



Rafael José da  
Silva Martins

Robô para Remoção de Ervas em Passeios







**Rafael José da  
Silva Martins**

## **Robô para Remoção de Ervas em Passeios**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica de António José Ribeiro Neves, Professor do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e de José Manuel Neto Vieira, Professor do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.



**o júri / the jury**

presidente / president

**Professor Doutor José Nuno Panelas Nunes Lau**

Professor Auxiliar, Universidade de Aveiro

vogais / examiners committee

**Professor Doutor Vítor Manuel Ferreira dos Santos**

Professor Associado C/ Agregação, Universidade de Aveiro (Arguente Principal)

**Professor Doutor António José Ribeiro Neves**

Professor Auxiliar, Universidade de Aveiro (orientador)



## agradecimentos / acknowledgements

A realização desta dissertação de mestrado contou com inúmeros incentivos e apoios de várias pessoas. Sem elas, seria muito difícil concluir a mesma.

Em primeiro lugar agradecer à minha mãe pela insistência e perseverança desde muito cedo comigo. Uma lutadora e que muito me ensinou na vida. Lutar, conquistar e nunca desistir são qualidades que irei guardar para sempre.

Em segundo lugar, ao meu pai que desde muito cedo me incutiu a arte e a astúcia de conseguir resolver problemas. Desde muito cedo que os engenhos e o prazer pela descoberta foram assuntos de conversa de almoço.

Obrigado aos dois pilares mais importantes e que me possibilitaram chegar até aqui.

Agradecer ao meu irmão pelas horas a fio de conversas sobre "trabalho" que tanto me ajudam a descobrir novas coisas e a querer explorar mais.

Quero dar um especial agradecimento ao Professor Doutor António José Ribeiro Neves. Sem ele, nada do que é aqui apresentado seria possível. Apesar de todas as complicações no decorrer do trabalho esteve sempre presente e pronto a ajudar. Obrigado pelas indicações e pela persistência.

Agradecer a todos os meus amigos e amigas que acompanharam o meu percurso académico e que em muito contribuíram para ser possível dizer que este é apenas o começo de uma nova temporada.



## Resumo

As ervas presentes em espaços públicos constituem um problema para as entidades públicas que as tentam remover regularmente. Uma prática bastante comum na resolução deste problema é a utilização de herbicidas. Estes colocam em causa a saúde pública e contaminam os espaços onde são aplicados. Devido ao problema da abusiva utilização de herbicidas, torna-se urgente pensar em soluções alternativas que possibilitem a remoção de ervas.

Nesta Dissertação damos início ao desenvolvimento de uma solução para a remoção de ervas em passeios e parques públicos baseada na utilização de robôs. A solução apresentada passa pela construção de um robô autónomo que possa navegar em passeios e detetar ervas. A superfície em que iremos focar o trabalho é a calçada Portuguesa pois apresenta um desafio pouco explorado e de maior dificuldade dado à diversidade de padrões existentes nas ruas de Portugal.

O trabalho apresentado é um ponto de partida para uma abordagem baseada em robótica com vista à remoção de ervas em calçada Portuguesa. Os resultados apresentados são ainda preliminares e carecem de trabalho futuro. No entanto, é possível através do robô desenvolvido e dos algoritmos implementados reconhecer a calçada Portuguesa e possibilitar que o robô se desloque em direção às ervas para proceder à extração das mesmas, assim que o sistema de extração seja desenvolvido e acoplado ao robô.



## **Abstract**

The herbs present in public spaces are a problem for the public entities that try to remove them regularly. A common practice in solving this problem is the use of herbicides. However, they create a public health problem contaminating the spaces where they are applied. Due to the problem of the abusive use of herbicides, it is urgent to think of alternative solutions that allow the removal of herbs.

In this Thesis we start the development of a solution for the removal of herbs in public parks based on the use of robots. The solution presented starts by the construction of an autonomous robot that is able to navigate in walks and detect herbs. The surface on which we will focus the work is the Portuguese pavement because it presents a little explored challenge of great difficulty due to the diversity of patterns existing in the streets of Portugal.

The work presented is a starting point for a robotic approach to the removal of herbs in Portuguese sidewalk. The results presented are still preliminary and require future work. However, it is possible using the developed robot and the implemented algorithms to recognize the Portuguese sidewalk and to allow the robot to move towards the herbs to proceed with the extraction, once the extraction system is developed and coupled to the robot.



# Conteúdo

<b>Conteúdo</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tabelas</b>	<b>v</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização e Relevância . . . . .	2
1.2 Contributos Semelhantes . . . . .	4
1.3 Problemas esperados . . . . .	6
1.4 Estrutura do Documento . . . . .	10
<b>2 Plataforma</b>	<b>11</b>
2.1 Sistema de Locomoção . . . . .	11
2.2 Modelo de Controlo . . . . .	12
2.3 Plataforma Final . . . . .	13
<b>3 Sistema de Visão</b>	<b>15</b>
3.1 Abordagem desenvolvida . . . . .	16
3.1.1 Calibração de Câmara . . . . .	16
3.1.2 Grelha de análise . . . . .	17
3.1.3 Histograma de Cor . . . . .	19
3.1.4 Concentração de textura . . . . .	19
3.1.5 Local Binary Patterns . . . . .	20
3.2 Resultados Experimentais . . . . .	21

<b>4</b>	<b>Desenvolvimento</b>	<b>25</b>
4.1	Arquitetura . . . . .	25
4.2	Modelo de Comunicação . . . . .	27
4.3	Modelo de Controlo e Decisão . . . . .	30
4.4	Ferramentas Desenvolvidas . . . . .	31
4.4.1	Servidores/ Serviços . . . . .	32
4.4.2	Aplicações Móveis . . . . .	32
4.4.3	Nó de processamento . . . . .	33
4.4.4	Serviço de Controlo de Cinemática . . . . .	34
<b>5</b>	<b>Conclusões</b>	<b>37</b>
	<b>Bibliografia</b>	<b>39</b>

# Lista de Figuras

1.1	Exemplo de calçada portuguesa. . . . .	2
1.2	Imagem do robô Bonirob . . . . .	5
1.3	Imagem do robô Tertill . . . . .	5
1.4	Exemplo de passeio degradado e padrão na calçada portuguesa. . . . .	9
1.5	Exemplo de local onde as ervas estão muito perto das paredes. . . . .	10
2.1	Representação de Modelo de Controlo. . . . .	13
2.2	Plataforma final. . . . .	14
3.1	Algoritmo de expansão. . . . .	18
3.2	Resultado dos cálculos dos histogramas. . . . .	20
3.3	Resultado de uma imagem após a utilização do algoritmo de Canny. . . . .	20
3.4	Algoritmo LBP com 8 bits. . . . .	21
3.5	Algoritmo genérico de deteção de semelhanças entre blocos na grelha de análise. . . . .	22
3.6	Testes ao algoritmo de deteção de passeios. . . . .	23
4.1	Diagrama da arquitetura do sistema desenvolvido. . . . .	26
4.2	Implementação do sistema de mensagens. . . . .	29
4.3	Descrição de funcionamento genérico do protocolo MQTT. . . . .	29
4.4	Máquina de estados do algoritmo de controlo automático. . . . .	30
4.5	Interfaces das aplicações desenvolvidas. . . . .	34



# Lista de Tabelas

4.1	Tabela de parâmetros configuráveis na unidade lógica. . . . .	35
4.2	Tabela de parâmetros configuráveis no serviço de cinemática. . . . .	36



# Capítulo 1

## Introdução

Os robôs são uma ferramenta cada vez mais utilizada no dia a dia, tanto em ambiente doméstico como industrial. Neste projeto propomos o desenvolvimento de um robô para a remoção de ervas em passeios. O equipamento deve ser autônomo e dotado de ferramentas que permitam a remoção eficaz de ervas em passeios ou espaços públicos. O equipamento que propomos deverá ser autônomo no que diz respeito à identificação de ervas, remoção das mesmas e na sua locomoção. Com vista ao cumprimento destes objetivos, serão estudados possíveis métodos para resolver os desafios apresentados e serão apresentadas as soluções implementadas.

Com o objetivo de analisar o meio envolvente ao robô e permitir ao sistema desenvolvido saber qual a posição exata da erva, de modo a que a remoção seja mais eficaz, bem como para que seja possível ao robô navegar em segurança no passeio, iremos utilizar uma câmara digital no espectro visível como principal sensor.

A solução apresentada terá em conta a localização absoluta do robô com base na utilização de um sistema *Global Positioning System* (GPS) e um comportamento reativo, isto é, calcula em cada instante a trajetória em direção ao próximo ponto do mundo detetado como erva. Para que este sistema seja funcional, serão estudados e desenvolvidos algoritmos de processamento de imagem para a correta delimitação de zonas livres e identificação de obstáculos e ervas. Devido a questões de usabilidade, a solução desenvolvida não pressupõe um conhecimento prévio do espaço onde vai ser utilizado.

## 1.1 Contextualização e Relevância

A calçada portuguesa, que remonta ao século XIX, é um marco histórico na cultura portuguesa, representando uma grande parte dos pavimentos, passeios e estradas públicas em Portugal. O facto de a calçada ser construída com base em várias pedras pequenas encostadas (Figura 1.1), estas criam pequenas fissuras que permitem o crescimento de ervas daninhas, danificando e sujando a via pública [1].



Figura 1.1: Exemplo de calçada portuguesa.

As entidades públicas estão neste momento divididas quanto aos métodos a utilizar para a eliminação de ervas em espaços públicos. Uma grande discussão está a ser levantada de momento devido ao uso de herbicidas para acabar com as ervas. No entanto, sabemos que o herbicida mais utilizado atualmente é o glifosato. Tal escolha deve-se não só pelos excelentes resultados que apresenta no controlo e prevenção de ervas mas, também, pelo baixo custo.

Contudo, o uso do glifosato está a levantar algumas questões. Por um lado, nos Estados Unidos da América, mais concretamente em São Francisco, o tribunal declarou uma sentença à empresa Monsanto de pagar uma coima no valor de 290 milhões de dólares ao agricultor Dewayne Johnson com um caso de cancro terminal. No entanto a Bayer (empresa que comprou a Monsanto) recorreu da sentença e afirmou que o glifosato não é cancerígeno [2] [3] [4]. Por outro lado, a UE (União Europeia) em 2009 lançou um despacho para os países membros que visava o controlo e aplicação de pesticidas que pretendia condicionar a utilização do glifosato [5]. No entanto, na altura não se encontravam soluções

viáveis para o combate deste problema, pelo que em Portugal, o glifosato foi permitido de novo [6]. Mais recentemente, em 2017 a utilização deste químico foi prolongada e aceite até 2022, mesmo que as opiniões se dividam nos países membros [7].

A OMS (Organização Mundial de Saúde) desaconselha o uso do produto em discussão, uma vez que nos vários estudos elaborados em diferentes países, os níveis de toxicidade no ser humano estão fortemente relacionados ao químico utilizado pelas entidades públicas e empresas de agricultura [8].

O glifosato é neste momento o herbicida mais vendido no mundo e é utilizado de forma massiva na agricultura, florestas, parques verdes e jardins [9]. O objetivo deste produto é produzir um bloqueio na enzima necessária para a planta criar proteínas e aminoácidos, evitando que estas se desenvolvam e acabem por morrer.

Pelo facto de as ervas crescerem de forma rápida, os municípios gastam quantias consideráveis de dinheiro no combate a este problema. Contudo, o maior problema não é a verba despendida, mas sim, o risco que a aplicação destes produtos expõe para a saúde pública e para o meio ambiente. Ao longo dos últimos anos, entidades políticas têm reunido esforços para que de alguma forma se possa controlar a utilização de pesticidas, sendo mesmo o objetivo primordial acabar por completo com o uso destes produtos. A utilização de produtos geneticamente modificados com base no glifosato têm um impacto no meio ambiente que a comunidade ainda não está ciente pois, segundo alguns estudos, o uso em massa de herbicidas apresenta graves problemas para a saúde pública. Segundo a OMS o glifosato é provavelmente cancerígeno em humanos e demonstrado em laboratório ser cancerígeno em animais [8]. Um outro estudo, conduzido na Espanha, indica que entre os anos de 1999 e 2009 foram diagnosticados cerca de 2.661 casos de cancro da mama na população feminina, dos quais 2.173 (81%) foram observados em áreas com níveis de contaminação elevados por pesticidas [10].

Pelo que podemos ver, existe neste momento uma necessidade para acabar com o uso de pesticidas nas áreas públicas. No entanto, ainda não existem soluções tecnológicas para a remoção de ervas em espaços públicos, sendo o objetivo deste trabalho poder dar um contributo significativo para a resolução deste problema.

## 1.2 Contributos Semelhantes

Existem projetos neste momento que visam o desenvolvimento de soluções robóticas para aplicação em agricultura, no entanto, nenhum se identifica completamente com as necessidades do problema referido anteriormente. Um dos projetos é o robô doméstico Tertill, que está a ser criado pela Franklin Robotics, LLC [11]. Este robô é completamente autónomo e através de um algoritmo de varrimento de área, vai eliminando as ervas à medida que percorre o espaço. Outro projeto é o Bonirob, que está a ser desenvolvido pela Bosch Deepfield Robotics [12] [13]. Este robô pretende substituir o homem nas tarefas diárias de um agricultor, como por exemplo, remover ervas, manter o solo limpo e avaliar se existe algum tipo de contaminação nos vegetais. Assim, através de tecnologia de localização, este robô é capaz de percorrer um terreno e identificar erva a erva e eliminá-la de forma completamente autónoma. Este último projeto ainda está em fase de testes e é apresentado sob forma de protótipo, não estando a ser comercializado. Relativamente aos dois projetos apresentados, é possível identificar alguns aspetos mais relevantes para o nosso objetivo:

- Bonirob: o sistema de identificação e categorização de plantas é um dos aspetos mais importantes de referir, pois permite ao robô estar constantemente a analisar as plantas e eliminar apenas aquelas que são efetivamente ervas. Este sistema é composto por uma campânula que passa por cima das plantas. Desta forma, controla a quantidade de luz e permite uma melhor aquisição de imagem por parte da câmara para a identificação das ervas daninhas.

Existem três formas de operação do robô (Figura 1.2): controlo remoto, localização e navegação por GPS e através da leitura das linhas do terreno.

A erva é eliminada através de um êmbolo mecânico que empurra a erva para de baixo do solo fazendo com que a erva não tenha mais contacto com a luz solar, impossibilitando o seu crescimento. Para um melhor desempenho, a empresa está a desenvolver um conjunto de êmbolos que trabalham independentes para que seja mais rápido a execução.

- Tertill: este equipamento deteta as ervas através de sensores que a empresa não revelou publicamente (Figura 1.3). No entanto, se a erva for maior do que o espaço entre o terreno e a parte inferior do robô a erva não é cortada. Este sistema permite eliminar



Figura 1.2: Imagem do robô Bonirob



Figura 1.3: Imagem do robô Tertill

---

a erva num estado prematuro do crescimento. Por outro lado, se as ervas tiverem uma dimensão maior o robô irá ignorá-las. Pelo facto de o robô eliminar as ervas da forma apresentada, se for plantada uma semente ou uma planta de um tamanho reduzido é necessário colocar algo à volta da planta para delinear esse espaço onde o robô não deve atuar.

O sistema de navegação do robô é por varrimento, ou seja, não existe nenhum mecanismo ou sensor que avalie o terreno e que preveja um determinado caminho mais apropriado. O robô sabe que deve virar quando bater em algo através dos sensores de toque laterais.

A erva é eliminada através de um fio de náilon que circula num eixo a altas rotações, um método normalmente utilizado por jardineiros. Esta forma de eliminar a erva é bastante simples e é utilizada de forma natural pelas empresas, sendo então uma

solução pouco dispendiosa. No nosso caso pode ser uma solução inicial para a remoção da erva mas que tem o problema de deixar a erva cortada para trás, sujando os passeios.

Ambos os projetos apresentam vantagens e desvantagens para a nossa discussão quanto a metodologias ou tecnologias a utilizar para a identificação de ervas daninhas. Podemos perceber que um robô tem um sistema de identificação de ervas bastante sofisticado e dispendioso (Bonirob), por outro lado, o projeto Tertil apresenta um sistema de sensores relativamente simples.

Seria interessante encontrar uma forma de combinar os métodos de deteção das ervas apresentados anteriormente, ou seja, identificar a erva com alguma distância através do uso de imagem digital com o objetivo de direcionar o robô para uma posição que permitisse a ativação de um outro sensor mais próximo da erva, para assim efetuar uma extração ou eliminação da erva de forma mais precisa.

Relativamente à extração de informação do ambiente circundante, o Bonirob utiliza dois sistemas: um sensor *Light Detection And Ranging* (LIDAR) e um sensor de GPS, que no nosso caso iria encarecer muito o projeto. Quanto ao projeto Tertil, este não apresenta qualquer tipo de sensor que lhe permita antecipar cálculos de rota. Estas duas abordagens são interessantes do ponto de vista em que devemos adaptar o trabalho à situação mais oportuna. Podemos retirar dos dois exemplos anteriores que seria interessante incorporar um sensor GPS para a localização absoluta do robô e uma câmara digital para estimar a movimentação, tendo assim uma solução não muito dispendiosa e que permita saber onde o robô se encontra e para onde se deve dirigir.

### 1.3 Problemas esperados

Devido ao facto de o robô operar num ambiente exterior, iremos encontrar alguns desafios na elaboração do projeto pois os fatores externos são mais difíceis de controlar. No entanto, podemos prever algumas situações de modo a obter um resultado melhor. De seguida serão enumerados alguns problemas que consideramos relevantes numa primeira fase de desenvolvimento:

- Variações de luz: a visão humana está tão habituada a identificar sombras que para nós não é um problema detetar qualquer tipo de sombra projetada no chão. Em

sistemas de visão por computador o problema inerente à existência de sombras é uma questão a ter especial atenção.

Um trabalho realizado na Universidade de Carnegie Mellon mostra uma forma de detetar sombras projetadas por objetos utilizando apenas uma imagem. O método apresentado assenta principalmente em classificação. Desta forma, os autores treinam um classificador com um conjunto de imagens contendo sombras. Após a classificação, são extraídas as características da imagem a avaliar e faz-se uma classificação tendo em conta o modelo treinado. Por fim, apresentam um segundo método que permite decidir se é efetivamente uma sombra ou não, tendo em conta o resultado do classificador. Os autores justificam este último método pelo facto de que existem nas imagens zonas que se assemelham a sombras, mas que na verdade não são. Os resultados apresentados neste projeto garantem 85% de eficácia em determinar sombras [14].

Uma vez que as ervas nascem muitas vezes junto às paredes de edifícios e acabam por criar sombras, é necessário tratar estas zonas com especial atenção. Através do apresentado no trabalho anterior, será possível detetar e identificar as ervas. No entanto, este método não foi utilizado devido ao facto de os classificadores apresentados pelos autores apresentarem maus resultados nos nossos cenários e não ter sido possível um novo treino dos mesmos, uma vez que seria necessário um conjunto de dados considerável para a análise, não disponíveis de momento.

- Detecção de obstáculos: quando falamos de deteção de objetos em robôs móveis que operam no exterior, para além das condições de luz natural, é necessário ter em consideração que existem dois movimentos importantes: o movimento da plataforma, bem como os sensores que nela estão inseridos, e o movimento dos próprios objetos que se deslocam no meio envolvente. Os métodos mais utilizados para resolver este problema são a utilização de estabilizadores de movimento acoplados à câmara, baseados no rastreio de características [15] e a computação do *optical flow* [16]. Estas aproximações focam-se em como obter a transformação que ocorre entre duas imagens.

Um estudo conduzido pela Universidade do Sul da Califórnia apresenta a possibilidade de serem detetados objetos em movimento em robôs móveis, o que se assemelha ao problema em análise, uma vez que é utilizada apenas uma câmara como principal

sensor. Este método divide o problema em duas partes: a primeira será corrigir o movimento da câmara; a segunda é estimar o movimento dos objetos no espaço. A forma de prever o movimento dos objetos no espaço é através de um filtro de Bayes e de um filtro de partículas adaptativo para uma estimativa interativa. Estes métodos são utilizados em conjunto. Numa primeira fase é calculada a diferença entre duas *frames*, por consequência resulta numa imagem com o deslocamento dos objetos. De seguida é utilizado o filtro de Bayes sobre o resultado anterior que permite calcular a probabilidade de movimento dos objetos. De seguida, é aplicado um filtro de partículas adaptativo, este método proposto por Dieter Fox em [17] permite obter resultados melhores que um filtro de partículas normais uma vez que utiliza informação sobre o deslocamento do robô. Foram realizados testes em três robôs diferentes neste estudo: O primeiro é um robô helicóptero, o segundo é um Segway RMP e finalmente um Pioneer2 AT. Estas plataformas apresentam diferentes movimentos de câmara e por isso foram escolhidos pelos autores para realizar os testes. Após os testes efetuados podemos ver que o robô helicóptero apresenta um rácio de deteção de 82,86%, o Segway RMP um rácio de 93,63% e o Pioneer2 AT de 66,28% [15]. Este estudo é bastante interessante uma vez que é possível obter a velocidade e direção dos objetos em movimento na cena. No entanto, este trabalho não se pode aplicar diretamente ao nosso problema uma vez que os próprios autores apresentam como dificuldade o facto de fazer o processamento em tempo real.

Um outro aspeto a ter em consideração é o pavimento onde o robô se desloca. Considerando que, em Portugal, é recorrente os passeios apresentarem alguma degradação, o robô deve evitar andar sobre esses espaços de modo a não ficar com o movimento preso (Figura 1.4).

- Minimização de custos associados à tecnologia: o objetivo deste projeto é que se torne numa solução passiva de ser comercializada. No entanto, a aquisição de alguns equipamentos iriam encarecer a solução final, pelo que o trabalho a ser desenvolvido terá como base alguns sensores com um custo mais baixo. Iremos utilizar como principal sensor uma simples câmara digital no espectro do visível e de um sensor GPS de baixo custo.
- Deteção de padrões na calçada Portuguesa: a calçada portuguesa é constituída por vários padrões e cores diferentes (Figura 1.4). Esta diversidade pode constituir alguns



Figura 1.4: Exemplo de passeio degradado e padrão na calçada portuguesa.

---

problemas, uma vez que o pavimento não é todo da mesma cor. A abordagem que iremos utilizar é baseada num algoritmo de deteção de pontos de fuga que permite detetar linhas com um ângulo subjacente, pelo que sabendo o ângulo das linhas, é possível saber se estamos perante uma linha horizontal, vertical ou oblíqua [18]. Desta forma será possível determinar os limites laterais e superiores da área em que o robô se esta a deslocar.

- Deteção de ervas: na agricultura de precisão são vastos os estudos realizados com o objetivo de controlar pragas e ervas. No entanto, estes trabalhos são conduzidos sobre a projeção de luz controlada [13] [19]. O que propomos é a realização de uma análise baseada em cor e padrões circundantes das ervas nas imagens recebidas. Uma vez que classificar todos os tipos de ervas iria acrescentar outros aspetos a ter em conta e encarecer a solução, não vamos considerar essa hipótese.

Para a deteção das ervas será avaliado o ambiente circundante, com o objetivo de detetar falhas num determinado padrão onde a erva está presente. Desta forma é feita uma avaliação da cor dessa falha. Se a cor da mesma corresponder a uma gama de cores verdes, então é identificada como erva.

- Remoção de ervas: as ervas encontram-se em locais onde normalmente não existe movimento ou passagem de humanos. Neste sentido, o mecanismo de remoção deverá ser algo que consiga remover a erva de locais como cantos, buracos no passeio, limites de passeio encostado a paredes (Figura 1.5), entre outros. Por motivos temporais este tema não será abordado, contudo é importante ressaltar o mesmo.



Figura 1.5: Exemplo de local onde as ervas estão muito perto das paredes.

---

## 1.4 Estrutura do Documento

Este documento apresenta a pesquisa e o trabalho desenvolvido em torno de um robô para remoção de ervas em calçada portuguesa. Desta forma surge a necessidade de dividir o problema em partes. O presente documento está dividido em seis capítulos representando os diferentes temas abordados. No Capítulo 2 são abordados os componentes de hardware utilizados que compõem o robô. No Capítulo 3 é abordado o sistema de visão implementado para resolver o problema de identificar os limites de um passeio e a identificação de ervas. No quarto capítulo apresentamos uma descrição do software desenvolvido, bem como a arquitetura proposta para a sistema. São apresentados também os modelos de comunicação, controlo e decisão do robô. No Capítulo 5 são mostradas todas as ferramentas, serviços e aplicações desenvolvidas que completam o sistema final numa camada de alto nível. Por fim, podemos encontrar as conclusões e sugestões de melhorias sobre o trabalho realizado.

# Capítulo 2

## Plataforma

Existem vários tipos de hardware ou plataformas que podem ser utilizadas para construir um robô móvel que possa ser utilizado em espaços exteriores. Para perceber qual a melhor plataforma a utilizar devemos analisar os requisitos necessários ao problema:

- O robô deve poder-se deslocar num pavimento irregular que contenha depressões ou supressões sem se danificar ou ficar preso.
- O equipamento deve poder-se deslocar e variar o seu sentido de deslocação de forma rápida e precisa com vista a poder posicionar-se para a próxima erva de forma eficaz.
- O custo de produção deve ter sido em conta, uma vez que o robô não deve ter um custo de produção muito elevado para que o mesmo seja passível de ser um produto comercializável.
- Devemos ter acesso à localização geográfica do robô em tempo real para que seja possível recolhe-lo por parte de um técnico ou responsável pelo equipamento.

Foi desenhada e desenvolvida uma plataforma de raiz com três rodas. A disposição de três rodas deve-se à facilidade de implementação e controlo. Desta forma, vamos ter duas rodas motrizes alinhadas na frente do robô e uma roda livre na retaguarda.

### 2.1 Sistema de Locomoção

Existem três categorias principais de locomoção robótica (rodas, lagartas e pernas) e uma quarta, considerada híbrida, onde são conjugadas estas três categorias principais.

Sabemos que os sistemas de locomoção com rodas são uma melhor opção de escolha, uma vez que são mais fáceis de implementar (dificuldade mecânica e complexidade de controlo) e mais eficientes (eficiência energética).

Outro aspeto que podemos considerar é que a plataforma terá de se deslocar em terrenos como a calçada portuguesa e não terá de subir ou escalar obstáculos [20]. Uma vez escolhido o tipo de locomoção, devemos agora perceber qual o tipo de rodas e geometria mais adequada ao problema. Existe uma quantidade considerável de geometrias possíveis a serem utilizadas com rodas, sendo que o mínimo de rodas necessárias para se fazer com que uma plataforma se desloque são duas. No entanto duas rodas para a nossa plataforma poderá apresentar vários problemas. Um deles será o facto de ser necessário transportar um mecanismo que faça a remoção e extração da erva, pelo que pode à partida introduzir alguns problemas quanto à estabilidade da plataforma. Uma geometria de quatro ou mais rodas é algo que podemos considerar, pois permite ter uma plataforma mais robusta e estável. No entanto, não terá qualquer tipo de sistema de amortecimento nos pontos de contacto com o solo. O pavimento em que o robô se vai deslocar, como descrito anteriormente, é bastante irregular pelo que iríamos muita vez perder o contacto de todos os pontos com o solo o que é algo não desejável. Outra solução será utilizar uma geometria de três rodas. A disposição que mais se destaca para o caso será a de duas rodas motrizes na frente e uma roda livre na traseira, uma vez que é um mecanismo de simples execução e acarreta menos custos de construção. As restantes geometrias de três rodas são igualmente interessantes e poderiam ser consideradas para o problema em causa, não obstante o modelo a utilizar será o apresentado anteriormente.

## 2.2 Modelo de Controlo

Para controlar os movimentos do robô foi utilizada a abordagem apresentada pelo estudo de Siegwart e Roland [20]. Este modelo de controlo considera que o robô se desloca sempre de um ponto inicial para um final em todas as iterações. Desta forma, são considerados dois planos, um global e um local para o robô. De seguida é calculada a distância entre a posição atual do robô e a final bem como os ângulos relativos ao desvio do robô para com a posição final. Este método permite também definir um modelo matemático que representa o deslocamento do robô no plano global de referência. Desta forma são considerados os valores de deslocamento obtidos pelos encoders dos motores em cada movimento

e vemos quanto é que o robô se desloca.

Podemos ver na Figura 2.1 como é representado o robô e quais os valores a serem considerados.

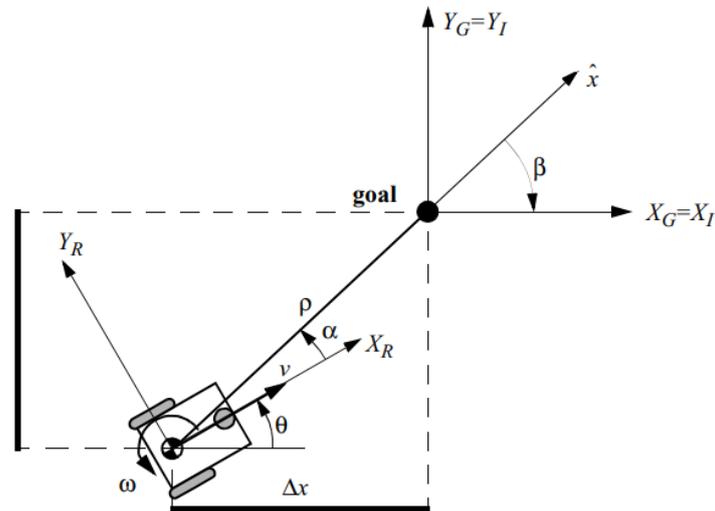


Figura 2.1: Representação de Modelo de Controlo [20].

## 2.3 Plataforma Final

Como descrito anteriormente, o robô é composto por duas rodas frontais que estão engrenadas num motor para cada uma. Atrás temos uma roda castor. A plataforma é composta ainda por alguns outros componentes:

- Uma bateria para alimentar os motores DC, o controlador dos mesmos e um ecrã LCD.
- Uma bateria para alimentar a unidade de processamento.
- Uma unidade de processamento Raspberry-pi III.
- Uma câmara digital.
- Uma Placa WIFI-USB.
- Um controlador para os motores elétricos.

A Figura 2.2 apresenta a plataforma final desenvolvida por nós de raiz e a disposição de todos os elementos descritos acima.



Figura 2.2: Plataforma final.

# Capítulo 3

## Sistema de Visão

A utilização de sistemas de visão artificial em ambientes exteriores é bastante desafiadora devido às geometrias irregulares, variações de luminosidade e muito ruído nas imagens recebidas. Contudo, esta é uma área muito explorada e que conta com algum trabalho desenvolvido.

Um estudo apresentado pela Universidade de Carnegie Mellon, apresenta uma solução bastante interessante na detecção de objetos e pavimentos utilizando apenas uma câmara digital. Os algoritmos apresentados baseiam-se nas cores dos objetos e do pavimento. Este estudo apresenta resultados muito bons para o tipo de pavimentos em que os testes foram aplicados (pavimentos regulares e sem padrões). Esta aproximação, tendo apenas em conta as cores dos objetos de interesse, será um ponto de partida para o nosso problema. Todavia, os padrões da calçada Portuguesa normalmente são expostos com cores diferentes e, tendo em conta este aspeto, o algoritmo proposto irá devolver os padrões da calçada como objetos e não como pavimento, pelo que não será suficiente para a resolução do problema [21].

O trabalho de Junseong [15] consiste em reconhecer passeios e pretende determinar pontos de fuga na imagem através de linhas de contorno ao redor dos passeios. Por sua vez, fizemos testes com este algoritmo e não obtivemos os resultados esperados. Devido à quantidade de detalhe, formas e cores a detecção de pontos de fuga fica condicionada devido à quantidade excessiva de linhas produzidas por tantas texturas e padrões. Como esperado, este algoritmo será uma excelente aproximação para pavimentos exteriores e interiores mas que não apresentem muita variação e detalhe.

Tirando partido dos trabalhos apresentados desenvolvemos uma solução que permite detetar de forma correta a calçada portuguesa.

## 3.1 Abordagem desenvolvida

Partindo do estudo dos trabalhos supracitados, a nossa abordagem será uma aproximação entre as texturas e cores apresentadas nos pavimentos exteriores. É intenção nossa também poder detetar pavimento com presença de ervas, padrões diferentes e condições de luz variáveis. Para que seja possível adquirir imagem com qualidade e que permita obter uma relação com o mundo físico, é necessário proceder à calibração da câmara a vários níveis.

### 3.1.1 Calibração de Câmara

A calibração da câmara será a primeira tarefa a ser considerada quando se coloca o robô em produção. Este é um procedimento importante e permite que sejam correlacionadas as unidades naturais da câmara e o mundo real (por exemplo milímetros) [22]. Foram utilizadas as ferramentas disponibilizadas pela biblioteca *OpenCV*. Esta ferramenta irá proceder às seguintes operações:

- Determinar a matriz de distorção, através do método da grelha de xadrez. A ideia é perceber como é que as linhas do tabuleiro de xadrez aparecem na câmara, que normalmente vão ser ligeiramente curvadas. Uma vez que na realidade são retas o algoritmo vai criar uma matriz que permite obter as relações entre a imagem original e a verdadeira.
- Obter uma *stream* de vídeo.
- Corrigir a imagem recebida da stream através da matriz de distorção.
- Ler a configuração de um ficheiro (XML/YAML).
- Guardar as configurações num ficheiro (XML/YAML).
- Calcular o erro de projecção.

Para que a calibração seja efetuada de forma correta deve ser utilizada uma grelha de xadrez. Todos os parâmetros de configuração de calibração estão disponíveis num ficheiro, permitindo assim configurar o tamanho da matriz de calibração, as dimensões de cada célula e o número de testes a efetuar para a calibração.

O sistema de visão irá ler a configuração previamente efetuada e o segundo passo no sistema de processamento será a aplicação de alguns algoritmos de pre-processamento. A imagem recebida pela câmara tem uma resolução de  $1296 \times 972$  pixels. Como não é necessário uma imagem com uma resolução tão elevada, iremos reduzir a mesma para  $512 \times 384$  pixels. Desta forma, vamos acelerar também o processamento da imagem.

A imagem é convertida do espaço de cor RGB para HSV. O espaço de cores HSV representa muito melhor a cor dos objetos de interesse, dando a possibilidade de configurar mais facilmente e com menos erro uma gama de cores para cada objeto. Posteriormente, a imagem é transposta para uma matriz que divide a mesma em blocos de  $n$  linhas e  $m$  colunas.

### 3.1.2 Grelha de análise

É utilizada uma matriz de  $n$  linhas e  $m$  colunas para a representação das várias áreas a analisar. A cada divisão chamamos de blocos.

$$A = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1j} \\ b_{21} & b_{22} & \dots & b_{2j} \\ \dots & \dots & \dots & \dots \\ b_{i1} & b_{i2} & \dots & b_{ij} \end{bmatrix}, \text{ com } 0 \leq i < n, \text{ e } 0 \leq j < m \quad (3.1)$$

Considerando uma imagem com  $l$  linhas e  $p$  colunas. Os pontos  $I$  (posição inicial) e  $F$  (posição final) representam uma seleção na imagem original. As aplicações  $\Theta(i, j)$  e  $\Omega(i, j)$  representam as transformações que permitem calcular os pontos  $I(x, y)$  e  $F(x, y)$  respetivamente às posições de  $A_{i,j}$  que definem cada bloco na imagem original. Os blocos de  $A_{i,j}$  são definidos pelas seguintes aplicações:

$$\Theta(i, j) = \begin{bmatrix} i \\ j \end{bmatrix} \begin{bmatrix} l & p \\ n & m \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (3.2)$$

$$\Omega(i, j) = \begin{bmatrix} i + 1 \\ j + 1 \end{bmatrix} \begin{bmatrix} l & p \\ n & m \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \end{bmatrix} \quad (3.3)$$

A análise começa no bloco mais próximo do robô e no centro  $A_{(i, \frac{m}{2})}$ . Os métodos que constituem a análise são descritos a seguir. No entanto, se a mesma retornar um valor de similaridade positivo, o algoritmo expande os próximos blocos, analisa os mesmos e compara com o anterior.

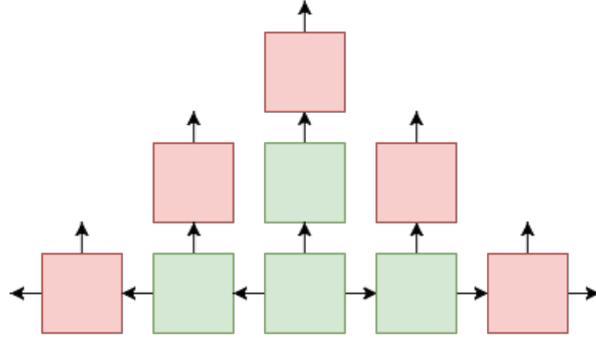


Figura 3.1: Algoritmo de expansão.

Após o cálculo do bloco  $A_{(i, \frac{m}{2})}$  vamos expandir os blocos  $A_{(i, \frac{m}{2}+1)}$ ,  $A_{(i, \frac{m}{2}-1)}$  e  $A_{(i-1, \frac{m}{2})}$ . Neste ponto do algoritmo, o bloco anterior  $A_{(i, \frac{m}{2})}$  é comparado com os valores de análise dos blocos expandidos. Se o valor retornado for positivo avançamos para os próximos blocos e assim sucessivamente. Este processo repete-se até que a análise retorne um valor negativo de similaridade e os próximos blocos não sejam expandidos. Se  $i = n$  então os blocos expandem para  $j - 1$  ou  $j + 1$  até se chegar a  $j = 0$  ou  $j = m$  respetivamente, mantendo-se o valor de  $i = n$ . Os blocos em  $i = n$  são também responsáveis por expandir os blocos em  $i - 1$  mantendo o valor de  $j$ . Os blocos com  $i < n$  expandem sempre em  $i - 1$  até  $i = 0$ , mantendo o valor de  $j$  inalterável.

Os valores de  $i$  e  $j$  devem ser considerados sempre consoante o pavimento a analisar. Ou seja, se queremos detetar um determinado padrão, devemos tentar que cada bloco de imagem inclua o máximo de detalhe do padrão.

Devemos ter em consideração que, ao aumentarmos o valor de  $i$  e  $j$ , estamos a diminuir a resolução da textura e dos padrões, o que, para valores mais baixos, o algoritmo poderá detetar uma grande parte da imagem como sendo a mesma textura, uma vez que a informação para cada bloco é maior. Aumentando os valores de  $i$  e  $j$  vamos também aumentar a complexidade do algoritmo e comprometer o desempenho deste. Neste sentido, devemos ter sempre em conta em que pavimento nos encontramos e como podemos tirar maior partido do algoritmo não comprometendo o desempenho geral.

Na Figura 3.1 podemos ver, de forma mais simples, que cada seta corresponde a uma comparação entre o bloco adjacente. Desta forma, o algoritmo só avança para o próximo bloco caso os três métodos seguintes retornem valores de similaridade aceitáveis dentro de um limite pré-definido.

### 3.1.3 Histograma de Cor

Os histogramas dão-nos a possibilidade de perceber a quantidade de uma cor numa dada imagem. Desta forma conseguimos avaliar a quantidade de cada cor respetivamente em cada bloco e compara-lo com os restantes blocos adjacentes. Desta forma, é mais simples perceber se um bloco adjacente apresenta uma distribuição similar em termos de cor. O cálculo de histogramas é feito em todos os blocos da grelha à medida que a propagação dos blocos ocorre. Com o intuito de perceber se a distribuição das cores de um nó é semelhante aos blocos adjacentes, os histogramas respetivos a cada bloco são comparados através do método *compareHist* fornecido pela biblioteca *OpenCV*. Esta comparação de valores é calculada através da distância de correlação entre os mesmos sendo o valor de retorno normalizado. Os valores de retorno da distância são sempre entre 0 e 1, em que 0 significa que não existe similaridade nenhuma e 1 total.

Podemos ver na Figura 3.2 três exemplos de imagens com os respetivos histogramas por baixo. Podemos reparar por análise dos gráficos que a primeira imagem (à esquerda), quando comparada com a imagem central, não apresenta muita relação de similaridade (na ordem dos 10%). No entanto, a imagem da direita, quando comparada com a do centro são bastante similares (semelhança de 85%). Neste caso estamos a avaliar a imagem por inteiro. A ideia deste exemplo é percebermos como se pode relacionar os histogramas. Os histogramas do centro e da direita (3.2) apresentam uma gama de cores semelhantes e daí o resultado de comparação apresentado ser maior. A solução implementada irá dividir a imagem original pela grelha de análise e calcular os histogramas de cada bloco para que se consiga perceber as similaridades a uma escala mais reduzida.

### 3.1.4 Concentração de textura

Para além da cor, é necessário perceber o detalhe das texturas presentes em cada bloco. O método de Canny [23], fornecido na biblioteca *OpenCV*, permite-nos com facilidade detetar arestas em imagens como apresentado na Figura 3.3. Este algoritmo, dada uma imagem inicial, calcula e devolve todas as arestas encontradas no bloco. Desta forma o procedimento utilizado é relativamente simples. Estamos à procura de arestas presentes em cada bloco, assim, o conjunto das arestas no bloco, definem padrões nas imagens. Após adquirir as arestas, calculamos a média das mesmas por bloco. Este valor é armazenado para futuros cálculos e de bloco para bloco os cálculos repetem-se. Com este método

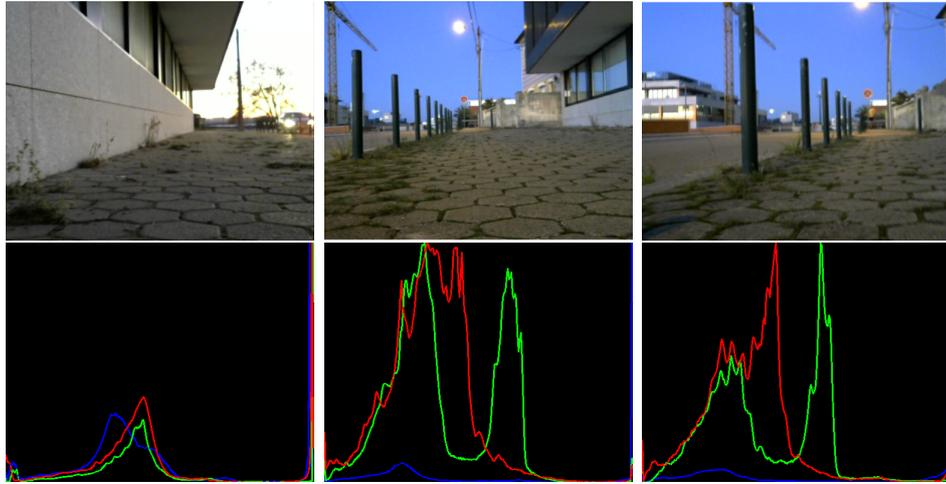


Figura 3.2: Resultado dos cálculos dos histogramas.

queremos mais uma forma de diferenciar os blocos e perceber se são semelhantes. Quando são comparados, se em dois blocos as médias foram iguais ou se estiverem dentro de um dado desvio padrão (pré-definido), os mesmos são considerados como semelhantes, caso contrário não são admitidos.



Figura 3.3: Resultado de uma imagem após a utilização do algoritmo de Canny.

### 3.1.5 Local Binary Patterns

Os descritores de imagem *Local Binary Patterns* (LBP) são bastante utilizados para extração de características com vista à utilização de algoritmos de classificação [24]. Este método consiste em simplesmente caracterizar a estrutura local de uma imagem comparando cada pixel com os seus vizinhos. Escolhendo um pixel no centro, comparamos diretamente

os pixels na sua vizinhança. Se o valor de intensidade dos vizinhos for menor que o centro atribuímos 0, se não, 1. Vamos, assim, acabar com um valor binário para cada pixel. Com 8 pixels na vizinhança temos  $2^8$  combinações possíveis, que são designadas por padrões binários locais (LBP no algoritmo original).

Com a utilização dos descritores LBP, é nossa intenção analisar a variação da textura num determinado bloco, comparar com a variação de textura do bloco seguinte e verificar a similaridade entre blocos, através de uma comparação dos histogramas. Assim sendo, como apresentado na Figura 3.4, vamos analisar pixel a pixel a imagem e comparar com os seus adjacentes. Daqui resulta um conjunto de códigos que serão normalizados após o cálculo do seu histograma. Os valores por fim são armazenados para comparação com outros blocos.

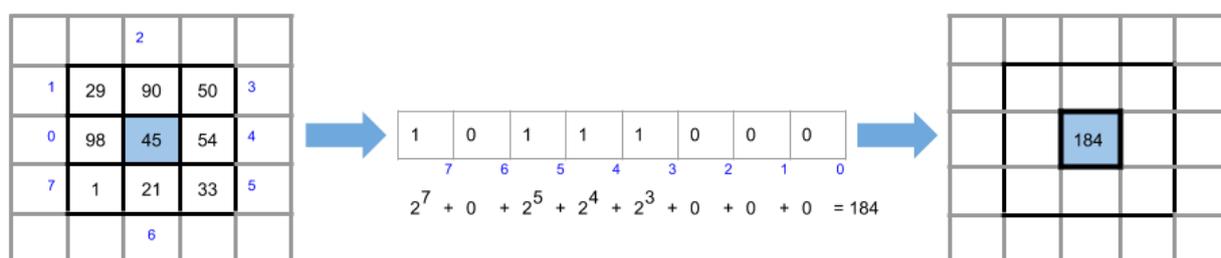


Figura 3.4: Algoritmo LBP com 8 bits.

Para que se consiga uma maior precisão da matriz de resultado, são considerados os três algoritmos em simultâneo para cada imagem adquirida. Após a conclusão dos cálculos, se os valores forem similares, é incrementado um valor na respetiva célula dando mais certeza. Quando o valor de certeza é superior a um determinado valor pré-definido, é considerado que o bloco atinge um valor válido. Em cada imagem, são atualizados todos os valores dos blocos. Podemos ver em mais detalhe na Figura 3.5 o algoritmo de forma geral. Se um bloco em análise deixa de ser válido por comparação com o seu vizinho, a célula respetiva é decrementada. Quando se obtiver um valor inferior ao aceitável por defeito, o bloco deixa de ser um bloco de zona confiável.

## 3.2 Resultados Experimentais

O algoritmo desenvolvido consegue expandir a textura de vários tipos de pavimentos em diferentes condições de luz como podemos ver na Figura 3.6. Os parâmetros de análise para

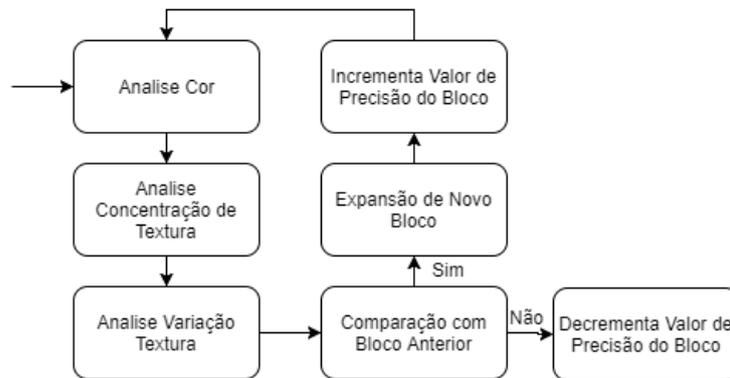


Figura 3.5: Algoritmo genérico de detecção de semelhanças entre blocos na grelha de análise.

os seguintes casos de teste não foram alterados com o intuito de obter melhores resultados, mas sim de perceber se o nosso algoritmo se adapta às várias condições diferentes sem a variação de parâmetros predefinidos. O resultado é, então, uma forma simples de detetar áreas semelhantes.

Como podemos perceber, o conjunto de métodos propostos permite definir uma área de interesse na frente do robô. Se existir uma mudança de padrão, cor ou textura, o algoritmo deixa de propagar os blocos, impedindo que seja uma área aconselhável. O algoritmo armazena os valores entre imagens. Assim podemos dar mais confiança nos resultados apresentados. No entanto, à medida que o robô se vai deslocando, se existir uma variação muito grande nos padrões, o mesmo deixa de andar e espera até ter resultados com mais confiança. Durante a execução dos testes podemos ver que o algoritmo não é 100% exato. Neste sentido, o problema das sombras mantém-se e deverá ser explorado como trabalho futuro. Podemos ver na imagem à esquerda da Figura 3.6 que a sombra é considerada um obstáculo quando na realidade não o é.

Uma vez que o robô analisa os primeiros blocos e os considera como o padrão por defeito a analisar, é preciso ter cuidado em colocar o robô numa posição inicial não obstruída ou à frente de sombras.

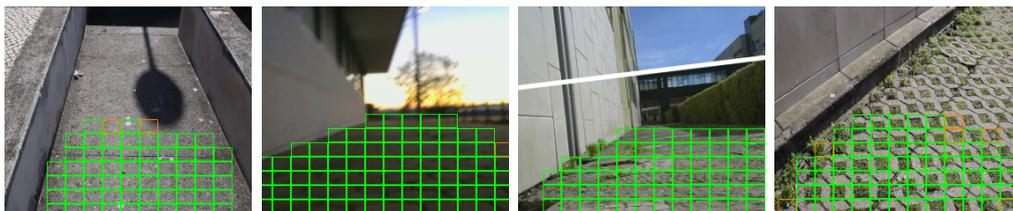


Figura 3.6: Testes ao algoritmo de detecção de passeios.



# Capítulo 4

## Desenvolvimento

Como apresentado até agora, existem vários componentes que constituem o sistema no geral, quer sejam de hardware ou de software. Para que o problema seja resolvido é necessário interligar todos estes sistemas para que seja um só. Neste capítulo iremos explicar como relacionamos todos estes componentes, quais os processos de comunicação, arquitetura de software escolhida e modelo de decisão.

### 4.1 Arquitetura

Queremos dar a possibilidade de descentralizar os processos e serviços do sistema, permitindo, assim, disponibilizar várias plataformas independentes de controlo. A arquitetura escolhida aproxima-se à de produtor/consumidor. Através desta, pretendemos dar a possibilidade a que um número de componentes independentes interajam e comuniquem entre si. Esta abordagem ao problema permite que seja fácil a substituição de componentes por outros ao longo do tempo, sem afetar o restante sistema, permitindo assim que exista uma quantidade de intervenientes variável e que possam interagir no sistema.

Um dos problemas que queremos resolver é dar a possibilidade de termos vários tipos de plataformas de controlo do robô no futuro, sejam elas aplicações móveis, site ou aplicação *desktop*. Através de uma arquitetura produtor/consumidor será mais fácil a interação entre os diferentes componentes [25]. Esta apresenta uma forte relação com sistemas distribuídos, uma vez que apenas temos de lidar com um tipo de comunicação previamente definido, permitindo assim que os sistemas interajam entre si mas não prejudiquem o comportamento dos outros diretamente se algo de errado acontecer. Podemos ver na Figura 4.1 como foi,

então, resolvido este problema.

Fazendo uma descrição de cada componente do diagrama de arquitetura, apresentado na Figura 4.1, destacamos os seguintes:

- Image Processing Device: unidade de processamento geral para determinar as ações seguintes. Recebe uma stream de vídeo do componente HTTP Vídeo Stream.
- HTTP Vídeo Stream: responsável por capturar o sinal de vídeo da câmara instalada no robô e partilhar um sinal online.
- Broker: serviço que é responsável por gerir as mensagens trocadas entre componentes.
- Controlo Cinemático: recebe instruções/comandos que devem ser executados colocando o robô a funcionar.
- Android Controlo: interface de interação para o cliente final.
- Cloud Storage: armazenamento de posições GPS ao longo do tempo.
- Android GPS: serviço a correr num aplicativo android que disponibiliza coordenadas GPS do dispositivo/ robô.

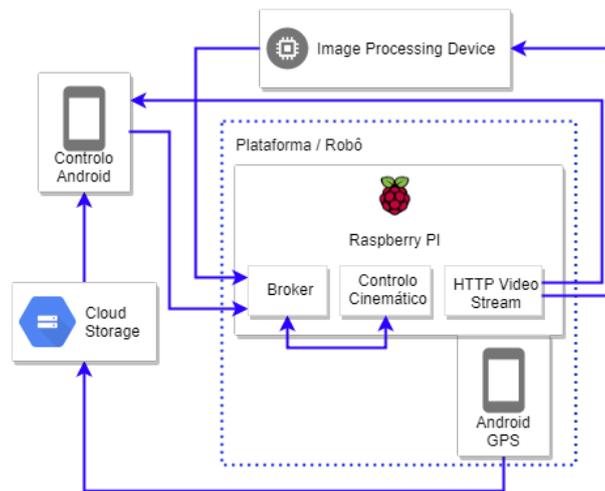


Figura 4.1: Diagrama da arquitetura do sistema desenvolvido.

O Raspberry Pi (instalado na plataforma do robô) é responsável por disponibilizar um broker de comunicação, um controlador cinemático e um servidor HTTP que disponibiliza

uma stream de vídeo capturada pela própria câmara. Na Plataforma está também instalado um smartphone android que incorpora uma aplicação que disponibiliza um sinal GPS e escreve num serviço de cloud.

O dispositivo de processamento de imagem é uma máquina instalada fora da plataforma do robô. Este é responsável por analisar a imagem fornecida pela stream de vídeo e dar instruções ao controlo cinemático através do broker.

A aplicação android de controlo é responsável por apresentar ao utilizador final um conjunto de opções e informações relativas ao sistema. Desta forma podemos dar início aos algoritmos de processamento, iniciar ou parar todas as tarefas do robô, bem como visualizar a posição absoluta do robô. Está também disponível a stream de vídeo fornecida pela plataforma em tempo real.

## 4.2 Modelo de Comunicação

Para que os diferentes componentes consigam comunicar é preciso escolher um meio e protocolo de comunicação. Uma vez escolhida a arquitetura as possibilidades de escolha ficam mais reduzidas. Contudo, podemos reparar que o modelo produtor consumidor está fortemente correlacionado com modelos de comunicação por filas de mensagens [25].

O protocolo *Message Queuing Telemetry Transport* (MQTT) é bastante simples e leve. Pelas mesmas razões é ideal para comunicação entre máquinas e dispositivos menos complexos. O protocolo proposto por Dr.c da IBM e Arlen Nipper da Arcom em 1999 foi desenhado para minimizar o impacto na largura de banda e maximizar as durações das baterias dos dispositivos [26]. Pelos motivos apresentados anteriormente, este será o protocolo escolhido para servir de comunicação entre dispositivos.

Uma tecnologia que utiliza o protocolo MQTT é o Eclipse paho, que disponibiliza uma vasta gama de clientes (produtores) escritos em diferentes linguagens. Desta forma, conseguimos uma tecnologia que se irá aplicar em todos os clientes desenvolvidos. Contudo, vamos analisar qual será o serviço a utilizar para receber as mensagens (broker). Este serviço será responsável por gerir a fila de mensagens e assegurar que as mesmas estão disponíveis para serem consumidas. O serviço escolhido foi o Eclipse Mosquitto. Este é um corretor de mensagens (message broker) open-source. Mais uma vez, este serviço foi desenhado para ser executado em servidores completos ou em pequenos dispositivos [26].

O protocolo funciona de maneira bastante simples. Podemos ver na Figura 4.3 que

existem dois intervenientes: o produtor (dispositivo 1), que produz mensagens num determinado tópico; e os consumidores (dispositivos 2 e 3), que consomem mensagens de determinado tópico se existirem. Numa primeira fase, os consumidores subscrevem um canal e aguardam até que haja mensagens para processar. Quando o produtor publica nesse tópico não espera que seja processada e segue a sua rotina.

No nosso caso a aproximação descrita atrás é muito interessante. A unidade de processamento de imagem (produtor) não tem de esperar que o controlador cinemático execute os seus cálculos para poder passar à próxima inspeção da imagem da stream. Desta forma o produtor inspeciona, envia a mensagem de comando e prossegue para a próxima inspeção. O controlo cinemático (subscritor/ consumidor) espera uma mensagem com o comando a efetuar, avalia a mesma e traduz em instrução. Esta abordagem é semelhante para a aplicação de controlo android. Podemos ver na Figura 4.2 a implementação do sistema de mensagens, onde toda a comunicação é feita no mesmo tópico. O controlo android e a unidade de processamento de imagem são os dois produtores de mensagens no nosso caso. Sendo que a última instância também é consumidora de mensagens. As mensagens produzidas são as seguintes:

- <START, i>: esta mensagem dá início ao controlador cinemático para ficar à espera de receber instruções de movimento. O parâmetro i é um inteiro que representa um valor booleano de 0 ou 1 que valida a instrução.
- <AUTO, i>: esta mensagem é direcionada à unidade de processamento de imagem. Indica que a mesma deve começar a inspecionar a stream de dados e publicar mensagens de movimento para o controlador cinemático. Apresenta também um valor i booleano que valida a mensagem.
- <FORWARD, X, Y, T>: esta mensagem é produzida pela unidade de processamento de imagem e é direcionada para o controlo cinemático indicando que o mesmo deve se direcionar para uma determinada posição. Os argumentos X e Y indicam a posição final desejada para o deslocamento. Já o argumento T indica o valor em graus de diferença entre o ponto atual e o final.
- <STOP, i>: a mensagem "STOP" indica paragem de todos os processos a decorrer. Destina-se à unidade de processamento de imagem e ao controlo cinemático. I é o argumento que valida a instrução.

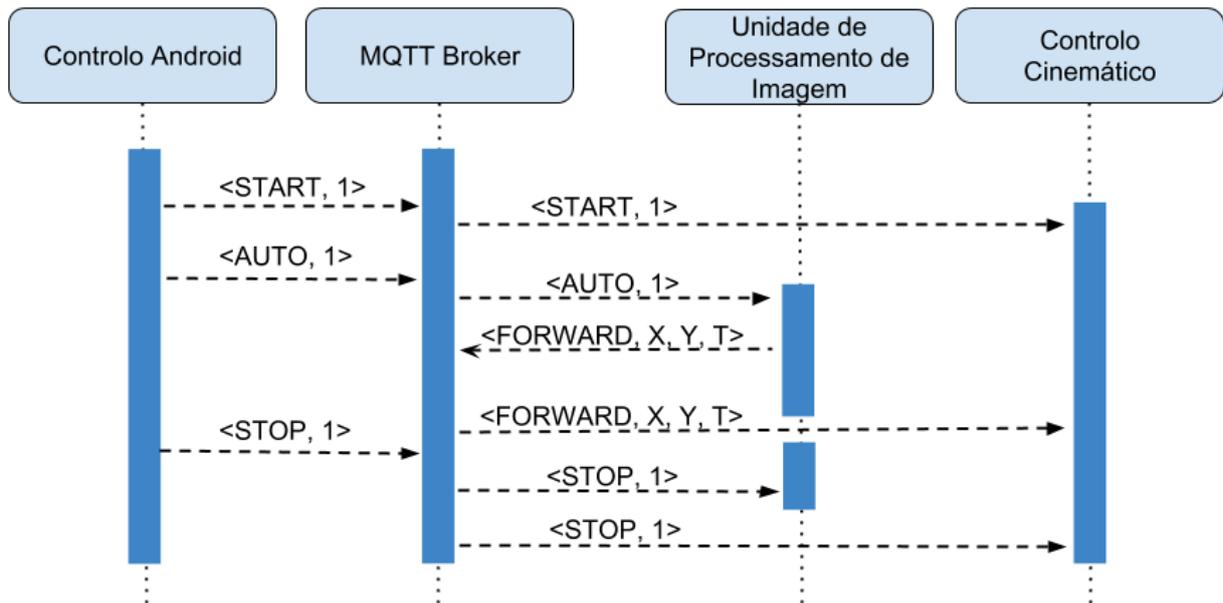


Figura 4.2: Implementação do sistema de mensagens.

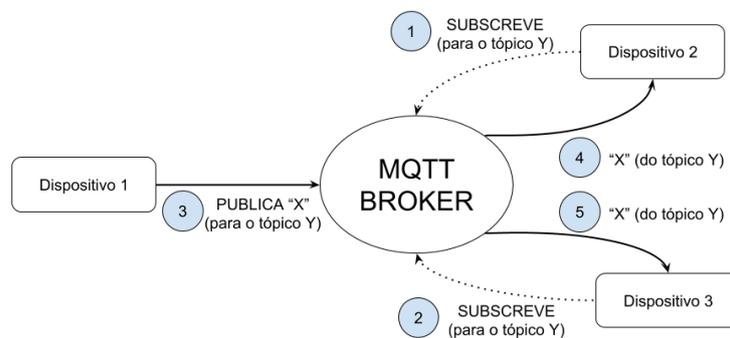


Figura 4.3: Descrição de funcionamento genérico do protocolo MQTT.

Foi também utilizada uma interface web que permite aceder à câmara instalada no robô. O serviço instalado foi o RPI-Cam-Web-Interface. Esta funciona sobre o servidor Web Apache e permite aceder à câmara através de um browser onde podemos configurar alguns parâmetros da câmara como por exemplo tempo de exposição, luminosidade, resolução entre outros aspetos [27]. O serviço disponibiliza também uma stream de vídeo MJPEG que permite aceder ao output da câmara através de um link direto para o mesmo [27].

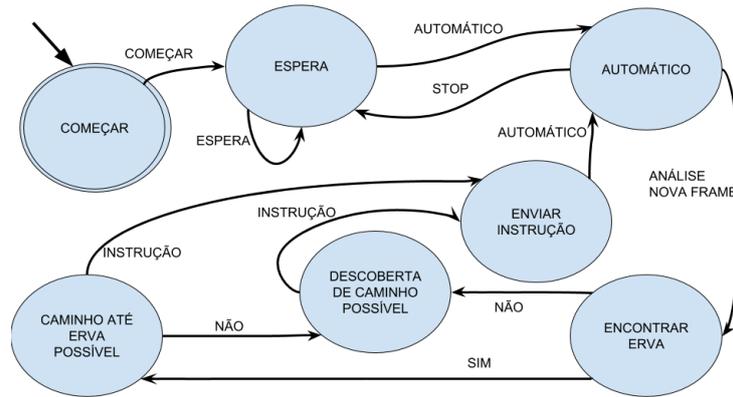


Figura 4.4: Máquina de estados do algoritmo de controlo automático.

### 4.3 Modelo de Controlo e Decisão

Vimos anteriormente como é que o robô se deve comportar quanto à sua deslocação. Contudo, falta descrever como e quando é que o mesmo se vai mexer de um ponto para o outro. Vamos descrever quem são os responsáveis por produzir mensagens que fazem com que o robô se desloque. Podemos considerar os dois responsáveis seguintes:

- O utilizador: consegue comandar o robô através de uma aplicação móvel que lhe permite enviar comandos como iniciar, parar e começar o movimento automático.
- A unidade de processamento: é responsável por adquirir imagem da câmara e calcular qual será a próxima posição que o robô deve se dirigir, desta forma o robô irá enviar comandos para o robô para que ele se movimente.

O utilizador na verdade é responsável por dar início ao processo de deslocamento, mas não interfere nada no mesmo, ou seja, fica da inteira responsabilidade da unidade de processamento a produção de mensagens de deslocamento. Podemos ver na Figura 4.4 a máquina de estados que traduz a lógica necessária para o robô se deslocar.

Inicialmente o algoritmo transita do estado iniciar para o estado de espera. Aqui, aguarda receber a instrução de reconhecimento automático. Aquando da mesma, começamos a avaliar uma frame da imagem proveniente da câmara do robô.

Começando pela pesquisa de ervas. A mesma é feita através de uma máscara. Esta permite detetar uma gama de verdes. A máscara retorna um conjunto de formas na imagem, são avaliadas consoante a sua área na imagem. As formas mais pequenas são

consideradas impurezas e são desprezadas. Por fim, são devolvidas as coordenadas da forma mais próxima do robô.

Após o reconhecimento da erva vamos encontrar um caminho válido até à erva. Se o mesmo for possível é então enviada a instrução de movimento e repete-se o algoritmo até que o mesmo receba a instrução STOP.

Caso a erva não seja detetada, vamos à procura de um caminho possível para prosseguir na descoberta.

Os caminhos até às ervas são calculados através de um algoritmo de pesquisa de custo mínimo A\*. É possível aplicar este algoritmo uma vez que quando a imagem é analisada, a grelha de análise é calculada de imediato e, assim, podemos relacionar a erva a um bloco e perceber se o caminho até ela é válido.

Se a erva não for encontrada vamos procurar um caminho livre para percorrer. Mais uma vez, recorrendo à grelha de análise vamos perceber em que sentido o robô se deve deslocar. Se o caminho estiver livre na frente do robô ele desloca-se nesse sentido. Caso esta condição não seja favorável, são analisados o lado direito e esquerdo da grelha e fazemos o robô se deslocar para o que estiver livre. No caso de todos os caminhos não estarem livres, deslocamos o robô para trás.

## 4.4 Ferramentas Desenvolvidas

Nesta secção vamos analisar as ferramentas de software desenvolvidas com vista a obter um sistema funcional. As principais ferramentas desenvolvidas são as seguintes:

- Aplicação Android para aquisição da localização do robô com base em GPS.
- Aplicação Android para controlo do robô.
- Servidor (nó de processamento), responsável pela inteligência artificial.
- Serviço de controlo de cinemática do robô.

Vamos analisar em mais detalhe nas seguintes secções como foram desenvolvidas as ferramentas e o que podemos esperar de cada uma delas. Vamos também descrever todos os serviços inerentes que são utilizados mas não desenvolvidos por nós.

#### 4.4.1 Servidores/ Serviços

Foram utilizados vários serviços e servidores:

- Servidor Apache: responsável por disponibilizar o serviço RPI-Cam-Web-Interface. Este permite apresentar uma interface web para controlar a câmara e ter acesso à mesma. Este servidor está instalado na unidade lógica a bordo no robô (Raspberry PI).
- Serviço Mosquitto: este serviço pode ser instalado quer na unidade de processamento do robô ou em outra máquina remota. Esta opção deve-se ao facto de a arquitetura ser distribuída e permitir que os serviços possam ser descentralizados. Para efeitos de teste o serviço foi colocado num nó de processamento remoto para que a unidade lógica do robô não tenha tanta sobrecarga de processamento, baixando assim o consumo do mesmo e aumentando a autonomia do robô.
- Serviço RPI-Cam-Web-Interface: interface de controlo que disponibiliza um serviço de stream de vídeo (MJPEG).
- Base de dados FireBase da Google: esta base de dados é responsável por armazenar os valores da geolocalização do robô em tempo real.

Todos estes serviços não foram desenvolvidos por nós, apenas utilizamos estas tecnologias para facilitar algum trabalho que poderia ser demoroso, pois estão testados e são versões estáveis de software.

#### 4.4.2 Aplicações Móveis

Foram desenvolvidas duas aplicações android para dispositivos móveis:

- Uma aplicação de geolocalização que funciona como um serviço. A única tarefa que a aplicação desempenha é atualizar a posição atual de um dispositivo num serviço externo da Google, onde é armazenado o último valor de geolocalização em tempo real. Esta aplicação não dispõe de interface gráfica. Apenas apresenta uma mensagem de que está a correr e podemos desativar o serviço clicando na mesma notificação na barra de atividade do dispositivo. Podemos ver a mensagem apresentada na imagem da esquerda na Figura 4.5.

- Aplicação de controlo do robô. Esta é mais complexa no sentido em que permite ao utilizador controlar em tempo real o que o robô está a fazer. Deve ser executada num dispositivo móvel de dimensões acima das 7 polegadas para que a interface seja mais perceptível, como podemos ver na imagem da direita na Figura 4.5. Esta interface apresenta duas imagens principais: a primeira é a representação em tempo real da trajetória descrita pelo robô; a segunda é a imagem em tempo real capturada pela câmara integrada no robô. Existe também a possibilidade de executar comandos diretamente no robô. Os mesmos são: *Connect*, conectar o dispositivo aos serviços externos; *Start*, comando para dar início à comunicação entre dispositivos; *Stop*, parar todas as interações com o robô no que diz respeito a movimento; *Lig./Desl.*, comando para começar o trabalho automático. Podemos configurar também os URL's de comunicação na parte de cima da interface.

O serviço de geolocalização foi desenvolvido numa aplicação móvel dado ao elevado custo de aquisição de um sensor GPS. Assim é disponibilizado o mesmo serviço a um preço de custo mais reduzido. Sabemos que a precisão destes equipamentos mais baratos não é elevada, dependendo dos dispositivos, podemos ir de 50 cm até metros de desvio. No entanto, para o efeito deste caso de estudo revelou-se uma solução viável.

A aplicação de controlo está desenhada para ser de fácil interação. A lógica da mesma é disponibilizar um conjunto de opções que permitam ao utilizador ter uma abstração de tudo o que acontece no sistema. Os botões de *Start*, *Stop* e *Lig./Desl.* só estão disponíveis após a conexão ser estabelecida, sendo que para isso, basta premir o botão *Connect*. Após a conexão estar estabelecida, estão disponíveis as opções de *Start* e *Stop*. O botão *Start* após ser premido envia uma mensagem para os outros dispositivos iniciando todos o processo. Esta opção disponibiliza também a opção *Lig./Desl.* que altera entre ligar e desligar o componente de lógica do robô. A qualquer momento podemos premir o botão *Stop* que irá fazer com que o robô pare instantaneamente.

### 4.4.3 Nó de processamento

O nó de processamento foi desenvolvido utilizando a linguagem de programação C++ e baseado na biblioteca *OpenCV*. Esta aplicação apresenta apenas uma janela onde podemos visualizar o resultado final de todos os algoritmos. A aplicação contém toda a lógica inerente à deslocação do robô.

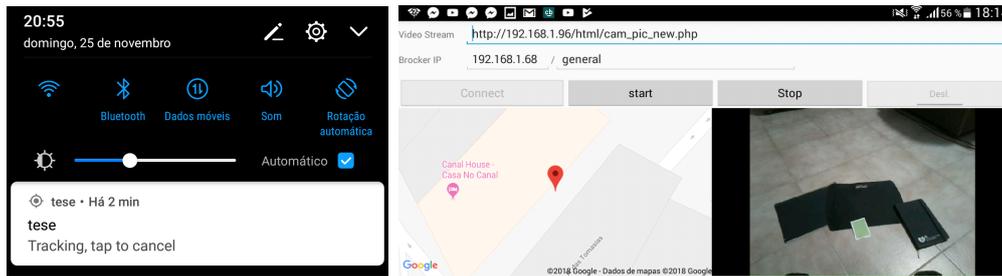


Figura 4.5: Interfaces das aplicações desenvolvidas. A imagem à esquerda é o serviço de geolocalização. A imagem à direita apresenta a interface de controlo desenvolvida para o utilizador final.

Podemos encontrar associados à aplicação dois ficheiros de configuração:

- O ficheiro *default.xml*: este permite configurar o tipo de calibração e os seus parâmetros; o URL da stream de dados; o local do ficheiro de configuração da câmara; url do servidor para os serviços de comunicação, o tópico a subscrever e o nome do cliente de comunicação.
- Ficheiro *settings.xml*: este permite configurar todos os parâmetros adjacentes aos algoritmos utilizados. Os parâmetros disponíveis são apresentados na Tabela 4.1.

#### 4.4.4 Serviço de Controlo de Cinemática

Este serviço está implementado em C++ e é executado na unidade lógica da plataforma (Raspberry Pi III). Contem toda a lógica necessária à cinemática do robô. Integra também um cliente MQTT que subscreve um canal no broker onde recebe as instruções de movimento. Este serviço no nosso caso arranca com o sistema operativo para que esteja sempre disponível. O mesmo não dispõe de interface gráfica.

Junto do serviço encontra-se um ficheiro de configuração (*config.conf*) com os valores necessários para que a aplicação arranque de forma automática. Os parâmetros são descritos na Tabela 4.2.

Parâmetro	Tipo de Dados	Valor por Defeito	Descrição
<i>bins</i>	Inteiro	30	Valor que define o número de divisões a utilizar nos histogramas calculados.
<i>overlay</i>	Inteiro	4	Valor de acumulação máximo, este é utilizado no grau de certeza de cada bloco na grelha de expansão.
<i>min – overlay</i>	Inteiro	2	Valor mínimo para aceitar um bloco como semelhante.
<i>canny – min</i>	Inteiro	100	Valor mínimo de threshold utilizado no algoritmo de Canny.
<i>canny – max</i>	Inteiro	500	Valor máximo de threshold utilizado no algoritmo de Canny.
<i>hist – val</i>	Decimal	0.5	Valor de diferença aceitável na comparação entre histogramas.
<i>textur – i</i>	Decimal	0.3	Valor de diferença aceitável na comparação entre texturas utilizado no algoritmo de Canny.
<i>textur – lbp</i>	Decimal	0.4	Valor de diferença aceitável na comparação entre os resultados do LBP.
<i>colunas</i>	Inteiro	15	Número de colunas da grelha de expansão.
<i>linhas</i>	Inteiro	15	Número de linhas da grelha de expansão.

Tabela 4.1: Tabela de parâmetros configuráveis na unidade lógica.

Parâmetro	Tipo de Dados	Valor por Defeito	Descrição
<i>devname</i>	String	/dev/ttyUSB0	Caminho para o dispositivo de comunicação série.
<i>mqttHost</i>	String	tcp://localhost:1883	Endereço e porta para o broker de comunicação.
<i>topic</i>	String	general	Tópico a inscrever no broker.
<i>clientID</i>	String	Server	Nome do cliente MQTT.

Tabela 4.2: Tabela de parâmetros configuráveis no serviço de cinemática.

# Capítulo 5

## Conclusões

Este trabalho apresenta uma aproximação a um problema atual e que deve ser solucionado. Estamos a falar do uso excessivo de herbicidas por parte das entidades reguladoras e responsáveis pela manutenção das vias públicas e espaços públicos. Sabemos agora também que este não é apenas um problema nacional. Muitas são as organizações governamentais e não-governamentais que estão preocupadas com a utilização excessiva destes produtos. O glifosato é apenas um químico de entre muitos que poderíamos abordar aqui. A verdade é que é o mais comercializado e por isso atrai mais atenção.

Os desafios a enfrentar num problema desta dimensão requerem sem dúvida a reunião de uma equipa multidisciplinar que permita resolver de forma eficiente todos os aspetos envolventes. De uma forma geral pensamos que o trabalho expõe, numa primeira instância, os desafios adjacentes à construção de uma plataforma que possa ser utilizada em ambientes exteriores, nomeadamente no que se refere à construção do robô e os obstáculos esperados ao trabalhar apenas com uma única fonte de dados de baixo custo.

Foram construídas as bases para um trabalho de continuação. Os resultados apresentados estão longe de estar concluídos ao ponto de se ter obtido um produto final e passível de ser comercializado. No entanto, foi testada uma nova abordagem de deteção de pavimentos que resultou numa interessante base de trabalho a ser mais aprofundada. Desta forma, é possível melhorar os algoritmos e introduzir novas metodologias ou algoritmos, a fim de extrair mais informação do mundo envolvente ao robô e melhorar os resultados obtidos. O algoritmo apresentado falha em algumas situações, nomeadamente quando existem sombras no ambiente de trabalho, sendo algo a melhorar no trabalho futuro.

Outro aspeto a considerar é a integração de mais sensores que permitam melhorar a

descrição do ambiente envolvente. Uma possibilidade seria ter mais uma fonte de dados que permitisse obter valores de profundidade, por exemplo.

A utilização de algoritmos de aprendizagem poderia ajudar também no reconhecimento de padrões, obstáculos e deteção de ervas. Esta é uma área em constante expansão e que apresenta resultados e técnicas inovadoras diariamente.

O facto de o sistema ter sido desenvolvido sobre a lógica de produtor consumidor, permite no futuro ter uma central de controlo física que permita controlar um conjunto de robôs espalhados numa cidade e desta forma possibilitar ter o controlo centralizado das máquinas por um único operador.

Em jeito de conclusão podemos dizer que o trabalho desenvolvido é um bom ponto de partida para trabalhos futuros nesta área ou problemas semelhantes. Todas as tecnologias são de código aberto e de livre utilização para mais facilmente permitir novos avanços.

# Bibliografia

- [1] Moura Francisco Amado Santos António Manuel Esteves, Henriques António A. Casal. *Manual the Portuguese Pavements*. Comedil – Comunicação e Edição, Lda. Rua Enfermeiras da Grande Guerra, 14-A 1170-119 Lisboa Telf: 218 123 753 – Fax: 218 141 900, 2009.
- [2] PÚBLICO Comunicação Social SA. Monsanto condenada a pagar indemnização milionária por causa de cancro causado por glifosato | EUA | PÚBLICO, 2018.
- [3] Glifosato. Monsanto recorre de condenação a pagamento a doente oncológico - Mundo - RTP Notícias, 2018.
- [4] Observador. Farmacêutica Bayer garante que glifosato é “seguro e não cancerígeno” – Observador, 2018.
- [5] Parlamento Europeu e o Conselho da União Europeia União. Directiva 2009/128/CE do Parlamento Europeu e do Conselho. *Jornal Oficial da União Europeia*, 309:71–86, 2009.
- [6] Ana Paula Félix and Miriam Cavaco. Manual de protecção fitossanitária para protecção integrada e agricultura biológica da vinha. *Direcção-Geral de Agricultura e Desenvolvimento Rural*, pages 1–126, 2009.
- [7] Sandra Salvado. União Europeia prolonga por mais cinco anos licença de utilização do glifosato - Mundo - RTP Notícias, 2017.
- [8] Kathryn Z. Guyton, Dana Loomis, Yann Grosse, Fatiha El Ghissassi, Lamia Benbrahim-Tallaa, Neela Guha, Chiara Scoccianti, Heidi Mattock, Kurt Straif, A. Blair, L. Fritschi, J. McLaughlin, C. M. Sergi, G. M. Calaf, F. Le Curieux, I. Baldi, F. Forastiere, H. Kromhout, A. 't Mannetje, T. Rodriguez, P. Egeghy, G. D. Jahnke,

- C. W. Jameson, M. T. Martin, M. K. Ross, I. Rusyn, and L. Zeise. Carcinogenicity of tetrachlorvinphos, parathion, malathion, diazinon, and glyphosate. *The Lancet Oncology*, 16(5):490–491, 2015.
- [9] Ana Maia. Pesticida perigoso usado por 89 câmaras - Diário de Notícias - Global Media Group, 2016.
- [10] Wissem Mnif, Aziza Ibn Hadj Hassine, Aicha Bouaziz, Aghleb Bartegi, Olivier Thomas, and Benoit Roig. Effect of endocrine disruptor pesticides: A review. *International Journal of Environmental Research and Public Health*, 8(6):2265–2303, 2011.
- [11] Franklin Robotics. Tertill: The solar powered weeding robot for home gardens by Franklin Robotics — Kickstarter, 2018.
- [12] Peter Biber, Ulrich Weiss, Michael Dorna, and Amos Albert. Navigation System of the Autonomous Agricultural Robot BoniRob. *Workshop on Agricultural Robotics: Enabling Safe, Efficient, and Affordable Robots for Food Production (Collocated with IROS 2012)*, pages 1–7, 2010.
- [13] W Bangert, A Kielhorn, F Rahe, A Albert, P Biber, S Grzonka, S Haug, A Michaels, D Mentrup, M Hänsel, D Kinski, K Möller, A Ruckelshausen, G Scholz, F Sellmann, W Strothmann, and Trautz. D. Field-Robot-Based Agriculture:“RemoteFarming.1” and “BoniRob-Apps”. *71th Conference AgEng Proceedings*, pages 2–9, 2013.
- [14] Jean-François Lalonde, Alexei A Efros, and Srinivasa G Narasimhan. Detecting Ground Shadows in Outdoor Consumer Photographs. *ECCV - School of Computer Science, Carnegie Mellon University*, pages 322–335, 2010.
- [15] Boyoon Jung and Gaurav S Sukhatme. Detecting Moving Objects using a Single Camera on a Mobile Robot in an Outdoor Environment. *International Conference on Intelligent Autonomous Systems*, pages 980–987, 2004.
- [16] D Castells, J M F Rodrigues, and J M H Du Buf. Obstacle detection and avoidance on sidewalks. *VISAPP 2010 - Proceedings of the International Conference on Computer Vision Theory and Applications*, 2:235–240, 2010.

- [17] Dieter Fox. KLD-Sampling : Adaptive Particle Filters. *Proceedings of the Sixteenth Annual Conference on Neural Information Processing Systems, Vancouver and Whistler, British Columbia, Canada*, pages 713–720, 2002.
- [18] Tim Dalglish, J. Mark G.. Williams, Ann-Marie J. Golden, Nicola Perkins, Lisa Feldman Barrett, Phillip J. Barnard, Cecilia Au Yeung, Victoria Murphy, Rachael Elward, Kate Tchanturia, and Edward Watkins. Sidewalk Direction Recognition and Multiple Color Sidewalk Area Detection. *Journal of Experimental Psychology: General*, 136(1):23–42, 2007.
- [19] Won Suk Lee, David C. Slaughter, and D. Ken Giles. Development of a machine vision system for weed control using precision chemical application. *International Conference on agricultural machinery engineering*, 3:802–811, 2001.
- [20] Siegwart, Roland and Illah R Nourbakhsh. *Introduction to Autonomous Mobile Robots*. 2004.
- [21] Tae Jae Lee, Dong Hoon Yi, and Dong Il Dan Cho. A monocular vision sensor-based obstacle detection algorithm for autonomous robots. *Sensors (Switzerland)*, 16(3):1–19, 2016.
- [22] Open Source Computer Vision. OpenCV: Camera calibration With OpenCV.
- [23] Shengxiao Niu, Jingjing Yang, Sheng Wang, and Gengsheng Chen. Improvement and parallel implementation of canny edge detection algorithm based on GPU. *Proceedings of International Conference on ASIC*, (6):641–644, 2011.
- [24] Simone Roberts, Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. Computer Vision Using Local Binary Patterns. In *Pharmacy News*, number 2/AUG., chapter 2, page 1. Springer-Verlag London, 2011 edition, 2007.
- [25] Len. Bass, Paul. Clements, and Rick. Kazman. *Software Architecture in Practice*. 2003.
- [26] Andrew Banks and Rahul Gupta. MQTT Version 3.1.1, 2014.
- [27] RPi-Cam-Web-Interface. RPi-Cam-Web-Interface - eLinux.org.

