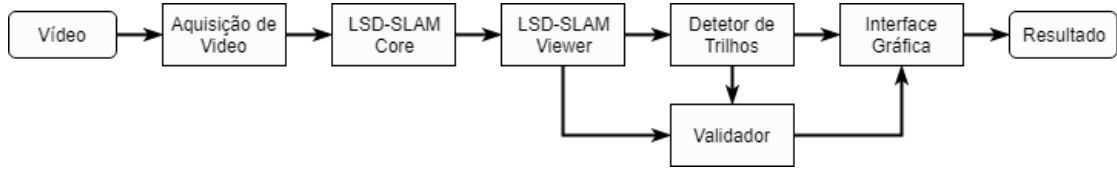


FIGURA 27: Descrição dos nós ROS utilizados pelo sistema proposto. Os retângulos representam os nós e as setas as trocas de mensagens através de tópicos ROS.

4.1.1 Calibração da câmara monocular

De modo a permitir ao LSD-SLAM obter uma correta aproximação da distância a que os objetos se encontram do robô (e.g., árvores), primeiro, é necessário calcular os parâmetros intrínsecos da câmara (e.g., distância focal) através de um processo de calibração. De referir que as imagens obtidas pela câmara já se encontram sem distorção ótica sendo assumido um modelo *pinhole* [71]. A Figura 28 ilustra parte do processo de calibração da câmara, que consiste em filmar um padrão, com dimensões conhecidas, em diferentes poses no campo de visão da câmara. Concretamente, o padrão assemelha-se a um tabuleiro de xadrez e é composto por uma matriz de 8x6 quadrados pretos e brancos, intercalados entre si, onde cada quadrado tem como dimensões 2,5 cm de altura por 2,5 cm de largura. O padrão é então movido ao longo do campo de visão da câmara a diferentes distâncias e ângulos até que tenha sido capturado em todas as regiões do campo de visão. Este processo de calibração é feito através de um pacote de software⁴ ROS que implementa um algoritmo da biblioteca OpenCV baseado nos trabalhos desenvolvidos por Zhang et al. [79] e Bouguet et al. [9].

⁴Pacote de software utilizado para calibração da câmara: http://wiki.ros.org/camera_calibration

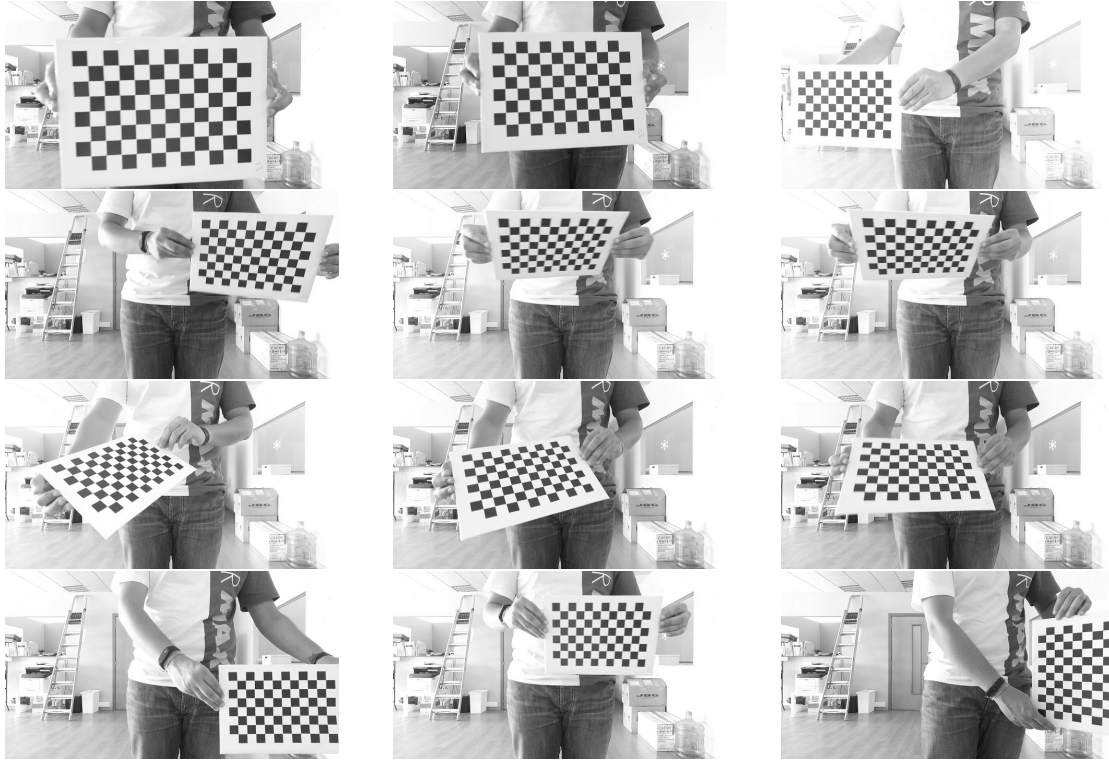


FIGURA 28: Sub-conjunto de imagens utilizadas para calibrar a câmara utilizada para adquirir o conjunto vídeos utilizado para validar o sistema proposto.

4.1.2 Parametrização do sistema

Após a calibração da câmara, os parâmetros intrínsecos obtidos são utilizados pelo LSD-SLAM no processo de estimação dos mapas de profundidade. São também definidos um conjunto de parâmetros de forma a melhorar os resultados obtidos pelo LSD-SLAM no conjunto de dados de teste utilizados. Concretamente, foram definidos empiricamente os seguintes valores: (a) $\text{minUseGrad} = 10 \in [1, 50]$, que define o gradiente mínimo na intensidade da imagem para um determinado pixel ser analisado; (b) $\text{cameraPixelNoise} = 5,0 \in [1, 50]$, que define o valor de ruído digital na intensidade da imagem utilizado na sua análise; (c) $\text{KFUsageWeight} = 6,0 \in [0, 20]$, que define a frequência com que são obtidas *keyframes* de forma melhorar o mapa de profundidade em relação à *keyframe* de referência; (d) $\text{KFDistWeight} = 16 \in [0, 20]$, define a quantidade de novas *keyframes* obtidas em relação à distância à última *keyframe* (i.e., *keyframe* de referência); e (e) $\text{depthSmoothingFactor} = 5 \in [0, 10]$, que define o fator utilizado na aplicação

TABELA 4.1: Parâmetros definidos empiricamente no sistema.

Número de nuvens de pontos que compõem a nuvem de pontos P_A .	τ	10
Valor somado ao limiar definido no primeiro filtro.	α	0,02
Número de pontos vizinhos utilizados no primeiro filtro.	k	25
Raio, grande, que define a área de pontos em relação ao ponto p , para cálculo das normais do segundo filtro.	r_l	0,06
Raio, pequeno, que define a área de pontos em relação ao ponto p , para cálculo das normais do segundo filtro.	r_s	0,02
Valor da distância mínima para um ponto pertencer ao plano (RANSAC).	tj	0,01
Número de máximo de iterações do RANSAC.	l	80
Valor somado à distância média, d_m , que define o limiar para os pontos pertencerem a nuvem P_{obs} .	v	0,07
Valor mínimo de feromona virtual, f_i , para o qual é considerado a presença de obstáculos.	ε	15
Valor somado à h_{max} calculado com base no ângulo formado entre a câmara e plano do chão.	β	20

de um filtro de *blur* aplicado ao mapa de profundidade. Os restantes parâmetros utilizados pelo sistema proposto encontram-se detalhados na Tabela 4.1.

O detetor de trilhos utiliza praticamente os mesmos valores definidos pelo detetor original [68] no qual se baseia, à exceção dos parâmetros modificados dinamicamente de forma a modelar o comportamento dos agentes virtuais (ver Secção 3.4). Concretamente, o valor que define a altura máxima na imagem até onde os agentes virtuais podem descrever a suas trajetórias, h_{max} , é definido de acordo com o ângulo de inclinação da câmara em relação ao plano do chão, θ , calculado ao longo da execução do sistema. O valor de feromonas virtuais depositado, f_i , é calculado de acordo com número de píxeis pertencentes aos obstáculos pelos quais os agentes virtuais passaram, ou seja, quanto maior for o número de píxeis pertencentes a obstáculos visitados pelo agente virtual menor será o valor de feromona virtual depositado. Por último, o filtro temporal NF é diretamente afetado pela máscara m_{obs} como forma de inibir os agentes virtuais, nas iterações seguintes, de avaliar as regiões da imagem onde foram previamente identificados obstáculos.

4.2 Conjunto de dados

Como forma de validar o sistema proposto, foram capturados um conjunto de 12 vídeos, ao longo de 5 percursos diferentes em diferentes alturas do dia (e.g., com o sol no seu ponto mais alto), com recurso a uma plataforma UAV DJI Phantom 3 Advance⁵ tele-operada. Após ter sido adquirido o conjunto de vídeos de teste, todo o processamento e análise dos mesmos foi feito em modo *off-line*, de forma a recolher um conjunto de dados que permitisse uma melhor análise aos resultados obtidos pelo sistema proposto. A Figura 29 ilustra três dos percursos efetuados pelo UAV, obtidos através da aplicação AirData⁶, no decorrer do processo de aquisição do conjunto de vídeos utilizados. Foram filmados um total de ≈ 43 minutos de vídeo, compreendidos numa distância total de ≈ 3 quilómetros, com uma distância média de ≈ 300 metros e um tempo médio de $\approx 3,5$ minutos por vídeo, com uma distância média do UAV ao solo de $\approx 2,5$ metros e com uma velocidade de $\approx 1\text{ms}^{-1}$. Todos os vídeos foram capturados a uma cadencia de 50 imagens por segundo com uma resolução de 1280×720 píxeis, e guardados com uma compressão de vídeo h264.

De referir que o conjunto de dados de teste utilizados por Santana et al. [65] não foi utilizado para validar o desempenho do sistema proposto, nesta dissertação, por não ter sido possível obter os parâmetros de calibração da câmara utilizada na aquisição dos vídeos. Sem esses parâmetros (intrínsecos) da câmara não é possível realizar uma correta projeção dos pontos tridimensionais para o plano da câmara.

4.3 Análise de Resultados

O detetor original, como demonstrado por Santana et al. [68], mostrou-se eficaz na identificação da região hipótese do trilha num vasto conjunto de situações. Contudo, a presença de distratores no trilha ou na sua envolvente tende a influenciar a tarefa de seguimento do mesmo ao induzir o detetor original a incorretamente

⁵DJI Phantom 3 Advanced: <https://www.dji.com/phantom-3-adv>

⁶AirData: <https://airdata.com>

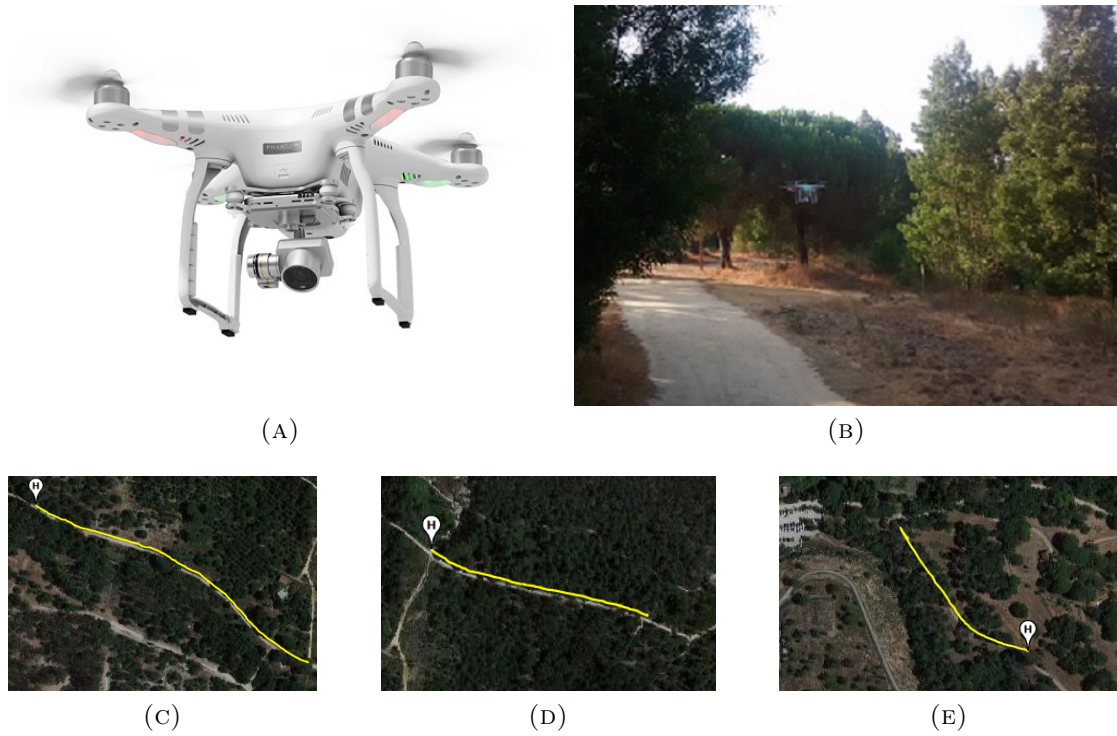


FIGURA 29: Plataforma UAV utilizada na aquisição dos vídeos de teste usados para validar o sistema proposto, e exemplos dos percursos realizados pelo UAV. (A) plataforma UAV DJI Phantom 3 Advance (imagem retirada do site oficial da DJI); (B) UAV no processo de aquisição dos vídeos de teste; (C) Percurso realizado para gravar o Vídeo 1: 15120 imagens, 347 metros percorridos e duração de 5 minutos e 03 segundos; (D) Percurso realizado para gravar o Vídeo 2: 8150 imagens, 195 metros percorridos e duração de 2 minutos e 72 segundos; (E) Percurso realizado para gravar o Vídeo 5: 7200 imagens, 172 metros percorridos e duração de 2 minutos e 40 segundos. O *H*, representa o ponto de partida do respetivo voo realizado pelo UAV.

identificar como sendo trilho certas regiões da imagem que sendo salientes não são trespassáveis pelo robô, como por exemplo árvores ou vegetação alta. Este problema é abordado nesta dissertação por meio do sistema proposto no Capítulo 3, que através da informação volumétrica recolhida do ambiente identifica as regiões na imagem onde existam elementos que se apresentam como obstáculos, permitindo assim ao sistema modelar o comportamento dos agentes virtuais do detetor original de forma a que estes não sejam influenciados pela presença dos obstáculos.

Como forma de validar os resultados obtidos pelo sistema proposto no conjunto de dados de teste, são analisadas cinco hipóteses:

- R 1: O sistema permite uma correta reconstrução tridimensional do ambiente através de uma câmara monocular;
- R 2: O sistema permite identificar obstáculos próximos do robô que se encontrem descritos no seu campo de visão;
- R 3: O sistema permite obter uma aproximação correta à orientação da câmara em relação ao solo;
- R 4: O sistema permite melhorar a detecção e seguimento de trilhos através da adição de informação tridimensional ao detetor original;
- R 5: O sistema permite validar, através de um grau de certeza, a correta identificação da região hipótese do trilho.

4.3.1 Avaliação da primeira hipótese (R1) em estudo

Na Secção 3.2 é proposta a utilização do pacote de software LSD-SLAM [25] como forma de obter informação volumétrica do ambiente, através da reconstrução tridimensional do mesmo, com recurso a uma câmara monocular montada num robô. A Figura 30 apresenta um resultado típico relativo à reconstrução tridimensional obtida diretamente do LSD-SLAM quando aplicada ao conjunto de dados de teste. O produto da reconstrução tridimensional é composto por uma nuvem de pontos acumulada e sem qualquer tipo de tratamento, que pode conter milhões de pontos. Após um conjunto de processos aplicados à nuvem de pontos acumulada, detalhados na Secção 3.3, é possível obter uma nuvem de pontos filtrada. A Figura 31 ilustra a reconstrução tridimensional dos diferentes elementos próximos do robô e que se encontram descritos no seu campo de visão (e.g., solo, árvores e outro tipo de vegetação). A densidade e organização das nuvens de pontos obtidas suportam a primeira hipótese (R1), ou seja, que o sistema de reconstrução monocular fornece uma representação tridimensional adequada no contexto do seguimento de trilhos.

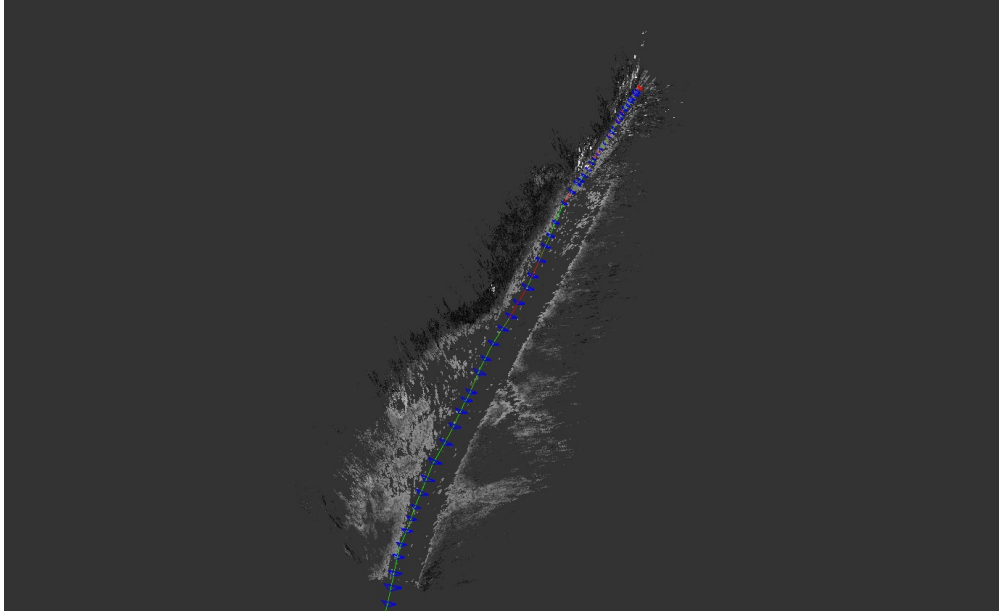


FIGURA 30: Exemplo da reconstrução tridimensional, em ambiente florestal, obtida pelo LSD-SLAM sem qualquer tipo de filtragem. As diferentes poses da câmara são representadas a azul e o trajeto através de segmentos de reta.

4.3.2 Avaliação da segunda hipótese (R2) em estudo

Como descrito na Secção 3.3.1, a nuvem de pontos é utilizada para identificar, de entre os elementos próximos do robô, quais os que se encontram mais elevados que o trilho (i.e., acima do plano do chão) e, como resultado, deverão ser considerados obstáculos. Os pontos da nuvem de pontos pertencentes a obstáculos são projetados no plano da câmara (imagem) de forma a criar uma máscara que identifique os obstáculos (ver Secção 3.3.2). A Figura 32 apresenta os resultados obtidos com esta metodologia no conjunto de dados de teste. Como é possível observar, estas figuras confirmam a segunda hipótese (R2), ou seja, que o sistema deteta corretamente os vários obstáculos presentes nas imediações do robô, produzindo dessa forma máscaras corretas no plano da câmara (imagem).

4.3.3 Avaliação da terceira hipótese (R3) em estudo

A terceira hipótese (R3) a ser testada nesta dissertação é a capacidade do sistema proposto em estimar o ângulo de inclinação da câmara em relação ao plano do chão, Θ , de inclinação da câmara em relação ao plano do chão, como detalhado na



FIGURA 31: Sub-conjunto de resultados típicos obtidos com o conjunto de dados de teste, que demonstra a reconstrução tridimensional do ambiente descrito em frente do robô ($\approx 8\text{m}$). As imagens com a reconstrução tridimensional, primeira e terceira colunas, ilustram as nuvens de pontos obtidas onde, os pontos a vermelho descrevem os obstáculos e os pontos a verde descrevem o plano do chão.

Secção 3.4. A Figura 33 apresenta dois gráficos para os percursos efetuados pelo UAV (Vídeo 1 (Figura 33a) e Vídeo 2 (Figura 33b)) onde é possível analisar os valores estimados para a inclinação da câmara ao longo das *keyframes* analisadas. É possível observar que na fase inicial, em que o sistema proposto é inicializado, existe uma grande variação no valor estimado para o ângulo da câmara em relação ao plano do chão, marca (Figura 33a/a). Isto deve-se ao facto de durante a inicialização do LSD-SLAM não ser possível obter uma correta reconstrução tridimensional do ambiente, pois não existe ainda translação suficiente da câmara que permita calcular os mapas de profundidade, o que por sua vez dificulta o cálculo do plano do chão, utilizado para estimar o ângulo de inclinação da câmara.

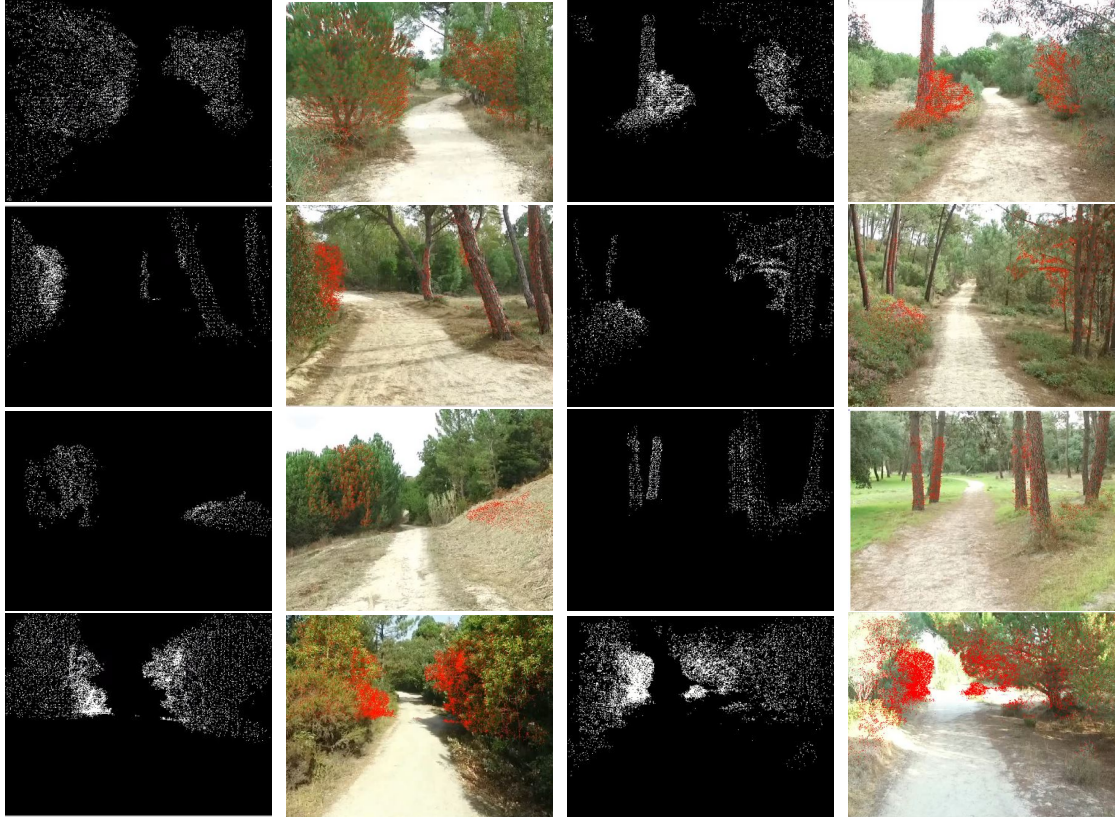


FIGURA 32: Sub-conjunto resultados típicos obtidos com o conjunto da dados de teste que demonstra uma correta identificação dos obstáculos próximos do robô, presentes no seu campo de visão. Os resultados são apresentados em pares de imagens, onde, à esquerda é apresentada a projeção dos pontos obtidos pela reconstrução tridimensional, no plano da câmara, e a direita os obstáculos descritos na imagem original como pontos vermelhos.

Contudo, é possível observar que após a fase inicial, referente ao processo de inicialização, marca (Figura 33a/a) e marca (Figura 33b/d), o ângulo θ da câmara assume um valor médio com um desvio padrão de $-10,31^\circ \pm 8,6^\circ$ para o Vídeo 1 e $-10,87^\circ \pm 9,74^\circ$ para o Vídeo 2, o que se aproxima ao valor esperado para as observações feitas durante o voo do UAV. Em concreto, o percurso associado ao Vídeo 1 apresenta uma topologia de terreno plano, que pode ser inferido através da análise do gráfico (Figura 33a), onde se pode observar um vasto conjunto de valores, de inclinação da câmara, próximos do valor médio de $\approx -10,31^\circ$ distribuídos ao longo das *keyframes* obtidas. Por outro lado, o percurso associado ao Vídeo 2 apresenta uma topologia de terreno irregular, com pequenos declives, e que pode de igual forma ser observado no gráfico (Figura 33b). Concretamente,

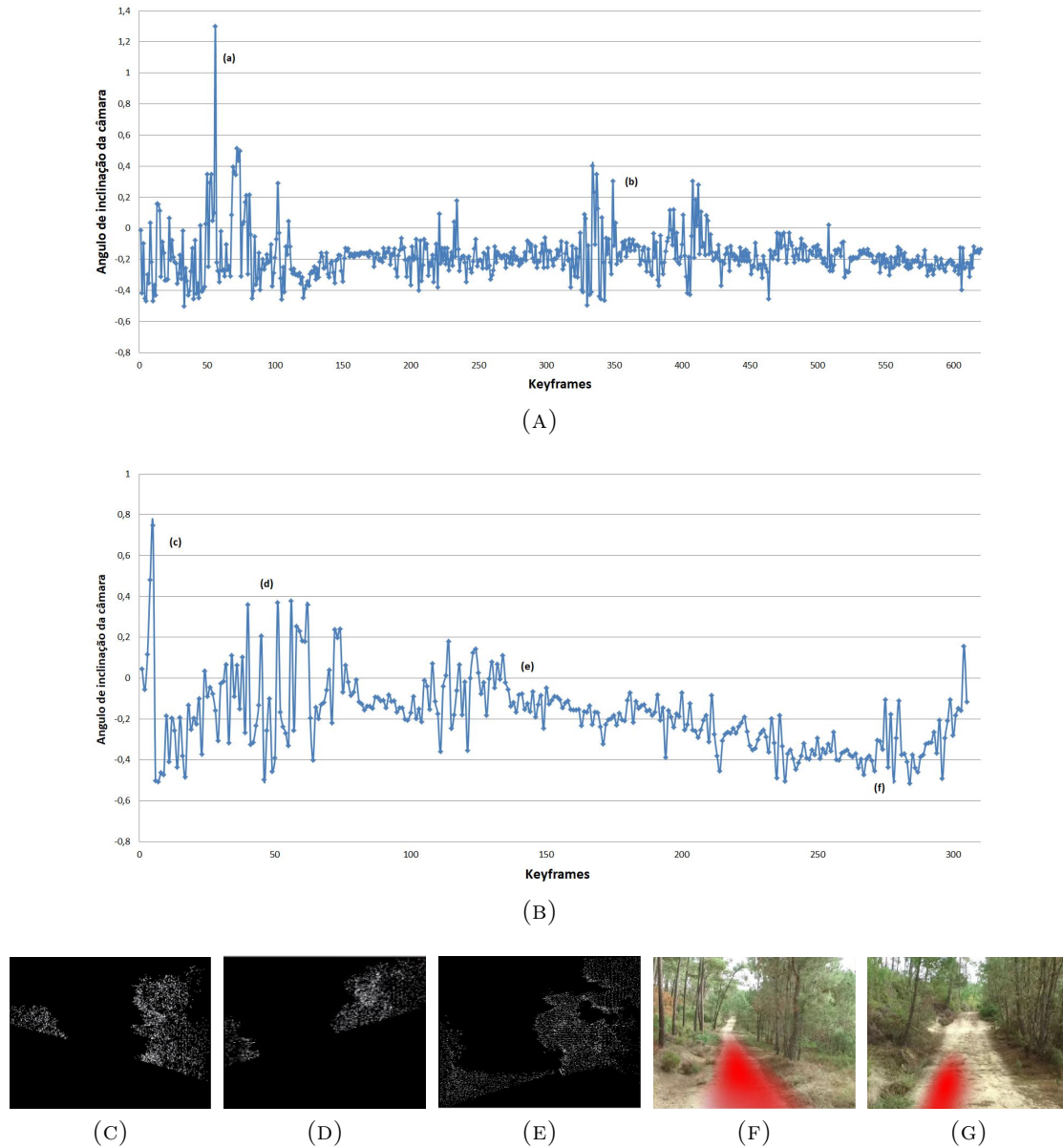


FIGURA 33: Dados obtidos referentes ao ângulo de inclinação da câmara em relação ao solo. O eixo coordenado x representa as *keyframes* analisadas, o eixo coordenado y representa o ângulo, θ , de inclinação da câmara em radianos. As referências (a,b,c,d,e,f) são utilizadas no texto como auxílio à explicação do gráfico. (A) Inclinação da câmara em relação ao plano do chão para o Vídeo 1; (B) Inclinação da câmara em relação ao plano do chão para o Vídeo 2; (C, D e E) Erro na estimativa do plano (RANSAC); (F) Declive descendente; (G) Declive ascendente.

pode-se observar o início de um declive descendente, marca (Figura 33b/e), e o início de um declive ascendente, marca (Figura 33b/f). A presença de *outliers*, como aqueles que podem ser observados em ambos os gráficos, marcas (Figura 33a/b) e (Figura 33b/d), originam de erros esporádicos na estimativa do plano do chão

por parte do algoritmo RANSAC, que ocorrem devido à quantidade limitada de informação tridimensional na região que descreve o plano do chão. Através de uma análise qualitativa dos resultados obtidos, é possível concluir que os valores médios estimados para o ângulo de inclinação da câmara, θ , em relação ao plano do chão, mostram uma boa aproximação às observações efetuadas durante o processo de aquisição do conjunto de dados de teste, validando assim a terceira hipótese (R3). Contudo, uma análise qualitativa devera ser feita em trabalho futuro de forma a corroborar as observações feitas.

4.3.4 Avaliação da quarta hipótese (R4) em estudo

A quarta hipótese (R4) analisada pretende validar se a inclusão de informação tridimensional do ambiente permite ao sistema proposto corrigir situações de falha no detetor original, concretamente, evitar que este identifique como região de trilho áreas da imagem com obstáculos presentes. Na Secção 3.4 é detalhado todo o processo de integração da informação tridimensional no detetor original que permite, ao sistema proposto modular o comportamento dos agentes virtuais. O objetivo é limitar o depósito de feromonas virtuais nas regiões da imagem classificadas como obstáculo, permitindo que regiões potencialmente perigosas para o robô (e.g., árvores) não sejam identificadas como trilho, levando o detetor a procurar soluções alternativas.

Na Figura 23 é possível observar as regiões caracterizadas como obstáculo no conjunto de dados de teste e a influencia destas no filtro temporal do detetor utilizado para identificar a região hipótese do trilho. Concretamente, observa-se que os agentes virtuais tendem a rejeitar regiões da imagem identificadas como obstáculos, ajudando dessa forma a produzir deteções mais corretas. A Figura 34 compara o sistema proposto e o detetor original na mesma situação do Vídeo 6 em condições iguais, ou seja, onde é apenas analisado por ambos detetores o conjunto de *keyframes* disponibilizadas pelo LSD-SLAM. É possível observar que o detetor original é influenciado pelas condições de luz que tornam a região à esquerda da imagem

(i.e., arbusto) mais saliente que o trilho, logo sendo selecionada como região hipótese do trilho. Por outro lado, o sistema proposto é capaz de identificar essa mesma região como obstáculo (linha cinco da Figura 34) e assim evitar que sejam depositadas feromonas virtuais nessa região (linha quatro da Figura 34), evitando que a mesma seja escolhida como região hipótese do trilho. Para que fosse feita uma comparação justa ao detetor original, o mesmo foi testado na sequência de imagens completa do Vídeo 12 (não apenas às *keyframes*), onde é possível observar, através da Figura 35, um resultado semelhante ao obtido anteriormente, ou seja, o detetor original continua a falhar na identificação da região da imagem onde se encontra o trilho. Outro exemplo pode ser observado na Figura 36, referente ao teste realizado com o Vídeo 3, onde o sistema proposto é capaz de determinar que a região identificada pelo detetor original como pertencente ao trilho encontra-se na verdade demasiado elevada para o ser.

Uma vez que o resultado obtido pelo sistema proposto, à semelhança do detetor original, consiste na identificação de uma região hipótese que deve estar contida numa parte do trilho e não uma segmentação completa do mesmo, o método de avaliação quantitativo escolhido segue uma abordagem semelhante à utilizada para avaliar o detetor original [68]. Concretamente, o trilho é considerado corretamente identificado caso a mancha de cor vermelha, que representa as regiões do filtro temporal (NF) que registam intensidades acima de 80% da intensidade máxima, se encontre localizada dentro dos limites do trilho e a sua distribuição espacial esteja alinhada com a direção do trilho. Em situações onde exista ambiguidade gerada pela ocorrência de duas manchas vermelhas na imagem resultante, ou seja, quando são identificadas pelo sistema duas regiões hipótese para o trilho em simultâneo, deve ser escolhida como resultado produzido pelo sistema a mancha que apresenta uma maior intensidade no filtro temporal.

A Tabela 4.2 descreve os resultados obtidos através dos testes realizados ao conjunto de dados apresentado na Secção 4.2, tendo por base o critério de sucesso identificado no parágrafo anterior. De forma a efetuar uma comparação justa entre ambos os sistemas, para cada vídeo foram feitos um conjunto de cinco testes no detetor original e cinco testes no sistema proposto, em condições iguais, onde

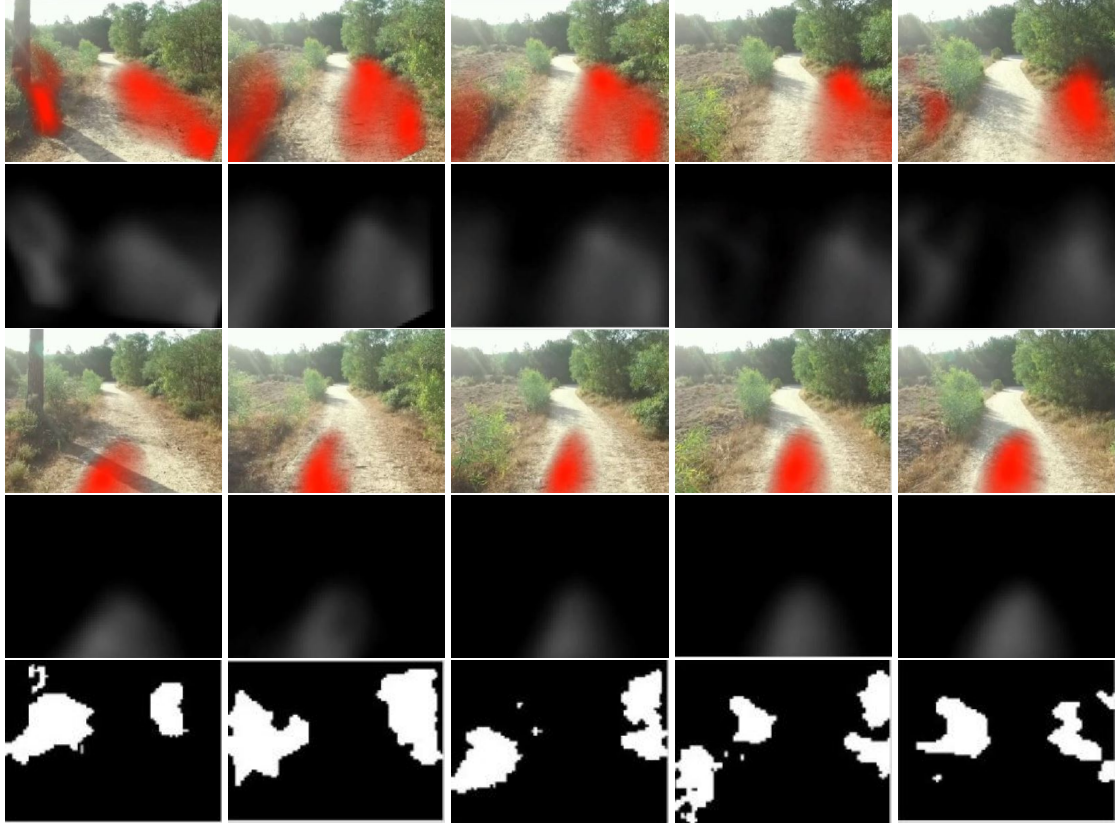


FIGURA 34: Imagens retiradas de testes realizados no Vídeo 6. A primeira linha ilustra o resultado obtido pelo detetor original, a terceira linha ilustra o resultado obtido pelo sistema proposto, a segunda e quarta linhas ilustram o filtro temporal, NF, que mostra atividade dos agentes virtuais em relação ao depósito de feromonas virtuais através de uma mancha vermelha e a linha cinco ilustra a detecção de obstáculos por parte do sistema proposto. O conjunto de resultados foi obtido correndo ambos os sistemas sobre as *keyframes* geradas pelo do LSD-SLAM.



FIGURA 35: Imagens do resultado obtido pelo detetor original numa situação de falha, no teste realizado à sequência de imagens completa do Vídeo 12. Na primeira linha é ilustrado o resultado obtido pelo detetor original, na segunda linha é ilustrado o resultado obtido pelo sistema proposto.

ambos os sistemas utilizaram como entrada as imagens RGB associadas às *keyframes* geradas pelo LSD-SLAM. A realização de múltiplas execuções por vídeo

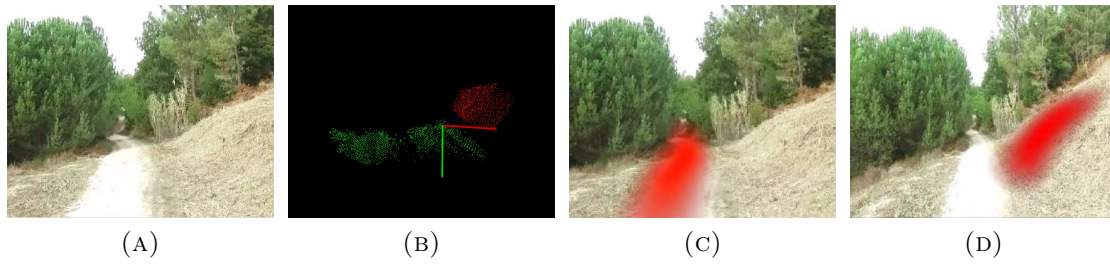


FIGURA 36: (A) Imagem analisada; (B) Informação tridimensional onde, os pontos verdes representam o plano do chão e os pontos vermelhos representam a região à direita do trilho que se encontra mais elevada; (C) Região hipótese do trilho identificada pelo sistema proposto; (D) Região hipótese do trilho identificada pelo detetor original.

TABELA 4.2: Conjunto de resultados obtidos nos testes realizados. Média e respetivo desvio padrão obtidos nos cinco testes realizados por vídeo, onde são avaliados os casos de sucesso em relação às *keyframes* analisadas.

Vídeo número	Número de <i>Keyframes</i>	Detetor original (<i>keyframes</i> corretamente classificadas [%])	Sistema proposto (<i>keyframes</i> corretamente classificadas [%])
1	585 ± 32	$89,98\% \pm 1,19\%$	$98,35\% \pm 1,05\%$
2	283 ± 27	$99,33\% \pm 0,92\%$	$100\% \pm 0,0\%$
3	470 ± 15	$77,67\% \pm 7,06\%$	$97,77\% \pm 0,52\%$
4	320 ± 20	$95,79\% \pm 3,04\%$	$95,61\% \pm 1,02\%$
5	235 ± 12	$86,54\% \pm 5,96\%$	$96,0\% \pm 3,83\%$
6	501 ± 25	$59,19\% \pm 21,52\%$	$98,64\% \pm 1,07\%$
7	482 ± 31	$94,94\% \pm 0,78\%$	$97,91\% \pm 1,16\%$
8	187 ± 19	$94,05\% \pm 4,77\%$	$97,60\% \pm 2,94\%$
9	411 ± 22	$93,18\% \pm 2,04\%$	$96,52\% \pm 1,02\%$
10	337 ± 23	$99,81\% \pm 0,39\%$	$100\% \pm 0,0\%$
11	507 ± 36	$95,79\% \pm 2,02\%$	$97,41\% \pm 1,14\%$
12	510 ± 11	$96,96\% \pm 3,04\%$	$97,77\% \pm 2,23\%$
Taxa de deteção [%]		$90,30\% \pm 4,22\%$	$97,78\% \pm 1,09\%$

visa compensar a influência das componentes estocásticas envolvidas na execução dos comportamentos dos agentes virtuais e do LSD-SLAM, uma vez que a quantidade de informação analisada está diretamente relacionada com a capacidade de processamento disponível em cada instante.

Uma análise aos resultados obtidos mostram que o detetor original obtém no conjunto de dados de teste uma taxa de sucesso de 90,30% com um desvio padrão de 4,22%. Confirma-se que o detetor original mostra falhar com maior frequência

quando influenciado pela presença de distratores nas regiões envolventes ao trilho, o que é agravado pelo facto de nessas situações o detetor aprender a aparência desses distratores como sendo a aparência do trilho, levando-o a propagar a falha ao longo das imagens seguintes e, inevitavelmente, a guiar o robô para regiões potencialmente ocupadas com obstáculos. Por outro lado, o sistema proposto apresenta um desvio padrão mais pequeno, de 1,09%, com uma taxa de sucesso na identificação da região hipótese do trilho de 97,78%. Com mais 7,86 % de sucesso que o detetor original, acompanhado de um menor desvio padrão, o sistema proposto comprova os benefícios de incluir informação tridimensional na modulação do comportamento do detetor original, validando assim a quarta hipótese (R4). A Figura 37 ilustra o conjunto de resultados típicos obtidos pelo sistema proposto, onde é possível observar a correta identificação da região hipótese do trilho.



FIGURA 37: Resultado obtido na identificação da região hipótese do trilho por vídeo. Da esquerda para a direita e de cima para baixo do Vídeo 1 até ao Vídeo 12.

4.3.5 Avaliação da quinta hipótese (R5) em estudo

Uma possível arquitetura para integração do sistema proposto é apresentada na Secção 3.5, onde é descrito o módulo Validador cuja funcionalidade principal é a

deteção de situações potencialmente perigosas para o robô. Este módulo é responsável por validar os resultados obtidos pelo detetor de trilhos através da informação tridimensional, o que permite obter um grau de certeza quanto à correção da identificação da região hipótese do trilho. Apesar de o detetor de trilhos proposto nesta dissertação mostrar uma maior robustez relativamente ao original em resultado da integração de informação tridimensional, como discutido na Secção 4.3.4, existem fatores que o poderão levar o mesmo a falhar. Por exemplo, mudanças repentinas do nível de iluminação podem induzir a falhas no método de cálculo da saliência visual, movimentos abruptos da câmara em condições de iluminação deficitárias podem levar a falhas no cálculo do fluxo ótico e mudanças abruptas na aparência do trilho podem levar a que o sistema aprenda a aparência errada, levando-o a desviar-se da correta identificação da região hipótese do trilho. O recurso à informação tridimensional do ambiente mostrou mitigar a ocorrência destas situações; contudo, em situações extremas como as identificadas anteriormente, a influência exercida pela informação de volumetria nos agentes virtuais pode não ser suficiente para impedir que regiões da imagem ocupadas por obstáculos sejam classificadas como região hipótese do trilho. A primeira linha da Figura 38 expõe uma destas situações, onde existe uma falha de fluxo ótico devido a um movimento rápido da câmara em conjugação com as condições de luz, o que leva a um espalhar da atividade do campo neuronal que induz o detetor de trilhos a classificar como trilho, de forma errada, uma região ocupada por obstáculos. Nesta situação, o módulo Validador, através da informação tridimensional, é capaz de sinalizar a ocorrência de uma falha na identificação da região hipótese do trilho. Por outro lado, em situações esporádicas, o algoritmo RANSAC pode falhar pela falta de informação tridimensional necessária para estimar o plano do chão, levando esta situação a que o sistema reporte, erradamente, a presença de obstáculos na região onde se encontra o trilho, como discutido mais à frente na Secção 4.3.6. Nesse caso o módulo Validador sinaliza como falha a discrepância entre a posição dos obstáculos detetados (erradamente em cima do trilho) e a posição do trilho estimada pelo detetor de trilhos, como é possível observar na segunda linha da Figura 38. O sistema proposto permite, assim, sinalizar situações de incorreção, sejam elas causadas por

problemas na reconstrução tridimensional, detecção do plano do chão ou problemas na identificação da região de trilho através da saliência visual, validando a quinta hipótese (R5) em estudo nesta dissertação. Apesar de não abordado nesta dissertação, esta sinalização pode ser utilizada por sistemas de navegação autônoma para robôs para, por exemplo, determinar em que instantes podem ou não recorrer ao detetor de trilhos para determinar a próxima ação.



FIGURA 38: Sub-conjunto de resultados típicos que demonstram o funcionamento do módulo Validador. A primeira linha ilustra a identificação de uma falha causada pela calculo do fluxo ótico. A segunda linha ilustra a identificação de uma falha causada por um errado sinalizar de obstáculos na região onde se encontra o trilho. A primeira coluna ilustra a projeção da informação da presença de obstáculos para o plano da câmara, a segunda coluna ilustra o resultado do detetor de trilhos, a terceira coluna ilustra a atividade do campo neuronal, NF , e a quarta coluna ilustra o resultado obtido pelo módulo Validador.

4.3.6 Casos de falha

O sistema proposto demonstrou, através dos resultados obtidos, ser capaz de identificar regiões potencialmente perigosas para o robô e de modelar com sucesso o comportamento do detetor original, produzindo assim uma detecção e seguimento de trilhos mais robusta. Contudo, o sistema proposto apresenta ainda alguns desafios que podem vir a ser resolvidos na sequência do trabalho realizado nesta dissertação. Um dos principais desafios identificados é o custo computacional exigido para a aquisição da informação tridimensional do ambiente, através de uma câmara monocular com recurso ao pacote de software LSD-LSAM. Diretamente relacionado com o custo computacional está o período de tempo necessário, pelo

LSD-SLAM, para gerar os mapas de profundidade, em média $1,154 \pm 0,27$ segundos. Contudo, com o avanço do poder computacional e o aparecimento de equipamentos como *Systems-on-a-Chip* (SOC) de alto desempenho, como o Nvidia TX2⁷, que permite um maior paralelismo na execução de tarefas, juntamente com a melhoria que se tem vindo a verificar na otimização de técnicas SLAM, este problema deveria ser resolvido num curto espaço de tempo. O sistema proposto foi desenhado de modo a ser modular, através da arquitetura ROS, contudo esta modularidade também traz um custo computacional associado à troca de mensagens entre os diferentes módulos, assim surge como solução alternativa a integração de todo o código num único pacote de software mais otimizado e que permitisse eliminar o custo associado à troca de mensagens, em detrimento da modularidade do sistema.

Além do tempo necessário entre aquisições dos mapas de profundidade, o sistema proposto necessita que haja translação suficiente da câmara para que o LSD-SLAM possa estimar a estrutura tridimensional do ambiente. A Figura 33 ilustra através das referências (Figura 33c), (Figura 33d) e (Figura 33e) erros na segmentação do plano do chão devido à falta de informação tridimensional, suficiente, necessária para que o algoritmo RANSAC possa estimar o plano de forma correta. Assim torna-se extremamente importante uma correta estimativa do plano do chão, onde o trilha se encontra descrito, de modo a segmentar os obstáculos que se encontram próximos do robô. Contudo, devido a irregularidades do trilha, que são por vezes observadas em ambientes florestais, a assunção de um terreno planar pode tornar-se frágil resultando por vezes em falsos positivos na deteção de obstáculos. Desta forma, o parâmetro que define o que é um obstáculo em função da altura ao plano do chão torna-se muito relevante em trilhos mais irregulares. A Figura 39 ilustra duas situações em que o valor definido para a altura não é suficientemente alto, o que promove a identificação de regiões pertencentes ao trilha como obstáculo. Consequentemente, os agentes virtuais são influenciados pela presença de um obstáculo que na realidade não existe, o que provoca uma

⁷Nvidia TX2: <http://www.nvidia.com/object/embedded-systems-dev-kits-modules.html>

interrupção no filtro temporal como é possível observar na quarta coluna da Figura 39. No entanto, o sistema proposto mostrou-se capaz de recuperar de tais situações nas imagens seguintes, devido à capacidade de exploração dos agentes virtuais e à forma como o filtro temporal é afetado pela máscara de obstáculos, m_{obs} , como discutido na Secção 3.4.

Na Secção 4.3.5 são também identificados casos de falha, que ocorrem tipicamente em situações extremas, onde fatores como a mudança repentina de luminosidade ou movimentos rápidos da câmara não permitem ao sistema proposto identificar corretamente a região hipótese do trilho. O módulo Validador mostrou-se capaz de mitigar estas situações através da sua identificação o que permite, por exemplo, notificar o sistema de navegação quando ao grau de certeza da correta identificação da região hipótese do trilho. Na ausência de informação tridimensional o sistema proposto comporta-se e falha nas situações onde o detetor original falha. Contudo, assim que a informação tridimensional é recuperada, o detetor aqui proposto tende a recuperar, não sendo esse o caso do detetor original.

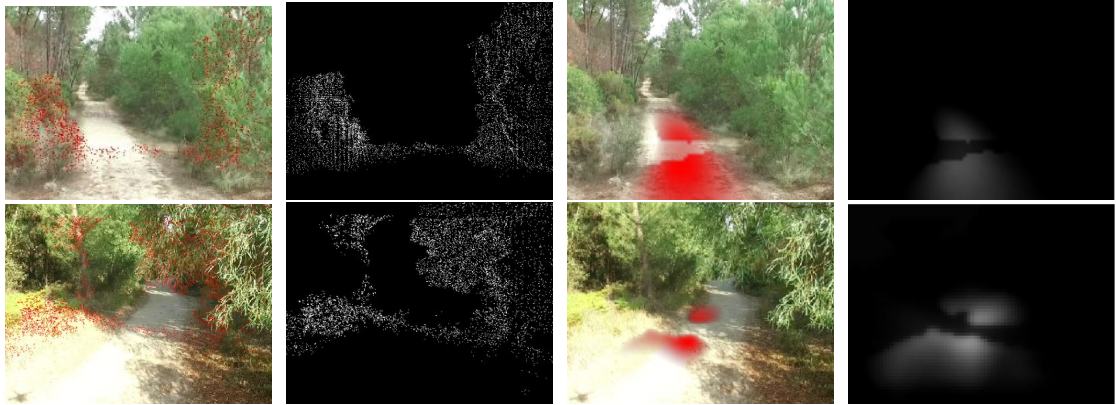


FIGURA 39: Casos de falha identificados durante a realização dos testes, onde primeira coluna ilustra sobreposição da informação tridimensional na imagem original, a segunda coluna ilustra a projeção da informação tridimensional no plano da câmara, a terceira coluna ilustra a sobreposição do filtro temporal, NF, na imagem original, e a quarta coluna ilustra o filtro temporal, NF.

Capítulo 5

Conclusões e Trabalho Futuro

Este capítulo apresenta um resumo dos resultados obtidos com a realização desta dissertação, e apresenta possíveis caminhos a seguir com o intuito de melhorar o sistema proposto.

5.1 Conclusões

O trabalho desenvolvido no âmbito desta dissertação consistiu no desenho e desenvolvimento de um sistema passível de ser utilizado como auxílio à navegação de um UAV, de pequena ou média dimensão em ambientes complexos, mais concretamente em ambiente florestal. O sistema proposto estende o trabalho desenvolvido no detetor original [68], ao qual foi adicionada a capacidade de utilizar informação tridimensional do ambiente, de modo a melhorar a sua robustez e permitir a sua utilização em UAV com recurso a uma câmara monocular.

Os resultados experimentais obtidos demonstraram que o sistema proposto é capaz de obter melhores resultados que o detetor original, no conjunto de dados teste utilizados. Concretamente, o sistema proposto obtém uma taxa de sucesso, na identificação da região hipótese do trilha, de 97,78%, que mostra uma melhoria de 7,48% em relação aos 90,30% obtidos pelo detetor original. Esta melhoria deve-se à inclusão de informação tridimensional sobre o ambiente, que permite mitigar a

identificação de regiões potencialmente perigosas para o robô pelo detetor original. Como discutido, na Secção 4.3, o detetor original na presença de distratores (e.g., árvores ou vegetação alta) que se apresentem mais salientes e com uma morfologia semelhante à região hipótese do trilho mostra falhar com maior frequência, o que é agravado pelo facto de nessas situações aprender a aparência desses distratores como sendo a aparência do trilho, propagando esse conhecimento ao longo do funcionamento do sistema.

Foi igualmente avaliada a capacidade do sistema proposto em estimar o ângulo de inclinação da câmara em relação ao solo (i.e., plano do chão), onde é possível concluir, através de uma análise qualitativa dos resultados, que o valor obtido mostra uma boa aproximação em relação ao valor esperado, para as observações realizadas durante o voo do UAV. Contudo, como forma de validar estes resultados torna-se necessário uma análise quantitativa, que deverá ser realizada em trabalho futuro.

Também foi possível validar a capacidade do sistema proposto em identificar situações em que as regiões potencialmente perigosas para o robô são identificadas como sendo a região hipótese do trilho. Por outro lado, a informação tridimensional utilizada para efetuar esta validação é a mesma utilizada para modelar o comportamento dos agentes virtuais, o que dificulta a capacidade do sistema proposto em desambiguar situações onde a reconstrução tridimensional não é corretamente obtida. Como forma de mitigar este problema deve ser avaliada, em trabalho futuro, a hipótese de o sistema proposto utilizar dois mecanismos de reconstrução tridimensional independentes, utilizados separadamente, para modelar o comportamento dos agentes virtuais e validar o resultado obtido pelo detetor de trilhos.

Outro resultado interessante é o facto da reconstrução tridimensional dos elementos distratores presentes no campo de visão do robô, ainda que não seja completa (i.e., a nuvem de pontos 3D é esparsa), permitir identificar os obstáculos com alguma resolução o que pode também ser explorado em trabalho futuro, por

exemplo, para identificar o tipo de objetos/elementos presentes no trilho ou nas regiões envolventes através da sua volumetria.

Contudo, apesar dos bons resultados obtidos, o sistema proposto apresenta um elevado custo computacional devido à aquisição da informação tridimensional do ambiente, o que consequentemente obriga o controlo a baixa velocidade do UAV (e.g., inferior a 1ms^{-1}). Por outro lado, o aumento na capacidade de processamento dos dispositivos moveis atuais, o aparecimento de sistemas integrados SOC de alto desempenho e o grande avanço que se tem verificado nos últimos anos no desenvolvimento e otimização das técnicas SLAM para câmaras monoculares [77], pode permitir que sistemas, como aquele que é proposto nesta dissertação, possam processar uma maior quantidade de informação e consequentemente permitir o controlo de UAV a maiores velocidades e igualmente aumentar a robustez no controlo a baixas velocidades. Outra possível solução para este problema, passa pela otimização do sistema proposto que permita uma análise completa à sequência de imagens, ao contrario do que se verifica atualmente onde só são analisadas as *keyframes* disponibilizadas pelo LSD-SLAM. Esta melhoria deve, assim, ser analisada e desenvolvida em trabalho futuro.

5.2 Trabalho Futuro

Com base nos resultados obtidos, esta secção apresenta as melhorias e complementos que podem ser implementados através da continuação do trabalho realizado no decorrer desta dissertação.

Como discutido na análise de resultados, Secção 4.3, os valores obtidos para o ângulo de inclinação da câmara em relação ao solo (i.e., plano do chão) são influenciados pela presença de *outliers* e pela incerteza associada ao cálculo do plano do chão, através do algoritmo RANSAC. Assim, deverá ser avaliada a aplicação de um estimador ótimo, como o filtro de Kalman [14], de forma de resolver

este problema. O filtro de Kalman poderá igualmente integrar informação de outros sensores, como a unidade inercial (IMU) do robô, de forma a permitir filtrar variações incoerentes nos valores estimados pelo sistema de visão.

Com vista a reduzir o custo computacional do sistema proposto, devem ser estudadas e avaliadas técnicas de processamento paralelo que possam ser integradas no mesmo, e que permitam a sua utilização nos UAV. Assim, surge a possibilidade de utilizar SOC de alto desempenho que permitam a utilização da *framework* CUDA. Esta *framework* recorre a uma GPU, integrada, que permite otimizar algoritmos de visão, tirando partido da sua elevada capacidade de paralelismo e do facto de estar preparada para lidar com operações matemáticas, como cálculo de matrizes e vetores. Contudo, a utilização desta *framework* implica uma transcrição do código utilizado para código específico para a plataforma CUDA. Outra possível solução para este problema, pode passar pela análise da região hipótese do trilho na sequência completa de imagens, partindo do pressuposto que a informação de translação da câmara, obtida pelo LSD-SLAM, permite projetar para todas as imagens da sequência a informação tridimensional obtida em cada *keyframe*.

Um possível complemento ao sistema proposto seria, com base nos resultados obtidos pela reconstrução tridimensional, catalogar os obstáculos de acordo com a sua volumetria. Foi possível observar que apesar de a nuvem de pontos tridimensional ser esparsa, os elementos caracterizados como obstáculo apresentam uma boa resolução que pode ser explorada de forma a identificar a que classe pertencem, como por exemplo distinguir uma árvore de um arbusto.

Por último, de modo a validar completamente o sistema proposto em operação real, o mesmo deverá ser integrado e testado com o sistema de navegação/planeamento de um robô, em particular para validar o auxílio à navegação de um UAV em ambiente florestal.

Bibliografia

- [1] Yaniv Alon, Andras Ferencz, and Amnon Shashua. Off-road path following using region classification and geometric projection constraints. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:689–696, 2006.
- [2] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz. Recent progress in road and lane detection: A survey, 2014.
- [3] Andreas Bartel, Frank Meyer, Christopher Sinke, Thomas Wiemann, Andreas Nuechter, Kai Lingemann, and Joachim Hertzberg. Real-time outdoor trail detection on a mobile robot. *Proceedings of the 13th IASTED International Conference on Robotics, Applications and Telematics*, pages 477–482, 2007.
- [4] Jens Berkmann and Terry Caelli. Computation of surface geometry and segmentation using covariance techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11):1114–1116, 1994.
- [5] Morten Rufus Blas, Motilal Agrawal, Aravind Sundaresan, and Kurt Konolige. Fast Color / Texture Segmentation For Outdoor Robots. *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference*, 1(d):22–26, 2008.
- [6] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. pages 1–9, 2016.

- [7] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems*. Number 1. Oxford university press, 1999.
- [8] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207, 2013.
- [9] Jean-Yves Bouguet. Matlab calibration tool, 2015.
- [10] Roland Brockers, Anthony Fragoso, Brandon Rothrock, Connor Lee, and Larry Matthies. Vision-based obstacle avoidance for micro air vehicles using an egocylindrical depth map. In *International Symposium on Experimental Robotics*, pages 505–514. Springer, 2016.
- [11] Alberto Broggi and Stefano Cattani. An agent based evolutionary approach to path detection for off-road vehicle guidance. *Pattern Recognition Letters*, 2006.
- [12] T. Chang and M. Shneier. Color model-based real-time learning for road following. *2006 IEEE Intelligent Transportation Systems Conference*, pages 939–944, 2006.
- [13] P Chaturvedi, A A Malcolm, and J Ibañez-Guzmán. Real-time road following in natural terrain. *2004 IEEE Conference on Cybernetics and Intelligent Systems*, pages 814–819, 2004.
- [14] SY Chen. Kalman filter for robot vision: a survey. *IEEE Transactions on Industrial Electronics*, 59(11):4409–4420, 2012.
- [15] H. D. Cheng, X. H. Jiang, Y. Sun, and Jingli Wang. Color image segmentation: Advances and prospects. *Pattern Recognition*, 34(12):2259–2281, 2001.
- [16] Ryan A Chisholm, Jinqiang Cui, Shawn KY Lum, and Ben M Chen. Uav lidar for below-canopy forest surveys. *Journal of Unmanned Vehicle Systems*, 1(01):61–68, 2013.

- [17] Dan Ciregan, Ueli Meier, and Jurgen Schmidhuber. Multi-column deep neural networks for image classification. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (February):3642–3649, 2012.
- [18] David Collogan. Uavs on the horizon. *Business & Commercial Aviation*, 2006.
- [19] J.D. Crisman and C.E. Thorpe. UNSCARF-a color vision system for the detection of unstructured\roads. *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, (April):2496–2501, 1991.
- [20] Jin Qiang Cui, Shupeng Lai, Xiangxu Dong, Peidong Liu, Ben M Chen, and Tong H Lee. Autonomous navigation of uav in forest. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 726–733. IEEE, 2014.
- [21] Hendrik Dahlkamp, Adrian Kaehler, David Stavens, Sebastian Thrun, and Gary Bradski. Self-supervised Monocular Road Detection in Desert Terrain. *Proc of Robotics Science and Systems RSS*, 2006.
- [22] Scaramuzza Davide and F Friedrich. Visual odometry: Part i: The first 30 years and fundamentals. *IEEE Robotics & Automation Magazine*, 2011.
- [23] J-L Deneubourg, Serge Aron, Simon Goss, and Jacques M Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of insect behavior*, 3(2):159–168, 1990.
- [24] Matthew Dunbabin and Lino Marques. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics & Automation Magazine*, 19(1):24–39, 2012.
- [25] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. *European C(Computer Vision – ECCV 2014)*:834–849, 2014.

- [26] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1456, 2013.
- [27] David Fernandez and Andrew Price. Visual detection and tracking of poorly structured dirt roads. *2005 International Conference on Advanced Robotics, ICAR '05, Proceedings*, 2005:553–560, 2005.
- [28] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [29] Patrick Fleischmann, Johannes Kneip, and Karsten Berns. An adaptive detection approach for autonomous forest path following using stereo vision. In *Control, Automation, Robotics and Vision (ICARCV), 2016 14th International Conference on*, pages 1–6. IEEE, 2016.
- [30] U. Franke, S. Mehring, A. Suissa, and S. Hahn. The Daimler-Benz steering assistant: a spin-off from autonomous driving. *Intelligent Vehicles Symposium*, pages 120–124, 1994.
- [31] Uwe Franke, Darius Gavrila, Steffen Görzig, Frank Lindner, Frank Paetzold, and Christian Wöhler. Autonomous driving goes downtown. *IEEE Intelligent Systems and Their Applications*, 13(6):40–48, 1998.
- [32] Alessandro Giusti, Jérôme Guzzi, Dan C Cirean, Fang-Lin He, Juan P Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, Davide Scaramuzza, and Luca M Gambardella. A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. pages 2377–3766.
- [33] Michael A Goodrich, Bryan S Morse, Damon Gerhardt, Joseph L Cooper, Morgan Quigley, Julie A Adams, and Curtis Humphrey. Supporting wilderness search and rescue using a camera-equipped mini uav. *Journal of Field Robotics*, 25(1-2):89–110, 2008.

- [34] PP Grassé. The automatic regulations of collective behavior of social insect and "stigmergy". *Journal de psychologie normale et pathologique*, 57:1–10, 1959.
- [35] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2:807–814, 2005.
- [36] Dirk Holz, Stefan Holzer, Radu Bogdan Rusu, and Sven Behnke. Real-time plane segmentation using rgb-d cameras. In *Robot Soccer World Cup*, pages 306–317. Springer, 2011.
- [37] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.
- [38] James Ingham. What is an agent. *Centre for Software Maintenance University of Durham*, 1997.
- [39] Yani Ioannou, Babak Taati, Robin Harrap, and Michael Greenspan. Difference of normals as a multi-scale operator in unorganized point clouds. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 501–508. IEEE, 2012.
- [40] Itseez. Open source computer vision library. <https://github.com/itseez/opencv>, 2015.
- [41] S.K. Kenue and General Motors Corporation. Research Laboratories. *LANELOC, Detection of Lane Boundaries and Vehicle Tracking Using Image-processing Techniques: Hough-Transform, region-tracing and correlation algorithms*. Number pt. 1 in Research publication. General Motors Research Laboratories, 1989.
- [42] Klaas Klasing, Daniel Althoff, Dirk Wollherr, and Martin Buss. Comparison of surface normal estimation methods for range sensing applications. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3206–3211. IEEE, 2009.

- [43] Jack Langelaan and Steve Rock. Towards autonomous uav flight in forests. In *Proc. of AIAA Guidance, Navigation and Control Conference*, 2005.
- [44] L Lucchese and S K Mitray. Color image segmentation: A state-of-the-art survey. *Proceedings of the Indian National Science Academy (INSA-A). Delhi, Indian: Natl Sci Acad*, 67:207–221, 2001.
- [45] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International Journal of Computer Vision*, 43(1):7–27, 2001.
- [46] Thomas Stützle Marco Dorigo, Mauro Birattari. Ant Colony Optimization . A Computational Intelligence Technique. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006.
- [47] Joel C. McCall and Mohan M. Trivedi. Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, 2006.
- [48] Anna Mulrine. Drones fill the troops gap in afghanistan. *US News & World Report*, 2008.
- [49] Montiel J. M. M. Mur-Artal, Raúl and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [50] Francesco Nex and Fabio Remondino. Uav for 3d mapping applications: a review. *Applied Geomatics*, 6(1):1–15, 2014.
- [51] Ryosuke Okuda, Yuki Kajiwara, and Kazuaki Terashima. A survey of technical trend of ADAS and autonomous driving. *Proceedings of Technical Program - 2014 International Symposium on VLSI Technology, Systems and Application, VLSI-TSA 2014*, pages 0–3, 2014.
- [52] Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

- [53] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. REMODE: Probabilistic, monocular dense reconstruction in real time. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [54] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. ROS : an open-source Robot Operating System. *ICRA workshop on open source software*, 3(Figure 1):5, 2009.
- [55] C. Rasmussen. Grouping dominant orientations for ill-structured road following. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 1:I–470, 2004.
- [56] Christopher Rasmussen. Appearance Contrast for Fast , Robust Trail-Following. pages 3505–3512, 2009.
- [57] Christopher Rasmussen, Yan Lu, and Mehmet Kocamaz. Trail following with omnidirectional vision. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pages 829–836, 2010.
- [58] Christopher Rasmussen, Yan Lu, and Mehmet Kocamaz. Integrating stereo structure for omnidirectional trail following. *IEEE International Conference on Intelligent Robots and Systems*, pages 4084–4090, 2011.
- [59] Christopher Rasmussen and Donald Scott. Shape-guided superpixel grouping for trail detection and tracking. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 4092–4097, 2008.
- [60] Christopher Rasmussen and Donald Scott. Terrain-Based Sensor Selection for Autonomous Trail Following. *International Workshop on Robot Vision, RobVis* 200:341–354, 2008.
- [61] Kocamaz M. Rasmussen C., Lu Y. Field and Service Robotics: Results of the 8th International Conference. *Springer Tracts in Advanced Robotics*, 92, 2014.

- [62] Stéphane Ross, Narek Melik-Barkhudarov, Kumar Shaurya Shankar, Andreas Wendel, Debadeepta Dey, J Andrew Bagnell, and Martial Hebert. Learning monocular reactive uav control in cluttered natural environments. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1765–1772. IEEE, 2013.
- [63] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
- [64] O. Sadano, H. Kawazoe, and M. Shimakage. Lane following vehicle control and process, October 8 2002. US Patent 6,463,369.
- [65] P Santana, R Mendonça, L Correia, and J Barata. Swarms for Robot Vision: The Case of Adaptive Visual Trail Detection and Tracking. *Proceedings of the European Conference on Artificial Life (ECAL)*, pages 712–719, 2011.
- [66] Pedro Santana, Nelson Alves, Luís Correia, and José Barata. A saliency-based approach to boost trail detection. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1426–1431, 2010.
- [67] Pedro Santana, Luís Correia, Ricardo Mendonça, Nelson Alves, and José Barata. Tracking natural trails with swarm-based visual saliency. *Journal of Field Robotics*, 30(1):64–86, 2013.
- [68] Pedro Santana, Ricardo Mendonca, Luís Correia, and José Barata. Neural-swarm visual saliency for path following. *Applied Soft Computing Journal*, 13(6):3021–3032, 2013.
- [69] Craig M Shakarji. Least-squares fitting algorithms of the nist algorithm testing system. *Journal of research of the National Institute of Standards and Technology*, 103(6):633, 1998.
- [70] Chanop Silpa-Anan and Richard Hartley. Optimised kd-trees for fast image descriptor matching. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

- [71] Peter Sturm. Pinhole camera model. In *Computer Vision*, pages 610–613. Springer, 2014.
- [72] Charles E. Thorpe. SCARF: A Color Vision System that Tracks Roads and Intersections. *IEEE Transactions on Robotics and Automation*, 9(1):49–58, 1993.
- [73] Sebastian Thrun and John J. Leonard. *Simultaneous Localization and Mapping*, pages 871–889. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [74] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars Erik Jendrossek, Christian Kolen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the DARPA Grand Challenge. *Springer Tracts in Advanced Robotics*, 36(9):1–43, 2007.
- [75] Teodor Tomic, Korbinian Schmid, Philipp Lutz, Andreas Domel, Michael Kassecker, Elmar Mair, Iris Lynne Grix, Felix Ruess, Michael Suppa, and Darius Burschka. Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. *IEEE robotics & automation magazine*, 19(3):46–56, 2012.
- [76] Luke Wallace, Arko Lucieer, Christopher Watson, and Darren Turner. Development of a uav-lidar system with application to forest inventory. *Remote Sensing*, 4(6):1519–1543, 2012.
- [77] Georges Younes, Daniel Asmar, and Elie Shamma. A survey on non-filter-based monocular visual slam systems. *arXiv preprint arXiv:1607.00470*, 2016.
- [78] Li Jun Zhang, He Ming Jia, and Da Peng Jiang. Sliding mode prediction control for 3D path following of an underactuated AUV. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2014.

- [79] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673. Ieee, 1999.
- [80] Jianyang Zheng, Yinhai Wang, and Nancy L Nihan. Quantitative evaluation of gps performance under forest canopies. In *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE*, pages 777–782. IEEE, 2005.
- [81] Song-Chun Zhu, Cheng-en Guo, Yizhou Wang, and Zijian Xu. What are Textons? *International Journal of Computer Vision*, 62(1-2):121–143, 2005.