Diogo Samuel Bento Amorim

Seguimento de um Humano por um Robô Companheiro

julho 2016

· U C ·

Universidade de Coimbra

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Seguimento de um Humano por um Robô Companheiro

Diogo Samuel Bento Amorim

Coimbra, julho 2016

Seguimento de um Humano por um Robô Companheiro

Orientador:

Prof. Doutor Rui Paulo Pinto da Rocha

Júri:

Prof. Doutor Jorge Manuel Moreira de Campos Pereira Batista (Presidente)

Prof. Doutor Manuel Fernando dos Santos Silva

Prof. Doutor Rui Paulo Pinto da Rocha

Dissertação submetida em realização do Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Coimbra, julho 2016

Agradecimentos

O trabalho, a dedicação, o esforço e a vontade de ir mais além são ingredientes essenciais no desenvolvimento de um projeto como este e no percurso académico percorrido que tanto exigiu em todos os parâmetros da vida pessoal.

A capacidade de desenvolver trabalho individualmente é uma arma poderosa, a maioria do trabalho deve ser desenvolvido nestes pressupostos ou não fosse esta tese de dissertação o culminar de um percurso académico individual.

No entanto, num processo de desenvolvimento e trabalho deste âmbito, as pessoas que nos rodeiam e que interagem connosco durante este processo exercem uma influência tal que se tornam elementos preponderantes e decisivos no desenvolvimento do trabalho.

Começo por agradecer ao meu orientador Prof. Rui Paulo Rocha que, com a sua experiência e sabedoria, soube aconselhar, encorajar e exercer pressão de forma a que eu não perdesse o rumo ao longo de um projeto tão extenso e demorado. A forma como interviu foi decisiva e essencial para o sucesso deste trabalho. Olhando para todo o percurso, percebo como este trabalho poderia ter ficado incompleto e muito aquém das expetativas se não fosse o seu encorajamento e sem a sua visão global de onde queria e acreditava que eu poderia chegar. Deste modo, deixo o meu maior agradecimento e apreço por me ter acolhido, orientado e disponibilizado incontáveis oportunidades para me ajudar.

Para os meus colegas de laboratório que sempre se mostraram disponíveis para intervir, fornecer a sua opinião e ajudar no que lhes fosse possível, demonstro respeito e agradecimento pela postura demonstrada ao longo da minha presença no laboratório. Apresento um especial agradecimento ao Gonçalo Martins que, em momentos chave, sempre se mostrou disponível para ajudar, demonstrando um largo conhecimento sobre o assunto e sempre muito assertivo.

Deixo também uma referência a vários colegas e companheiros desta jornada de cinco anos e que se encontram no desenvolvimento do seu trabalho de dissertação, um agradecimento por sempre demonstrarem interesse, companheirismo e oferecerem sempre o seu suporte.

Devo dar também uma palavra de apreço a um amigo de muitos anos, Jorge Coelho que se mostrou disponível para realizar um revisão ortográfica ao documento e que se revelou muito efetiva.

Por fim, não posso deixar de mencionar o suporte a todos os níveis dos meus pais, que sempre acreditaram em mim, investiram em mim e que provavelmente são as pessoas que mais se vão orgulhar por ter alcançado este objetivo.

Resumo

Cada vez mais surgem robôs a partilhar o mesmo espaço físico que os humanos. Deste modo é importante pensar em como os robôs interagem com um humano e na partilha do mesmo espaço que eles. A forma como se movem e reagem passa a ter cada vez mais relevância e é uma componente cada vez mais presente em sistemas robóticos móveis. Aspetos relacionados com a movimentação do robô que não estão diretamente relacionados com comandos rígidos podem ter um impacto muito significativo na forma como a pessoa percepciona a presença do robô no mesmo espaço (*e.g.* agressividades, suavidade ou hesitação).

No desenvolvimento deste trabalho, alguns destes aspetos são tidos na programação do robô móvel para estabelecer uma relação adequada entre a perceção do humano pelo robô e a navegação (atuação) do robô no mesmo espaço físico que o humano.

Infelizmente, a maioria das aplicações desenvolvidas não se focam na forma como o robô atua em termos de comportamento e forma de estar em espaços físicos habitualmente ocupados pelos humanos. Deste modo não é obtido um bom resultado de interação entre o robô e o humano.

É precisamente com a noção apresentada que surge a vontade de trabalhar neste domínio e de realizar alguns progressos nesta área. Para isso foi desenvolvido um sistema de classificação de atividades humanas por forma a dotar o robô móvel de um nível mais avançado de decisão, baseado nas atividades que são reconhecidas da pessoa a ser seguida.

Assim, neste trabalho, é apresentado um sistema robótico móvel implementado em ROS (*Robot Operating System*) que integra uma capacidade de percepção, baseada num sistema de deteção do esqueleto humano e na classificação das suas atividades. Esta informação é integrada numa camada de software que trata de aspetos comportamentais do robô, resultando num sistema de navegação desenvolvido tendo em consideração os requisitos específicos da aplicação pretendida: seguimento de uma pessoa. O robô é composto por alguns componentes que capacitam o sistema de realizar as particularidades mencionadas.

Por fim, o sistema implementado no robô móvel sofreu um processo de validação experimental que sustenta todos os resultados apresentados em diversas situações. Primeiramente as componentes individuais do sistema foram alvo de uma análise individual de forma a validar as componentes de forma separada, tornando-as peças de um sistema maior que, por sua vez, podem ser reutilizadas em outros projetos de interação humano-robô. Uma análise mais

intensiva foi realizada ao robô seguidor em plena integração das suas funções em operação, corroborando assim a sua funcionalidade.

Abstract

The number of robots that share the same human's physical space is increasingly arising, thus is important think about the way that robots interact with humans and how they share the same space. The way how robots are, move and react became a very relevant aspect and it is an element even more present in mobile robotic systems. Issues related to robot's movement which are not directly associated with rigid commands, may have a significant impact on how a person perceives the robot's presence in the same space (*e.g.* aggressiveness, softness or hesitation).

To perform this work, some of these aspects were taken into consideration in mobile robot programming to establish an appropriate relationship between the perception of the human by the robot and navigation (operation) of the robot in the same physical space as the human.

Unfortunately, most of the developed applications do not focus on how the robot operates in accordance with the behavior and the way to be in physical spaces usually occupied by humans. Thus, it is not obtained good interaction result between the robot and the human.

It is precisely with the notion presented that comes the desire to work in this field and make some progress. For this, it was developed a human activity classification system in order to give it a more advanced decision level based on activities that are recognized of the person to be followed.

Thus, this work presents a mobile robot system implemented in ROS (*Robot Operating System*) and integrates a capacity of perception which is based on tracking systems of human's skeleton and classification of their activities, integrating this information into a software layer that deals with behavioral aspects of the robot. This results in a developed naviagtion system taking into account the specific requirements of the intended application: tracking a person. The robot is composed by a few components that enable the system to perform the aforementioned characteristics.

Finally, the system implemented in mobile robot underwent an experimental validation process that supports all the results presented in different situations. First, system individual components were the target of an individual analysis in order to validate the components separately, making the parts of a larger system, which in turn can be reused to other projects of human-robot interaction. A more intensive analysis was performed to the follower robot

with full integration of its functions in operation, thus confirming its functionality.

"Computers are like Old Testament gods, lots of rules and no mercy."

— Joseph Campbell, *The Power of Myth*

Conteúdo

Agradecimentos **ii**

Resumo **iv**

Abstract **vi**

Lista de Acrónimos **xii**

Lista de Figuras **xiv**

Lista de Tabelas **xv**

1 Introdução **1**

1.1 Motivação e Contexto 1

1.2 Objetivos 3

1.3 Estrutura da Dissertação 4

2 Fundamentos e Estado da Arte **7**

2.1 Comportamento de um Robô Seguidor 7

2.2 Aprendizagem Automática Supervisionada 9

2.2.1 Sistemas e Algoritmos de Classificação 11

2.3 Sumário 16

3 Seguimento de Pessoas por um Robô **17**

3.1 Robótica de Serviços 17

3.2 Interação Humano-Robô 18

3.3 Estimação da Pose do Esqueleto Humano 18

3.3.1 Microsoft Kinect 19

3.3.2 Sensor de Profundidade 19

3.3.3 *PrimeSense, OpenNI e NITE* 20

3.3.4 Pacote *OpenNI Tracker* 20

3.4 Extração de Características 3D 21

3.5 Considerações Sobre o Sensor Kinect 25

3.5.1 Campo de Visão 25

3.5.2 Capacidade de Hardware . . . 26
3.5.3 Ambiente . . . 26
3.6 Distância Pessoal . . . 27
3.7 Sumário . . . 28

4 Sistema Proposto **29**
4.1 Opções de Projeto e Implementação . . . 29
4.2 Recursos de *Hardware* . . . 30
4.3 Visão Global do Sistema . . . 32
4.4 Sensorização . . . 33
4.5 Classificação . . . 34
4.6 Navegação . . . 36
4.7 Sumário . . . 40

5 Validação Experimental e Análise de Resultados **41**
5.1 Testes Comportamentais . . . 41
5.2 Desempenho do Sistema de Classificação com o Sensor Estático . . . 42
5.3 Desempenho do Sistema Integrado . . . 44
5.3.1 Seguimento de um Humano pelo Robô . . . 45
5.3.2 Desempenho do Sistema de Classificação com o Sensor em Movimento 47
5.4 Sumário . . . 50

6 Conclusão **51**
6.1 Trabalho Futuro . . . 52

7 Bibliografia **53**

A Anexos **59**

B Guia de Procedimentos **61**

Lista de Acrónimos

BSFS	Binocular Sparse Feature Segmentation
CPU	Central Processing Unit
EKF	Extended Kalman Filter
FN	False Positive
FP	False Negative
HRI	Human Robotic Interactions
IR	InfraRed
LRF	Laser Range Finder
LSH	Locality Sensitive Hashing
PSDK	Prime Sense Development Kit

RAM	Random Access Memory
RANSAC	Random Sample Consensus
RGB	Red Green Blue
ROC	Receiver Operating Characteristic
ROS	Robot Operating System
SDK	Software Development Kit
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous localization and mapping
SVC	Support Vector Classification
SVM	Support Vector Machine
TN	True Positive
TP	True Negative
tf	Transform

Lista de Figuras

2.1 Demonstração da alocação de dados do algoritmo SVM (imagem retirada de [41]). 15

3.1 Sistema de cognição de um robô (informação extraída de [2]). 17
3.2 15 juntas do modelo do esqueleto humano disponibilizado pelo pacote **OpenNI Tracker** (imagem retirada de [6]). 20
3.3 Relação dos diferentes espaços de coordenadas. 24
3.4 Campo de visão global da Microsoft Kinect, imagens retiradas de [1]. 26
3.5 Visualização de todas as zonas estabelecidas pela investigação de Edward T. Hall (1966) [39], imagem extraída de [5]. 27

4.1 Dispositivos de *hardware* utilizados neste trabalho. 31
4.2 Arquitectura de software 32
4.3 Módulo de perceção. 33
4.4 Módulo de classificação. 35
4.5 Conjunto de atividades propostas no sistema de classificação. 36
4.6 Representação vetorial da incidência dos dados de sensorização obtidos na navegação desenvolvida. 38
4.7 Módulo de Navegação. 40

5.1 Testes realizados a aspetos comportamentais do robô. 42
5.2 Resultados do sistema de classificação representados em matrizes de confusão. 43
5.3 Ilustração das experiências realizadas com o sistema integrado. 46
5.4 Imagem da entrada do ISR com um dia de sol intenso. 47
5.5 Imagem de um corredor do ISR com um pórtico de iluminação natural. . . . 47
5.6 Matrizes de confusão do sistema de classificação quando opera no sistema integrado para seguimento de um humano por um robô. 48

A.1 Microsoft Kinect, imagem extraída de [4]. 59
A.2 Hardware da Microsoft Kinect, imagem extraída de [4]. 60
A.3 Resumo dos elementos constituintes da plataforma. 60

Lista de Tabelas

2.1 Alguns dos principais trabalhos desenvolvidos com o tema Robô Seguidor. . 9
2.2 Preço de monitores em função do tamanho 10
2.3 Preço de monitores em função de um maior número de características. 11

5.1 Análise individual das classes constituintes do sistema de classificação. 44
5.2 Análise individual das classes constituintes do sistema de classificação sujeito a novas condições. 49

1 Introdução

Os desenvolvimentos na área da Robótica que se verificaram nas últimas décadas contribuíram com muitas tecnologias que hoje estão presentes em diversas aplicações relevantes para a sociedade. A Robótica tem hoje em dia uma forte presença na indústria, mas também já está presente em aplicações militares, proteção civil (e.g. busca e salvamento), na cirurgia minimamente invasiva, no entretenimento e em serviços (*e.g.* robôs guia, robôs tarefeiros, hospitais e escritórios, etc.). Prevê-se que nos próximos anos a robótica de serviços alcançará uma forte penetração no mercado, tornando os robôs móveis objetos de consumo massificados e cada vez mais presentes e ubíquos no quotidiano das pessoas, incluindo ambientes domésticos. Estes factos despertam o interesse em ir sempre mais além.

1.1 Motivação e Contexto

No que se refere ao tema deste trabalho, apesar de já existirem desenvolvimentos tecnológicos relevantes, ainda não foram exploradas todas as potencialidades que este demonstra ter.

O desenvolvimento de serviços robóticos capazes de assistir humanos nas mais diversas tarefas do dia-a-dia passou a ser foco de muito interesse e, portanto, de desenvolvimento.

Possíveis aplicações deste tipo de robôs são facilmente identificáveis, como por exemplo numa missão de busca e salvamento realizada num edifício, em que o robô tenha por missão assistir o agente humano na exploração do ambiente e localização de dados relevantes (e.g. vítimas, focos de incêndio, etc.). A capacidade de seguir o humano, aliada à capacidade de exploração autónoma é fundamental no início da missão para o robô acompanhar o humano desde uma estação base até ao local em que tem início a tarefa de exploração colaborativa, dentro do teatro de operações.

A utilidade de um robô com estas características num ambiente doméstico surge quando se identificam necessidades especiais. Um exemplo da sua utilidade é no auxílio de pessoas que apresentem algum tipo de debilidade (e.g. pessoas idosas, pessoas com deficiência motora ou psicológica, doentes, etc). O robô pode prestar um serviço de assistência pessoal (e.g. para poder monitorizar permanentemente o seu estado de saúde, para efetuar lembretes, para transportar objetos de um local para o outro da casa, etc.), melhorando assim a qualidade de vida destes utilizadores.

Aplicações não domésticas, tais como serviços públicos (e.g. enfermeira assistente) e

industriais (e.g. transporte de cargas em armazém) são mais alguns exemplos das possíveis aplicações de um robô com estas características.

Aplicações com naturezas muito variadas também já são parte integrante de diferentes estruturas da sociedade, como a assistência a idosos em ambientes domésticos, utilização de robôs em serviços públicos, industriais/privados e mesmo para aplicações militares (e.g. serviços hospitalares, transporte de cargas autónomas ou guiadas por um utilizador, etc.). Portanto, são inúmeras as aplicações que se podem identificar, surgindo assim a motivação para trabalhar neste tipo de robôs em específico.

A ideia central que se pretende desenvolver nesta dissertação é programar um robô móvel para localizar espacialmente/temporalmente um utilizador humano e segui-lo durante a sua locomoção, a uma distância socialmente aceitável, confortável para o utilizador, de forma robusta e segura em relação a possíveis obstáculos estáticos e dinâmicos com que o robô se possa deparar.

Navegação e perceção são duas partes fundamentais e interrelacionadas no desenvolvimento de qualquer robô móvel com o objetivo de realizar tarefas em ambientes dinâmicos e em estreita interação com pessoas.

"Siga em frente cerca de um quilómetro e vire à esquerda na rua Oliveira do Carmo... depois de duas ou três casas, penso eu... vire à direita, irá ver um parque com um repuxo, a loja que procura tem uma faixa verde na sua frente e encontra-se ao lado da esplanada de um café... é muito fácil de chegar lá!"

Serão necessários grandes avanços em termos tecnológicos para que seja possível um robô chegar ao destino com dados ou informações como as que existem nesta descrição em linguagem natural, que contém um determinado grau de incerteza e ambiguidade.

A principal diferença entre navegação robótica e navegação humana está nas capacidades de perceção e interpretação de dados/informações. Os humanos podem detetar, classificar e identificar ambientes sob forte variação das condições que os rodeiam, independentemente da orientação ou distância a que se encontrem. Atualmente, robôs estão habilitados a detetar obstáculos estacionários e dinâmicos, no entanto têm uma capacidade de perceção e de decisão muito limitada, em nada comparáveis à dos humanos. Muitos estudos e investigações foram desenvolvidos e muito tem sido feito por forma a melhorar o desempenho da navegação robótica, nomeadamente elementos fundamentais como perceção e decisão. Têm emergido nos últimos anos novos conceitos, técnicas e abordagens que tentam elevar para outro patamar os avanços efetuados neste campo. Alguns exemplos podem ser vistos na secção 2, onde é apresentado o estado da arte das técnicas relacionadas com o tema desta dissertação.

Uma pessoa reconhecer um semelhante no ambiente que o rodeia representa uma simples tarefa para nós humanos, no entanto esta tarefa é extremamente complexa quando se pretende que seja uma máquina a realizá-la. A estimação da pose de um humano tem sido um problema que ainda nos dias de hoje apresenta dificuldades na área de visão por computador. A captura de movimento e reconhecimento de atividades humanas são exemplos

óbvios. Surgem ainda questões complexas devido ao elevado número de dimensões no espaço de procura, elevado número de graus de liberdade, restrições envolvidas - tais como não permitir penetração das partes do corpo entre si e não permitir posições impossíveis para uma pessoa. Outros desafios incluem a variação desordenada do ambiente, nomeadamente questões de iluminação, tais como a sua variação e sua origem, e por fim parâmetros relativos à forma corporal de uma pessoa.

A robótica é uma área em desenvolvimento, que já deu provas do seu potencial. Embora muito já tenha sido desenvolvido, o futuro potencial deste tema apresenta indícios animadores, pois está a um passo de explodir a nível aplicacional. A indústria foi o setor que mais tirou proveito da robótica, mas nas restantes áreas ainda está muito longe de atingir o seu elevado potencial.

Mais objetivamente, a utilização de robôs seguidores pode ter inúmeras aplicações, o que podia facilitar a vida de muitas pessoas. Imaginemos uma situação específica de uma pessoa que sofre de insuficiência respiratória: esta pessoa tem a necessidade de deslocar consigo garrafas de oxigénio. Facilmente se pode imaginar as limitações desta pessoa e o que um robô assistente poderia fazer por ela. Este caso em particular é um exemplo entre muitos que se podem encontrar e que fundamenta a necessidade do desenvolvimento desta área. É desta perspetiva que surge a motivação de desenvolver trabalho nesta área.

1.2 Objetivos

A proposta principal desta dissertação passa por estudar o problema de seguimento de um humano por um robô companheiro e implementar uma solução viável, robusta e concretizável. Para isso será implementado um algoritmo com o objetivo de seguir um humano e que obtenha a localização da pessoa a seguir no sistema de coordenadas do robô. Desta forma permite realizar uma perseguição apresentando uma interação com o humano adequada, com um comportamento socialmente aceitável e mantendo uma distância de segurança em função da localização da pessoa, transmitindo assim uma sensação de conforto, aceitação e segurança na presença do robô.

Para o desenvolvimento deste projeto será realizada a implementação de um sistema em ROS que integra diferentes módulos com as diferentes funções necessárias ao correto funcionamento do robô seguidor. Dois elemento preponderantes estão na base de desenvolvimento, o primeiro é módulo de perceção capaz de fazer o reconhecimento de um humano, com este módulo será possível detetar, seguir, localizar o esqueleto humano e classificar suas actividade com a Microsoft Kinect, assim como também um sensor LRF que trará informação para o sistema sobre o ambiente que o rodeia. O segundo módulo é o de Navegação, este módulo cuidará de toda a ação do robô tendo em consideração múltiplos aspectos comportamentais.

O sistema apresentado assenta no desenvolvimento de três estágios fundamentais que constituem os principais objetivos deste trabalho de dissertação:

- Perceção

Implementação de um algoritmo de deteção, seguimento e localização de um humano em relação ao robô, fazendo uso da Microsoft Kinect.

Reconhecimento de atividades humanas através de um sistema de classificação.

- Navegação

  Implementação de técnicas de navegação já conhecidas.

  Implementação de um nó de navegação que satisfaça as necessidades específicas da aplicação.

- Integração

  Desenvolvimento de um sistema robótico móvel que combina perceção e ação de forma a alcançar um comportamento de seguimento de um humano robusto e natural, que mantém uma distância socialmente aceitável entre um humano e o robô, baseado na teoria *proxemics* de Hall, 1966 [39].

Desta forma, espera-se obter algumas contribuições com o desenvolvimento deste trabalho. A implementação de um robô móvel capaz de realizar um serviço de seguimento de um humano é uma das principais metas a atingir. Com a finalidade de melhorar a qualidade da informação obtida e aumentar o nível de perceção do humano/utilizador, um sistema de reconhecimento de atividades humanas será adicionado ao robô, dotando-o de uma maior capacidade de perceção.

Com a combinação e utilização de todos os recursos estabelecidos anteriormente, a contribuição central deste trabalho visa a criação de um sistema robótico móvel funcional que pode ser futuramente usado como base para mais complexos estudos em interação humano-robô (HRI) e futuras investigações em HRI e equipas de robôs que explorem os recursos de HRI.

1.3 Estrutura da Dissertação

Este documento de dissertação é organizado de forma a introduzir o leitor no tema. No Capítulo 2, o leitor será contextualizado no que já foi feito no passado por outros autores e com os principais fundamentos essenciais para a compreensão deste trabalho.

De seguida, no Capítulo 3, é abordado um conjunto de tópicos relacionados com o tema robô seguidor, que fazem a ponte entre os diferentes e importantes assuntos que influenciaram e foram tidos em consideração no desenvolvimento deste trabalho. Este capítulo insere o utilizador já num nível mais profundo de conhecimento sobre o tema aqui discutido e sob a abordagem seguida no desenvolvimento deste trabalho. Desta forma, no Capítulo 4 é dado a conhecer em detalhe a forma como foi implementado este projeto.

No Capítulo 5 encontram-se conclusões, fundamentadas em resultados obtidos através de testes e experiências levadas a cabo a fim de obter um indicador de performance e eficiência do sistema.

Por fim, no Capítulo 6 são retiradas conclusões finais assim como é realizada uma reflexão sobre o trabalho futuro que pode vir a ser realizado com base no que aqui foi desenvolvido.

2 Fundamentos e Estado da Arte

Nesta secção são abordados os fundamentos científicos que estão na base do desenvolvimento deste trabalho de dissertação, de forma a que o leitor possa entender os temas presentes sem necessidade de consulta externa, auxiliando e conduzindo-o numa visita a este tema de uma forma agradável e sem dificuldades na sua compreensão.

No sentido de contextualizar o leitor relativamente ao trabalho já desenvolvido na área, é dada uma visão global do estado da arte. Deste modo, pretende-se apresentar o tema de forma genérica e os principais avanços científicos já obtidos no seu âmbito.

2.1 Comportamento de um Robô Seguidor

O que esperamos de um robô que utiliza o mesmo espaço que um humano, que tem alguma interação com um humano e que, de alguma forma, realiza uma tarefa para um humano, ou em função deste?

Na área científica de robótica, à medida que avanços tecnológicos são realizados, mais robôs vão sendo construídos com múltiplas finalidades, destinados a espaços domésticos, públicos e industriais. Estes robôs prestam, na sua generalidade, algum tipo de serviço. Deste modo, as pessoas que coabitam com este tipo de robôs tendem a dar forma ou características humanas quando contactam com eles frequentemente. As pessoas ficam especialmente sensíveis à forma como o robô se movimenta, age e se comporta no seu espaço e na interação com um humano no desempenho das suas funções.

Vamos ver como exemplo um robô seguidor de um humano, tratado neste trabalho. Ao perseguir uma pessoa, se este não se deslocar com algum cuidado, tendo em atenção características como hostilidade, a forma como é percebida a sua presença e empatia, o robô não será muito bem aceite na partilha do mesmo espaço com a pessoa.

O ponto de vista anterior pode parecer excessivo na medida em que sabemos que estamos a lidar com uma máquina, mas se o robô tiver em consideração, por exemplo, o facto de andar mais lentamente, pode apresentar um comportamento mais educado. Verificamos que este exemplo ilustra bem a importância comportamental de um robô no desempenho das suas tarefas. Sem dúvida, podemos afirmar que a forma como um robô se movimenta e a forma como se comporta perante diferentes situações vai influenciar fortemente a aceitação de um robô por um humano na partilha de um mesmo espaço fisico.

No âmbito deste tema já alguns trabalhos foram realizados, entre os quais alguns de

maior destaque. Gockley, Forlizzi e Simmons [26] aprofundaram o estudo de comportamentos sociais para robôs, permitindo que pessoas se sintam confortáveis na presença destes e compreendam as intenções deles. Satake et al. [48] elaboraram um estudo muito completo sobre estratégias para robôs sociais interagirem com pessoas e como o deverão começar a fazer. Calvo et al. [11] desenvolveram uma estratégia que se baseia na implementação de uma arquitetura de perceção-controlo.

Se o leitor tiver curiosidade em conhecer mais sobre o tema, Fong et al. [23] realizaram uma pesquisa de robôs que desempenham tarefas de interação social, abordando uma série de robôs sociais já existentes.

Técnicas de deteção de uma pessoa é um outro tópico associado ao tema Robô Seguidor que terá necessariamente de implementar alguma técnica de deteção de uma pessoa. A Tabela 2.1 apresenta um apanhado dos principais trabalhos desenvolvidos na deteção de uma pessoa nos últimos anos. Esta tabela pretende dar uma visão comparativa, referenciando o hardware utilizado, metodologias e o que efetivamente é feito.

O primeiro trabalho listado pertence a S. Shaker et al. [50]. Este trabalho faz uso de um *Laser range finder* e baseia-se na deteção das pernas da pessoa alvo. O trabalho propõe um sistema de inferência baseado em lógica difusa *fuzzy* que tem como objetivo controlar e suavizar o movimento do robô enquanto este persegue uma pessoa, considerando uma distância de segurança em relação à pessoa que está sendo seguida. Apesar de eficiente, a deteção de pernas deixa o sistema um pouco limitado pois não é referenciado o que acontece quando mais que uma pessoa está presente no campo de visão do laser.

Z.Chen e S.T.Birchfield [14] apresentaram um algoritmo chamado Binocular *Sparse Feature Segmentation* (BSFS) destinado a ser a base de visão que suporta um robô seguidor móvel. O algoritmo deteta e combina pontos característicos entre um par de imagens estéreo e uma sucessiva sequência de imagens, tornando possível a deteção de pontos 3D em tempo real.

J. Satake e J.Miura [47] exploram um método de deteção e seguimento de múltiplas pessoas para um robô móvel usando a informação de distância adquirida por câmara estéreo. É apresentada uma teoria em que o robô estima continuamente a posição e velocidade das pessoas no seu campo de visão. Apesar de muito promissora, esta abordagem peca por possuir demasiadas limitações, nomeadamente não considerar o caso de quando duas pessoas estão muito próximas, sugerindo para isso que uma nova fonte de informação fosse necessária. Por último, também o facto de não considerar qualquer tipo de obstáculos no seu ambiente, estando limitado o robô a operar apenas em instalações livres.

J.Satake et al. [46] investigaram uma técnica de identificação de pessoas para um robô móvel que tem como função seguir uma pessoa específica. Quando existe a necessidade de distinguir várias pessoas no mesmo ambiente é uma forte opção a considerar, no entanto exige alocação de muitos recursos em termos de hardware e software.

B.Ilias et al. [32] desenvolveram na sua pesquisa um robô seguidor de uma enfermeira, utilizando para isso um sistema de visão de alta velocidade. A ideia fundamental é dotar o

robô de seguir uma pessoa (*e.g.* uma enfermeira) com sucesso. O processamento é realizado em três microcontroladores para lidar com toda a informação provida por uma Microsoft Kinect. Este considera obstáculos no meio que o rodeia e evita-os. A finalidade de um sistema de visão de alta velocidade num robô móvel surge pela semelhança que este sistema de visão apresenta em relação à visão humana, pois este sistema disponibiliza uma visão 3D do ambiente que o rodeia. O sistema de visão é utilizado para identificação e seguimento contínuo de uma pessoa. A Microsoft Kinect foi a escolha mais óbvia para a realização deste trabalho devido à sua capacidade de detetar a pose de uma pessoa. Ao nível de limitações verificam-se os ambientes externos ou que contenham a presença de luz infravermelha, pois como explicado na secção 3.5 esta câmara não pode operar nesses ambientes. No entanto é apontada, pelo autor do trabalho, uma solução para esta limitação.

Autor	Hardware Utilizado	Método Aplicado	O que é Detectado na Pessoa
S.Shaker, J.J. Saade, e D. Asmar (2008)	Laser range finder	Sistema de inferência Fuzzy, Algoritmo de deteção de pernas	Pernas da pessoa alvo
Z.Chen, S.T.Birchfield (2007)	Câmara estéreo	Abordagem Lucas Kanade, Algoritmo BSFS, RANSAC, algoritmo de detecção de face	Disparidade e fluxo óptico
J. Satake e J.Miura (2009)	Câmara estéreo e rastreamento EKF	Verificador baseado em SVM	Parte superior do corpo através de imagens de profundidade
J.Satake, M. Chiba e J.Miura (2012)	Câmara estéreo, rastreamento EKF e Laser range finder	Verificador baseado em SVM, Informação de cor, características SIFT, RANSAC	Parte superior do corpo através de imagens de profundidade, características SIFT para diferenciar 2 pessoas com roupas semelhantes
B.Ilias, S.A. Abdul Shukor, A.H.Adom e M.H Razali (2014)	Kinect: sensor de alta velocidade	Processing.org software, Método de detecção do esqueleto humano	Método detecção do esqueleto humano, o humano alvo necessita de levantar as suas mãos

Tabela 2.1: Alguns dos principais trabalhos desenvolvidos com o tema Robô Seguidor.

2.2 Aprendizagem Automática Supervisionada

Primeiramente é necessário entender a diferença entre os conceitos de aprendizagem automática supervisionada [33] e não supervisionada [38], suportando assim a opção pelos algoritmos utilizados, como terá a oportunidade de ver nas secções 2.2.1.1, 2.2.1.2 e 2.2.1.3.

Em aprendizagem automática supervisionada, é fornecido um determinado conjunto de dados em que é definido o conhecimento de como se espera que seja a saída, tendo uma clara ideia da relação que existe entre a entrada e a saída.

Em aprendizagem automática não supervisionada, por outro lado, permite uma abordagem do problema com uma pequena ou mesmo sem nenhuma ideia da constituição das classes.

Um algoritmo de aprendizagem supervisionada analisa dados de treino ou de referência e produz uma formulação matemática que melhor se aproxima ao conjunto de dados providos ao algoritmo. Depois do algoritmo treinado, vai permitir distinguir corretamente uma classe com origem numa fonte desconhecida.

Descendo ao nível mais básico, podemos olhar para a aprendizagem automatizada supervisionada como uma aproximação matemática, linear, polinomial ou de outra natureza, de forma a conseguir aproximar uma equação a um conjunto de dados conhecido. Um exemplo é um algoritmo para prever o preço de monitores de computador em função do seu tamanho.

Dimensão Monitor	Preço
15”	70€
17”	90€
19”	100€
21”	120€
22”	150€
24”	220€
27”	300€

Tabela 2.2: Preço de monitores em função do tamanho

Na tabela 2.2 podem ser observados um conjunto de dados puramente fictícios, em que facilmente se identifica a propriedade de linearidade. Uma simples equação matemática linear descreve o comportamento da variação de preço em função do tamanho dos monitores. Para este problema temos então como entrada o tamanho do monitor e como saída a previsão de um preço como resultado.

No entanto, como é evidente, este exemplo é de extrema simplicidade. Imaginemos agora que pretendemos prever o preço de um monitor da mesma forma, mas agora com mais características técnicas que afetam o preço do produto, como por exemplo a resolução, contraste, tempo de resposta, n^0 de cores e luminosidade, como é possível verificar na tabela 2.3.

Com esta nova tabela verificamos que uma relação matemática ja não se pode aplicar, como no exemplo da tabela 2.1. Todas as novas características devem ser consideradas agora como novas entradas influenciando o resultado ou previsão.

Porém, este problema continua a ser simples. Suponhamos agora outra característica: a localização. Diferenciais entre países e até mesmo entre cidades, devido a múltiplos fatores, aumentam muito a complexidade do problema. Portanto, aplicar um algoritmo para resolver este problema passa a ser uma tarefa muito complicada.

Nesta fase, a utilização de técnicas de aprendizagem automática ganha força e podem ser utilizadas tanto para problemas imensamente complexos como para problemas mais simples.

Nestes problemas são reunidos um grande número de dados, geralmente denominados de dados de treino, que como o próprio nome indica, servem para treinar a máquina classificadora.

Posteriormente são utilizados um conjunto de algoritmos que se vão encarregar de analisar os dados, prevendo assim qual o resultado (saída) que melhor corresponde a um conjunto de

Dimensão Monitor	Resolução	Contraste	T. Resposta	Nº Cores	Lumin.	Preço
15”	720p	300:1	10ms	500K	$90cd/m^2$	70€
17”	720p	500:1	10ms	500K	$100cd/m^2$	90€
19”	1080p	1000:1	5ms	500K	$150cd/m^2$	100€
21”	1080p	1000:1	5ms	1M	$250cd/m^2$	120€
22”	1080p	80000000:1	2ms	1M	$250cd/m^2$	150€
24”	4k	80000000:1	2ms	16.7M	$300cd/m^2$	220€
27”	4k	80000000:1	1ms	50M	$500cd/m^2$	300€

Tabela 2.3: Preço de monitores em função de um maior número de características.

dados de treino providos (entrada) ao algoritmo - neste caso, o preço.

A utilização e criação dos dados de treino é a razão pela qual estas técnicas de aprendizagem automática se consideram supervisionadas.

2.2.1 Sistemas e Algoritmos de Classificação

A classificação é uma das aplicações da aprendizagem automática supervisionada. Classificação é o processo de utilizar algum tipo de entrada (dados de referência) e devolver um resultado que pertence a um subconjunto do conjunto disponibilizado como entrada. Um sistema de classificação é usado, geralmente, quando se pretende classificar alguma coisa (*e.g.* atividades humanas, objetos, etc.). É a distinção de uma classe das demais possíveis, ou seja, uma simples escolha do resultado mais provável, que mais se adequa a um subconjunto existente.

Um algoritmo de classificação é um procedimento para seleção de uma hipótese de um conjunto de alternativas que melhor se adequa ao conjunto de observações.

Se o leitor pretender mais informação sobre este tópico, poderá obter um conhecimento mais profundo em [27].

Classificação é um tema muito relevante e é fortemente usado em sistemas de visão por computador e de informação. Devido à sua emergente necessidade e suas possíveis aplicações é alvo de muito interesse e investigação.

Viajando dos primeiros trabalhos até aos mais recentes, Gavrila [25] propôs uma separação entre modelos 2D e 3D. Em Aggarwal e Cai [36] uma nova taxonomia foi apresentada com base na análise de movimento de uma pessoa. Com uma abordagem semelhante à anterior, Wang et al. [54] desenvolveram uma categorização hierárquica de ação. Mais tarde Moeslund et al. [37] surgiram com uma abordagem baseada na pose de um humano para reconhecer a sua atividade. Ele propôs a *fourfold taxonomy*, que se baseia na inicialização de movimento, seguimento, estimação da pose e reconhecimento de atividades.

Turaga et al. [52] trouxeram uma nova abordagem: propuseram a diferenciação entre o significado de *ação* e *atividade*, onde o reconhecimento de atividade é categorizado de acordo com o grau de complexidade da mesma. Poppe [42] caracterizou os métodos de reconhecimento de atividades em duas categorias distintas, *top-down* e *bottom-up*. Aggarwal e Ryoo [8], também aqui com a caracterização de uma nova abordagem, distinguiram o reconhecimento de atividades em duas categorias: uma abordagem *single layer* e *hierarchical*.

Anos mais tarde surgiram novas formas de realizar o reconhecimento de atividades, nomeadamente através de modelação 3D. Para isso, utilizaram-se câmaras com captação de imagens de profundidade, extraindo assim a pose das juntas do esqueleto humano. Exemplo disso é o trabalho de Chen et al. [13], Ye et al. [56] e Aggarwal e Xia [9].

Após muita investigação e muito trabalho desenvolvido em reconhecimento de atividades humanas é de salientar que na sua grande maioria os sistemas são de sequências de imagens.

No entanto, realizar esta tarefa através de imagens estáticas ou de dados recorrendo a um só instante continua a ser um desafio para a comunidade científica que aborda este tema. Guo e Lai [28] é um exemplo a consultar no que se refere a este tópico.

É possível encontrar muitos outros estudos e abordagens distintas pela comunidade científica mas, por agora, o leitor ficou com um conhecimento dos principais avanços realizados no reconhecimento de atividades de um humano.

Nas subsecções seguintes são apresentados e explicados os fundamentos e a origem dos algoritmos de classificação utilizados neste trabalho.

2.2.1.1 Naive Bayes

O teorema Bayes foi reconhecido depois de Thomas Bayes, que estudou como calcular o parâmetro da distribuição binomial de forma a que este descreva o melhor possível um conjunto de observações. Após a sua morte, o seu amigo Richard Price editou e apresentou o seu trabalho em 1763, como um ensaio para resolver um problema no *Doctrine of Chances* [17].

Os métodos baseados na formulação Naive Bayes são um conjunto de algoritmos de aprendizagem automática supervisionada com base na aplicação do teorema de Bayes, teorema este com fortes (simples) suposições de independência estatística.

Com base no conhecimento adquirido de [44], [41] e [58], dada uma classe variável y e um vetor função dependente $x_1...x_n$, o teorema de Bayes fornece a seguinte relação:

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)} \tag{2.1}$$

através da suposição de (simples) independência

$$P(x_i|y, x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) = P(x_i|y) \tag{2.2}$$

assim para qualquer i, a equação (2.1) pode ser simplificada em

$$P(y|x_1, ..., x_n) = \frac{P(y)\Pi_{i=1}^{n} P(x_i|y)}{P(x_1, ..., x_n)} \tag{2.3}$$

porque $P(x_1, ..., x_n)$ é uma dada entrada constante, pode ser usada a seguinte regra de classificação:

$$P(y|x_1, ..., x_n) \propto \prod_{n=1}^{n} P(x_i|y) \tag{2.4}$$

$$\Downarrow$$

$$\hat{y} = arg\max_{y} P(y)\prod_{n=1}^{n} P(x_i|y) \tag{2.5}$$

é possível desta forma utilizar a estimação *maximum a posteriori* (MAP) [24] para estimar $P(y)$ e $P(x_i|y)$.

A diferença entre os tipos de algoritmos de classificação baseados na teoria de Bayes é principalmente devido às suposições que são impostas na distribuição de $P(x_i|y)$.

Este tipo de algoritmos difere de outros mais sofisticados pela sua simplicidade e rapidez. Apesar da sua simplicidade, os resultados em situações reais são surpreendentemente bons.

Depois de muita investigação e trabalho desenvolvido com o mesmo, está provado [58] que é um bom classificador, no entanto não é um bom estimador e, como tal, os resultados de estimação do algoritmo - entenda-se probabilidade de acerto - não devem ser considerados fiáveis.

2.2.1.2 K-Nearest Neighbours

Existem muitos algoritmos baseados na vizinhança dos pontos, no entanto este tipo de algoritmo caracteriza-se pela necessidade de grande proximidade dos dados de referência com os dados analisados. Algoritmos que implementam de forma eficiente a ideia de comparar diretamente uma série de pontos com cada ponto não são rudimentares quando comparados com algoritmos designados de *brute force* [35].

Quando o objetivo não é obter um resultado com extrema exatidão mas sim perceber uma aproximação de um dado padrão a um já referenciado na base de dados, a probabilidade que um padrão apresenta relativamente a a outro em termos de semelhança, é suficiente para que se possa trabalhar com este tipo de resultado em muitas aplicações.

Locality Sensitive Hashing (LSH), ver [10] e [41], primeiramente proposto por P. Indyk e R. Motwani [29], é a técnica que resultou da tentativa de desenvolver um algoritmo com a variação relatada no parágrafo anterior. A ideia principal baseia-se na probabilidade de colisão, ou seja, dois objetos com uma semelhança na compilação das suas informações.

O algoritmo é então definido como se segue.

Uma família H de funções de domínio S para uma gama U é chamada de sensível (r, e, p_1, p_2), com $r,\ e > 0$, $p_1 > p_2 > 0$, se para qualquer $p,\ q \in S$ (sendo p e q quaisquer dois pontos) as seguintes condições se mantiverem (com D representando a função de distância):

- Se $D(p, q) <= r$, então $P_H[h(p) = h(q] >= p_1$
- Se $D(p, q) <= r(1 + e)$, então $P_H[h(p) = h(q] <= p_2$

Desta forma, pontos dentro da distância r irão apresentar entre si uma probabilidade p_1 bastante elevada. Por outro lado, pontos que se encontrem separados por uma distância

$r(1+e)$ terão uma probabilidade p_2 reduzida o suficiente para uma análise de diferenciação em relação a p_1. Existe assim uma família de funções LSH denominada de H onde o índice é atribuído da seguinte forma:

1. São escolhidas de H, k funções $h_1, h_2, ..., h_k$ ao acaso. Assim, para qualquer $p \in S$ é alocado p, ficando referenciado como $g(p) = (h_1(p), h_2(p), ..., h_k(p))$.

2. Independentemente é executado o primeiro passo l vezes, de forma a construir l estimadores com as funções $g_1, g_2, ..., g_l$.

São concatenadas as funções no primeiro passo com o objetivo de reduzir a probabilidade de pontos distantes quando estes coincidem, mitigando assim o mais possível a sua probabilidade e reduzindo, consequentemente, o erro deste tipo. A probabilidade de p_2 decresce, tornando-se mais insignificante quanto maior a ordem de k. A escolha de k deve ser avaliada caso a caso, com experiência, adequando o seu valor à aplicação em que é usada.

2.2.1.3 Support Vector Machine

A designação *Support Vector Machine* (SVM), também muito usualmente conhecida por *Support Vector Networks*, trata o desenvolvimento de um algoritmo destinado a tratar problemas de classificação. Em 1936, R.A. Fisher [22] desenvolveu a primeira abordagem a reconhecimento de padrões. Em 1962, Rosenblatt [45] explorou mais profundamente este tipo de algoritmos de aprendizagem automática, surgindo então os conceitos de *Perceptrons* (um modelo de computador ou máquina computorizada concebido para representar ou simular a capacidade do cérebro reconhecer e discriminar) e redes neurais, aproximando este conceito da forma como é conhecido nos dias de hoje. Até então, nenhum algoritmo minimizava o erro num conjunto de vetores necessário para ajustar o peso de cada elemento da rede.

Na década de 80 surgiram os primeiros algoritmos que tinham em consideração todos os pesos dos elementos da rede neural, realizando as adaptações necessárias de forma a minimizar o erro de um certo conjunto de vetores. Esses avanços foram desenvolvidos por Rumelhart, Hinton e Williams em 1986, 1987 [55], Parker em 1985 [40] e LeCun em 1985 [34], altura em que foi descoberto o algoritmo de propagação de retorno, *Back Propagation Algorithm*, ver [15].

Em 1995, por Corinna Cortes e Vladimir Vapnik, ver [16], alcançou-se o conceito que hoje se denomina de *Support Vector Networks*, que foi o primeiro algoritmo exclusivamente dedicado a problemas de classificação.

A ideia fundamental resume-se à construção de um hiper plano ou um conjunto de hiper planos, ou seja, um infinito espaço dimensional. Este espaço serve para alocar dados e consequentemente proceder à classificação dos mesmos. Desta forma é alcançada uma boa separação dos dados a classificar pelo hiper plano que distancia pontos de diferentes classes.

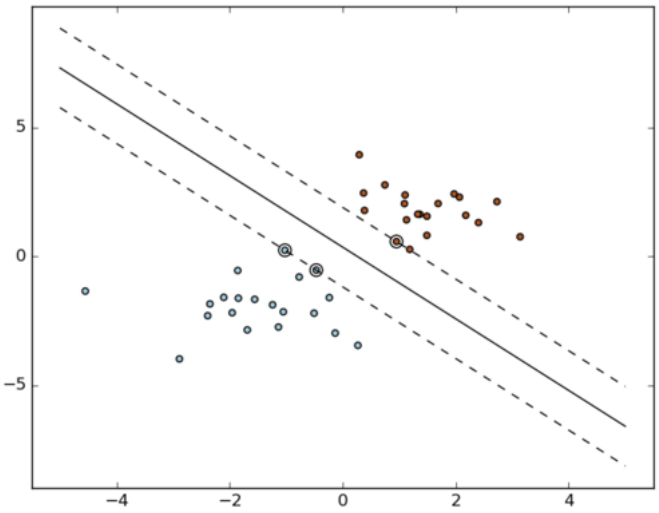


Figura 2.1: Demonstração da alocação de dados do algoritmo SVM (imagem retirada de [41]).

Assim, quanto mais largo é o distanciamento entre pontos de diferentes classes, menor é o erro de classificação, como se pode verificar na Figura 2.1. É então que surge uma sequela do algoritmo SVM que foi desenvolvida especificamente para problemas de classificação denominada de SVC. Este algoritmo é formulado da seguinte forma:

Dado um vetor de referência, $x_i \in \Re^p$, $i = 1, ..., n$, em duas classes, e um vetor $y \in \{1, -1\}^n$, SVC resolve o seguinte problema:

$$\arg\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \tag{2.6}$$

sujeito a,

$$y_i(w^T \phi(x_i) + b) \geq 0 - \zeta_i \quad , \quad \zeta_i \geq 0, \quad i = 1, ..., n$$

que é equivalente a,

$$\arg\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \tag{2.7}$$

sujeito a,

$$y^T \alpha = 0 \quad , \quad 0 \leq \alpha_i \leq \quad , \quad i = 1, ..., n$$

onde o e é um vetor de uns, $C > 0$ é o limite superior, Q é uma matriz nxn semidefinida positiva, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$ e $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ é o núcleo. Desta forma os vetores são implicitamente mapeados num espaço dimensional super elevado através da função ϕ.

A função de decisão será então:

$$sgn(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + \rho) \tag{2.8}$$

De notar que muitos modelos utilizam C como parâmetro de regulação, outros utilizam α. Desde que esteja presente o conhecimento de que $C = \frac{n}{\alpha}$, sendo n o número de amostras, nenhum problema terá na interpretação desta solução.

A formulação matemática aqui apresentada tomou como base de referência as fontes [12] e [18].

2.3 Sumário

O capítulo que o leitor acabou de ler apresentou um pouco do que já foi feito na área científica de robôs seguidores assim como também na área de sistemas de classificação.

Esta noção do trabalho e autores que já desenvolveram trabalho nesta área serve para que o leitor compreenda o ponto de situação atual em termos de desenvolvimento.

Simultaneamente foram apresentados conceitos fundamentais para a compreensão do trabalho aqui realizado.

Para contextualizar um pouco mais o leitor com o tema e com alguns dos principais tópicos de interesse na realização deste trabalho de dissertação, no capítulo seguinte serão apresentado e abordado muitos e diferentes assuntos que fazem a ligação com a implementação do sistema apresentado no Capítulo 4.

3 Seguimento de Pessoas por um Robô

Um robô que forneça um serviço de seguimento pode revelar-se um robô com muita utilidade a nível aplicacional. Disponibilizar algum tipo de serviço robótico sem necessidade de teleoperação, torna essa tarefa simples, intuitiva e principalmente cómoda devido a um eficiente seguimento de uma pessoa. Um sistema deste tipo foca necessariamente dois pontos fundamentais: seguimento de uma pessoa e navegação.

3.1 Robótica de Serviços

Os serviços inteligentes de robôs têm evoluído de uma forma acentuada com o surgimento de novas tecnologias e avanços nas mesmas. Como consequência, cada vez mais surgem robôs com a habilidade de assistir e de suportar o humano de alguma forma, ou seja, prestando um serviço.

Com o objetivo de desenvolver serviços de robótica cada vez mais sofisticados, conceitos como computação adaptativa, integração de informação num ambiente distribuído e modelação cognitiva para interação Robô-Humano são cada vez mais utilizados.

O que é um robô com capacidade cognitiva?

Figura 3.1: Sistema de cognição de um robô (informação extraída de [2]).

Trata-se da combinação de diferentes sinergias inerentes a um sistema inteligente, nomeadamente a perceção, planeamento de movimento e navegação, compreensão e aprendizagem

automática. Estes elementos constituem um sistema que se caracteriza de cognitivo, ver Figura 3.1.

Os avanços efetuados nesta área, mais especificamente a aprendizagem automática, permitem a robôs aprender autonomamente ações complexas através do ambiente em que se inserem e das pessoas que interagem com o robô. Isto tudo só é possível porque o robô é dotado da capacidade artificial de observar e aprender.

3.2 Interação Humano-Robô

Interação define-se por existir algum tipo de comunicação entre duas entidades. A área de estudo que se dedica a este assunto denomina-se por interação Humano-Robô, sigla em inglês **HRI**.

Esta área científica tem a atenção de investigadores desde à relativamente pouco tempo. A grande dificuldade de desenvolver trabalho neste campo está na multiplicidade de áreas científicas que engloba, tal como ciência cognitiva, ciência de computação, interação Humano-Computador, fatores humanos, psicologia, etc..

Uma HRI pode ser processada fundamentalmente por meio de uma interação física, social ou teleoperada, sendo esta última fortemente associada a interações remotas, pelas quais um robô móvel pode ser controlado à distância por um humano. Para este controlo ser processado são utilizados sensores que requerem algum tipo de contacto físico (*e.g.* comando de consola de jogos).

Recentemente surgiram recursos tecnológicos, como por exemplo a Microsoft Kinect, que introduziram um novo nível de abstração na interação Humano-Robô. Este novo recurso veio facilitar a forma como uma máquina pode percecionar o meio que a rodeia. Este facto é tão relevante que no âmbito deste trabalho é possível afirmar que um humano envia comandos de controlo para o robô com **total** nível de abstração. O que isto significa no âmbito deste trabalho? O humano/utilizador apenas efetuando a tarefa de caminhar de um ponto A para um ponto B, sem necessidade de pensar em mais nada que não seja o seu percurso pessoal, está em constante interação com o robô, estando este a realizar um serviço de seguimento.

Um novo nível de HRI é alcançado aqui, pois apesar de um humano não o efetuar de uma forma direta, está a dar informação ao robô da sua localização.

3.3 Estimação da Pose do Esqueleto Humano

Neste trabalho é realizada a função de perceção através da deteção do esqueleto do corpo humano.

Assim, será analisado um pouco mais detalhadamente, nesta secção, como esse processo é realizado, o hardware necessário, que recursos são utilizados, tecnologias envolvidas e o que é imprescindível para ligar tudo isto.

3.3.1 Microsoft Kinect

Destinada a jogos (Xbox 360), a 4 de novenbro de 2010, a Microsoft Kinect, Figura A.1 que se encontra em anexo A, foi lançada para possibilitar ao utilizador uma nova experiência de interação com um jogo.

Esta câmara, foi inicialmente projetada para ser um dispositivo de entrada da Xbox 360, surgindo desta forma com um novo tipo de controlador. O movimento do corpo humano é então uma nova entrada de dados para uma máquina de jogos.

Assim que esta foi lançada no mercado de videojogos, foram debruçados sobre a Microsoft olhares curiosos e muita atenção pela forma como, tecnologicamente, funcionava a câmara.

A comunidade científica identificou imediatamente o grande potencial desta nova tecnologia. Todo o tipo de aplicações e projetos surgiram cada vez mais fazendo uso desta nova câmara. O sucesso foi tal que, em muito pouco tempo, surgiram *hacks* que permitiam aceder ao processamento da Kinect. Não demorou muito para que a Microsoft anunciasse planos para lançamento de um Kit de software de desenvolvimento com um lançamento comercial.

A um ritmo muito acentuado surgiram mais e mais projetos fazendo uso da câmara. Este facto levou à exploração de todos os recurso a nível aplicacional, levando cada vez mais as suas possibilidades ao limite.

A câmara disponibiliza imagens RGB e informação de profundidade através de um sensor, o que torna possível a sua aplicação na área da robótica. Possui hardware interessante com muitos tipos de recursos, nomeadamente: uma câmara RGB, um emissor de infravermelhos (IR) e um sensor de profundidade (câmara IR), dois microfones, 4 fontes digitais de áudio e um sistema de inclinação, ver Figura A.2 que se encontra em anexo A.

É possível obter informação mais detalhada sobre as características e especificidades do dispositivo em [3].

3.3.2 Sensor de Profundidade

A informação de profundidade referida na secção anterior é proveniente de um emissor/recetor de infravermelhos, criando através do emissor um padrão estruturado de luz infravermelha para uma câmara proceder à interpretação da reflexão dos infravermelhos emitidos. Estes são deformados pela presença de formas no campo de visão da câmara. Ao serem interpretados estes infravermelhos, são gerados dados que nos fornecem informação de profundidade. A teoria por trás da geração destes valores baseia-se em conceitos de triangulação, relacionando a posição do emissor, da câmara que os receciona e as formas que se encontram no campo de visão da câmara.

A emissão de infravermelhos é feita a 30*fps*, que fornece imagens de 640 x 480 com indicação de um valor de profundidade para cada pixel da imagem. O campo de visão da câmara é de 43,5° na vertical por 57° na horizontal. É possível obter informação mais detalhada sobre as características e especificidades do dispositivo em [3].

3.3.3 *PrimeSense, OpenNI e NITE*

Nos meses entre o lançamento oficial da Microsoft Kinect e da disponibilização de um software de desenvolvimento pela Microsoft (SDK) foi oferecida uma recompensa, publicamente, para quem desenvolvesse software que fosse capaz de interpretar a saída de dados da Microsoft Kinect num computador.

É então que em novembro de 2010, um *hacker* chamado Hector ganhou a oferta proposta pela Adafruit, que oferecia um prémio para a primeira pessoa que descobrisse como criar drivers para utilizar os recursos da Microsoft Kinect e os tornasse livres para a comunidade.

A **PrimeSense Technology** [49], que pertence atualmente à Apple, criou uma tecnologia de sensores 3D que permite a qualquer dispositivo uma interface de interação entre pessoas e uma máquina, de forma simples e intuitiva. Esta empresa desenvolveu o hardware e software que hoje é possível encontrar na Microsoft Kinect, concebendo também um Kit de desenvolvimento designado de *Prime Sensor Development Kit* (PSDK).

Atualmente, o software que realiza a interface com o hardware continua a ser baseado no originalmente lançado depois do lançamento da Microsoft Kinect (2010), apenas sofrendo diversas atualizações, o que depois de vários anos demonstra a robustez da tecnologia desenvolvida pela PrimeSense.

3.3.4 Pacote *OpenNI Tracker*

OpenNI Tracker [21] é um pacote desenvolvido para o sistema ROS. Este pacote faz uso dos recursos falados na secção 3.3.3 discutida anteriormente.

Resumidamente, este pacote permite detetar pessoas utilizando a Microsoft Kinect. Assim, consegue fornecer a pose de 15 juntas do esqueleto humano, relativamente ao sistema de coordenadas da câmara.

De acordo com a documentação oficial [7] são fornecidas as posições das 15 juntas do esqueleto humano, ver Figura 3.2, nomeadamente do torço, dos braços, das pernas, etc...

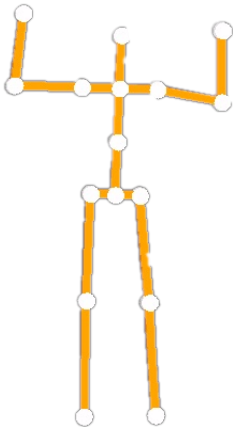


Figura 3.2: 15 juntas do modelo do esqueleto humano disponibilizado pelo pacote **OpenNI Tracker** (imagem retirada de [6]).

Então o pacote publica no sistema ROS os dados relativos ao posicionamento e orientação

das juntas do esqueleto humano detetado. Cada junta terá a sua própria coordenada no espaço que é publicada pelo pacote **OpenNI Tracker** através de um conjunto de dados que, no sistema ROS, são designadas por (tf). Uma tf são transformações geométricas entre todos os referenciais, necessárias para converter as coordenadas das juntas do corpo nas coordenadas relativas ao sistema de coordenadas da Microsoft Kinect e que, posteriormente, possibilitam alocar o sistema de coordenada da câmara dentro do sistema de coordenada do mapa de referência relativo à navegação.

3.4 Extração de Características 3D

Depois de apresentado na secção anterior um dos recursos utilizados neste trabalho, será abordada a forma como usar os dados fornecidos de maneira a poder determinar a pose de um humano e classificar as suas atividades.

Será necessário trabalhar os dados recebidos pelo pacote **OpenNI Tracker** pois estes dados são recebidos sem critério em relação à forma como se pretende que sejam utilizados.

Portanto, primeiro será realizada a extração das características 3D do esqueleto humano que vão ser determinantes para a determinação da pose de uma pessoa e para utilização dos algoritmos de classificação mencionados nas secções 2.2.1.1, 2.2.1.2 e 2.2.1.3. A escolha para a utilização do esqueleto 3D disponibilizado pela Microsoft Kinect é fundamentada pelo facto de ser uma forma simples, completa e compacta de representar o corpo humano.

Utilizando estes dados, diferentes tipos de características podem ser geradas. Um pequeno detalhe que tem extrema importância é o facto de que a criação de diferentes características devem ser **invariantes** no espaço, ou seja, deverão ser independentes da localização da câmara, do deslocamento do robô ou da posição do humano no espaço. Por outras palavras, o sistema deve ser capaz de obter dados com as mesmas propriedades, independentemente da posição ou orientação em que se encontre. Isto acontece porque todo o tipo de características geradas têm como referência elementos do corpo humano, que são dinâmicos no espaço.

De seguida são listados alguns tipos de características geradas neste contexto, com base em alguns autores [9] que relatam o trabalho desenvolvido na área [53] [31] [57]. Os autores mencionados criaram um conjuntos de características que podem ser processadas com os dados obtidos das juntas do esqueleto humano nas condições de **invariância** referidas anteriormente.

A formulação matemática, cuja descrição se segue, está na base da metodologia seguida para a criação de características. Esta encontra-se de forma genérica e permite relacionar juntas - quer uma relação entre juntas do esqueleto de uma pessoa, quer uma relação de juntas entre dois esqueletos presentes. Portanto, fica assim fundamentada a possibilidade de se vir a desenvolver mais trabalho em HRI e interação entre mais do que uma pessoa, reforçando as potencialidades que a utilização desta metodologia oferece. Assim sendo, i e j são quaisquer juntas das pessoas x e y, $t \in T$, sendo que para uma única pessoa $x = y$. $< p^y_{j,t}, p^y_{k,t}, p^y_{l,t} >$ indica o plano gerado por p^y_j, p^y_k, p^y_l.

$< p^y_{j,t}, p^y_{k,t}, p^y_{l,t} >_n$ indica o vector normal do plano gerado por p^y_j, p^y_k, p^y_l.

- Localização da junta:

$$F' = p_{i,t} \tag{3.1}$$

- Distância entre juntas:

$$F^{jd}(i,j;t) = \| \, p^x_{i,t} - p^y_{j,t} \, \| \tag{3.2}$$

- Orientação das juntas:

  Quaternião de rotação para cada uma das juntas.

$$q = [q_0 \quad q_1 \quad q_2 \quad q_3]^T \tag{3.3}$$

  Esta representação justifica-se por não sofrer de um bloqueio no seu manuseio, em inglês denominado por "**Gimbal Lock**". Embora de maior dificuldade para a interpretação humana, possui melhores características numéricas para ser usado em processamento computacional. Os ângulos de Euler, estão relacionados com um quaternião da seguinte forma:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(2(q_0 q_1 + q_2 q_3)), 1 - 2(q_1^2 + q_2^2)) \\ \arcsin(2(q_0 q_2 - q_3 q_1)) \\ \text{atan2}(2(q_0 q_3 + q_1 q_2)), 1 - 2(q_2^2 + q_3^2)) \end{bmatrix} \tag{3.4}$$

  Sendo:

  ϕ - *roll*: Rotação no eixo X

  θ - *pitch*: Rotação no eixo Y

  ψ - *yaw*: Rotação no eixo Z

- Movimento da junta:

$$F^{jm}(i,j;t1,t2) = \| \, p^x_{i,t1} - p^y_{j,t2} \, \| \tag{3.5}$$

- Característica do plano formado entre juntas e sua normal:

$$F^{pl}(i,j,k,l,t) = dist(p^x_{i,t}, < p^y_{j,t}, p^y_{k,t}, p^y_{l,t} >) \tag{3.6}$$

$$F^{npl}(i,j,k,l,t) = dist(p^x_{i,t}, < p^y_{j,t}, p^y_{k,t}, p^y_{l,t} >_n) \tag{3.7}$$

- Velocidade das juntas e sua normal:

$$F^{jv}(i,j,k,t) = \frac{v^x_{i,t}(p^x_{j,t} - p^y_{k,t})}{\parallel p^x_{l,t} - p^y_{l,t} \parallel} \tag{3.8}$$

$$F^{nv}(i,j,k,l,t) = v^x_{i,t} \cdot \hat{n} < p^y_{j,t}, p^y_{k,t}, p^y_{l,t} > \tag{3.9}$$

Como é possível verificar na listagem anterior, é possível gerar diferentes naturezas de características com os dados disponibilizados. Estas características podem ser simples e, tal como a distância entre juntas, outros tipos consideram a orientação e o movimento das juntas combinando dados de origem espacial e temporal. Por fim também é possível ter outras mais complexas como as de planos.

No trabalho aqui desenvolvido não são utilizados todos estes tipos de características, fundamentalmente são utilizadas características de distância com referência ao torso, orientação entre juntas do esqueleto e, por último, também de velocidade. A abordagem aqui tomada baseia-se na análise instante a instante dos dados, como tal, não é considerada a componente temporal mostrada anteriormente. Deste modo, será necessário reformular a teoria geral de forma a se adequar ao pretendido neste trabalho.

Para implementação do classificador são utilizados três tipos de características:

- Distância entre juntas.
- Orientação entre juntas.
- Velocidade das juntas.

Para a distância entre juntas são processadas as características mencionadas. Cada junta é representada por J_i que é um **vetor tridimensional** da junta i representada no espaço de coordenadas do mapa gerado pelo sistema. Na Figura 3.3 é possível observar os diferentes espaços de coordenadas existentes. Todos os espaços de coordenadas estão relacionados geometricamente ao mapa e, deste modo, todos os dados obtidos são representados tendo uma referência comum. Esta característica é o que permite relacionar os dados tratados nos diferentes módulos e nas diferentes fontes de origem.

Tanto a pessoa como a base da câmara são elementos dinâmicos, isto significa que em diferentes instantes uma determinada pose do esqueleto pode assumir valores posicionais referentes a qualquer lugar do espaço existente. Assim, como tudo no sistema é dinâmico, é necessário descobrir um elemento que sirva de referência.

A solução é fixar uma junta do esqueleto humano, que desta forma fornece uma referência dinâmica. Uma compensação posicional é portanto realizada por se assumir este pressuposto. De uma forma genérica, para um esqueleto com P juntas, J_0 será a junta de referência e J_2 uma segunda junta de referência para possibilitar uma normalização dos dados. Habitualmente é utilizada a junta do torso e do pescoço que, por serem elementos centrais e com menos flexibilidade, assumem um papel de fortes candidatos a serem referências escolhidas.

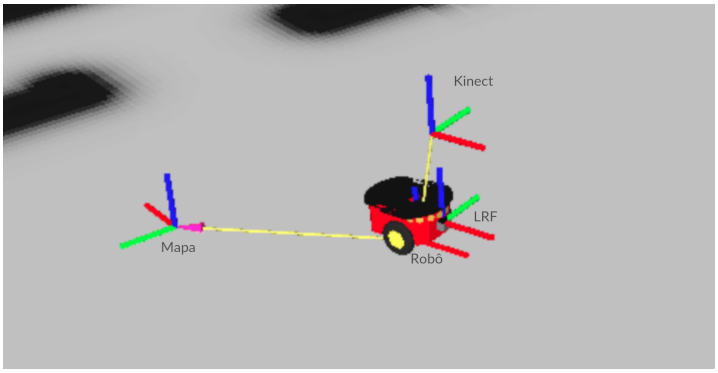


Figura 3.3: Relação dos diferentes espaços de coordenadas.

Formulando então o que foi dito, para a junta i do esqueleto é gerado d_i, vetor que representa a distância entre a junta i denominada J_i e a junta de referência J_0, normalizada pela distância entre J_2 e J_0:

$$d_i = \frac{J_i - J_0}{\parallel J_2 - J_0 \parallel}, \qquad i = 1, 2, ..., P-1 \tag{3.10}$$

No sistema implementado, J_0 corresponde à junta do torso do esqueleto humano e J_2 pertence à junta do pescoço, sendo estas duas juntas elementos de referência que permitem não só a dinâmica de movimento como também a normalização dos dados obtidos. A junta J_i é uma das 14 juntas analisadas individualmente.

O valor d_i é então uma distância que não sofre variação pelo dinamismo da pessoa no campo de visão da câmara, assim como também não sofre pela deslocação do robô.

$$f = [d_1, d_2, d_3, ..., d_{P-1}] \tag{3.11}$$

A característica f é um vetor constituído pelas distâncias de todas as juntas P do esqueleto relativo a um instante.

A característica de distância entre juntas é uma boa forma de classificar gestos mas, para a classificação de posturas, a orientação das juntas do esqueleto assim como a especificação de relações entre elas claramente fornece dados melhor classificáveis. Isto trará maior eficácia aos resultados obtidos, sendo outra das características fundamentais, pois cada atividade desempenhada por um humano regista diferentes orientações entre o torso e a junta respetiva.

Por último, são também utilizadas características de velocidade que permitirão classificar atividades humanas incrementando mais um nível de diferenciação.

No entanto, porque o sistema foi implementado em ROS, os recursos do mesmo podem e devem ser utilizados para minimizar a complexidade do sistema, portando é recorrendo ao sistemas ROS que a orientação das juntas é obtida e o cálculo das velocidades é realizado. As potencialidades que o sistema oferece para manipulação e personalização dos dados disponibilizados sobre tf do sistema são de grande valor e libertam o programador de cálculos excessivamente exaustivos.

Concluindo, o sistema de classificação é compostos por três distintas fontes de informação

geradas de forma a serem invariantes no espaço, dotando o sistema de um conjunto de características base que, pelas suas diferentes naturezas, permitem a obtenção de dados distintivos que vão estar na base de uma boa classificação.

3.5 Considerações Sobre o Sensor Kinect

Ao planear o início de um projeto em que se pretende utilizar a Microsoft Kinect, algumas questões devem ser revistas e consideradas.

Se o projetista negligenciar e decidir ignorar estes aspetos poderá vir a entrar num caminho sem saída e ter que repensar o seu projeto. Para que isso não se suceda na realização deste trabalho são analisados nas seguintes subsecções os tópicos com mais relevância.

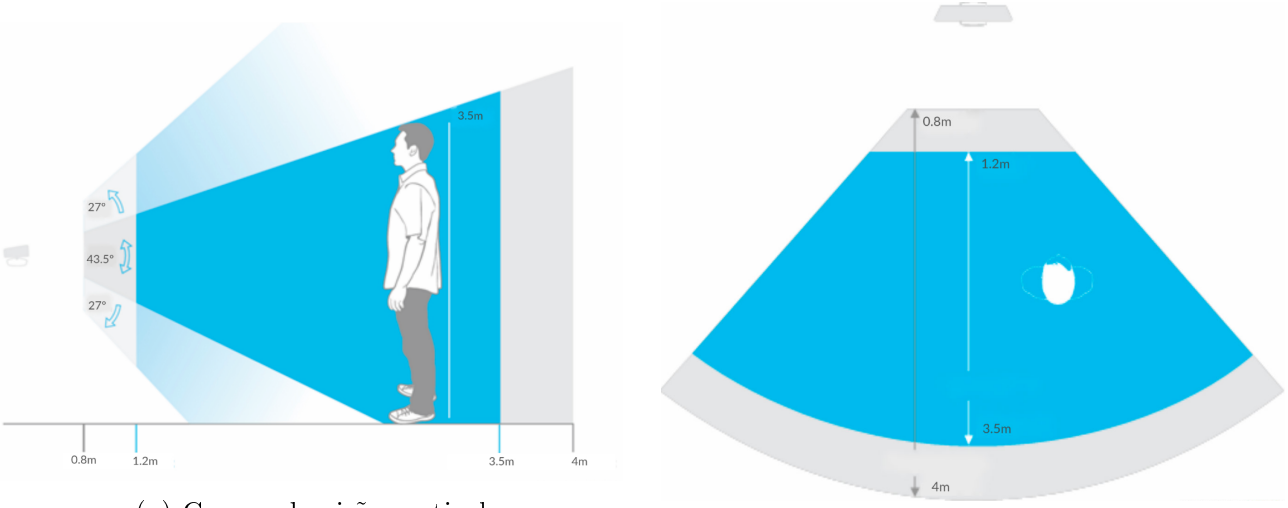
3.5.1 Campo de Visão

Para uma aplicação que tenha como finalidade captar uma pessoa e trabalhar com a informação daí obtida o campo de visão é um aspeto crucial. Primeiramente analisa-se questões de proximidade recorrendo a informação oficial (Microsoft) [1] e a ensaios realizados em laboratório visto não existir uma informação oficial [7] que mencione alguma coisa a respeito de detalhes sobre o campo de visão na utilização do pacote **OpenNi Tracker** e este ser software de código aberto. Desta forma é corroborada a informação disponibilizada pela Microsoft em que se pode afirmar que a Kinect garante sensibilidade para a detecção de uma pessoa se esta estiver a uma distância mínima de 0,8m de proximidade da Kinect. No caso da pessoa se aproximar mais do que o limite estabelecido, a deteção da pessoa não será possível, ver Figura 3.4. Ao interpretar as duas imagens é possível verificar que se a pessoa se aproximar demasiado da Kinect sairá do campo de visão da mesma.

Se o robô utilizado na realização do projeto possuir dimensões elevadas, facilmente se minimiza este problema colocando a câmara numa localização atrasada em relação à frente do robô. Para pequenos robôs resta apenas ter em mente que terá de ser acautelada a aproximação ao mesmo.

No que respeita ao alcance, num raio de 4 metros e dentro do raio de visão angular horizontal da câmara de 57° a pessoa será captada pela câmara. Na verdade é possível atingir os 5 metros de distância em condições ótimas de luminosidade. Verticalmente, a câmara possui uma gama angular de 43,5° graus, com a possibilidade de um ajuste de $+/-$ 27°. Estes valores de campo de visão da Kinect dececionam comparativamente a outras câmaras, mas não se pode deixar de mencionar que esta gama reduzida deve-se em particular ao sensor que possui. Para uma câmara com um sensor deste tipo, na realidade até são valores bons.

Comparativamente a outros sensores como o Hokuyo ou Lidar, a gama angular é realmente muito reduzida, no entanto não se pode esquecer de que os dados obtidos pela câmara são dados 3D e não apenas 2D como os outros sensores mencionados.



(a) Campo de visão vertical

(b) Campo de visão horizontal

Figura 3.4: Campo de visão global da Microsoft Kinect, imagens retiradas de [1].

Uma solução alternativa perante a necessidade de aumentar o campo de visão passa por alterar a disposição da câmara de forma a que fique colocada numa zona superior ao robô. Assim o campo de visão localiza-se no sentido de cima para baixo. Desta forma, o campo de visão melhora substancialmente em relação à área que é suposto cobrir. Contudo, esta solução retira alcance em profundidade à câmara.

A melhor solução terá de ser encontrada em função da aplicação.

3.5.2 Capacidade de Hardware

Uma questão muito relevante para o projeto é perceber se é possível usar esta câmara em termos energéticos. A câmara consome cerca de 2A de corrente, o que para um robô pequeno que não tenha baterias com grande capacidade pode significar um grande problema pois, desta forma, retira muita autonomia energética ao robô.

3.5.3 Ambiente

A Microsoft Kinect é provida de um emissor de infravermelhos que fornece uma nova dimensão de profundidade. Desta forma surge um grande problema: a luz solar. A Kinect funciona pela projeção de um laser de infravermelhos num padrão estruturado e a câmara de IR captura a distorção do padrão emitido determinando assim a distância dos pontos refletidos. Com a incidência da luz solar, o padrão é difundido e distorcido pela luz solar e portanto a informação contida nos feixes de luz infravermelha é completamente perdida.

Em oposição a este importante facto, a câmara não necessita de luz ambiente rigorosamente nenhuma. Mais ainda, pode operar em total escuridão, dotando o aparelho de uma interessante característica.

Resta dizer que existem grandes limitações na utilização desta câmara e que a sua utilização deve ser previamente estudada. Em contrapartida, são disponibilizadas características

muito interessantes por esta câmara, o que, aplicadas de forma conveniente, proporciona um grande potencial de aplicações em que pode ser inserida, nomeadamente no desenvolvimento de projetos e investigação científica.

3.6 Distância Pessoal

O conceito *proxemics* [39] descreve a distância entre pessoas nas suas situações sociais e a forma como interagem. Assim são classificados de forma distinta diferentes espaços que nos relacionam uns com os outros, nomeadamente: espaço íntimo, espaço pessoal, espaço social e espaço público. Ver Figura 3.5 para uma melhor visualização e compreensão do conceito.

Em 1959, Edward T. Hall estabeleceu o conceito *proxemics*, mencionado no parágrafo anterior. Primeiramente a ideia foi introduzida num dos seus livros *The Hidden Dimension* (1966). O seu estudo não só definiu distâncias e classificação dos diferentes espaços que nos rodeiam mas também como esse espaço pode ser usado. Este conceito define como diferentes circunstâncias alteram a nossa noção de espaço pessoal e como podem manipular a forma como nos sentimos.

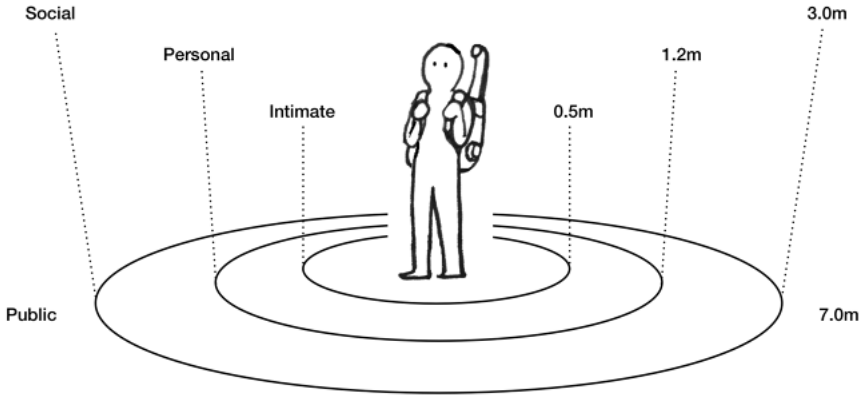


Figura 3.5: Visualização de todas as zonas estabelecidas pela investigação de Edward T. Hall (1966) [39], imagem extraída de [5].

Qual a relevância deste tema? É justamente devido ao conceito de *proxemics* ser um dos elementos mais fundamentais de comportamentos sociais.

Para um robô móvel que interage socialmente com outras pessoas e no mesmo espaço, a forma como o robô se desloca e a forma como se localiza em relação às pessoas com que socializa tem um impacto muito relevante no seu desempenho assim como na sua aceitação no mesmo espaço que os humanos. Um outro tópico com muita relevância, ao invés de analisar apenas o ponto de vista do humano em relação ao robô, considerações do robô em relação ao humano também devem ser tidas em atenção, designadamente as necessidades do robô para o desempenho das suas funções globais de perceção. O robô comporta características limitativas que poderão comprometer a aquisição de dados, tais como o ambiente e a necessidade de manter o utilizador no seu capo de visão. Com os dois aspetos em conta, a forma

como o robô se move e se localiza deve ser otimizada em prol de encontrar um equilíbrio entre as considerações mencionadas.

3.7 Sumário

A ideia fundamental com a exposição deste capítulo foi abordar o tema recorrendo a diferentes tópicos que têm especial relevância no tema seguimento de pessoas por um robô. Outros tópicos que realizam a ponte entre o tema e a implementação realizada neste trabalho foram discutidos com o intuito de contextualizar o leitor.

A apresentação de conceitos que estiveram na base do trabalho, especificação de características técnicas e considerações importantes acerca do tema e do trabalho são apresentadas de forma a ligar diferentes tópicos ao trabalho realizado.

No capítulo seguinte vai ser apresentado o sistema implementado, levando o leitor a conhecer progressivamente a forma como está projetado.

4 Sistema Proposto

Neste capítulo é apresentado o sistema implementado. É transmitida informação ao leitor de forma progressiva, ou seja, é apresentada inicialmente ao leitor uma visão de alto nível e simplista da estrutura do sistema e depois, individualmente, são explicados em pormenor as suas características.

Pretende-se assim que o leitor consiga entender claramente as diferentes fases de implementação assim como todas as informações relevantes.

4.1 Opções de Projeto e Implementação

Todo o projeto foi desenvolvido para operar sobre o sistema ROS [43]. A razão principal que fundamenta a opção por este sistema é a existência de suporte e uma comunidade internacional que projetam uma maior confiança na sua utilização. O ROS disponibiliza ferramentas e bibliotecas para ajudar os desenvolvedores de *software* a investigar, criar e melhorar aplicações para robôs. O nível de abstração relativo ao *hardware* é bom e útil.

Devido ao ROS estar licenciado sob fonte aberta (Licença BSD), qualquer desenvolvedor tem acesso a recursos oficiais, pode instalar o sistema no seu computador e pode ter acesso a informação oficial sobre os vários conceitos, bibliotecas e questões técnicas relacionadas com o ROS. Pode ainda consultar tutoriais para, passo-a-passo, aprender ROS em contato direto com o sistema, integrar-se com a comunidade ROS, submeter o seu próprio repositório e ter acesso a suporte técnico oficial. Estas são vantagens que tornam vantajosa a escolha.

A adicionar às vantagens mencionadas no paragrafo anterior, e por consequência da utilização do sistema por uma grande comunidade internacional, verifica-se um maior desenvolvimento do mesmo, o que faz com que várias distribuições de ROS sejam lançadas periodicamente. O conceito de *pacote* veio facilitar o sistema e consequentemente a sua partilha, assim como um conjunto de ferramentas providas pelo sistema que facilitam muito a vida do desenvolvedor (e.g. *debugging*). Existem também robôs já projetados especificamente para operar em ROS, o que pode libertar por completo o desenvolvedor de aspetos relacionados com *hardware*. Também os *drivers* para utilização de dispositivos externos são um ótimo recurso e hoje em dia já existem para quase todo o tipo de sensores e controladores.

Por fim destaca-se a divulgação e partilha de conhecimento entre a comunidade, desde livros e artigos publicados com implementações de código aberto, documentação e tutoriais com código fonte igualmente aberto, cursos que utilizam ou ensinam ROS e também eventos

relacionados com o tema.

Com um sistema desenvolvido de uma forma tão alargada, com suporte livre e com uma comunidade ativa de grande dimensão, a escolha deste sistema tornou mais fácil as tarefas de desenvolvimento levadas a cabo neste trabalho de dissertação.

No que respeita às opções tomadas para o desenvolvimento de um sistema de classificação, depois de uma análise sobre as possibilidades existentes, foi decidido desenvolver este módulo em linguagem de programação Python. A razão pelo seu desenvolvimento em Python deve-se ao facto de que, desta forma, seria possível usar os algoritmos de classificação (ver secção 2.2.1) contidos na biblioteca de aprendizagem automática *scikit-learn* [41]. A razão pela opção tomada deve-se, fundamentalmente, ao facto de ser uma ferramenta simples e eficiente para análise de dados. A biblioteca *scikit-learn* disponível em Python é muito fácil de utilizar, simplificando muito um projeto deste tipo de sistema. Está, assim, acessível para qualquer pessoa em termos de complexidade. Estes algoritmos são bastante dinâmicos e podem ser usados e reutilizados em muitos e diversos contextos. Foram construídos sobre *NumPy*, *SciPy* e *matplotlib*, ferramentas de análise, cálculo e visualização indispensáveis para trabalhar com os mesmos. Por fim, o facto de ser uma biblioteca destinada à livre utilização e comercialmente utilizável sob a licença BSD, fornece ao projetista condições ideais para optar por esta solução.

Em comparação a outras opções como o Matlab, que disponibiliza condições semelhantes, o facto da opção pela biblioteca *scikit-learn* possibilitar a operação do mesmo sobre o sistema ROS, onde as restantes componentes operam, permite a integração de todos os módulos num só sistema. Desta forma, a comunicação entre diferentes módulos é realizada apenas sobre um sistema que possui serviços de comunicação entre nós muito intuitivo.

É assim apresentado um sistema a operar sobre o ROS que contempla diferentes módulos desenvolvidos em C++ e Python.

4.2 Recursos de *Hardware*

A plataforma utilizada para a realização deste trabalho é constituída por quatro elementos chave com funções e finalidades específicas.

Para a locomoção é utilizado um robô móvel, o Pioneer P3-DX (ver Figura 4.1a). O Pioneer é um robô diferencial, robusto, destinado ao meio académico. Este robô é versátil, confiável e a sua durabilidade faz dele uma plataforma muito popular no meio académico. Esta plataforma pode atingir a velocidade de 1,6 m/s e transportar uma carga máxima de 23 kg.

O Hokuyo (ver Figura 4.1b), é um sensor *laser range finder* (LRF) com boa precisão (Distância 20 mm a 4000 mm : ±3% da medida) e resolução (0,36°), com um ângulo máximo de varrimento de 240°, apresentando-se como uma solução interessante em muitas aplicações de robôs móveis, comparado com outros LRF com melhores características mas muito mais dispendiosos. São muito utilizados para realizar *Simultaneous localization and mapping*

(a) Robô Móvel Pioneer P3-DX.

(b) Hokuyo URG-04LX.

(c) Microsoft Kinect.

(d) Computador Portátil.

Figura 4.1: Dispositivos de *hardware* utilizados neste trabalho.

(SLAM). Compacto, leve e com um baixo consumo energético possibilita uma fácil aplicação em qualquer ambiente e está preparado para longos períodos de operação. É um sensor muito robusto e não sofre quase influência nenhuma negativa em ambientes de iluminação extremos, como muito brilho ou escuridão. Tem ainda a vantagem de que a sua operação não viola a privacidade nem o bem estar de qualquer pessoa que esteja no mesmo ambiente que o sensor. A sua grande finalidade neste projeto será dados que possibilitam o robô de realizar a navegação com o conhecimento do ambiente fornecido pelo LRF.

A Microsoft Kinect (ver secção 3.3.1) é o elemento que possibilita a deteção da pose de um humano e disponibiliza informação a todo o sistema de classificação.

Por último, todo o processamento é centralizado num computador portátil e, portanto, todos os restantes elementos constituintes da plataforma comunicam diretamente com o computador.

O computador portátil utilizado para a operação é um Sony VPCCB2M8E, com 8 GB de RAM, processador Intel Core i5-2410M, CPU de 2.3 GHz de 4 núcleos, sob o sistema operativo Ubuntu 14.04 LTS e ROS Indigo. De notar que surgiram dificuldades de operação com a Kinect em outros computadores testados, quando se encontravam a operar apenas com a sua bateria. Assim, sugere-se que para a operação com este sistema seja utilizado um computador com uma bateria robusta, que assegure os níveis energéticos normais de operação do computador. Caso isso não se suceda, surgirão problemas na utilização da Microsoft Kinect.

A utilização de um computador portátil deve-se exclusivamente a questões de simplicidade. O poder de processamento, portabilidade, acesso a ambiente gráfico e possibilidade de usar ambientes simulados são as grandes vantagens para a sua utilização. No entanto, a utilização de unidades de processamento (*e.g.* odroid e raspberry pi3) com dimensões reduzidas é uma possibilidade, sendo possível tornar a plataforma mais compacta a nível de *hardware*. Esta opção é aconselhável numa fase final, em que todo o desenvolvimento está concluído, pois estas unidades de processamento não possuem processamento gráfico, não permitindo utilizar todos os meios de desenvolvimento possíveis com um computador portátil.

Para resumir, a Figura A.3 no anexo A, ilustra a plataforma equipada com todos os recursos de *hardware* apresentados anteriormente. Assim, de uma forma intuitiva, o leitor pode consultar os diferentes elementos que a constituem e as suas principais finalidades.

4.3 Visão Global do Sistema

Nesta secção é apresentada uma visão geral do sistema, com a qual se pretende que, a um nível de abstração mais elevado, o leitor perceba a organização do sistema desenvolvido neste trabalho, preparando-o assim para entrar em mais detalhes nas secções seguintes.

O sistema é constituído por três componentes fundamentais (ver Figura 4.2). São eles **sensorização**, **classificação** e **navegação**.

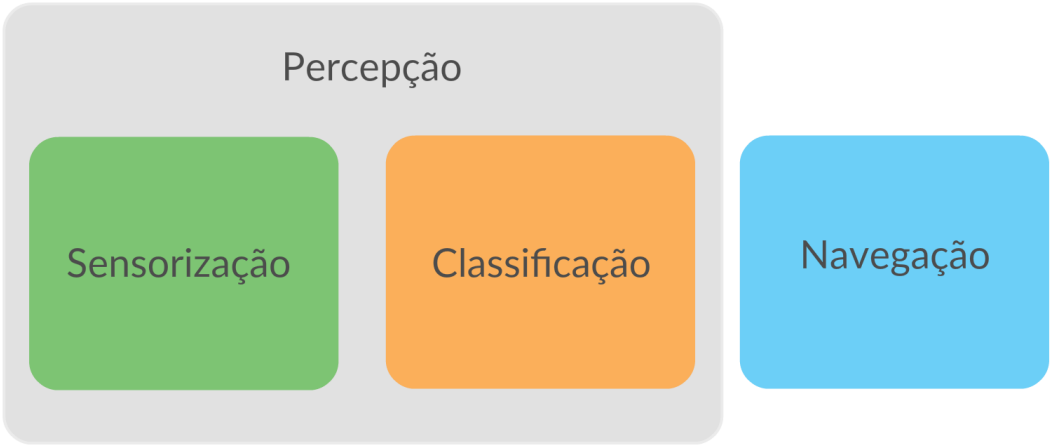


Figura 4.2: Arquitectura de software

Como qualquer sistema robótico, a percepção do ambiente que o rodeia, independentemente da sua função, é uma componente de extrema relevância sem a qual o robô não conseguirá interagir com elementos externos (*e.g.* obstáculos, presença humana, *etc.*). Portanto neste sistema aqui apresentado a primeira componente trabalhada foram os elementos de sensorização. Para percepcionar o seu ambiente, no robô, foram instalados um sensor LRF e uma câmara Microsoft Kinect (ver anexo A, Figura A.3). Esta componente é responsável por prover o sistema com informação útil que estará na origem das ações desempenhadas pelo robô.

A componente de classificação tratará de capacitar o robô de reconhecimento de atividades humanas, implementando um sistema de classificação baseado em aprendizagem automática supervisionada, como explicado na secção 2.2. Para desempenhar estas funções, o módulo de classificação utiliza os dados provenientes do módulo de sensorização, procedendo ao tratamento de dados para que seja possível proceder a uma classificação de atividades de uma pessoa.

Por último, ao sistema pertence também uma componente de navegação. É neste módulo onde toda a ação do robô é gerada, tendo em consideração múltiplos aspetos para a produção de movimento. O movimento é condicionado e tratado de forma a ter em atenção questões como obstáculos no seu ambiente e distância social do robô em relação ao utilizador, como abordado na secção 3.6. Este módulo usa informações provenientes do módulo de sensorização, nomeadamente dados relativos à pose do humano seguido e informação do ambiente providas pelo LRF.

Como foi explicado anteriormente, mesmo apenas com uma noção geral do sistema, facilmente se compreende que todas as componentes estão profundamente relacionadas.

Nas secções seguintes irá perceber com maior detalhe como estes módulos operam e foram projetados.

4.4 Sensorização

A componente de sensorização é constituída por uma câmara Microsoft Kinect a operar em ROS. Para ser possível usar esta câmara é utilizado um *driver* denominado de *OpenNI* disponibilizado pela comunidade científica para livre utilização.

Para tornar mais fácil a sua utilização e possibilitar um maior nível de abstracção dos programadores em relação a aspetos relacionados com o *hardware*, é utilizada a *middleware library PrimeSense NiTE 2.0*. Assim, um desenvolvedor pode focar-se especificamente no propósito da sua aplicação sem necessitar de ter que realizar procedimentos de calibração ou de identificação da pose inicial. Recorde-se que na secção 3.3.3 foram abordadas algumas características agora mencionadas com mais detalhe.

Através dos recursos mencionados, a Microsoft Kinect fornece diferentes fontes de dados (*e.g.*: dados da nuvem de pontos 3D, dados da imagem RGB e informações relativas à câmara), como se pode visualizar na Figura 4.3. Esta informação flui para um nó denominado *OpenNi Tracker* que procede à deteção do esqueleto de um humano, complementando com um contínuo seguimento da pessoa enquanto esta se situar ao alcance do campo de visão da câmara. Esta informação é publicada sob a forma de transformada */tf* do sistema ROS, com a informação relativa a 15 juntas do esqueleto humano. Cada *tf* publicada possui a informação de posicionamento e orientação de uma junta, resultando assim na completa informação da sua pose. Sempre que se verifique uma deteção, novas publicações são geradas com uma frequência de 30 Hz. A informação é imediatamente difundida pelos módulos de Classificação e Navegação (ver secção 4.5 e 4.6), por forma a ser utilizada de diferentes formas e para diferentes finalidades como se verá nas respetivas secções.

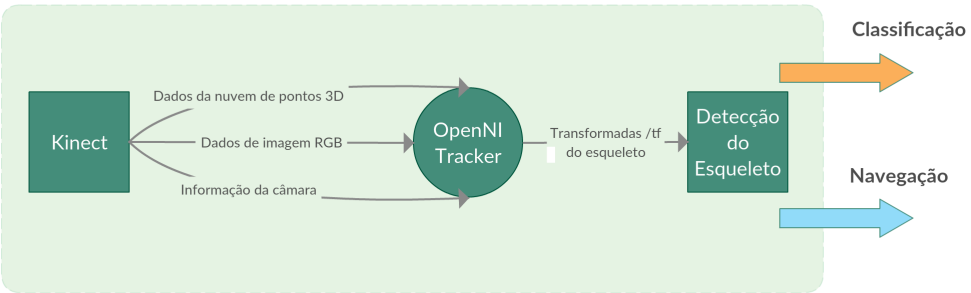


Figura 4.3: Módulo de perceção.

É importante salientar que a deteção do esqueleto do corpo humano da pessoa a ser seguida pelo robô é realizada com recurso à utilização do pacote *OpenNI Tracker*, de domínio

público, sendo este recurso o único com origem externa e não desenvolvido no âmbito deste trabalho de dissertação. Todo o restante sistema e trabalho apresentado foi completamente desenvolvido de raiz.

Todo o processo de deteção do esqueleto de um humano é baseado na obtenção de informação 3D dos píxeis da imagem, só possível pela tecnologia do sensor da Kinect. No entanto, será necessário estar ciente das considerações mencionadas na secção 3.5 acerca das restrições da sua utilização.

Um outro elemento que obtém dados de percepção é o LRF utilizado na plataforma. Pela sua importância no módulo de navegação e porque apesar de ser um recurso de perceção é utilizado somente no módulo de navegação, será visto mais em pormenor na secção 4.6 onde serão analisadas todas as particularidades do referido módulo.

4.5 Classificação

O sistema proposto possui a capacidade de realizar reconhecimento de atividades humanas através de um sistema de classificação baseado em aprendizagem automática supervisionada (ver secção 2.2).

O módulo de classificação foi desenvolvido de raiz em linguagem de programação Python. A razão pelo seu desenvolvimento em Python deve-se às opções de implementação mencionadas na secção 4.1.

Segundo as características fundamentadas na secção 3.4, este sistema consegue extrair informações de diferentes origens (*e.g.* posição das juntas, velocidade das juntas, orientação da juntas e orientação do movimento realizado) o que o capacita a realizar o reconhecimento de atividades de uma pessoa.

Desta forma pretende-se, com este sistema de classificação, realizar o reconhecimento de atividades de uma forma simples e eficiente.

A Figura 4.4 apresenta o sistema de classificação proposto. Inicialmente é processada cada tipologia de características de forma a conseguir transformar os dados fornecidos pelo módulo de sensorização em dados melhor classificáveis. Na imagem, o passo nº1 do processo apresenta o processo de transformação dos dados de posicionamento das juntas do esqueleto humano, correspondendo a C_1 a situação demonstrada no primeiro passo da Figura 4.4. A C_2 corresponderá o processo realizado para o tratamento de dados realizado com os dados fornecidos da orientação de juntas do esqueleto humano, a C_3 a velocidade das juntas e assim sucessivamente.

O processo realizado em cada tipologia ($C_1, C_2, C_3, ...$) foi abordado com maior detalhe na secção 3.4.

Prosseguindo, depois de extraídas todas as tipologias de características, no passo nº2 é necessário seleccionar as diferentes tipologias para agrupar no passo 3 o conjunto de classes classificáveis num único vetor C_i. Em simultâneo, ainda no passo nº3, um vetor designado de vetor A será o conjunto de dados a classificar que poderá ser uma simples amostra de

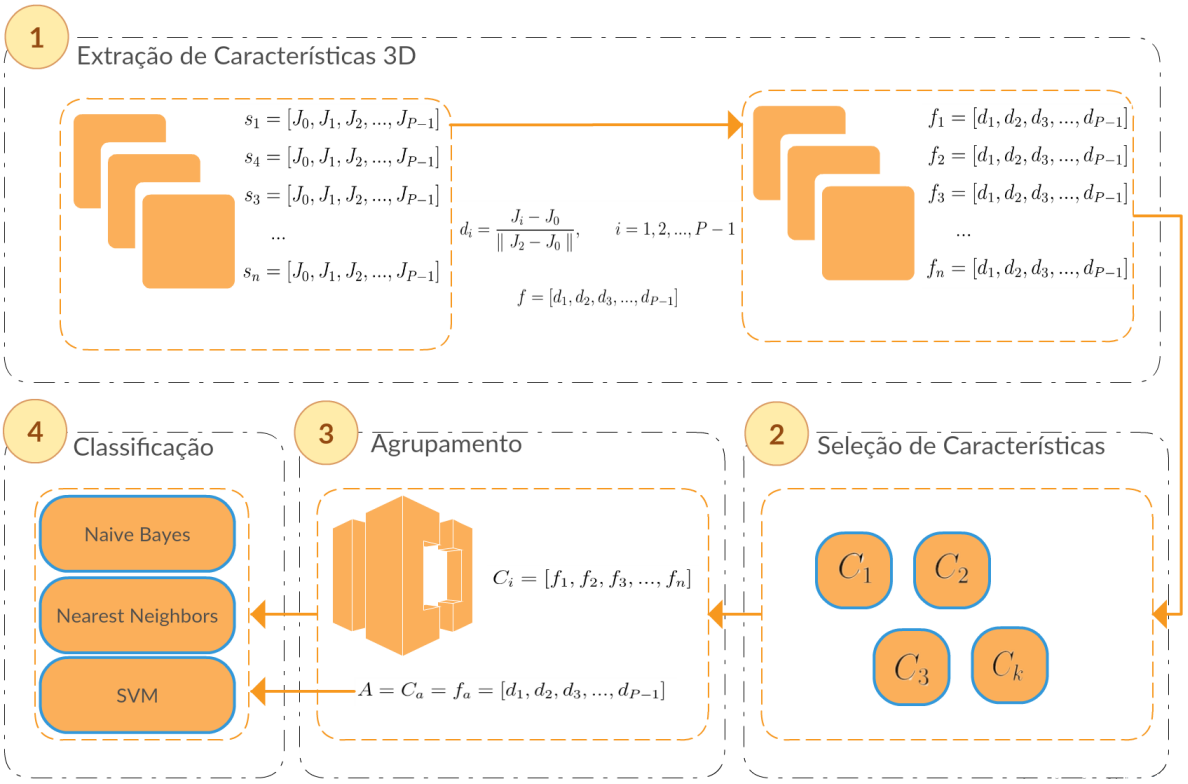


Figura 4.4: Módulo de classificação.

uma situação de teste ou os dados a classificar em tempo real.

Por fim, três diferentes algoritmos de classificação são utilizados, ver fundamentos na secção 2.2.1, depois dos dados terem sido devidamente tratados e estruturados. Para a utilização destes algoritmos será apenas necessário inserir corretamente o vetor C_i com as diferentes classes (dados de treino) e o vetor A. Um resultado será obtido por cada algoritmo utilizado. Depois de obtido o vetor resultante da classificação, uma ponderação será feita com pesos semelhantes. Em caso de um resultado que não demonstre coerência entre os algoritmos, a repetição desse processo específico é novamente realizada com a obtenção de um novo conjunto de dados da situação atual.

Na Figura 4.5 é possível observar as diferentes classes correspondentes a diferentes atividades humanas. O resultado da classificação será um dos identificados na imagem, nomeadamente: permanecer de pé, sentado, braços abertos, caído no chão, a saltar e a andar.

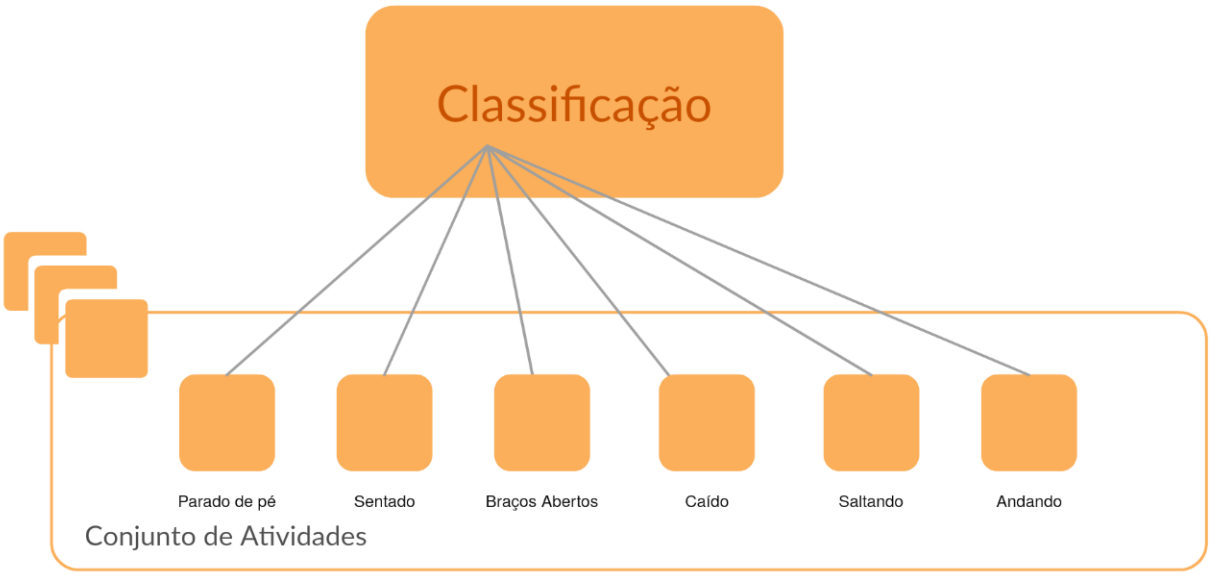


Figura 4.5: Conjunto de atividades propostas no sistema de classificação.

4.6 Navegação

A componente de navegação envia comandos de velocidade para o robô móvel para que este exiba o comportamento adequado à situação percecionada através dos outros módulos do sistema. Em termos de opções para a realização de uma navegação adaptada ao que é proposto neste trabalho — seguimento de um humano — não existiu nenhuma solução que se afigurasse, à partida de forma óbvia, como sendo a melhor. Deste modo, e desconhecendo se qualquer escolha tomada teria os resultados pretendidos, diferentes técnicas foram testadas a fim de perceber se era possível adaptar as técnicas utilizadas às necessidades do problema.

Inicialmente foram testados pacotes de navegação muito conhecidos na comunidade ROS, nomeadamente o *move_base* em associação com um pacote de SLAM, como por exemplo o *gmapping*. Sem dúvida que a realização de SLAM e geração de trajetórias em função do objetivo de destino operam de uma forma eficiente. No entanto, surgiram pelo facto da pessoa a ser seguida constituir um obstáculo com uma proximidade relevante. Depois de um tempo de investigação, foi descoberto que era possível minimizar o efeito da presença de uma pessoa através de um serviço que consegue limpar o mapa local de uma forma periódica, impedindo então que um objeto dinâmico interfira na geração de trajetória. É uma solução possível, mas que peca pela ineficiência da necessidade de proceder a limpezas do mapa local muito frequentemente e pela necessidade de enviar novos objetivos de navegação regularmente, obrigando o algoritmo utilizado pelo *move_base* a gerar novas trajetórias muito frequentemente. Esta necessidade traz ineficiência à aplicação.

Após muitas tentativas para se encontrar uma solução para esta ineficiência emergiu a necessidade de criar uma aplicação de navegação desenvolvida de raiz que atendesse às necessidades impostas pelo objetivo do trabalho.

A aplicação desenvolvida em C++ baseia-se na aplicação da teoria de campos de potenci-

ais virtuais [30] como principal meio de navegação do robô móvel, provendo assim o robô da capacidade de navegar autonomamente. A principal característica desta abordagem consiste na atribuição de um valor de potencial aos elementos do ambiente e este ser propagado pelo ambiente, gerando um campo de potencial. Este campo de potencial possibilita, posteriormente, a orientação do sistema autónomo no sentido de atingir o objetivo especificado que, neste caso, será o de seguir uma pessoa.

Para atribuição de valores de potencial aos elementos localizados no ambiente do robô, são usadas as seguintes equações [30]:

- O vetor de força de repulsão é definido da seguinte forma:

$$F_r = \sum_{i=0}^{N} \begin{cases} \frac{K_r}{d^2}\left[\frac{x_{s_i}-x_r}{d}\hat{\boldsymbol{x}} + \frac{y_{s_i}-y_r}{d}\hat{\boldsymbol{y}}\right] & , se \quad r > 0,5m \\ \\ 0 & , se \quad r < 0,5m \end{cases} \tag{4.1}$$

  em que K_r corresponde a uma constante de força repulsiva com valor negativo, d corresponde à distância entre o robô e um objeto detetado pelo sensor, (x_r, y_r) são as coordenadas da localização do robô no mundo (informação obtida pelo *driver* do robô da odometria) e (x_{s_i}, y_{s_i}) são as coordenadas do objeto detetado pelo sensor correspondente à orientação i da gama de medidas N fornecida pelo sensor.

  Só é atribuído um valor significativo caso r seja maior que 0,5 m. A variável r é o raio calculado entre a localização da pessoa e uma medida fornecida pelo sensor, definido como $r = \sqrt{(x_h - x_{s_i})^2 + (y_h - y_{s_i})^2}$, em que (x_h, y_h) são as coordenadas da localização da pessoa. Tem como objetivo filtrar os dados fornecidos pelo sensor, negligenciando a pessoa como um objeto dinâmico e garantindo que da sua localização não são geradas forças de repulsão.

- O vetor de força de atração é determinado pela seguinte equação:

$$F_a = K_a \left[\frac{x_h - x_r}{d}\hat{\boldsymbol{x}} + \frac{y_h - y_r}{d}\hat{\boldsymbol{y}}\right] \tag{4.2}$$

  A localização da pessoa é um dado sempre presente no sistema de navegação. À localização da pessoa é atribuído um vetor de atração e desta forma o robô possui um objetivo, que varia no tempo consoante o deslocamento da pessoa no tempo.

- Força resultante:

$$F_t = F_a + F_r \tag{4.3}$$

  A força resultante total que irá determinar a velocidade e orientação do robô num determinado momento corresponde à soma da força atrativa com a resultante das forças repulsivas. Tendo em conta que os cálculos de cada componente de força $\hat{x}$ e $\hat{y}$

Figura 4.6: Representação vetorial da incidência dos dados de sensorização obtidos na navegação desenvolvida.

foram feitos em separado, a força resultante também é definida nas componentes $\hat{x}$ e $\hat{y}$.

Na Figura 4.6 é possível observar a incidência dos elementos presentes ao redor do robô, a sua influência e o modo como são consideradas. É uma figura meramente exemplificativa que se destina apenas a dar ao leitor uma visão simplificada do processo.

Desta forma é possível adequar a navegação desenvolvida tendo em consideração as exigências do projeto.

O aspeto comportamental inserido no sistema de navegação tem uma importante missão que visa afetar as ações de navegação com base nos acontecimentos percecionados pelo robô relativamente ao humano. A exercer influência nas decisões de navegação, paralelamente à teoria de campo de potenciais aplicada, existem considerações comportamentais.

Assim, os aspetos comportamentais do robô são os seguintes:

- Manter uma distância socialmente aceitável.

  O robô mantém uma distância socialmente aceitável, não violando o espaço íntimo nem pessoal do utilizador (ver secção 3.6), que garante 1,2 metros de distância entre o robô e o utilizador.

- Manter a pessoa no campo de visão do robô.

Se o utilizador violar esse espaço intencionalmente, o robô roda sobre si próprio (sem movimento de translação) de modo a manter o utilizador no seu campo de visão.

- Adaptação da velocidade do robô à velocidade do utilizador.

  A velocidade de deslocamento do utilizador é sempre tida em consideração por forma a ajustar a velocidade de deslocamento do robô à velocidade do utilizador. Para obtenção da velocidade do utilizador foi utilizada uma ferramenta denominada de *lookupTwist* membro da classe *tf::Transformer* disponibilizada pelo sistema ROS. Assim, para o seu cálculo, foi necessário definir alguns parâmetros e realizar algumas considerações. Primeiramente, a velocidade de deslocamento da pessoa é determinada através da informação relativa ao posicionamento do torso da pessoa. Com base neste pressuposto, é necessário definir alguns parâmetros, nomeadamente *tracking_frame*, *observation_frame*, *reference_frame*, *reference_point*, *reference_point_frame*, tempo (t) e o período (T). Para aplicação é assumido que o ponto de referência é a origem do *tracking_frame* e a *reference_frame* é o mesmo que a *observation_frame*. Simplificando, é o ponto observado para o cálculo da velocidade, considerando diferentes *frames* escolhidas pelo programador. O tempo é o corrente e o período é de $0,1s$, podendo assim obter a velocidade média no intervalo definido $(t - \frac{T}{2}, t + \frac{T}{2})$.

  Para integração da velocidade da pessoa no sistema de navegação, a equação do cálculo de velocidade linear do robô sofreu modificações de forma a contemplar a velocidade da pessoa no cálculo da velocidade de seguimento da mesma, da seguinte forma:

$$V_l = K_v H_v \sqrt{F_{t_x}^2 + F_{t_y}^2} \tag{4.4}$$

  em que V_l é a velocidade linear do robô, K_v é uma constante de velocidade e H_v é a velocidade de deslocamento da pessoa. As variáveis F_{t_x} e F_{t_y} são as componentes do vetor resultante da aplicação do cálculo dos campos de potencial.

  Desta forma é minimizada a perda do utilizador pelo robô. Apesar de a velocidade máxima do robô ser igual a 1,6 m/s, esta é demasiado elevada quando comparada com a velocidade típica de locomoção de uma pessoa, que pode revelar-se perigosa para os humanos que partilhem o mesmo espaço físico com o robô. A velocidade é portanto limitada a 0,6 m/s pela aplicação, mesmo na necessidade do robô se deslocar com maior velocidade.

- Utilização do reconhecimento de atividades humanas para decisões de navegação.

  O sistema de classificação foi desenvolvido pela necessidade de obter mais informação acerca do utilizador de forma a poder dotar um robô de um melhor nível comportamental. Neste caso específico, a informação da atividade da pessoa é tida em conta para a necessidade do robô se aproximar, não só através do conhecimento da distância a que se encontra o utilizador, mas também para perceber se realmente existe necessidade de deslocamento (*e.g.* o utilizador encontra-se a uma distância de 3 metros, mas parado).

A informação resultante do reconhecimento de atividades provida pelo sistema de classificação é publicada num sistema de mensagens do sistema ROS com um período de 2 segundos, baseada nos resultados obtidos durante esse mesmo período. Esse resultado denomina-se de *class_ result*. O nó de navegação vai ler a mensagem sempre que uma nova mensagem seja publicada e a informação é utilizada para decidir se o robô se deve movimentar ou permanecer parado, dependendo da atividade reconhecida.

Figura 4.7: Módulo de Navegação.

A Figura 4.7 apresenta o módulo de navegação composto pelos dois nós mencionados, que faz uso de toda a informação proveniente da plataforma a fim de atingir a navegação personalizada desejada e desenvolvida para este trabalho. Depois de processada a informação nos dois nós mencionados, são gerados comandos de ação para o robô através do tópico */cmd vel* existente para comunicar com a plataforma.

4.7 Sumário

Este capítulo foi progressivamente apresentado de forma a dar a conhecer o sistema implementado neste trabalho. Inicialmente foram apresentadas as escolhas tomadas para o desenvolvimento deste sistema, foi apresentada a plataforma sob a qual o sistema foi desenvolvido e apresentada uma visão geral da constituição e arquitetura do mesmo, explorando detalhadamente como foram implementados.

Espera-se assim que o leitor tenha conseguido obter um conhecimento profundo sobre o sistema desenvolvido e de todas as suas particularidades.

No capítulo que se segue será apresentado um conjunto de dados conclusivos que demonstram e comprovam o desempenho e eficiência em circunstâncias reais e experimentais.

5 Validação Experimental e Análise de Resultados

Neste capítulo será avaliado o desempenho das diferentes componentes que constituem o sistema apresentado na secção anterior.

Inicialmente será avaliado o desempenho da ação de seguimento do robô em termos comportamentais e, de seguida, o sistema de classificação de atividades humanas será validado independentemente do resto do sistema. Por fim, é realizada uma avaliação do sistema a operar com todas as suas funções. Desta forma pretende-se perceber se o sistema de classificação de atividades sofre influência pelo movimento do robô e observar o desempenho da função de seguimento em diferentes experiências realizadas.

5.1 Testes Comportamentais

O movimento do robô não é uma ação rígida de movimento entre a sua localização atual e o seu objetivo final. No contexto em que este robô se insere, vários aspetos comportamentais são levados em consideração.

De seguida, na Figura 5.1, são apresentados três testes que demonstram a forma como o robô atua em diferentes circunstâncias[1].

No teste 1 é realizado um ensaio para a situação em que o robô deve manter uma distância de segurança em relação ao seu utilizador em diferentes circunstâncias.

No teste 2 é avaliado o caso do utilizador violar a distância de segurança por iniciativa própria. Nessa situação, o robô apenas controla a sua orientação, rodando em torno de si próprio. O robô mantém-se, assim, sempre direcionado para o utilizador tendo-o no seu campo de visão e mantém, consequentemente, o seu seguimento.

Por último, no teste 3, analisa-se a variação de velocidade do robô a desempenhar a tarefa de seguimento em função da proximidade do utilizador. Apenas de referir que o robô encontra-se limitado a uma velocidade máxima de segurança e, portanto, o utilizador deve ter em atenção a sua velocidade de locomoção.

Para concluir, estas três situações - em que o robô precisa ter definido a sua forma de agir - são fundamentais para manter um funcionamento mais estável e robusto. São também

[1] É possível visualizar um vídeo ilustrativo destas três situações através do URL: https://www.youtube.com/watch?v=1wGg9ePPVgs

Figura 5.1: Testes realizados a aspetos comportamentais do robô.

essenciais para manter o equilíbrio entre as necessidades que o robô exige para manter o humano no seu campo de visão e uma aceitação, na forma como se comporta e atua, por parte das pessoas que partilham o mesmo espaço em que este se encontra.

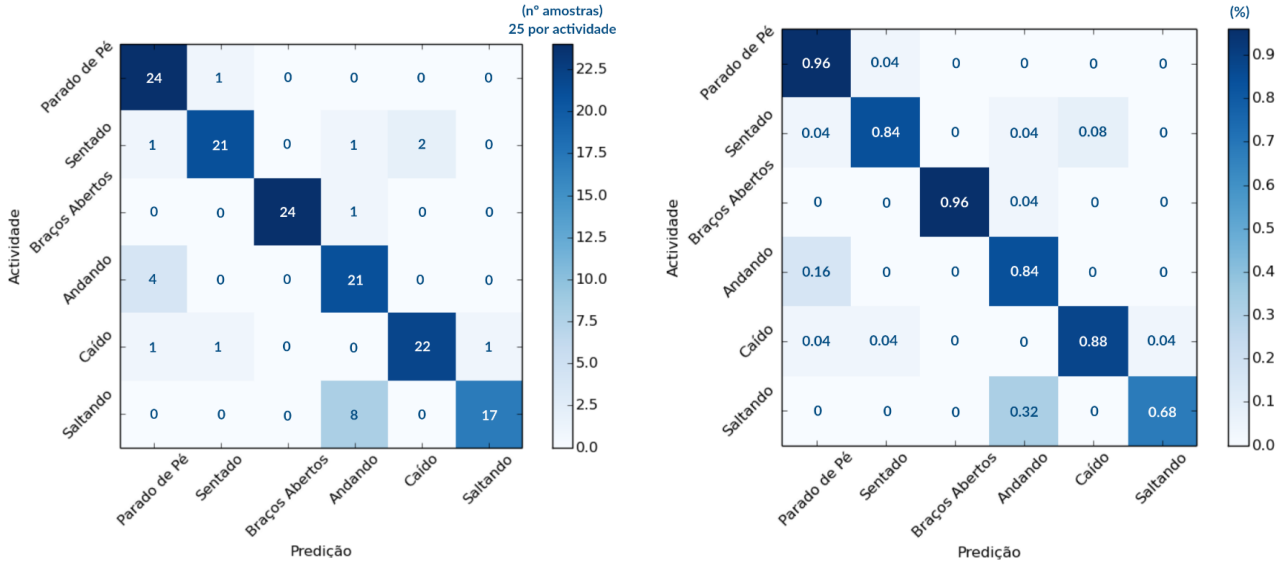
5.2 Desempenho do Sistema de Classificação com o Sensor Estático

Os resultados que vão ser apresentados nesta secção surgem de testes de laboratório do sistema de classificação, com independência do restante sistema implementado neste trabalho. Desta forma, é possível trabalhar com maior certeza numa fase inicial e adquirir resultados que não sofrem com a alta dinâmica de movimento revelada pelo sistema como um todo.

Assim, é apresentado um sistema de classificação independente com a possibilidade de ser reutilizado no âmbito de outros projetos futuros na área de interação humano-robô, sendo a sua eficiência corroborada como apresentado de seguida. Na secção seguinte será testado o sistema de classificação a operar simultaneamente com a ação de seguimento de um humano. Nele será verificada a influência da dinâmica do robô no sistema de classificação. Os dados de treino dos algoritmos são obtidos em ambiente de laboratório, em condições de operação consideradas ótimas (ver secção 3.5). Para a constituição dos dados de teste são realizadas

um conjunto de amostras baseado em 5 pessoas com diferentes alturas e estaturas, por forma a obter resultados que apresentem robustez relativamente à constituição física do utilizador. Os testes aqui apresentados são, desta forma, baseados em 150 amostras distribuídas por 5 pessoas, sendo que 4 delas não conheciam previamente nem nunca tinham tido contacto com o sistema.

Para a representação de resultados são utilizadas matrizes de confusão. Uma matriz de confusão é uma boa forma de classificar verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN). A existência de mais do que duas classes torna intuitiva a compreensão dos resultados de classificação.



(a) Matriz de confusão.

(b) Matriz de confusão normalizada.

Figura 5.2: Resultados do sistema de classificação representados em matrizes de confusão.

Na Figura 5.2 são apresentados os resultados do sistema de classificação de atividades humanas. Este foi treinado para distinguir as diferentes atividades em tempo real.

Na diagonal das matrizes de confusão encontram-se os verdadeiros positivos, ou seja, reportando à Figura 5.2b é a percentagem de acerto que o sistema revelou depois de obtidas as amostras. Os resultados revelam que existe em geral uma boa probabilidade de reconhecer uma atividade corretamente[1]. Deve-se salientar que o reconhecimento da atividade de saltar revelou-se a mais crítico, levando o sistema a apresentar constantes falsos positivos pela dificuldade em obter dados com características distintivas desta ação do humano, que tem uma dinâmica superior às outras ações que se pretenderam classificar.

Quando se procurava perceber como analisar este tipo de sistema de classificação com múltiplas classes, surgiu o problema ao tentar utilizar, para análise, a métrica *Receiver Operating Characteristic* (ROC) [20]. Esta métrica é frequentemente utilizada para avaliar a qualidade de um sistema de classificação. Geralmente, este sistema é utilizado para problemas binários por definição.

[1]É possível visualizar um vídeo ilustrativo do sistema de classificação de actividades humanas através do URL: https://www.youtube.com/watch?v=Dcj6DNCfIk4

Para adaptação, a solução passa por estender a análise que era anteriormente feita de um sistema binário para um sistema de múltiplas classes. Para isso é necessário binarizar a saída uma vez que está construída uma matriz de confusão. Através dos resultados obtidos, todos os dados que são necessários para calcular todos os parâmetros de análise (precisão, exatidão, sensibilidade, especificidade e *F-score*) relativos a cada classe. Através da análise individual de cada classe é possível obter resultados do sistema global.

Como fundamento para os cálculos e informação apresentados na Tabela 5.1, destacam-se as referências [20] e [51], por serem bastante conceituadas, por disponibilizarem a metodologia de cálculo aqui apresentada e ainda por comprovarem os resultados obtidos. Estas mesmas referências servirão também para o leitor consultar o significado de alguns dos parâmetros apresentados, se necessário.

Parâmetro	Parado de Pé	Sentado	Braços Abertos	Andando	Caído	Saltando
TP	24	21	24	21	22	17
TN	105	108	105	108	107	112
FP	6	2	0	10	2	1
FN	1	4	1	4	3	8
Exatidão=(TP+TN)/(TP+FP+TN+FN)	94,85%	95,56%	99,23%	90,21%	96,27%	93,48%
Precisão=TP / (TP + FP)	80,00%	91,30%	100,00%	67,74%	91,67%	94,44%
Sensividade = TP / (TP + FN)	96,00%	84,00%	96,00%	84,00%	88,00%	68,00%
Especifidade = TN / (FP + TN)	94,59%	98,18%	100,00%	91,53%	98,17%	99,12%
F-score = 2*TP /(2*TP + FP + FN)	87,27%	87,50%	97,96%	75,00%	89,80%	79,07%

Tabela 5.1: Análise individual das classes constituintes do sistema de classificação.

A Tabela 5.1 apresenta os resultados obtidos para cada classe, revelando resultados de uma análise mais detalhada. Estes dados estão na base dos resultados finais do sistema de classificação de atividades humanas.

Deste modo, a avaliação final deste sistema é baseada em algumas suposições realizadas para sistemas de múltiplas classes em que os TP e TN possuem o mesmo valor e são o somatório da diagonal da matriz de confusão, $TP = TN = 129$. Por coincidência, o nº de FP e FN também é o mesmo, $FP = FN = 21$.

Assim, baseado no mesmo processo de cálculo dos parâmetros de análise ROC presentes na Tabela 5.1, o sistema de classificação de atividades humanas apresenta uma exatidão de 86.00%, uma sensibilidade de 86.00%, uma especificidade de 86.00% e um F-score de 86.00%. A razão da existência de resultados iguais para todos os parâmetros deve-se ao facto de, por coincidência, $FP = FN$.

Para concluir, verifica-se que a diferente estatura das pessoas envolvidas nos testes não é um fator relevante para o desempenho do sistema de classificação uma vez que todos os dados envolvidos no sistema de classificação encontram-se normalizados e têm como referência o torso da cada pessoa, minimizando as diferenças entre as estaturas das pessoas analisadas.

5.3 Desempenho do Sistema Integrado

Com a apresentação desta secção pretende-se analisar o desempenho de todas as componentes do sistema apresentado em operação. O interesse em avaliar o seu desempenho surge

pela necessidade de averiguar se a dinâmica do robô, causada pelo seu constante movimento, influencia o desempenho das diferentes componentes que o constituem.

Deste modo, são apresentados um conjunto de três novas experiências em diferentes percursos, avaliando, desta vez, típicos casos em que o robô frequentemente pode operar. De seguida, uma nova avaliação do sistema de classificação de atividades humanas é realizada com a finalidade de verificar se este sofre com o movimento do robô e com as adaptações necessárias para o incluir no sistema completo.

5.3.1 Seguimento de um Humano pelo Robô

Um conjunto de três experiências foram realizadas a fim de validar a eficiência da ação de seguimento de um humano pelo robô. As experiências ilustradas na Figura 5.3 apresentam três situações comuns de operação do robô. Cada situação está representada por uma série de quatro imagens que representam a sequência de movimentos da experiência.

A experiência 1 visa verificar a eficiência da função de seguimento a passar por cantos. É uma situação que surge com muita frequência em operação. A influência dos obstáculos no robô tem que ser comedida uma vez que movimentos bruscos do robô podem resultar na perda do utilizador pelo campo de visão da câmara. Considerando este aspeto, o robô apenas sofre influêcia de um obstáculo, muito suavemente, evitando mudanças bruscas de direção.

Um dos maiores problemas com a aplicação da teoria de campos de potencial, é a ocorrência de *mínimos locais* durante a navegação. Pela natureza da aplicação, este é um factor que não carece de atenção, pois, durante a realização do trabalho nunca ocorreram. A proximidade da pessoa inerente à natureza da navegação realizada pelo sistema que se pretendia desenvolver evita a sua ocorrência.

Na experiência 2 foi testada a atividade de seguimento ao percorrer um corredor. Esta experiência, apesar de ser também um caso muito frequente em qualquer localização em que o robô se encontre a operar, é provavelmente a tarefa mais simples a desempenhar pelo robô. Como tal, o seu desempenho foi ótimo e não apresentou qualquer nota negativa de registo.

Por fim, a experiência 3 relata o desempenho de um circuito criado em laboratório, realizando um teste de percurso. Como esta experiência é uma representação de todas as situações comportamentais vistas na secção 5.1 e das duas experiências relatadas anteriormente, é a experiência realizada mais completa[1]. O resultado foi muito positivo, não obstante dos problemas de robustez encontrados, que têm origem no pacote *OpenNI Tracker*.

É importante referir que o pacote de navegação *move_base* foi utilizado também em todas as experiências de forma a verificar se o desempenho do robô melhorava em relação ao modelo implementado de campos de potencial. É perfeitamente possível fazer uso desse recurso, contudo duas notas negativas são de registar. A primeira prende-se com o facto de ser necessário enviar novos objetivos para o pacote de navegação que, por sua vez, tem a

[1] É possível visualizar um vídeo ilustrativo do sistema integrado no URL: https://www.youtube.com/watch?v=D90CCMUF0wA

Figura 5.3: Ilustração das experiências realizadas com o sistema integrado.

necessidade de recalcular uma nova trajetória para o novo objetivo, causando instabilidade na navegação do robô e tornando-o pouco "amigável". Isto implica também uma maior carga computacional associada ao cálculo recorrente de novas trajetórias. A segunda nota de relevância deve-se à detecção de obstáculos que, com este pacote, sofre muita variação nos movimentos de componente rotacional levando a problemas de seguimento do utilizador pela sua elevada variação.

Problemas associados à luz solar, mesmo em ambientes interiores, também foram alvo de análise. Este aspecto revela influência na navegação do robô pelo facto de esta utilizar dados providos pela perceção da pessoa pela câmara Kinect que, devido ao seu princípio de funcionamento envolvendo a projeção de um padrão de luz infravermelha, sofre com a presença de luz solar no espaço de trabalho.

Na Figura 5.4 e 5.5 é possível verificar algumas amostras de imagens de profundidade retiradas de contextos em que se detetavam anomalias no seguimento de uma pessoa. Antes de se realizar uma análise deve-se salientar o facto de que neste tipo de imagem, um pixel branco representa um ponto distante e um pixel preto representa um ponto da imagem muito próximo. Seguindo este raciocínio, ao olhar para as imagens das Figuras 5.4 e 5.5 é possível observar que na presença de luz solar os pontos distantes ficam pretos, a presença de vidros no ambiente, que pela sua natureza translúcida não refletem a luz infravermelha projectada pela Kinect, inviabilizam a medição da profundidade dos píxeis associados às superfícies vidradas. Deste modo deparam-nos com a perda de dados por estes dois factores que não

Figura 5.4: Imagem da entrada do ISR com um dia de sol intenso.

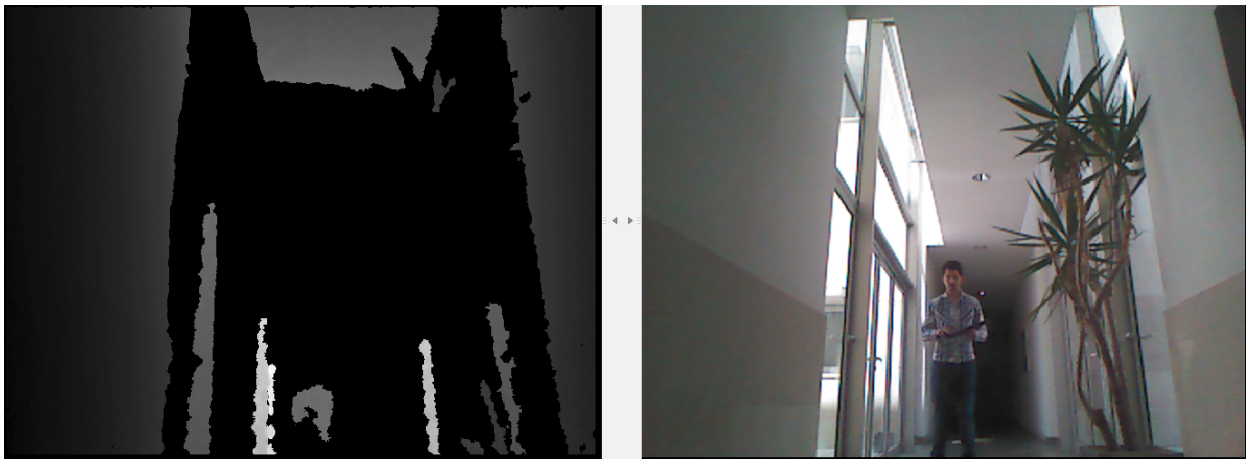


Figura 5.5: Imagem de um corredor do ISR com um pórtico de iluminação natural.

permitem a correta operação do sensor.

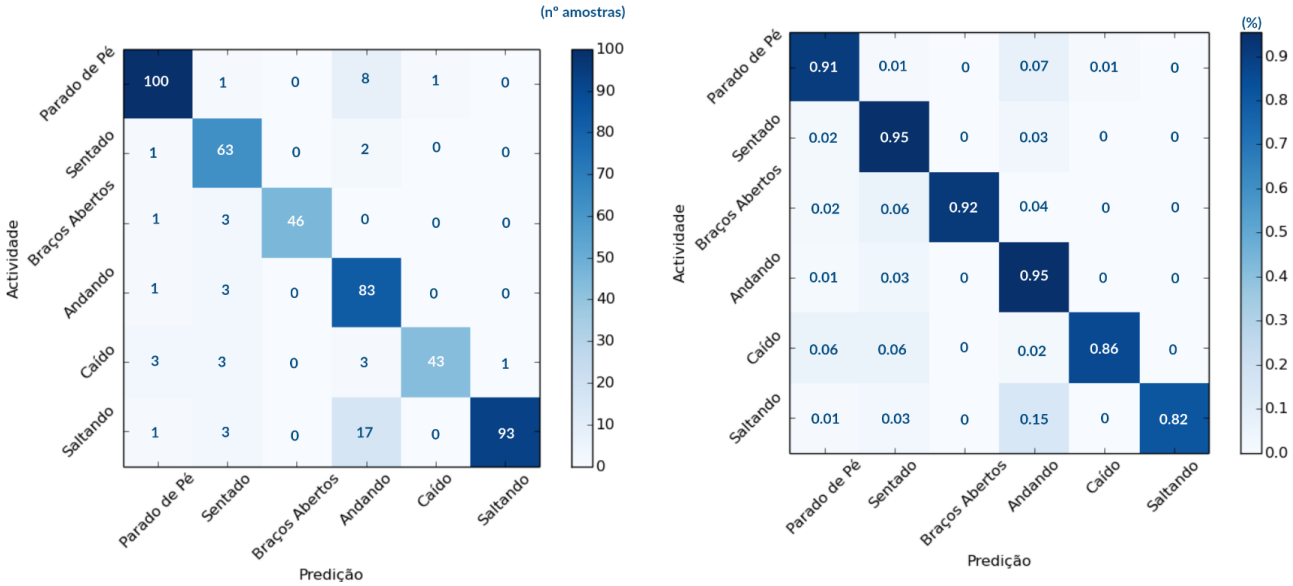
Para concluir, o comportamento de seguimento do humano por parte do robô móvel apresenta um desempenho aceitável. O maior problema verificado prende-se com questões de robustez identificados no pacote de deteção do esqueleto humano. Este problema surge pelo facto do pacote ter sido dimensionado para um ambiente estático. No caso deste projeto, o ambiente de operação é altamente dinâmico e a tarefa de seguimento é um pouco afetada por movimentos de rotação realizados pelo robô. Esta situação não retira funcionalidade ao sistema pois é perfeitamente possível operar com este pacote, contudo está-se sujeito a uma probabilidade maior de se perder o humano a ser seguido, sobretudo quando o humano realiza manobras de mudança brusca de direção (*e.g.* ao dobrar uma esquina de dois corredores ortogonais entre si).

5.3.2 Desempenho do Sistema de Classificação com o Sensor em Movimento

Depois de analisado o sistema de classificação separadamente das restantes componentes do projecto (ver secção 5.2), de forma a validar o seu desempenho para qualquer aplicação

onde se deseje implementar, surge a necessidade de realizar uma nova análise, desta vez com o sensor Kinect em movimento, a bordo do robô móvel durante a execução da tarefa de seguimento de um humano.

A necessidade de uma nova avaliação resulta de dois fatores principais. O primeiro deve-se ao facto do sistema de classificação operar sobre uma grande dinâmica de movimento do robô que, em oposição à validação realizada na secção 5.2, pode ou não sofrer com o movimento do robô. O segundo fator visa avaliar algumas alterações na utilização dos resultados fornecidos pelo sistema de classificação, no sentido de tornar o sistema menos sensível a eventuais erros. Como o sistema de classificação fornece um resultado a cada período de operação (100 ms), o sistema está sujeito a alterações comportamentais muito frequentes devido a falhas de classificação, a erros na aquisição de dados ou flutuações constantes nos resultados devido a transições entre atividades. Com vista a solucionar a situação, utiliza-se um processo de votação, obtendo desta forma, um resultado a cada dois segundos, ou a cada 20 períodos, com base nos 20 últimos resultados fornecidos pelo sistema de classificação. Em comparação a outros sistemas de classificação [19] implementados em experiências reais, geralmente baseadas nos últimos segundos de atividade, no caso do projecto mencionado esse valor é de 5 s, não é um retrocesso no tempo de produção de um resultado. Esta é uma prática que visa estabilizar o sistema, e o tempo utilizado nesse processo é um investimento na produção de um resultado mais assertivo e robusto.



(a) Matriz de confusão.

(b) Matriz de confusão normalizada.

Figura 5.6: Matrizes de confusão do sistema de classificação quando opera no sistema integrado para seguimento de um humano por um robô.

Assim, segundo os novos pressupostos apresentados, é realizada uma nova análise sob as mesmas circunstâncias da análise anterior (ver secção 5.2). No entanto, agora são tidos em consideração os dois fatores mencionados que certamente exercem grande influência nos resultados finais, como se pode verificar em seguida.

Os resultados apresentados na Figura 5.6, através de matrizes de confusão, são basea-

dos em 481 amostras realizadas em ambiente de laboratório sob as alterações mencionadas anteriormente. O número de amostras de cada atividade foi obtido indiscriminadamente garantindo apenas que cada atividade tinha um conjunto de amostras com no mínimo 50 unidades.

Este modelo de análise possui uma dimensão amostral superior devido ao facto desta avaliação estar acoplada ao sistema global apresentado neste projeto e, como tal, merece um processo de testes mais intensivo.

Parâmetro	Parado de Pé	Sentado	Braços Abertos	Andando	Caído	Saltando
TP	100	63	46	83	43	93
TN	328	365	382	345	385	335
FP	7	13	0	28	1	0
FN	10	3	4	4	7	21
Exatidão=(TP+TN)/(TP+FP+TN+FN)	96,18%	96,40%	99,07%	93,04%	98,17%	95,32%
Precisão=TP / (TP + FP)	93,46%	82,89%	100,00%	74,77%	97,73%	100,00%
Sensividade = TP / (TP + FN)	90,91%	95,45%	92,00%	95,40%	86,00%	81,58%
Especifidade = TN / (FP + TN)	97,91%	96,56%	100,00%	92,49%	99,74%	100,00%
F-score = 2*TP /(2*TP + FP + FN)	92,17%	88,73%	95,83%	83,84%	91,49%	89,86%

Tabela 5.2: Análise individual das classes constituintes do sistema de classificação sujeito a novas condições.

A Tabela 5.2 apresenta os resultados obtidos para cada classe, revelando resultados de uma análise mais detalhada. Esta análise segue exatamente as mesmas suposições realizadas anteriormente em que os TP e TN possuem o mesmo valor e são o somatório da diagonal da matriz de confusão, $TP = TN = 428$. O nº de $FP = 49$ e de $FN = 48$.

Assim, baseado no mesmo processo de cálculo dos parâmetros de análise ROC presentes na Tabela 5.1, o sistema de classificação de atividades humanas nas novas circunstâncias já mencionadas anteriormente nesta secção, apresenta uma exatidão de 89.73%, precisão de 89.73%, sensibilidade de 89.92%, especificidade de 89.73% e um F-score de 89.82%.

Para concluir a análise, é possível verificar que, em geral, a percentagem de sucesso na classificação melhorou. A primeira ilação a retirar diz respeito ao primeiro fator mencionado nesta secção. Todo o sistema implementado opera relacionando todas as suas componentes a um mapa comum. Este mapa serve de referência a todos os elementos do sistema e, portanto, todos os dados obtidos que constituem as características processadas têm como referência o mesmo sistema de coordenadas. Como tal, as medidas realizadas, seja com o robô/câmara estática ou em movimento, estão mencionadas no mesmo mapa de referência do sistema. Deste modo verificou-se que a aquisição de dados pela câmara Kinect, relativas ao esqueleto humano, não sofre influência pelo deslocamento do robô, logo não se verificou perda de desempenho devido a este fator.

Relativamente ao segundo fator que visava estabilizar o resultado obtido à saída do sistema de classificação através de algumas modificações, verificou-se que trouxe melhorias em termos de desempenho. A ocorrência de erros de classificação, erros na aquisição de dados e flutuações do resultado na transição de atividades deixaram de exercer uma influência significativa no resultado de classificação final pois, como já referido, a atribuição de um resultado de classificação sofreu algumas modificações de adaptação a este sistema.

5.4 Sumário

Neste capítulo foi apresentada uma validação experimental do sistema, primeiramente através de uma abordagem específica que serviu para verificar alguns componentes do sistema de forma individual, e por fim avaliou-se o desempenho do sistema a operar fazendo uso de todos os seus recursos.

As experiências realizadas permitiram validar com sucesso o sistema e o objetivo principal — o desenvolvimento de um robô móvel com a capacidade de seguir um humano — foi concretizado.

6 Conclusão

Com o desenvolvimento deste trabalho que ambicionava o desenvolvimento de um robô seguidor com uma capacidade de perceção elevada, alcançaram-se algumas expectativas que eram propostas desde o início do projeto. No entanto, também novas metas foram sendo definidas e alcançadas ao longo do seu desenvolvimento. Alcançou-se assim o objetivo de desenvolver um robô seguidor que contemplasse múltiplos fatores comportamentais e com uma capacidade de perceção mais elevada através do sistema de classificação de atividades humanas. A sua combinação permitiu obter resultados positivos e abriu uma nova janela de oportunidades para mais e melhor desenvolvimento nesta área.

Mesmo com uma ideia base bem estruturada desde a idealização deste trabalho, sempre esteve sujeito a melhorias e a novos desafios. Após a ideia estar bem definida e delineada, o trabalho resumiu-se a três componentes fundamentais: sensorização, classificação e navegação. Estes três elementos estão na base do trabalho, dos objetivos e da estrutura de implementação do sistema.

O sistema de navegação personalizado, desenvolvido de forma a satisfazer as necessidades da aplicação, revelou um desempenho muito positivo e possibilita a adaptação de mais aspetos relacionados com o comportamento que se deseje que o robô tenha assim como outras características a nível de navegação. A ação do robô é, assim, um processo suave, que reage a diferentes circunstâncias de diferentes modos, apresentando um comportamento socialmente aceitável pelas pessoas que partilham o mesmo espaço que o robô.

O sistema de classificação de atividades humanas foi inicialmente desenvolvido separadamente das restantes funções do sistema. Desta forma, foi validado um sistema de classificação de atividades humanas com resultados corroborados experimentalmente. A vantagem desta análise acentua-se pelo facto de que este módulo pode ser facilmente desacoplado das restantes partes e adaptado no âmbito de qualquer outro projeto sem necessidade de uma nova e exaustiva fase de testes.

Com o sistema a operar na totalidade das suas funções, algumas características evidenciaram-se, nomeadamente o facto de nenhuma das suas componentes (sensorização, classificação e navegação) afetar direta e indesejavelmente as outras componentes do sistema. Este era um facto desconhecido e que só com a realização de experiências e testes foi possível observar.

O processo de integração das componentes mencionadas revelou-se simples e eficiente precisamente devido ao facto de não se terem verificado interferências importantes entre os diferentes módulos. Apenas a necessidade de comunicação entre as componentes mereceu

especial atenção no processo de integração.

6.1 Trabalho Futuro

Ao seguir este caminho algumas expetativas foram surgindo, não só expetativas pessoais como expetativas relativas à comunidade científica em que este trabalho está inserido. Assim, é de referir que as principais contribuições deste trabalho passaram pelo desenvolvimento de um robô móvel capaz de realizar a tarefa de seguimento de um humano, o desenvolvimento de um pacote de navegação desenvolvido de raiz, a implementação de um sistema de classificação capaz de realizar a tarefa de reconhecimento de atividades humanas e, por fim, a criação de uma aplicação base destinada ao desenvolvimento de estudos em HRI.

Apesar dos bons resultados apresentados no Capítulo 5, o sistema pode ser sujeito a melhorias. A melhoria mais evidente é o desenvolvimento de um pacote que realize a deteção de uma pessoa baseada noutras técnicas, substituindo o pacote *OpenNI tracker* utilizado para realizar essa tarefa e anulando os problemas encontrados na presença de iluminação natural. O desenvolvimento deste pacote traria também a vantagem de ter um sistema completamente desenvolvido de raiz, lebra-se que o pacote de deteção da pessoa é o único que não foi desenvolvido pelo autor desta dissertação.

Finalmente, existe uma grande abertura de oportunidades para entrar na área de desenvolvimento HRI. É uma área científica em desenvolvimento, atual e que ainda peca pela escassez do seu desenvolvimento. Outra vertente importante é também a possibilidade de desenvolver mais investigação em HRI para equipas de robôs comandadas de forma mais intuitiva por um utilizador.

7 Bibliografia

[1] Skeletal Tracking. https://msdn.microsoft.com/en-us/library/hh973074.aspx. Accessed: 2016-06-13.

[2] Service robots: The next big productivity platform. http://www.pwc.com/us/en.html. Accessed: 2016-06-02.

[3] Kinect for Windows Sensor Components and Specifications. https://msdn.microsoft.com/en-us/library/jj131033.aspx, . Accessed: 2016-06-07.

[4] Xbox 360 Kinect Teardown. https://www.ifixit.com/Teardown/Microsoft-Kinect-Teardown/4066/1, . Accessed: 2016-06-07.

[5] ALAN, İlişkilerde Mesafeniz Kaç Santimetre? http://pozitifid.blogspot.pt/2015/06/alan-iliskilerde-mesafeniz-kac.html. Accessed: 2016-06-13.

[6] BVH Motion Capture Guide. http://www.kinecthacks.com/guides/bvh-motion-capture-guide/. Accessed: 2016-06-07.

[7] Openni Tracker Documentation. http://www2.technologyreview.com/tr50/primesense/. Accessed: 2016-06-07.

[8] Aggarwal, Jake K and Ryoo, Michael S. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

[9] Aggarwal, Jake K and Xia, Lu. Human activity recognition from 3d data: A review. *Pattern Recognition Letters*, 48:70–80, 2014.

[10] Andoni, Alexandr and Indyk, Piotr. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 459–468. IEEE, 2006.

[11] Calvo, Roberto and Cañas, José María and García-Pérez, Lía. Person following behavior generated with jde schema hierarchy. In *ICINCO*, pages 463–466, 2005.

[12] Chang, Chih-Chung and Lin, Chih-Jen. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[13] Chen, Wei and Xiong, Caiming and Xu, Ran and Corso, Jason. Actionness ranking with lattice conditional ordinal random fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 748–755, 2014.

[14] Chen, Zhichao and Birchfield, Stanley T. Person Following With a Mobile Robot Using Binocular Feature-Based Tracking. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 815–820. IEEE, 2007.

[15] Cilimkovic, Mirza. Neural Networks and Back Propagation Algorithm. *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin*, 15, 2008.

[16] Cortes, Corinna and Vapnik, Vladimir. Support-Vector Networks. *Machine learning*, 20(3):273–297, 1995.

[17] De Moivre, Abraham. The Doctrine of Chances. In *Annotated Readings in the History of Statistics*, pages 32–36. Springer, 2001.

[18] Fan, Rong-En and Chang, Kai-Wei and Hsieh, Cho-Jui and Wang, Xiang-Rui and Lin, Chih-Jen. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[19] Faria, Diego R and Vieira, Mario and Premebida, Cristiano and Nunes, Urbano. Probabilistic human daily activity recognition towards robot-assisted living. In *Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on*, pages 582–587. IEEE, 2015.

[20] Fawcett, Tom. An introduction to ROC analysis. *Pattern recognition letters*, 27(8): 861–874, 2006.

[21] Field, Tim. Openni tracker ROS package (groovy), 2013.

[22] Fisher, Ronald Aylmer. *Statistical methods for research workers*. Genesis Publishing Pvt Ltd, 1925.

[23] Fong, Terrence and Nourbakhsh, Illah and Dautenhahn, Kerstin. A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3):143–166, 2003.

[24] Gauvain, Jean-Luc and Lee, Chin-Hui. Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. *Speech and Audio processing, ieee transactions on*, 2(2):291–298, 1994.

[25] Gavrila, Dariu M and Davis, Larry S. 3-D model-based tracking of humans in action: a multi-view approach. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 73–80. IEEE, 1996.

[26] Gockley, Rachel and Forlizzi, Jodi and Simmons, Reid. Natural person-following behavior for social robots. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 17–24. ACM, Association for Computing Machinery, 2007.

[27] Goldberg, David E and Holland, John H. Genetic Algorithms and Machine Learning. *Machine Learning*, 3(2):95–99, 1988.

[28] Guha, Tanaya and Ward, Rabab Kreidieh. Learning sparse representations for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(8):1576–1588, 2012.

[29] Har-Peled, Sariel and Indyk, Piotr and Motwani, Rajeev. Approximate Nearest Neighbor: Towards Removing the Curse of Dimensionality. *Theory of computing*, 8(1):321–350, 2012.

[30] Hellström, Thomas. Robot navigation with potential fields. *Department of Computing Science, Umea University, Tech. Rep*, 2011.

[31] IBBT, ESAT-PSI. Does Human Action Recognition Benefit from Pose Estimation?

[32] Ilias, B and Shukor, SA Abdul and Yaacob, S and Adom, AH and Razali, MH Mohd. A Nurse Following Robot with High Speed Kinect Sensor. *ARPN Journal of Engineering and Applied Sciences*, 9(12):2454–2459, 2014.

[33] Kotsiantis, Sotiris B and Zaharakis, I and Pintelas, P. Supervised Machine Learning: A Review of Classification Techniques, 2007.

[34] LeCun, Yann. Une procedure d'apprentissage pour reseau a seuil asymmetrique (a Learning Scheme for Asymmetric Threshold Networks). 1985.

[35] Li, Shengren and Amenta, Nina. Brute-Force k-Nearest Neighbors Search on the GPU. In *Similarity Search and Applications*, pages 259–270. Springer, 2015.

[36] Moeslund, Thomas B and Granum, Erik. A survey of computer vision-based human motion capture. *Computer vision and image understanding*, 81(3):231–268, 2001.

[37] Moeslund, Thomas B and Hilton, Adrian and Krüger, Volker. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2):90–126, 2006.

[38] Niebles, Juan Carlos and Wang, Hongcheng and Fei-Fei, Li. Unsupervised Learning of Human Action Categories using Spatial-Temporal Words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[39] Nina Brown. Proxemic Theory, 1966. In *Proxemic Theory by Nina Brown*. 2001.

[40] Parker, David B. Optimal algorithms for adaptive networks: Second order back propagation, second order direct propagation, and second order Hebbian learning. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 2, pages 593–600, 1987.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python.

[42] Poppe, Ronald. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[43] Quigley, Morgan and Conley, Ken and Gerkey, Brian and Faust, Josh and Foote, Tully and Leibs, Jeremy and Wheeler, Rob and Ng, Andrew Y. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[44] Rish and Irina. An Empirical Study of the Naive Bayes Classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, volume 3, pages 41–46. IBM New York, 2001.

[45] Rosenblatt, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[46] Satake, Junji and Chiba, Masaya and Miura, Jun. A SIFT-based person identification using a distance-dependent appearance model for a person following robot. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 962–967. IEEE, 2012.

[47] Satake, Junji and Miura, Jun. Robust stereo-based person detection and tracking for a person following robot. In *ICRA Workshop on People Detection and Tracking*, pages 1–10, 2009.

[48] Satake, Satoru and Kanda, Takayuki and Glas, Dylan F and Imai, Michita and Ishiguro, Hiroshi and Hagita, Norihiro. How to approach humans? Strategies for social robots to initiate interaction. In *Human-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on*, pages 109–116. IEEE, 2009.

[49] Sense, Prime. The primesensor tm reference design 1.08. *Prime Sense*, 8, 2011.

[50] Shaker, Samir and Saade, Jean J. and Asmar, Daniel. Fuzzy Inference-Based Person-Following Robot. *International Journal of Systems Applications, Engineering and Development*, 2(1):29–34, 2008.

[51] Tsoumakas, Grigorios and Katakis, Ioannis and Vlahavas, Ioannis. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2009.

[52] Turaga, Pavan and Chellappa, Rama and Subrahmanian, Venkatramana S and Udrea, Octavian. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.

[53] Wang, Jiang and Liu, Zicheng and Wu, Ying and Yuan, Junsong. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1290–1297. IEEE, 2012.

[54] Wang, Liang and Tan, Tieniu and Ning, Huazhong and Hu, Weiming. Silhouette analysis-based gait recognition for human identification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(12):1505–1518, 2003.

[55] Williams, DE Rumelhart GE Hinton RJ and Hinton, GE. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[56] Ye, Mao and Zhang, Qing and Wang, Liang and Zhu, Jiejie and Yang, Ruigang and Gall, Juergen. A survey on human motion analysis from depth data. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 149–187. Springer, 2013.

[57] Yun, Kiwon and Honorio, Jean and Chattopadhyay, Debaleena and Berg, Tamara L and Samaras, Dimitris. Two-person interaction detection using body-pose features and multiple instance learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 28–35. IEEE, 2012.

[58] Zhang and Harry. The Optimality of Naive Bayes. *American Association for Artificial Intelligence (www.aaai.org)*, 1(2):3, 2004.

Apêndice A

Anexos

Figura A.1: Microsoft Kinect, imagem extraída de [4].

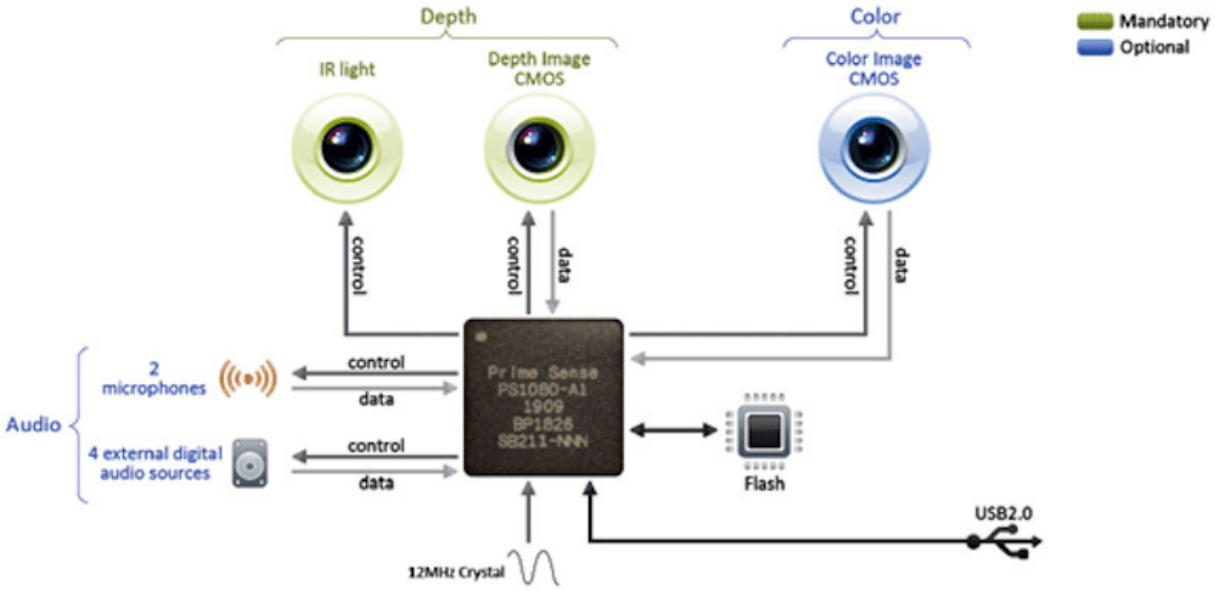


Figura A.2: Hardware da Microsoft Kinect, imagem extraída de [4].

Figura A.3: Resumo dos elementos constituintes da plataforma.

Apêndice B

Guia de Procedimentos

ROBOTFOLLOWER – PERSON FOLLOWING BEHAVIOR BY A COMPANION ROBOT

Diogo Amorim's M.Sc. Project

Figure 1. Available platform to run this project.

SYNOPSIS

This work presents a system implemented in Robot Operating System (ROS) and integrates a large capacity of perception which is based on tracking systems of human's skeleton and classication of their activities, integrating this information into a software layer that deals with behavioral aspects of the robot. This results in a completely customized navigation system, according to the required needs. The robot is composed by a few components that enable the system to perform the aforementioned characteristics.

RESEARCH TOPIC AND KEYWORDS

(Research topic: Robot follower, Perception, Human Activity Recognition, Classification, Navigation)

DURATION:

As long as we wish that the robot follow us.

CONTACT PERSONS

- Diogo Amorim (diamorim@isr.uc.pt)
- Rui P. Rocha (rprocha@isr.uc.pt)

REQUIREMENTS

- Hardware
  - A robotic platform Pioneer P3-DX;
  - A laser range finder;
  - A Microsoft Kinect;
  - A laptop with a good battery performance;

- Software
  - ROS indigo;
  - Drivers for robotic platform and LRF (laser range finder);
  - Install OpenNI drivers, Primesense drivers and the NITE middleware;
  - Install Python 3.5.0;
  - Install the scikit-learn software library;

ASSEMBLY

1. Make sure that you have a laptop with a good battery performance;

2. Set the laptop on the robotic platform;

3. Plug all the components of the platform into the laptop:

   3.1. Turn on the platform;

   3.2. Plug the platform into the computer;

   3.3. Plug the laser LRF into the computer;

   3.4. Plug the Microsoft Kinect into the computer;

LAUCHING THE SYSTEM

To launch the system you need make sure that everything is correctly setting up, and all the components are running well. After that you need follow these steps:

1. Open a terminal, set the correct workspace and run:

```
$ roslaunch mrl_pioneer pioneer_lrf.launch
```

That launch will launch the pioneer robot and the hokuyo node, It will also publish all the transforms relationships of all components not defined in the system (*e.g.* Microsoft Kinect).

2. If you have permissions problems, as I had, please run these commands:

```
$ sudo chmod 666 /dev/ttyACM0
$ sudo chmod 666 /dev/ttyUSB0
```

3. Open a new terminal and launch the gmapping:

```
$ rosrun gmapping slam_gmapping
```

The gmapping will allow the use of a global map of the system.

4. Launch the Microsoft Kinect:

```
$ roslaunch openni_launch openni.launch
```

5. Launch the Tracker node:

```
$ rosrun openni_tracker openni_tracker
```

6. Launch the Naviagtion node:

```
$ rosrun gazebo_random_walk potential_fields
```

7. Launch the Classification node:

```
$ rosrun openni_tracker subscribe_pose.py
```

If everything goes well, you just need stand in front of the robot about 2 seconds, and the robot will start follow you and will proceed to the classification of the activities recognized.