



Daniela Filipa Aires Nobre

RoboCops II - Patrulhamento de Edifícios com Equipas de Robôs Móveis

Grau de Mestre em Engenharia Eletrotécnica e de Computadores

Setembro 2016



UNIVERSIDADE DE COIMBRA



Departamento de Engenharia Eletrotécnica e de Computadores
Faculdade de Ciências e Tecnologia
Universidade de Coimbra

RoboCops II - Patrulhamento de Edifícios com Equipas de Robôs Móveis

Daniela Filipa Aires Nobre

Coimbra, Setembro de 2016



RoboCops II - Patrulhamento de Edifícios com Equipas de Robôs Móveis

Daniela Filipa Aires Nobre

Orientador:

Prof. Doutor Rui Paulo Pinto da Rocha

Júri:

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

Prof. Doutor Jorge Nuno de Almeida e Sousa Almada Lobo

Prof. Doutor Rui Paulo Pinto da Rocha

Dissertação submetida como requisito parcial para obtenção do grau de Mestre em Engenharia
Eletrotécnica e de Computadores

Coimbra, Setembro de 2016

AGRADECIMENTOS

No dia em que escolhi fazer esta dissertação estava ciente de que estava a embarcar numa grande aventura da qual iria sair com uma grande bagagem de conhecimento e experiência.

Este trabalho foi o culminar de um percurso académico individual, mas não solitário, e com esta fantástica experiência surgiram grandes amizades, partilhas de conhecimento, entreaajuda e companheirismo. Durante esta fase pude contar com pessoas incríveis, que me apoiaram, ajudaram, não me deixaram desistir nem baixar os braços, mesmo nas fases mais complicadas, e com as quais pude discutir diversos assuntos que me ajudaram a progredir quer profissionalmente, quer pessoalmente.

Neste pequeno texto quero agradecer a todas as pessoas que fizeram parte deste meu percurso, e sem as quais nada disto teria sido possível.

Em primeiro lugar quero agradecer ao meu orientador Prof. Rui Rocha, por ter aceite este desafio, por me ter acolhido e ajudado. Sei que este percurso foi atribulado, mas ele nunca me deixou baixar os braços nem desistir e tinha sempre uma palavra de encorajamento para me dar. Pude contar com a sua sabedoria e experiência para me guiar ao longo deste percurso e me manter no caminho certo, sei que sem ele esse caminho teria sido muito mais sinuoso e difícil de percorrer. Interveio nas alturas certas e sempre se mostrou disponível em todas as situações. O meu muito obrigada por tudo!

Em segundo lugar quero agradecer à melhor mãe do mundo Maria Luísa Nobre por ter estado sempre comigo e ser uma fantástica mulher. Ouviu os meus lamentos, partilhou as minhas alegrias, e deu-me sempre muita força para atingir os meus objetivos. Mesmo com todas as provações e vicissitudes que a vida lhe tem dado tem sido uma fonte de exemplo e inspiração e a ela me agarrei muitas vezes.

Agradeço ao meu irmão Pedro Nobre, que é o meu exemplo a seguir, por todas as vezes que me deu na cabeça e me orientou para que me focasse nos meus objetivos e os pudesse atingir.

Ao meu pai, que já não estando entre nós deve estar muito orgulhoso de tudo o que conquistei e a quem em momentos de aflição recorri, na esperança de obter um sinal que me fizesse prosseguir e de quem tenho muitas saudades.

Quero agradecer ao meu namorado Elísio Sousa por me ter aturado nesta etapa. Fui difícil, chata, pessimista, e muitas vezes compliquei o que era fácil. Soube ouvir-me, incentivar-me, ajudar-me e manteve-se comigo sempre que precisei. É o meu pilar e a minha força. Obrigada!

À mãe do meu namorado Ana Paula Sousa por me ter ouvido, aconselhado e animado em muitas alturas, é uma grande mulher e sempre me transmitiu muita força.

Quero agradecer aos meus colegas de laboratório por me terem ouvido, por terem discutido assuntos comigo e me terem proporcionado bons momentos. Um especial e enorme agradecimento ao Gonçalo Martins que foi uma peça chave nesta dissertação. Discutimos muitos assuntos, muitas ideias e o conhecimento e experiência que ele possui foram sem dúvida uma ajuda preciosa e indispensável. Sem ele esta etapa teria sido muito mais difícil. Soube ouvir-me e ajudar-me sempre que precisei e recebeu-me de braços abertos desde o primeiro dia, quando se mostrou logo disponível para ajudar.

Ao meu colega e amigo António Simões que me deu ideias fundamentais para a execução deste trabalho e que sempre esteve disponível para me ajudar, inclusive para tirar fotografias.

Quero agradecer aos meus amigos e colegas que me ajudaram a tirar as fotografias necessárias, é um trabalho demorado e cansativo e eles mostraram-se sempre disponíveis para tal, Fábio Carreira, Marta Batista, Filipa Guia, Miguel Mira, Ângelo Honório e mais uma vez Elísio Sousa e António Simões.

Agradeço ao Instituto de Sistemas e Robótica, mais precisamente ao Laboratório de Robótica Móvel por me ter dado todas as ferramentas necessárias à execução deste trabalho.

Por fim, mas não menos importante, aos meus amigos(as) que estiveram comigo em todos os momentos da minha vida, mas que se revelaram um apoio e um suporte essencial na execução deste trabalho, por todos os momentos de distração que me proporcionaram e por toda a força que me transmitiram.

Este trabalho é meu, mas é sobretudo de todos aqueles que me rodearam durante estes últimos meses. Obrigada!

RESUMO

Cada vez mais os robôs são uma ferramenta importante para auxiliar os seres humanos nas suas tarefas do quotidiano, seja por estas serem perigosas, como testes de colisões de carros e desarmamento de bombas ou por serem monótonas, como vigilância e patrulhamento. No contexto do patrulhamento pode-se encontrar algumas vantagens no uso de robôs móveis ao invés de pessoas ou mesmo de câmaras fixas. Eles não se cansam, não se desconcentram, e caso avariem, todos os locais continuam a ser patrulhados. Além disso, a inserção/remoção de um robô num ambiente é uma tarefa relativamente fácil e sem custos acrescidos.

Este trabalho surge então na tentativa de dar resposta a um dos problemas do patrulhamento com equipas de robôs, a perceção artificial. Para que o patrulhamento feito por robôs seja o mais autónomo possível, é necessário que estes tenham a capacidade de reconhecer a existência de anomalias no ambiente, e se possível, serem capazes de as identificar. Neste trabalho são apresentadas várias técnicas de visão por computador, entre elas o uso do descritor SURF juntamente com a métrica EMD, que têm sido utilizadas para resolver problemas semelhantes e que, em ambientes controlados, podem fornecer resultados interessantes.

Como é de esperar, num sistema deste género existem sempre erros associados à navegação, que impossibilitam os robôs de pararem sempre exatamente na mesma posição e orientação nos locais de patrulhamento. Ao longo da execução deste trabalho verificou-se que algumas técnicas convencionais como a comparação de histogramas, a deteção de arestas ou a técnica anteriormente referida não são invariantes à luminosidade e às diferenças de pontos de vista nos mesmos locais.

A solução apresentada para o problema passa pelo uso de uma técnica de aprendizagem automática. Isto permite que o sistema aprenda que diferenças de iluminação e de pontos de vista não são anomalias e por isso torna-se mais robusto a essas condições. Neste sentido, é usado o descritor HoG para extrair características da imagem, por ser invariante à luminosidade, e posteriormente usa-se SVM para classificar imagens como sendo anómalas ou não.

Esta solução foi inicialmente validada com imagens de teste obtidas em ambiente controlado, adquiridas em diferentes locais, onde apenas se alterava a orientação do robô. Posteriormente foram realizadas experiências utilizando imagens adquiridas numa missão de patrulhamento real.

No final deste trabalho são apresentados os resultados experimentais obtidos e os mesmos são alvo de análise. Por fim, no último capítulo, apresenta-se uma conclusão do trabalho efetuado e possíveis melhorias futuras.

Palavras Chave: Visão por Computador, Aprendizagem Automática, Detecção de Anomalias.

ABSTRACT

Robots have been an increasingly important tool in assisting humans beings in their daily tasks, due to their dangerousness, such as car crash tests and bomb disarming, or simply due to their monotony, such as surveillance and patrolling. In the context of patrolling, some advantages can be found in the use of mobile robots instead of people or even fixed cameras. The robots are tireless and conscientious, individual malfunctions will not impede the patrolling and inserting/removing a robot in a patrolling environment is a relatively easy and cost-effective task, etc.

This work aims to solve one of patrolling's biggest problems when it comes to using a team of mobile robots, which is artificial perception. In order to make the patrolling as autonomous as possible, it is necessary for the robots to have the capability of recognizing the existence of anomalies in the environment and, if possible, be able to identify them. In this work, various computer vision techniques are presented, namely the use of the SURF feature descriptor along with the EMD metric, which have been used to solve similar problems and can provide interesting results when used in controlled environments.

Navigation errors are to be expected in a system of this kind, which makes it impossible for the robots to stop exactly in the same position and orientation at the patrolling places each time. During the execution of this work, some conventional techniques such as histogram comparison, edge detection or the previously mentioned technique have been verified to not be luminosity-invariant and are also greatly affected with the shift in the robot's point of view. The solution presented to this problem is based on a machine learning technique. This allows the system to learn that not all differences in lighting and points of view are anomalies, and because of that it becomes much more robust under these navigation errors. In this sense the HoG descriptor is used to extract the images' features, due to its invariance to luminosity, and then SVM is used to classify the images as anomalous or non-anomalous.

This solution was initially validated with test images obtained under controlled conditions, taken in different places where only the robot's orientation was shifted. Later on, more experiments were conducted using images taken during real patrolling missions.

The results that were obtained using this technique are displayed at the end of this work. These results were analysed and the last chapter presents a conclusion of the all the work that has been done and how the system can possibly be improved in the future.

Keywords: Computer Vision, Machine Learning, Anomaly Detection.

“If you can't fly then run, if you can't run then walk, if you can't walk then crawl, but whatever you do you have to keep moving forward.”

-Martin Luther King Jr.

CONTEÚDO

Agradecimentos	VII
Resumo	IX
Abstract	XI
Lista de Acrónimos.....	XVI
Lista de Figuras	XVII
Lista de Tabelas	XX
1. Introdução.....	1
1.1 Motivação.	1
1.2 Objetivos.....	3
1.3 Estrutura da Dissertação.	3
2. Enquadramento Teórico.....	4
2.1 Videovigilância.....	4
2.2 Sistema Multi-Robô.....	5
2.2.1 Coordenação Multiagente em Patrulhamento	6
2.3 Detecção de Anomalias	7
2.4 Classificação de Objetos.....	10
2.5 <i>Open Source Computer Vision Library</i>	11
2.6 Sumário.....	12
3. Técnicas Baseadas na Definição de um Limiar	13
3.1 Abordagens Para Definir um Limiar	13
3.2 SURF e EMD.....	15
3.3 Comparação de Histogramas	24
3.4 Segmentação Baseada num Grafo	26
3.5 Sumário.....	30
4. Sistema Proposto	31
4.1 Aprendizagem automática	31
4.2 Máquina de Vetores de Suporte	33
4.3 Histograma de Orientação de Gradientes	36
4.4 Visão Global do Sistema	37
4.5 Sumário.....	41

5. Validação Experimental e Análise de Resultados	42
5.1 Experiências Preliminares	42
5.1 Experiências em Situações Reais	45
5.1 Classificação das Anomalias	52
5.1 Sumário.....	54
6. Conclusão	55
6.1 Trabalho Futuro	56
7. Bibliografia	57
8. Anexos	61

LISTA DE ACRÓNIMOS

Acrónimo	Significado em Inglês	Significado em Português
EMD	<i>Earth Mover's Distance</i>	Distância de Terra Movida
GMM	<i>Gaussian Mixture Models</i>	Modelos de Misturas de Gaussianas
HoG	<i>Histogram of Oriented Gradient</i>	Histogramas de Orientações de Gradientes
ISR-UC	<i>Institute of Systems and Robotics - University of Coimbra</i>	Instituto de Sistemas e Robótica – Universidade de Coimbra
LRF	<i>Laser Range Finder</i>	Medidor de Distância a Laser
MAS	<i>Multiagent Systems</i>	Sistemas Multiagente
ML	<i>Machine Learning</i>	Aprendizagem Automática
OpenCV	<i>Open Source Computer Vision Library</i>	--
PCA	<i>Principal Component Analysis</i>	Análise de Componentes Principais
ROS	<i>Robot Operating System</i>	--
SIFT	<i>Scale-Invariant Feature Transform</i>	--
SURF	<i>Speeded Up Robust Features</i>	--
SVM	<i>Support Vector Machine</i>	Máquina de Vetores de Suporte

LISTA DE FIGURAS

Figura 2.1 – Exemplo de um sistema de videovigilância.	5
Figura 2.2 – Imagem apresentada em [44] onde está representado o mapa métrico e sobreposto o grafo de patrulhamento (nós e arestas representados a azul e vermelho, respetivamente) do ISR – UC.	6
Figura 2.3 – Imagem de um local do ISR - UC; (a) sem anomalia; (b) com anomalia; note-se que a imagem (b) é tirada de um ponto de vista diferente da (a) sendo resultado do patrulhamento. A anomalia a considerar é a pessoa presente em (b).	8
Figura 3.1 – Imagem explicativa do algoritmo baseado na definição de um limiar de similaridade; (a) aquisição do <i>dataset</i> de imagens para cada local e obtenção do limiar; (b) comparação de uma imagem do local (obtida durante o patrulhamento) com a imagem de referência.	13
Figura 3.2 – Imagens de um local teste (local T) do ISR - UC, tiradas em diferentes dias, com diferentes pontos de vista; (a) sem anomalia; (b) com pessoa; (c) com mala; (d) com pessoa.	14
Figura 3.3 – Exemplos de imagens tiradas no local A; (a) sem anomalia; (b) com anomalia-pessoa; (c) com anomalia-porta aberta; (d) com anomalia-cadeira; (e) com anomalia-pessoa + cadeira + porta aberta.	19
Figura 3.4 – Valores da EMD calculada entre a imagem 1 (a_1) e as restantes imagens (incluindo ela própria) no local A.	19
Figura 3.5 – Exemplos de imagens tiradas no local B; (a) sem anomalia; (b) com anomalia-pessoa; (c) com anomalia-mala; (d) com anomalia-pessoa + mala + porta aberta.	20
Figura 3.6 – Valores da EMD calculada entre a imagem 1 (b_1) e as restantes imagens (incluindo ela própria) no local B.	20
Figura 3.7 – Exemplos de imagens tiradas no local C; (a) sem anomalia; (b) com anomalia-pessoa; (c) com anomalia-mala; (d) com anomalia-pessoa + mala.	20
Figura 3.8 – Valores da EMD calculada entre a imagem 1 (c_1) e as restantes imagens (incluindo ela própria) no local C.	21
Figura 3.9 – Imagens do local A com deslocamento de 5° sobre as quais foi aplicada a homografia.....	23
Figura 3.10 – Imagens do local A com deslocamento de 15° sobre as quais foi aplicada a homografia.	23

Figura 3.11 – Valores da EMD calculada entre a imagem 1 (a_1) e as restantes imagens (incluindo ela própria) no local A após correção das imagens com homografia.	23
Figura 3.12 – Imagens do local A corrigidas com homografia e seus histogramas.....	25
Figura 3.13 – Imagens do corredor do ISR - UC tiradas com 3 segundos de diferença sem deslocamento.....	25
Figura 3.14 – Imagem explicativa do algoritmo de compressão de caminho.	27
Figura 3.15 – Imagens exemplo resultantes da implementação do algoritmo de segmentação de Felzenszwalb e Huttenlocher (2004).	27
Figura 3.16 – Imagens exemplo resultantes da implementação do algoritmo de segmentação de Felzenszwalb e Huttenlocher (2004); (a) imagens das segmentações de imagens do local C, com uma anomalia e correção de deslocamento; (b) imagens das segmentações de imagens do local A, sem anomalia e sem deslocamento.	29
Figura 4.1 – Visualização da separação entre duas classes de dados recorrendo a um hiperplano. Fonte: wikipédia.	33
Figura 4.2 – Representação dos módulos do sistema.	37
Figura 4.3 – Sistema proposto; (a) Visão global do sistema; (b) Comportamento detalhado num local exemplo (local 3).	38
Figura 4.4 – Treino para cada local.	39
Figura 4.5 – Exemplo de anomalias a considerar.	40
Figura 4.6 – Funcionamento do sistema na classificação das anomalias.	40
Figura 5.1 – Imagens dos três locais utilizados nesta experiência; (a) imagem do local A; (b) imagem do local B; (c) imagem do local T.	42
Figura 5.2 – Mapa do ISR - UC onde estão marcados os 10 locais onde cada robô para e as direções em que os robôs param e captam as fotografias.	46
Figura 5.3 – Imagens exemplo de 5 locais distintos do ISR - UC; (a) imagem do local F sem anomalia; (b) imagem do local G com anomalia (pessoa); (c) Imagem do local H com anomalia (porta aberta); (d) imagem do local I com anomalia (mala); (e) imagem do local J com anomalia (pessoa).	46

Figura 5.4 – Gráfico com os resultados da experiência 1 utilizando dados reais; apresentação do número de verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN) para cada local.	47
Figura 5.5 – Gráfico com os resultados da experiência 2 utilizando dados reais; apresentação do número de verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN) para cada local.	49
Figura 5.6 – Gráfico com os resultados da experiência 2 utilizando dados reais e homografia; apresentação do número de verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN) para cada local após se usar homografia.	51
Figura 5.7 – Imagem do local C com uma anomalia (mala).	53
Figura 5.8 – Imagem do local D com uma anomalia (pessoa).	53
Figura 5.9 – Ilustração do funcionamento esperado do protótipo numa situação real de patrulhamento. (a) imagem da cena; (b) imagem capturada pelo robô; (c) informação apresentada pela estação base.	54

LISTA DE TABELAS

Tabelas 5.1 e 5.2 – Matrizes de confusão para os locais A e B, respetivamente, referentes à experiência 1.	43
Tabela 5.3 – Matriz de confusão para o local T referente à experiência 1.	43
Tabelas 5.4, 5.5 e 5.6 – Matrizes de confusão para os locais A B e T, respetivamente, referentes à experiência 2.	44
Tabela 5.7 – Tabela com o número de fotografias tiradas para cada local.	47
Tabelas 5.8 – Tabela com a exatidão média obtida em cada local.	48
Tabelas 5.9 – Matriz de confusão do sistema de deteção de anomalias global.	48
Tabelas 5.10 – Tabela com a exatidão média obtida em cada local.	49
Tabelas 5.11 – Matriz de confusão do sistema de deteção de anomalias global.	49
Tabela 5.12 – Matriz de confusão do sistema de deteção de anomalias global, após se usar a homografia.	51
Tabela 5.13 – Matriz de confusão para o classificador de anomalias.	52

1. INTRODUÇÃO

Nos últimos anos tem-se assistido a avanços técnicos significativos e em especial na área da Robótica onde se observa um grande crescimento na investigação em sistemas robóticos, seja em manipuladores robóticos ou em robótica móvel. Tem havido um grande esforço de investigação para dotar os robôs de mobilidade e capacidade de percepção artificial e decisão, necessárias para se adaptarem e reagirem adequadamente a ambientes dinâmicos e por vezes desconhecidos dando origem a muitos e novos desafios. Estas capacidades permitem que os robôs executem diversas tarefas e, por isso, são cada vez mais utilizados no quotidiano do ser humano.

1.1. Motivação

A introdução de robôs móveis em ambientes reais prende-se com a necessidade de auxiliar o ser humano em tarefas pesadas, repetitivas, perigosas ou difíceis de realizar. Como exemplos tem-se a vigilância, a distribuição de medicamentos e alimentos a doentes em enfermarias hospitalares, a indústria, a monitorização ambiental, os cenários de busca e salvamento em ambientes de catástrofe e de guerra, etc. Muitas vezes, as tarefas realizadas com robôs móveis são realizadas por equipas de robôs dotados da capacidade de se localizarem no ambiente e navegarem de forma autónoma, libertando assim os seres humanos de tarefas monótonas e repetitivas como o patrulhamento de edifícios.

Navegação e percepção são então duas partes fundamentais para que qualquer robô móvel seja capaz de realizar tarefas em ambientes dinâmicos com reduzida interação com humanos. De facto, a principal diferença entre navegação robótica e navegação humana está na capacidade de aquisição e interpretação de dados do meio ambiente. Um ser humano tem a capacidade de perceber tudo o que o rodeia mesmo que a aquisição de dados seja em ambientes cujas condições variam constantemente, uma vez que consegue decodificar a informação recebida pelos seus sentidos. São os olhos que permitem ao ser humano conhecer e ter ampla percepção do mundo à sua volta. Quatro quintos de todos os dados recebidos pelo cérebro chegam através das imagens captadas pelos olhos. Já os robôs, têm grandes limitações no que toca à tomada de decisão e percepção do ambiente, uma vez que não possuem capacidade de decodificar a informação obtida, tendo ainda assim, uma grande capacidade para contornar e se desviar de obstáculos.

O patrulhamento está associado ao ato de percorrer um determinado ambiente com o objetivo de o proteger ou de simplesmente o supervisionar. Muitos estudos têm sido realizados neste âmbito, com grandes evoluções verificadas nos algoritmos de coordenação e decisão usados em

grupos de robôs de patrulhamento com o intuito de otimizar o escalonamento pelos vários robôs nos locais a visitar no decurso da missão. Segundo David Portugal e Rui P. Rocha [45] têm sido propostos recentemente algoritmos de patrulhamento para equipas de robôs móveis cooperativos que, com base na representação do ambiente a patrulhar através de um grafo, permitem obter um elevado desempenho na execução automática da tarefa, funcionar de forma completamente distribuída, escalar para equipas de grande dimensão, e ser tolerantes a falhas de robôs individuais.

Para que os robôs possam executar tarefas como vigilância ou patrulhamento não podem só possuir capacidade de navegação autónoma, devem também estar dotados de alguma perceção artificial que lhes permita detetar automaticamente anomalias no ambiente. Desta forma os robôs ficam aptos para executarem autonomamente a tarefa do patrulhamento em cenários reais.

Uma vez que a melhor maneira de perceber o mundo é através da visão, é importante estudar algoritmos de visão por computador que consigam dotar os robôs de capacidades que lhes permita detetar situações de alarme nos locais de interesse a patrulhar. Felizmente têm sido feitos vários estudos nesta área e, atualmente, existe um elevado número de técnicas e algoritmos que permitem implementar numa máquina um sistema artificial, capaz de obter e codificar informação através de imagens. Estas técnicas estão em constante evolução, uma vez que só a partir do final da década de 70 é que começaram os estudos mais profundos, quando os computadores já podiam processar grandes conjuntos de dados, como imagens.

A maioria dos sistemas de vigilância/patrulhamento utilizam câmaras fixas para executarem essas tarefas, descartando as funcionalidades e vantagens dos robôs. Uma vez que recorrendo a técnicas de visão por computador é possível dotar os robôs de capacidades de perceção artificial é possível, recorrendo às capacidades dos robôs, obter sistemas de patrulhamento mais robustos e flexíveis do que redes de câmaras fixas. É fácil reconfigurar os locais a patrulhar por um grupo de robôs móveis, que não requerem uma infraestrutura de câmaras. Não é fácil, senão inviável, reconfigurar uma rede de câmaras. É nesta perspetiva que surge a motivação para este trabalho. Pegando no protótipo desenvolvido por David Portugal [44], onde os robôs navegam autonomamente num ambiente representado por um grafo, de forma a coordenarem adequadamente as duas tarefas e maximizarem a frequência de passagem nos locais a patrulhar, o objetivo é dotar os robôs de perceção artificial para que, aquando de uma paragem, possam concluir se existe ou não alguma anomalia no local em apreço.

1.2. Objetivos

A proposta desta dissertação passa por estudar o problema do patrulhamento, mais concretamente da perceção artificial, de forma a implementar uma solução robusta e viável para a deteção de anomalias num determinado ambiente.

O principal objetivo deste trabalho passa então por estudar e implementar um algoritmo robusto que possibilite aos robôs perceberem o ambiente a ser patrulhado e detetarem anomalias que possam ocorrer. Como objetivo secundário pretende-se que o robô seja capaz de classificar a anomalia detetada. Por forma a que seja tomada uma decisão com base na informação processada pelo robô, estes devem emitir uma mensagem de alarme para o supervisor humano que irá ter contacto permanente com uma aplicação remota numa estação base que lhe permite monitorizar toda a missão levada a cabo pelos robôs. O objetivo é que com este trabalho se possa implementar um sistema de patrulhamento para que possa ser utilizado numa situação real, por exemplo numa galeria de arte, para deteção de intrusos ou ausência de algum artefacto.

1.3. ESTRUTURA DA DISSERTAÇÃO

Esta dissertação encontra-se dividida em seis capítulos. No capítulo 2 são referidos alguns trabalhos feitos anteriormente nesta área e são referidos alguns conceitos e algumas técnicas utilizadas em problemas semelhantes. No capítulo 3 são abordadas algumas técnicas de visão por computador que foram ensaiadas neste trabalho como potenciais soluções do problema, mas que não forneceram os resultados pretendidos. No capítulo 4 apresenta-se o sistema proposto, explicando a técnica utilizada e descrevendo a metodologia seguida. No capítulo 5 são apresentados os resultados obtidos com a realização de diversas experiências e são extraídas algumas conclusões. Por fim, no capítulo 6 apresenta-se uma conclusão refletindo-se sobre o percurso feito ao longo deste trabalho. São apresentadas algumas linhas viáveis de trabalho futuro.

2. ENQUADRAMENTO TEÓRICO

Nesta secção são discutidos alguns trabalhos desenvolvidos nas áreas subjacentes a esta dissertação, videovigilância, coordenação multiagente em missões de patrulhamento, deteção de anomalias e classificação de objetos. São referidas algumas técnicas presentes nesses trabalhos. Mais tarde, será feita uma descrição detalhada de algumas das técnicas referidas neste capítulo e que foram implementadas aquando do desenvolvimento do trabalho.

2.1. VIDEOVIGILÂNCIA

Videovigilância é uma área de investigação bastante desenvolvida quer no setor comercial quer no tecnológico sendo cada vez mais uma realidade ubíqua, apesar de ainda levantar muitas questões éticas e até jurídicas, relativamente à aquisição de imagens em espaços públicos sem o consentimento das partes envolvidas [1]. A tecnologia atingiu um nível de maturação que permite colocar câmaras para capturar imagens de vídeo com um baixo custo, mas encontrar recursos humanos para monitorizar essas imagens tem um custo elevado. Encontram-se facilmente câmaras de videovigilância em estabelecimentos comerciais, de ensino, de serviços públicos, etc., tendo a capacidade de capturar imagens e/ou vídeos e de os guardar em ficheiros. Ocorrido um evento indesejável, os investigadores policiais, podem analisar as imagens desse momento e verificar o que realmente aconteceu. Para evitar esses eventos é necessária uma monitorização e uma análise contínua dos dados de vídeo para alertar os responsáveis de que existe algo suspeito, para que estes possam agir antes de ocorrer o evento [18].

É impraticável um ser humano vigiar em permanência múltiplas câmaras. Por isso foi necessário dotar os sistemas de vídeo de alguma capacidade de perceção. Primordialmente contagem de pessoas, leitura de matrículas, controlo remoto de câmaras, restrição de perímetro [2] eram algumas das atividades que se podiam encontrar nos sistemas de videovigilância. Hoje em dia, graças aos avanços tecnológicos, é possível fazer reconhecimento de pessoas, objetos, medir o nível de atenção de um segurança, entre outras.

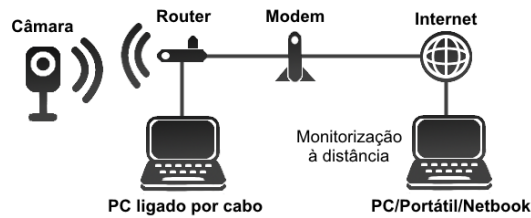


Figura 2.1 – Exemplo de um sistema de videovigilância.

2.2. SISTEMAS MULTI-ROBÔ

Em geral, para uma execução eficaz desta atividade é necessária a alocação de agentes de patrulhamento, para que seja varrida a maior área possível por unidade de tempo e que seja assegurada uma frequência mínima de passagem de patrulha em cada zona a patrulhar. Assim, geralmente existe a necessidade da tarefa ser executada por um grupo de agentes que assegurem a cobertura e frequência de passagens necessárias, tratando-se, portanto, de um problema multiagente [44]. Neste contexto, um sistema de múltiplos robôs deverá ser capaz de executar tarefas que um único robô não conseguiria suportar.

Existem vantagens em usar equipas de robôs ao invés de usar apenas um. Sem a ajuda de outros robôs, um robô fica mais vulnerável em ambientes hostis e uma equipa de robôs consegue garantir uma maior frequência de cobertura de todos os locais e uma maior robustez a nível do patrulhamento. Por outro lado, quando um problema é complexo é útil dividi-lo em diferentes tarefas e dividi-las pelos diferentes robôs da equipa. A execução de tarefas particionáveis com cooperação eficaz é uma grande vantagem dos sistemas multiagente (MAS – *Multiagent Systems*). Esta vantagem pode ser utilizada, por exemplo, na exploração de ambientes desconhecidos. Para aumentar a robustez e fiabilidade podem existir diversos robôs dotados de diferentes habilidades e sensores e assim criar-se uma equipa heterogénea onde cada robô é relativamente simples, o que reduz o seu custo global.

Outra vantagem de usar uma equipa de robôs em vez de apenas um, tem a ver com a autonomia do robô. Um robô que executa um conjunto de tarefas sozinho irá necessitar de uma maior bateria para as poder executar todas, ao contrário de uma equipa de robôs que necessita apenas da suficiente para cada um executar a sua tarefa. Por outro lado, para que um robô possa executar diversas tarefas diferentes, necessita de estar dotado de um conjunto de diferentes sensores, e por isso, torna-se mais complexo e inevitavelmente mais caro. Além disso, uma falha num robô de uma equipa de robôs não é tão grave como uma falha num robô que opera sozinho. Numa equipa de robôs pode-se redirecionar, para outro robô da equipa, a tarefa que estava a ser executada por aquele que falhou, enquanto a falha é resolvida ou enquanto o robô é substituído

por outro. Quando ocorre uma falha num robô que opera sozinho a execução da tarefa pára. Por todas estas razões é muito mais vantajoso usar uma equipa de robôs do que apenas um.

2.2.1. COORDENAÇÃO MULTIAGENTE EM PATRULHAMENTO

Usar câmaras fixas para videovigilância pode ter algumas desvantagens, nomeadamente, quando se pretende vigiar um lugar que anteriormente não o era, ou quando se quer deixar de vigiar um determinado local. Para que tal aconteça é necessário instalar ou desinstalar uma câmara de um determinado local o que pode acarretar custos extra. Por outro lado, quando uma câmara avaria, o local onde esta se encontrava deixa de estar vigiado, tendo de se proceder à reparação da mesma o mais rápido possível. É neste contexto que usar robôs se torna mais vantajoso, bastando alterar o percurso para que deixem de vigiar um determinado local, ou passem a vigiar outro que não o era até então. Caso um robô fique sem bateria ou avarie, todos os locais vão continuar a estar vigiados, sem comprometer a segurança do ambiente, pois os restantes robôs continuam a seguir a trajetória e a “substituir o robô inapto”.

No âmbito da tese de doutoramento de David Portugal [44] na qual o autor exemplifica algoritmos de navegação para os robôs, foi desenvolvido um protótipo de uma equipa de robôs para patrulhamento, permitindo-lhes navegar autonomamente no Instituto de Sistemas e Robótica – Universidade de Coimbra (ISR – UC), recorrendo para isso a um grafo representativo do ambiente. Nesse protótipo, as posições iniciais que os robôs têm de tomar no início da missão são previamente conhecidas, assim como todos os pontos de interesse onde os robôs têm de parar para adquirirem informação, por exemplo uma imagem, de modo a realizarem o patrulhamento/vigilância. Em [13] a abordagem utilizada por Beeson é semelhante, também recorrendo a um grafo para representar o ambiente.

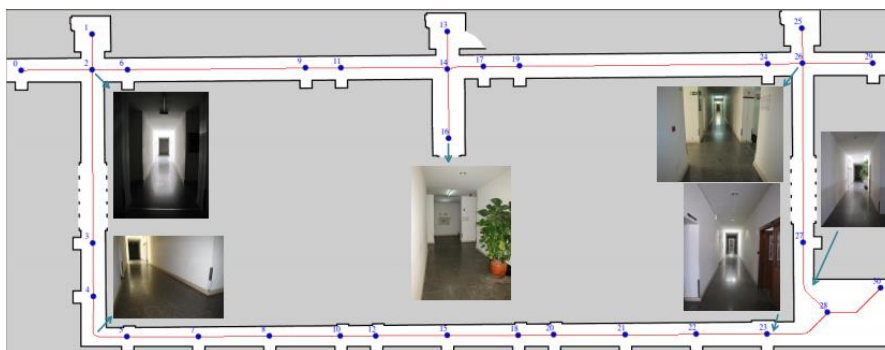


Figura 2.2 – Imagem apresentada em [44] onde está representado o mapa métrico e sobreposto o grafo de patrulhamento (nós e arestas representados a azul e vermelho, respetivamente) do ISR - UC.

Se toda a área de um ambiente não pode ser monitorizada o tempo todo, então, cada local alvo dessa área deve ser visitado o maior número de vezes possível. De forma semelhante a [44], também em [30] se aborda a temática do patrulhamento com equipas de vários robôs. Nesse trabalho, a área de atuação é representada como sendo um polígono, com k pontos e é usada uma equipa de N robôs que visitam repetidamente cada ponto do polígono. A área de trabalho é decomposta em células, e cada célula é do tamanho dos robôs. É gerado um caminho cíclico com todas as áreas a visitar e a distância entre cada robô é calculada dividindo o tamanho total do ciclo pelo número total de robôs, N , dando uma distância igual entre cada dois robôs permitindo, deste modo, distribuir os robôs pela área de uma forma equidistante, e assim passar por todos os locais o maior número de vezes, espaçado em intervalos de tempo regulares. Em [42] é proposta uma métrica para melhor distribuir os robôs pelos locais de patrulhamento, mas também baseado num grafo representativo do ambiente.

A área de investigação destinada ao desenvolvimento de algoritmos para coordenação de multiagentes está amplamente estudada e existem algoritmos eficientes que resolvem esse problema. Para que os robôs consigam verificar se existe alguma anomalia onde param, por exemplo, uma pessoa onde não deveria estar ninguém, e analisar essa anomalia de modo a que o responsável possa agir em conformidade, é necessário implementar técnicas de visão por computador que, em conjunto com o algoritmo de patrulhamento, fornecem um sistema de patrulhamento completo.

2.3. DETEÇÃO DE ANOMALIAS

A ideia base por detrás da grande maioria das técnicas de deteção de novidades é a de fazer corresponder os dados correntes, obtidos pelo sensor, com os dados obtidos previamente. Em [35] e [36] são apresentadas várias técnicas que podem ser utilizadas para detetar novidades, como abordagens usando modelos de misturas de gaussianas (GMM - *Gaussian Mixture Models*), abordagens usando redes neuronais, e outras abordagens de aprendizagem automática (ML - *Machine Learning*) onde se pode incluir máquina de vetores de suporte (SVM – *Support Vector Machine*), entre outras. Em [15] os autores apresentam um relatório técnico sobre deteção de anomalias, onde são exemplificados tipos de anomalias, deteção de intrusos, deteção de fraude, etc., onde apresentam diferentes técnicas para deteção de anomalias, como técnicas baseadas em classificação, como SVM e redes Bayesianas, técnicas baseadas em *clustering*, técnicas baseadas em abordagens estatísticas, entre outras.

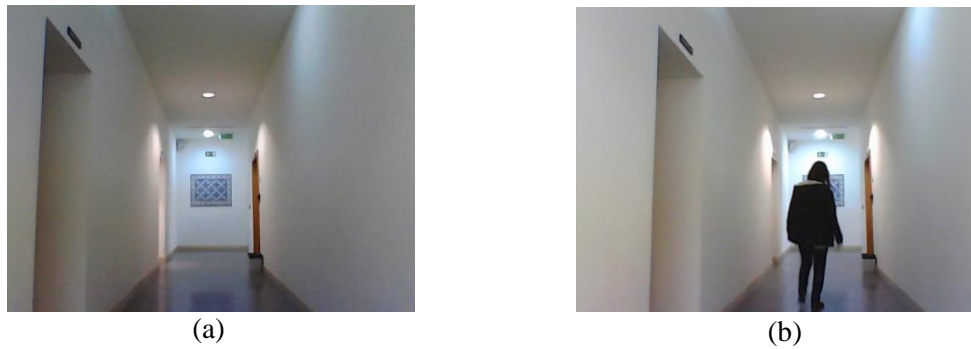


Figura 2.3 – Imagem de um local do ISR - UC; (a) sem anomalia; (b) com anomalia; note-se que a imagem (b) é tirada de um ponto de vista ligeiramente diferente da (a) sendo resultado do patrulhamento. A anomalia a considerar é a pessoa presente em (b).

Em situações específicas pode-se utilizar a informação obtida através do *laser range finder* (LRF) usado pelo robô para a tarefa de navegação, para se implementar algoritmos de detecção de anomalias. Por exemplo em [31] e [21], é usada informação obtida pelo *laser scanner* 3D do robô para construir uma nuvem de pontos. É usado um GMM para agrupar todos os pontos encontrados e, posteriormente, é aplicada a métrica da distância de terra movida (EMD – *Earth Mover's Distance*) [47] para calcular a distância entre duas distribuições de pontos. Se existir alguma diferença, então a EMD permite quantificar essa diferença. Em [40] a informação extraída do *laser* é agrupada em pontos usando GMM e posteriormente esses pontos são representados num grafo. Os grafos são depois comparados através do seu clique máximo. Dois nós de um grafo são adjacentes se forem ligados por uma aresta. O clique de um grafo é o conjunto de nós, onde todos eles são adjacentes, e o clique máximo é o maior entre todos os cliques de um grafo.

Em [10], os autores usam a informação obtida pelo LRF e por uma câmara colocada no robô para reconstruir o ambiente. Esta abordagem introduz a utilização de descritores de imagens. Neste caso, foi utilizado o descritor *Scale-Invariant Feature Transform* (SIFT) para fazer corresponder uma imagem a outra (imagem de referência) através da correspondência de características, ou *features*, locais.

Em [11] os autores definem uma métrica que mede a similaridade de dois pares de imagens tendo por base o tamanho da concatenação de um par de imagens comprimidas. Explicam que um par de imagens semelhantes concatenado e comprimido possui tamanho inferior a um par de imagens diferentes e, dessa forma, definem um limiar para a mesma cena que separa imagens de situações anómalas de imagens sem anomalia.

Quando se faz o patrulhamento com robôs móveis é necessário ter em conta dois movimentos diferentes; em primeiro lugar, a movimentação dos próprios robôs que origina a

movimentação dos sensores resultando em alterações no ponto de vista em que a informação é recolhida; em segundo lugar, a movimentação dos objetos do próprio ambiente [24].

Quando a informação obtida são apenas imagens do ambiente, então a sua análise torna-se mais suscetível ao ruído. Analisar imagens que tenham sido tiradas em diferentes momentos do dia, com diferentes condições atmosféricas, sem nenhuma anomalia, mas em diferentes condições de iluminação e com diferentes pontos de vista, pode gerar resultados erróneos. Nestes casos a análise deve ser feita com algoritmos robustos à rotação, à translação e à luminosidade.

Este é o ponto fulcral deste trabalho: implementar uma técnica de visão por computador que permita detetar anomalias no ambiente tendo em conta as diferenças de luminosidade e os diferentes pontos de vista em que imagens da mesma cena são tiradas, durante uma missão de patrulhamento. Felizmente, vários estudos têm sido realizados, resultando em algoritmos e soluções para os problemas de visão por computador subjacentes.

• **Subtração do fundo (*background subtraction*):**

Uma das técnicas comumente utilizadas para a deteção de anomalias é a subtração do fundo. Em [23] os autores definem a subtração de fundo como sendo um método usado para segmentar regiões de movimento numa sequência de imagens tiradas por câmaras estáticas comparando cada nova imagem com uma imagem anterior do cenário. Apresentam um modelo que estima a probabilidade de observar em cada píxel da imagem um valor de intensidade, com base numa amostra de valores para cada píxel.

Em [41] o autor define subtração do fundo como sendo uma abordagem para detetar objetos em movimento através da diferença entre a imagem atual e uma imagem de referência chamada de “imagem de fundo”. Esta imagem, deve ser uma representação do cenário sem objetos em movimento e deve ser atualizada regularmente para se adaptar às condições de luminosidade e aos ajustes de geometria.

Em [17], [49], [46] e [38] são utilizadas câmaras fixas e é realizada subtração do fundo para detetar eventuais anomalias nas diferentes cenas. Esta técnica é utilizada sobretudo em câmaras fixas. Se a câmara de vigilância for ligeiramente movida, então esta técnica irá indicar imagens sem anomalia como sendo anómalas, porque toda a imagem está deslocada [11]. É por isso que, em câmaras fixas, o problema está automaticamente resolvido, não tendo em conta o ruído das câmaras. Fatores como a iluminação e sombras podem trazer muitas dificuldades [29], e rapidamente se chegou à conclusão de que esta técnica não era indicada para este trabalho. Para que esta técnica funcione é necessário ter um ambiente controlado, que num ambiente real é difícil obter. Usando robôs móveis torna-se impossível garantir que estes tiram as fotografias de cada cena do mesmo ponto de vista, pois qualquer alteração a esse nível irá ser uma fonte de erros.

Sendo impossível garantir que não há diferenças de luminosidade ou pontos de vista diferentes, optou-se por estudar métodos de correspondência de imagens. Com esta abordagem, evita-se os problemas do modelo de subtração do fundo e consegue-se obter informação mais realista.

• Correspondência de imagens:

Correspondência de imagens ou comparação de imagens com o objetivo de obter uma medida de similaridade é um problema importante de visão por computador, com várias aplicações como a recuperação de imagens, o reconhecimento de objetos e de cenas, entre outros. No âmbito desta dissertação, algoritmos de correspondência de imagens podem ser úteis na medida em que medem similaridade entre duas imagens, permitindo estabelecer uma relação entre um par de imagens e, assim, separar imagens de situações anómalas de imagens de situações normais.

A correspondência de imagens pode ser feita através da comparação de histogramas de cor, da comparação de pontos característicos [26], as chamadas *features*, extração de arestas e comparação das mesmas, entre outros. Posteriormente, os vetores que contêm as características de cada imagem são comparados recorrendo para isso a métricas que permitem medir a similaridade das imagens. Pode usar-se a norma Euclidiana, a correlação, entre outras. Em [26] é utilizado o descritor SIFT para obter os pontos característicos da imagem e posteriormente é utilizada EMD que permite medir a similaridade entre duas distribuições discretas – distribuições de *features*. Em [39] é utilizado o descritor *Speeded Up Robust Features* (SURF) para extrair pontos característicos da imagem. Em [27] os píxeis são agrupados em regiões e essas regiões são representadas por *clusters*. As imagens são então representadas como GMM e posteriormente é utilizada a distância de Kullback-Leibler para medir a similaridade de distribuições e então fazer correspondência das imagens.

2.4. CLASSIFICAÇÃO DE OBJETOS

Detetar uma anomalia, numa situação de patrulhamento, pode não ser suficiente. Nesta dissertação surge como objetivo secundário classificar a anomalia detetada, uma vez que essa informação pode ser crucial para que o operador humano/vigilante se desloque ao local A em detrimento do local B sem comprometer a segurança do ambiente vigiado. Na literatura existem vários métodos para classificação de objetos. Em [51] por exemplo, são mencionadas algumas técnicas usadas atualmente, como métodos *clustering*, deteção de arestas, subtração do fundo, entre outros, para segmentar imagens e, posteriormente, recorrendo à análise de componentes principais (PCA – *Principal Component Analysis*), fazem correspondência de objetos utilizando a

segmentação feita anteriormente. Esta abordagem necessita que os objetos que aparecem numa imagem também apareçam na outra, para que possam ter correspondência.

Em [50] Tzotsos utiliza SVM para classificar objetos. David Lowe [34] utiliza descritores SIFT para extrair pontos característicos de uma imagem; tendo imagens dos objetos que aparecem numa imagem teste, e extraíndo os pontos tanto dos objetos como da imagem teste faz correspondência dos objetos com os objetos presentes na imagem. Em [43] os autores utilizam uma base de dados de imagens, *Columbia Object Image Library* (COIL), com 7200 imagens de 100 objetos diferentes e, posteriormente, utilizam SVM para reconhecimento dos objetos. Em [19] utilizam o descritor SIFT e, logo após, SVM e *Naïve Bayes* para categorizar objetos. Comparando as duas abordagens, SVM mostrou melhores resultados.

Mais uma vez em [20] recorre-se ao uso de SVM, mas desta vez para se detetarem pessoas. São utilizados histogramas de orientações de gradientes (HoG – *Histogram Of Oriented Gradient*) para descrever a imagem como um histograma de gradientes e posteriormente utiliza-se SVM para detetar humanos nas imagens.

2.5. OPEN SOURCE COMPUTER VISION LIBRARY

OPEN SOURCE COMPUTER VISION LIBRARY (OpenCV) [3] é uma plataforma totalmente livre para uso académico e comercial que desenvolve aplicações na área de visão por computador, inicialmente desenvolvida pela Intel em 2000. O OpenCV possui módulos de processamento de imagem de vídeo I/O, estruturas de dados, álgebra linear, interface gráfica de utilizador e inúmeros algoritmos de visão por computador como, filtros de imagem, calibração de câmara, reconhecimento de objetos entre outros. Esta biblioteca foi desenvolvida nas linguagens de programação C/C++ mas também dá suporte a utilizadores de Java, Python entre outros. Os objetivos passavam não só por avançar a pesquisa de visão por computador para promover código aberto e otimizado para tarefas básicas de visão, de modo a que o código pudesse ser portátil mas também por avançar aplicações baseadas em visão por computador e disponibilizar código fonte gratuitamente.

Esta é uma biblioteca muito conceituada em visão por computador e que foi amplamente usada nesta dissertação na versão 3.0 lançada a 04/06/2015. Foi necessário estudar o funcionamento do OpenCV e as estruturas de dados específicas desta biblioteca para posterior utilização. Possui muitos dos algoritmos que foram utilizados no decorrer deste trabalho e foi por isso uma ferramenta muito importante no desenvolvimento do mesmo.

2.6. SUMÁRIO

Neste capítulo apresentaram-se alguns dos trabalhos desenvolvidos na área do patrulhamento e algumas técnicas implementadas para resolver situações específicas.

No próximo capítulo irão ser apresentadas e explicadas algumas das técnicas que foram testadas no decorrer deste trabalho, mas que infelizmente não forneceram os resultados pretendidos.

3. TÉCNICAS BASEADAS NA DEFINIÇÃO DE UM LIMIAR

Nesta secção é apresentado todo o trabalho realizado no âmbito desta dissertação até chegar à solução final apresentada no capítulo 4. São apresentadas todas as abordagens que foram implementadas até esse ponto e são dadas breves explicações sobre cada uma delas. São as abordagens utilizadas inicialmente, mas que não produziram os resultados mais satisfatórios. Contudo, foram importantes para aquisição de conhecimento e permitiram chegar à abordagem proposta.

3.1. ABORDAGENS PARA DEFINIR UM LIMIAR

Com base na afirmação “Uma cena é considerada anómala quando a similaridade máxima entre a cena e todas as cenas do mesmo local, observadas anteriormente, fica abaixo de um dado limiar” [24] optou-se como primeira abordagem por implementar técnicas que permitissem definir um limiar que separasse imagens de uma situação anómala de imagens sem anomalia. Estas técnicas são implementadas tendo por base um conjunto de imagens das duas classes.

São técnicas que permitem detetar se uma imagem tem ou não anomalia, mas que requerem uma técnica posterior para que a anomalia seja classificada. Como a deteção de anomalia é o passo mais importante tentou-se encontrar e implementar uma técnica robusta que permitisse primeiro detetar anomalias. Só depois de se estar perante uma técnica que permitisse ter uma boa deteção é que se passaria ao estudo de classificadores.



Figura 3.1 – Imagem explicativa do algoritmo baseado na definição de um limiar de similaridade; (a) aquisição do *dataset* de imagens para cada local e obtenção do limiar; (b) comparação de uma imagem do local (obtida durante o patrulhamento) com a imagem de referência.

Na figura 3.1 vê-se o procedimento que é levado a cabo na implementação das abordagens que são descritas nesta secção. Numa primeira fase de implementação são capturadas imagens de um mesmo local, com e sem anomalia, mas com diferentes pontos de vista. Quando um robô pára numa missão de patrulhamento, num determinado local, nunca pára exatamente na mesma posição em que parou anteriormente, podendo parar mais à frente/atrás e/ou mais para a direita/esquerda. Por isso, as fotografias capturadas têm um ligeiro desvio relativamente às que foram capturadas anteriormente.

Depois das fotografias todas tiradas, é implementada uma métrica que compara cada uma dessas fotografias com a de referência (sem anomalia). É feito um estudo rigoroso para que seja obtido um limiar que separe fotografias de situações anómalas de fotografias sem anomalia. Posteriormente, aquando de uma missão de patrulhamento, o robô tira uma fotografia e essa fotografia é comparada com a fotografia de referência, utilizando a mesma métrica que foi usada para definir o limiar. As imagens são comparadas com essa métrica e verifica-se se a diferença entre elas é ou não superior ao limiar encontrado anteriormente. Se sim, então as imagens são diferentes e por isso existe anomalia. Caso contrário não existe anomalia.

O objetivo é que exista em cada local um conjunto de fotografias e um limiar para que quando o robô patrulhar e tirar uma fotografia, possa fazer o processamento dessa fotografia com as fotografias/limiar desse local, sabendo para isso o ponto do grafo em que se encontra e assim qual o *dataset* a que tem de aceder. Nesse sentido e numa fase experimental, apenas para saber se as técnicas (apresentadas de seguida) são robustas o suficiente, foi tirado um conjunto de imagens, com diferentes ângulos, para três locais distintos do ISR - UC, local A, B e C. Relativamente a uma posição de referência, 0° , fazem-se incrementos de 5° , 10° e 15° e decrementos de -5° , -10° e -15° . Para cada ângulo (0° , $\pm 5^\circ$, $\pm 10^\circ$, $\pm 15^\circ$) é então adquirido um conjunto de fotografias, com e sem anomalia, como se mostra na figura 3.2.



Figura 3.2 Imagens de um local teste (local T) do ISR - UC, tiradas em diferentes dias, com diferentes pontos de vista; (a) sem anomalia; (b) com pessoa; (c) com mala; (d) com pessoa.

3.2. SURF E EMD

A implementação desta técnica foi baseada em [26] e [39]. Em [39] os autores utilizam um descritor SURF em vez de um descritor SIFT [26]. Foi tomada a decisão de se utilizar o descritor SURF nesta dissertação uma vez que é um descritor de imagem inspirado em parte pelo descritor SIFT mas é várias vezes mais rápido e mais robusto. Nesta secção são então apresentados o descritor SURF e a métrica EMD bem como técnicas para compensar as diferenças de pontos de vista de imagens do mesmo local.

- **SURF**

Desenvolvido por Herbert Bay et al. [12], é um descritor e detetor invariante à rotação e à escala e que aproxima ou supera a nível da repetibilidade, robustez e clareza de outros detetores propostos anteriormente, mas possui um menor custo computacional.

A tarefa de encontrar pontos de interesse e de correspondência entre imagens com o mesmo objeto ou cena é uma parte importante de visão por computador que é útil em diversas áreas como registo de imagem, calibração de câmaras, reconhecimento de objetos, entre outras. Os pontos de interesse a encontrar podem ser cantos, circunferências, juntas, etc.

O SURF é baseado na matriz Hessiana e utiliza também o determinante dessa matriz para seleccionar tanto a escala como a localização dos pontos de interesse.

A matriz Hessiana utiliza originalmente as derivadas Gaussianas de segundo grau, e para um ponto $\mathbf{x} = (x, y)$ numa imagem \mathbf{I} uma matriz Hessiana em \mathbf{x} com escala σ é dada por:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix}$$

onde $L_{xx}(x, \sigma)$ é a convolução da derivada Gaussiana de segundo grau $\frac{\partial^2}{\partial x^2} g(\sigma)$ com a imagem \mathbf{I} no ponto \mathbf{x} .

As derivadas Gaussianas de segundo grau precisam de ser discretizadas e cortadas, e para além disso, os efeitos de *aliasing* acontecem sempre que as imagens resultantes são subamostradas. Para contrariar este efeito o SURF aproxima as derivadas Gaussianas de segundo grau com filtros retangulares de dimensões $m \times m$. Neste sentido, é preciso também equilibrar os pesos relativos na expressão para calcular o determinante da matriz Hessiana, e para este efeito é utilizada a norma Frobenius, $|\mathbf{x}|_F$, da seguinte forma:

$$K = \frac{|L_{xy}(\sigma)|_F |D_{xx}(m)|_F}{|L_{xx}(\sigma)|_F |D_{xy}(m)|_F},$$

onde D_{xx} , D_{xy} e D_{yy} são as aproximações às derivadas Gaussianas de segundo grau.

Assim o determinante da matriz Hessiana aproximada é:

$$\det(H_{approx}) = D_{xx}D_{yy} - (KD_{xy})^2$$

Após detetados os pontos de interesse, o SURF determina a orientação e os componentes do descritor que o descreve.

Para determinar a orientação dos pontos de interesse, calculam-se as transformadas de Haar nas direções x e y numa vizinhança circular em redor desses pontos. Estes resultados são representados como vetores num espaço 2D, e a orientação final é calculada somando os resultados das transformações dentro de uma janela deslizante com um ângulo de abertura de $\frac{\pi}{3}$, resultando num vetor por cada soma feita. O maior destes vetores é o vetor dominante e consequentemente é o vetor que determina a orientação final do ponto de interesse.

Para determinar os componentes que descrevem o ponto de interesse, constrói-se uma região quadrada centrada no ponto de interesse com orientação equivalente à calculada anteriormente. Esta região é subdividida em 4x4 e em cada sub-região são novamente calculadas as transformadas de Haar nas direções horizontal e vertical (relativamente à orientação do ponto de interesse). Para cada sub-região, são calculadas as somas dos resultados destas transformadas $\sum d_x$ e $\sum d_y$ (para as direções horizontal e vertical, respetivamente). São também considerados os valores das somas dos valores absolutos das mesmas $\sum |d_x|$ e $\sum |d_y|$.

Cada sub-região irá então ter um vetor \mathbf{v} de quatro dimensões que o caracteriza:

$$\mathbf{v} = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right)$$

Deste modo, o ponto de interesse irá ter um vetor que o descreve de comprimento 64, constituído pelos 16 vetores das 4x4 sub-regiões.

- **Implementação do SURF em OpenCV**

Existe no OpenCV uma função que fornece os vetores descritores de uma imagem encontrados através do descritor SURF. A função `surf->detectAndCompute()` detecta os pontos de interesse e computa os descritores fornecendo os vetores descritores de cada imagem.

- **EMD**

Uma assinatura, ou *signature*, $\{s_j = (\mathbf{m}_j, \mathbf{w}_{m_j})\}$, representa um conjunto de clusters de *features*. Cada cluster é representado pela sua média, \mathbf{m}_j , e pela sua fração de píxeis \mathbf{w}_{m_j} que pertencem a esse *cluster*. Em abordagens anteriores, a distância entre duas imagens em escala de cinzentos pode ser computada da seguinte maneira: cada píxel é representado por n “pedras” onde n é um inteiro que representa nível de cinzento desse píxel [47]. Depois de normalizar duas imagens com o mesmo número de pedras, a distância entre elas é computada como sendo o custo mínimo de correspondência de pedras entre as duas imagens. É com base nesta distância que surge a EMD [47].

Intuitivamente, dadas duas distribuições onde uma pode ser vista como a massa de terra espalhada no espaço e a outra como um conjunto de buracos no mesmo espaço, a EMD mede a menor quantidade de trabalho necessária para preencher os buracos com a terra. Uma unidade de trabalho corresponde a transportar uma unidade de terra por unidade distância entre a terra e o buraco.

Computar a EMD é uma solução para o conhecido problema de transporte. Supondo que existem vários fornecedores com uma dada quantidade de mercadorias e que são solicitados para fornecer vários consumidores, cada um com capacidade limitada. Nesse caso o problema de transporte é encontrar o fluxo menos caro de mercadorias dos fornecedores, necessárias para satisfazer as necessidades dos consumidores.

A correspondência de *signatures* pode ser vista então como um problema de transporte, e por isso a EMD pode ser utilizada para encontrar a semelhança entre duas imagens, uma vez que pode definir-se uma *signature* como sendo um fornecedor e outra como sendo um consumidor, e definir o custo como sendo a distância entre um elemento na primeira *signature* e um elemento na segunda. Intuitivamente, a solução é minimizar a quantidade de trabalho necessário para transformar uma *signature* noutra.

Então, sendo $P = \{(\mathbf{p}_1, \mathbf{w}_{p_1}), \dots, (\mathbf{p}_m, \mathbf{w}_{p_m})\}$ a primeira *signature* com m clusters onde \mathbf{p}_i é um *cluster* representativo e \mathbf{w}_{p_i} é o peso do *cluster* e $Q = \{(\mathbf{q}_1, \mathbf{w}_{q_1}), \dots, (\mathbf{q}_n, \mathbf{w}_{q_n})\}$ é a segunda *signature* com n clusters; $D = [d_{ij}]$ a matriz de distâncias onde d_{ij} é a distância entre o

cluster \mathbf{p}_i e o *cluster* \mathbf{q}_j . Quer encontrar-se um fluxo $F = [f_{ij}]$, com f_{ij} igual ao fluxo entre \mathbf{p}_i e \mathbf{q}_j que minimize o custo total:

$$\text{Trabalho} (P, Q, F) = \sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij} \quad \text{Eq. 3.1}$$

Com as seguintes restrições:

$$f_{ij} \geq 0 \quad 1 \leq i \leq m, \quad 1 \leq j \leq n \quad (1)$$

$$\sum_{j=1}^n f_{ij} \leq w_{\mathbf{p}_i} \quad 1 \leq i \leq m \quad (2)$$

$$\sum_{i=1}^m f_{ij} \leq w_{\mathbf{q}_j} \quad 1 \leq j \leq n \quad (3)$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min \left(\sum_{i=1}^m w_{\mathbf{p}_i}, \sum_{j=1}^n w_{\mathbf{q}_j} \right) \quad (4)$$

- (1) Permite mover apenas “mercadorias” de P para Q e não vice-versa;
- (2) Limita os *clusters* em P a não enviarem maior quantidade de “mercadorias” que os seus pesos;
- (3) Limita os *clusters* em Q para não receberem mais “mercadorias” que os seus pesos;
- (4) Força a movimentação da maior quantidade de “mercadorias” possível.

Então a EMD pode ser definida como o trabalho (Eq 3.1) normalizado pelo fluxo total:

$$EMD (P, Q) = \frac{\sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}} \quad \text{Eq. 3.2}$$

Isto é necessário quando duas *signatures* têm diferentes pesos totais, para evitar favorecer os mais pequenos.

O código desta métrica foi disponibilizado em [4] pelos autores e foi utilizado no âmbito desta dissertação.

As *signatures* utilizadas são os vetores descritores de cada imagem extraídos com o SURF e atribuiu-se o mesmo peso a cada uma. O objetivo é comparar todas as imagens com a de referência e utilizar o valor dado pela EMD para definir o limiar de separação.

- **Resultados**

Infelizmente, como é possível observar pelos resultados seguintes, não foi possível definir um limiar que separe imagens de situações anômalas de imagens sem anomalia com a exatidão esperada. Foram utilizados três locais diferentes do ISR - UC onde se realizaram as experiências.

➤ Local A:

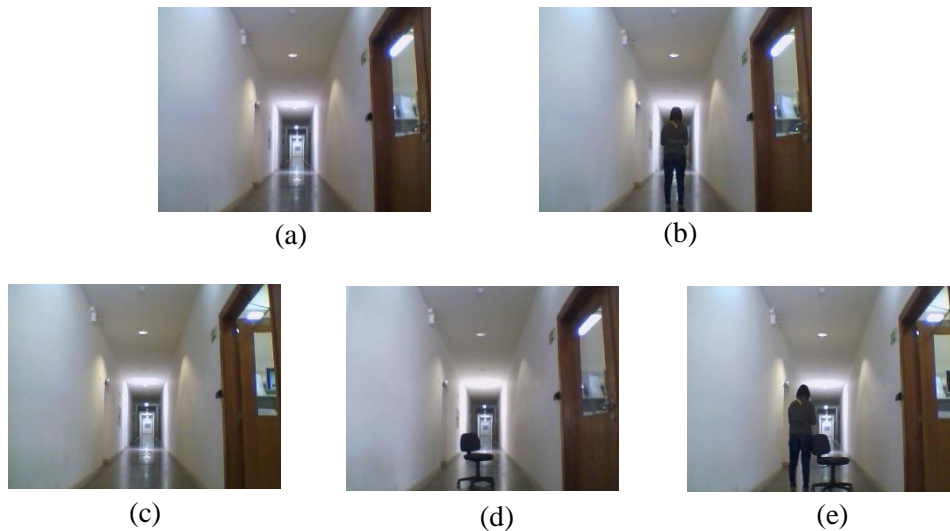


Figura 3.3 – Exemplos de imagens tiradas no local A; (a) sem anomalia; (b) com anomalia-pessoa; (c) com anomalia-porta aberta; (d) com anomalia-cadeira; (e) com anomalia-pessoa + cadeira + porta aberta.

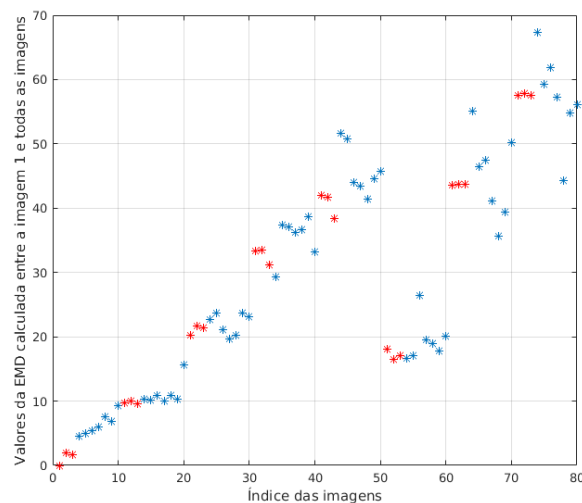


Figura 3.4 – Valores da EMD calculada entre a imagem 1 (a_1) e as restantes imagens (incluindo ela própria) no local A.

Nota: A vermelho estão representados os valores da EMD calculados entre as imagens sem anomalia e a imagem 1. A azul estão os valores das restantes EMD. De 10 a 50 estão índices de imagens deslocadas para a direita (5° , 10° , 15° e no caso do local A, 20° também). De 50 a 70 estão os índices das imagens capturadas ligeiramente para a esquerda (-5° , -10° e -15°). De 0 a 10 estão

os índices das imagens capturadas na posição inicial (0°). De 10 a 20 os índices das imagens adquiridas quando se rodou o robô 5° para a direita (posição 5°), e assim sucessivamente, estando as imagens capturadas na posição -15° representadas nos índices 70 a 80.

➤ Local B:

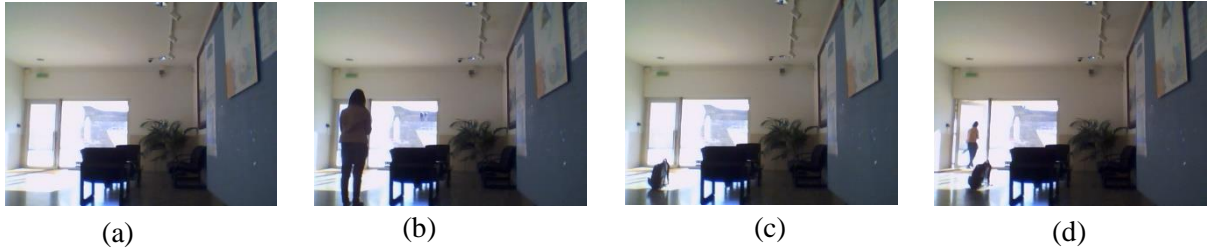


Figura 3.5 – Exemplos de imagens tiradas no local B; (a) sem anomalia; (b) com anomalia-pessoa; (c) com anomalia-mala; (d) com anomalia-pessoa + mala + porta aberta.

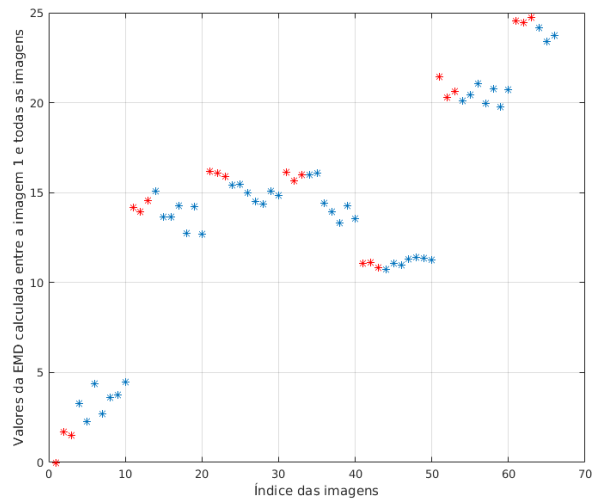


Figura 3.6- Valores da EMD calculada entre a imagem 1 (b_1) e as restantes imagens (incluindo ela própria) no local B.

➤ Local C:



Figura 3.7 – Exemplos de imagens tiradas no local C; (a) sem anomalia; (b) com anomalia-pessoa; (c) com anomalia-mala; (d) com anomalia-pessoa + mala.

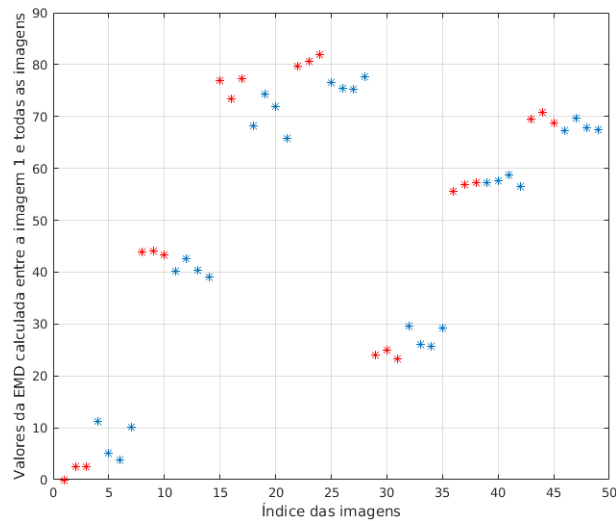


Figura 3.8- Valores da EMD calculada entre a imagem 1 (c_1) e as restantes imagens (incluindo ela própria) no local C.

Como é perceptível nos três locais, não é possível obter um limiar que separe as imagens. Os pontos a vermelho representam imagens sem anomalia e o objetivo seria obter representações onde esses pontos estivessem todos ou quase todos abaixo do nível dos pontos azuis. No entanto, tal não se verifica. Pode verificar-se também, que no primeiro conjunto de pontos (nos 3 gráficos), imagens tiradas da posição inicial (0°), é possível separar as imagens anómalas das restantes. Quando se adiciona movimento (alteração do ponto de vista) obtêm-se os restantes conjuntos de pontos e aí já não se torna possível fazer a separação. No caso deste trabalho é necessário considerar os movimentos de translação e de rotação dos robôs, provocados pelos erros de navegação, que originam desvios entre as imagens de um mesmo ponto.

Para translação pura, pode ser criado um modelo do ambiente juntando diferentes panoramas para criar uma imagem de fundo. Podem ser utilizadas matrizes de homografia para descrever a transformação entre imagens diferentes [29]. Em [52], [28], [37] utilizam a homografia para fazer registo de imagem. As imagens da mesma cena, mas com pontos de vista diferentes, aplicam um descritor para extrair pontos de interesse, fazem a correspondência desses pontos e posteriormente utilizam a homografia para juntar as imagens e obter uma única imagem de fundo. Em [52] utilizam o detetor de cantos formulado por Harris, em [28] utilizam o descritor SURF e em [37] o descritor SIFT. Posteriormente é então usada a homografia para formar uma imagem de fundo.

No caso destas experiências, verificou-se que nas imagens obtidas com o mesmo ponto de vista, o algoritmo fornece os resultados esperados. Contudo, em imagens tiradas de diferentes pontos de vista tal não se verifica. Como não é possível garantir que os robôs capturem as fotografias todas na mesma posição é necessário utilizar um método/algoritmo que possa compensar o movimento/diferença de ponto de vista em duas imagens da mesma cena.

Seguindo a abordagem utilizada em [28], utilizam-se os pontos obtidos com o SURF, mas em vez de se calcular diretamente a EMD sobre esses pontos, faz-se a correspondência dos pontos usando a homografia para compensar o deslocamento. Posteriormente, é usada a EMD para estimar a similaridade das imagens.

• Homografia

Um ponto 2D (x, y) numa imagem pode ser representado como um vetor 3D, $\mathbf{x} = (x_1, x_2, x_3)$ onde $x = \frac{x_1}{x_3}$ e $y = \frac{x_2}{x_3}$. Isto é a representação homogénea de um ponto e situa-se no plano de projeção P^2 . A homografia é um mapeamento invertível de pontos e linhas no plano de projeção $P^2 \rightarrow P^2$ [22]. Um mapeamento $P^2 \rightarrow P^2$ é uma projeção se e só se existir uma matriz H 3x3 não singular tal que para qualquer ponto em P^2 representado por um vetor \mathbf{x} haja um ponto mapeado correspondente $H\mathbf{x}$. Isto permite concluir que para calcular a homografia que mapeia cada x_i para o x_i' correspondente é preciso apenas calcular a matriz de homografia H .

Tipicamente, as homografias são estimadas entre imagens através de correspondência de pontos de interesse entre as mesmas. Os algoritmos comumente utilizados fazem uso de correspondências deste género, embora também haja algoritmos que utilizam outras características tais como linhas ou formas.

Neste caso em específico optou-se por relacionar todas as imagens com uma imagem de referência. Cada vez que uma imagem é comparada com a imagem de referência, calcula-se a homografia entre elas, descartando a informação que não é comum a ambas. Assim as imagens ficam com a mesma informação, isto é, ficam com o mesmo ponto de vista e sobre elas é aplicado o algoritmo anterior (SURF + EMD).

• Determinação da homografia com OpenCV

O OpenCV possui uma função que calcula a homografia tendo por base os pontos encontrados com o descritor SURF. Os pontos encontrados são correspondidos entre as duas imagens utilizando a função `match()` e posteriormente os pares de pontos correspondentes são utilizados na função `findHomography()`. Esta função permite obter a matriz de homografia que relaciona duas imagens.

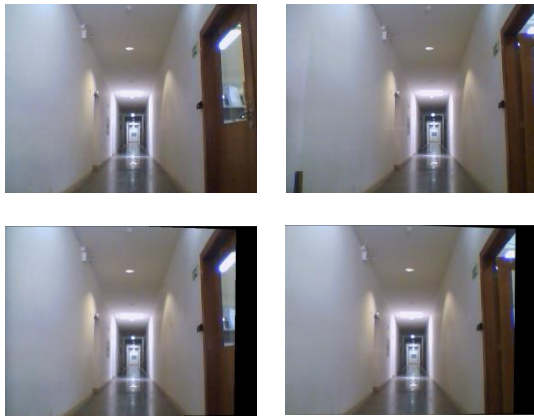


Figura 3.9 – Imagens do local A com deslocamento de 5° sobre as quais foi aplicada a homografia.

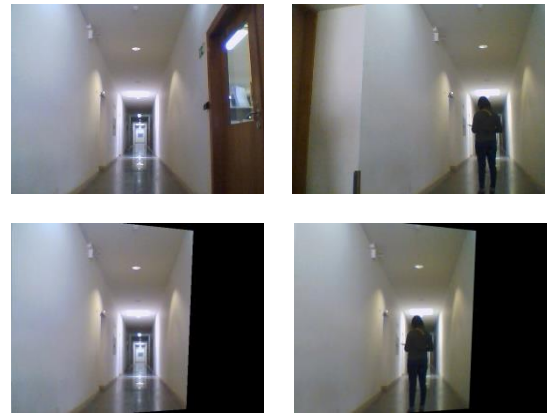


Figura 3.10 – Imagens do local A com deslocamento de 15° sobre as quais foi aplicada a homografia.

Este algoritmo mostra-se eficaz quando o deslocamento ocorrido é significativo mesmo que existam anomalias no cenário. As imagens ficam então com o mesmo ponto de vista.

- **Resultados (Homografia + SURF + EMD)**

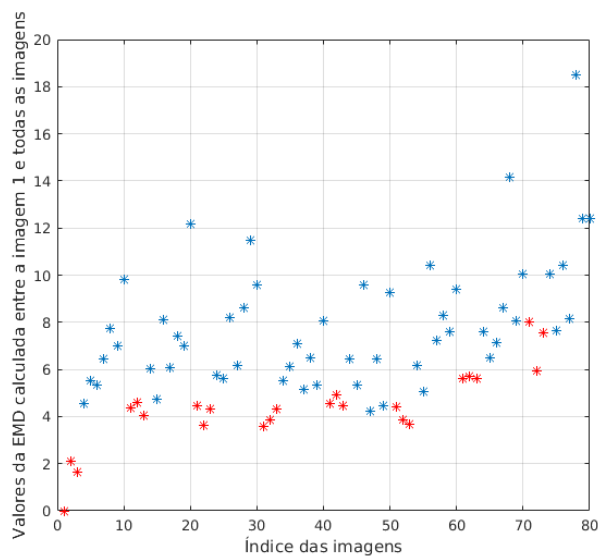


Figura 3.11- Valores da EMD calculada entre a imagem 1 (a_1) e as restantes imagens (incluindo ela própria) no local A após correção das imagens com homografia.

Infelizmente, mais uma vez não foi possível encontrar um limiar para separar as duas categorias de imagens. No local A, no qual se obtiveram melhores resultados com esta abordagem, verificou-se uma acentuada melhoria, ainda assim para imagens deslocadas para a esquerda (10° e

15°) os resultados não permitem definir um limiar. Iriam existir muitos falsos negativos e alguns falsos positivos no limiar que fosse definido. Nos outros locais os resultados foram bastante menos favoráveis do que aqueles obtidos neste local.

Esta técnica foi testada de várias formas numa tentativa de obter resultados mais satisfatórios. Foram utilizadas funções existentes no OpenCV para extrair as *features* a comparar com EMD (diferentes do descritor SURF), também se utilizou o fluxo ótico em vez da homografia para compensar o deslocamento, mas sem o sucesso pretendido. Olhando para a figura 3.13 verifica-se que, ao existir um deslocamento acentuado, também existe uma grande diferença na luminosidade. Se para o olho humano essa diferença é significativa, então para a máquina torna-se pior. Face aos resultados obtidos, optou-se por utilizar outras técnicas baseadas em visão por computador, nomeadamente a comparação de histogramas e extração de arestas.

3.3. COMPARAÇÃO DE HISTOGRAMAS

A comparação de histogramas é uma das várias maneiras disponíveis para se obter informação acerca das imagens. Um histograma é definido como a quantidade do número de píxeis para cada valor de intensidade de uma imagem ou uma representação gráfica da distribuição da intensidade. É uma das formas comumente utilizada para representar a distribuição dos níveis de cinzento de uma imagem, e possivelmente a mais útil em processamento de imagem [32]. São também conhecidos como distribuição de intensidade e Função de Densidade de probabilidade já que, estatisticamente, o histograma representa a probabilidade de se achar um tom de cinzento (neste caso) de um dado valor, dentro da imagem.

Ao observar-se um histograma têm-se uma noção instantânea das características da imagem. Dessa forma, quando se comparam dois histogramas de duas imagens adquire-se uma noção mais clara da similaridade entre elas. Utilizando o método referenciado em [32] optou-se por comparar histogramas utilizando a norma qui-quadrado:

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)}$$

- **Comparação de histogramas em OpenCV**

Utilizando a função `compareHist()` do OpenCV e utilizando a norma qui-quadrado comparou-se o histograma de cada imagem de teste com o da imagem de referência. É de notar que antes deste procedimento, as imagens foram corrigidas com a homografia para que a informação fosse comum nas duas.

- **Resultados**

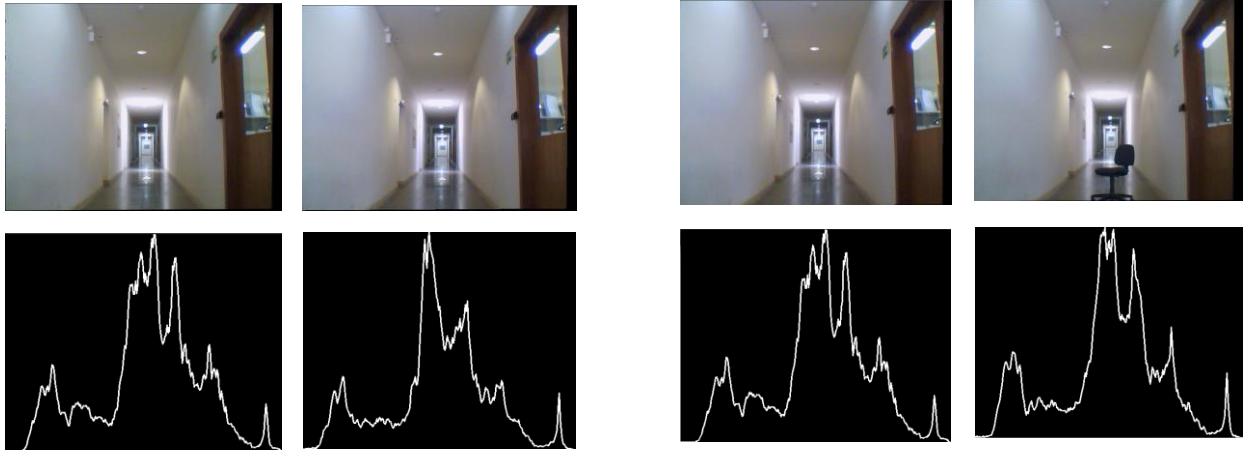


Figura 3.12 – Imagens do local A corrigidas com homografia e seus histogramas.

Mais uma vez não foi possível definir um limiar, isto porque existem casos onde imagens da mesma cena, mas sem anomalia, apresentam diferenças maiores que imagens da mesma cena, mas com anomalia. Também se utilizou a detecção de arestas e comparou-se as imagens resultantes recorrendo aos seus histogramas, mas mais uma vez sem sucesso.



Figura 3.13 – Imagens do corredor do ISR - UC tiradas com 3 segundos de diferença sem

Ao olhar para a figura 3.13 verifica-se a existência de uma grande diferença de luminosidade nas imagens capturadas com 3 segundos de diferença sem que exista nenhum deslocamento da câmara/robô, resultante do balanceamento automático do branco por parte da câmara. Este resultado é problemático no contexto deste problema, uma vez que as técnicas estudadas até então baseiam-se na informação da cor das imagens e, por isso, os resultados por elas produzidos são erróneos.

3.4. SEGMENTAÇÃO BASEADA NUM GRAFO

Numa outra tentativa, e após alguma pesquisa, encontrou-se um método de segmentação proposto por Pedro Felzenszwalb e Daniel Huttenlocher [25] que divide uma imagem em regiões tendo por base um grafo representativo da imagem. O método é baseado em selecionar arestas de um grafo, onde cada nó desse grafo é representado por um píxel, e onde píxeis vizinhos estão ligados por arestas. O peso de cada aresta mede a similaridade entre píxeis. O método mede a existência de uma fronteira entre duas regiões comparando duas quantidades: uma baseada nas diferenças de intensidades nessa fronteira e outra baseada na diferença de intensidade entre píxeis vizinhos. As diferenças de intensidades ao longo da fronteira entre duas regiões são percentualmente importantes se forem grandes relativamente à diferença de intensidade numa dessas duas regiões.

Supondo que se tem um grafo $G = (V, E)$ com vértices $v_i \in V$ (conjunto de elementos a serem segmentados) e arestas $(v_i, v_j) \in E$. Neste método, cada aresta (v_i, v_j) tem um certo peso $w((v_i, v_j))$, que é uma medida não negativa de similaridade entre os vértices ligados por ela, v_i e v_j .

A segmentação S é uma partição de V em componentes, ou regiões, $C \in S$ correspondente a uma região conectada em $G' = (V, E')$ com $E' \subseteq E$. Por outras palavras, uma segmentação é induzida por um subconjunto de arestas E . Existem diferentes formas de medir a qualidade de uma segmentação, mas no geral, o que se pretende é que os elementos que formam um componente sejam semelhantes e que elementos em diferentes componentes sejam diferentes. Significa que arestas que ligam dois vértices do mesmo componente devem ter pesos baixos, e arestas que ligam vértices de componentes diferentes devem ter pesos altos.

Em [25] inicialmente é criado um grafo com todos os píxeis da imagem onde cada píxel se liga a todos os seus píxeis vizinhos. Posteriormente, o utilizador define um limiar para o peso da aresta. Com base nesse limiar verifica-se se dois píxeis ligados pertencem ou não à mesma região. Píxeis da mesma região interligam-se entre si ligando-se ao pai correspondente - píxel encontrado com o algoritmo união-pela-raiz (*link-by-rank*) – e posteriormente utiliza-se o algoritmo de compressão de caminho (*path compression*) [5] para que todos os nós de uma região se liguem diretamente ao pai, obtendo um grafo semelhante ao da figura 3.14.

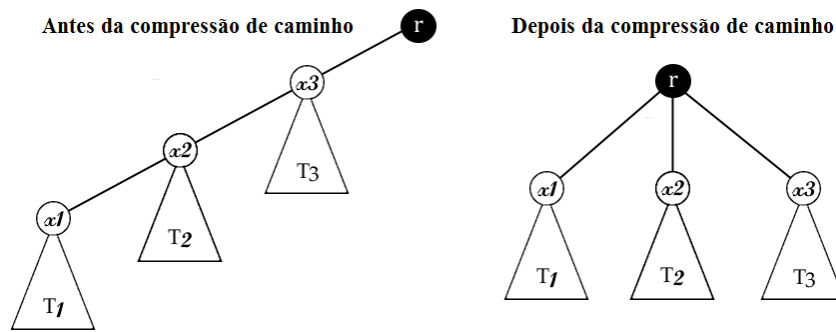


Figura 3.14 – Imagem explicativa do algoritmo de compressão de caminho.

Depois deste processamento a imagem fica dividida em diversas regiões, cada uma com um conjunto de píxeis de intensidade semelhante, como se pode verificar na figura 3.15.

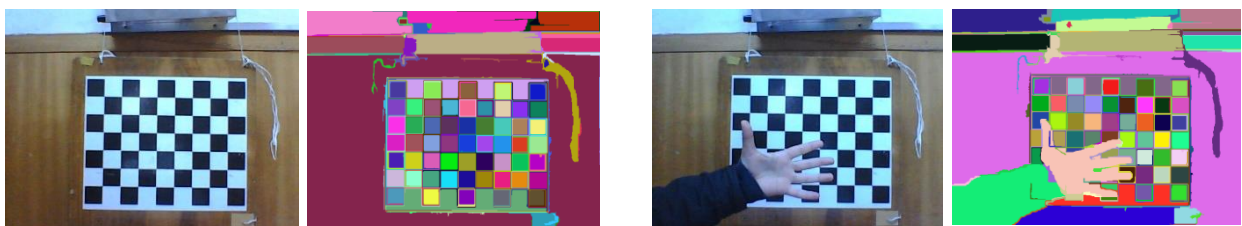


Figura 3.15 – Imagens exemplo resultantes da implementação do algoritmo de segmentação de Felzenszwalb e Huttenlocher (2004).

O código fonte deste algoritmo foi disponibilizado pelos autores em [6].

- **Alterações efetuadas ao código para medir a semelhança entre imagens**

O código anterior fornece um grafo por cada região, chamado componente. O componente tem as coordenadas do píxel pai e o número de píxeis filhos a ele ligado. Para detetar anomalias, pegando neste algoritmo, criou-se um que mede a similaridade das imagens tendo por base o número de píxeis em cada par de regiões correspondentes num par de fotografias:

1. Tendo duas imagens da mesma cena, mas tiradas em momentos diferentes, aplica-se o algoritmo da segmentação e obtém-se um grafo para cada imagem. Cada imagem por sua vez é segmentada em componentes;
2. Extraí-se a média das posições dos píxeis de cada componente (isto é, o píxel central/médio do componente);
3. Definiu-se um limiar d , por exemplo $d=50$, que maximiza a distância possível entre dois componentes correspondentes. Isto é, o centro do componente A da imagem 1 tem de ficar a uma distância absoluta menor que d do centro do componente correspondente na imagem 2.

Definiu-se ainda um limiar α para medir a similaridade entre dois componentes (por exemplo $\alpha = 0.8$) e um limiar β para medir a similaridade entre duas imagens (por exemplo $\beta = 0.8$);

4. Criaram-se estruturas de dados para armazenar os componentes. Uma para armazenar os pares de componentes correspondidos, outra para armazenar os componentes sem correspondência da imagem 1 e outra para armazenar componentes da imagem 2 que não tinham correspondência;
5. Para cada componente foi criada uma matriz de uns e zeros (Z_i), com o tamanho total da imagem, e elementos dessa matriz são um só e só se o píxel da imagem pertencer a esse componente. Ou seja, se o píxel da imagem na posição (x,y) pertence ao componente em questão, então o elemento da matriz na posição (x,y) terá o valor um;
6. Pegando em cada componente C_i da imagem 1, percorrem-se todos os componentes C_j da imagem 2 e verifica-se quais desses têm o seu píxel médio uma distância inferior a d do píxel médio do componente C_i . Desses componentes, verifica-se aquele cuja matriz Z_j tem maior número de uns nas mesmas posições que os uns da matriz Z_i . Se a relação entre o número de uns no mesmo sitio é superior a α então os componentes são semelhantes e insere-se esse par de componentes na estrutura de dados correspondente. Se após acabar o ciclo o componente C_i não tem correspondência então é inserido na estrutura de dados correspondente.
7. No final restam componentes da imagem 2 que não tiveram correspondência. Esses são inseridos na estrutura de dados correspondente.
8. No final do algoritmo têm-se as estruturas de dados preenchidas. Pegando na estrutura que armazena os pares de componentes correspondidos, verifica-se o número total de píxeis dos componentes C_j . Se a relação entre o número total de píxeis na imagem 1 e o número total de píxeis dos componentes C_j que tiverem correspondência nos componentes C_i for superior a β então as imagens são correspondentes.

Este algoritmo funcionou quando os resultados da segmentação produzida eram favoráveis, mas infelizmente os parâmetros da própria segmentação juntamente com os parâmetros deste algoritmo não podiam ser constantes para o mesmo local, isto é, parâmetros iguais, para o mesmo local, sem nenhuma anomalia, produziam segmentações diferentes. Mais uma vez acredita-se que se deve à luminosidade, como se pode observar na figura 3.16 onde são apresentados dois exemplos de segmentações em 2 locais diferentes, um com correção da homografia e com anomalia e outro sem.



Figura 3.16 – Imagens exemplo resultantes da implementação do algoritmo de segmentação de Felzenszwalb e Huttenlocher (2004); (a) imagens das segmentações de imagens do local C, com uma anomalia e correção de deslocamento; (b) imagens das segmentações de imagens do local A, sem anomalia e sem deslocamento.

Nas imagens da figura 3.16 é possível verificar que imagens semelhantes produzem segmentações muito diferentes e que imagens com anomalia produzem segmentações também elas diferentes mas, onde a anomalia não é detetada não se consegue perceber onde ocorre. Estes resultados repetem-se muitas vezes. Fotos da mesma cena sem nenhuma anomalia e sem grandes diferenças de iluminação produzem segmentações erróneas que induzem o algoritmo de similaridade em erro, e fotos da mesma cena, mas com anomalia, produzem segmentações semelhantes. Foram realizados alguns testes, ajustando ao máximo os parâmetros, mas os resultados repetiam-se e, por isso, não foi possível definir o tal limiar de separação.

Além das técnicas supracitadas, foram testadas outras, como por exemplo o compensar do balanceamento automático do branco da câmara. Para isso, encontrou-se uma razão entre as intensidades máximas de duas imagens e a todos os píxeis da imagem com menos intensidade, multiplicou-se o seu valor com a razão encontrada, isto numa tentativa de normalizar as imagens. Essa técnica também não resultou e, por isso, abordagens que trabalhem tendo por base a cor das imagens são um risco. Por isso optou-se por utilizar um algoritmo baseado em técnica de aprendizagem automática apresentado no capítulo seguinte.

3.5. SUMÁRIO

Neste capítulo deu-se a conhecer algumas das técnicas utilizadas inicialmente para a tentativa de resolução do problema. Estas técnicas infelizmente não conduziram aos resultados esperados uma vez que o *dataset* de imagens no ambiente real tem grandes variações de iluminação e em cada ponto de patrulhamento existe variação do ponto de vista de onde as fotografias são capturadas. De facto, para solucionar este problema é necessário um algoritmo robusto e capaz de lidar com estas condições.

4. SISTEMA PROPOSTO

Todas as condições adversas que se verificaram mostraram ser decisivas no funcionamento de um algoritmo para detetar anomalias. Deste modo e pegando nas abordagens estudadas, optou-se por utilizar um algoritmo de aprendizagem automática com provas dadas [36]. Neste capítulo, é apresentada uma proposta para o sistema final assim como uma breve explicação sobre aprendizagem automática e sobre a técnica usada, SVM. É também explicado o funcionamento global do sistema.

4.1. APRENDIZAGEM AUTOMÁTICA

As técnicas de aprendizagem automática permitem que os sistemas se integrem e se adaptem aos dados que tentam modelar. Estes sistemas, uma vez treinados, são usados para fazer previsões. Por outras palavras, pode dizer-se que uma vez treinados, estes sistemas são utilizados para operar sobre dados semelhantes aos que foram utilizados para a sua formação, para atingir os seus resultados. Podem ser usados para estimação ou para classificação. Em termos gerais, aprendizagem automática consiste em treinar o sistema com dados que lhe permita fazer previsões de dados do mesmo tipo.

Geralmente existem duas formas de aprendizagem: aprendizagem supervisionada e aprendizagem não supervisionada. Aplicações em que os dados de treino são compostos por vetores de entrada, juntamente com os vetores alvo correspondentes (por exemplo etiquetas) são conhecidos como problemas de aprendizagem supervisionada. Casos como o reconhecimento de dígitos, por exemplo, em que o objetivo é corresponder a cada vetor de entrada uma categoria de um conjunto de categorias discretas, são chamados problemas de classificação. Noutros problemas de reconhecimento de padrões, denominados de problemas de aprendizagem não supervisionada, os dados de treino consistem num conjunto de vetores de entrada x sem quaisquer valores de saída correspondentes e o algoritmo tem de encontrar a estrutura de dados de entrada, fazendo reconhecimento de padrões, aumentando assim a complexidade computacional. O objetivo neste tipo de problemas poderá ser o agrupamento de exemplos semelhantes dos dados de entrada, denominado de *clustering*, ou ainda a determinação de distribuição dos dados de entrada, denominada estimação de densidade.

Técnicas intermédias ocupam o espaço entre estas duas, como *reinforcement learning*, onde por exemplo um programa de computador interage com um ambiente dinâmico onde deve executar um certo objetivo, tal como a condução de um veículo sem alguém que diga se chegou ou não ao seu objetivo ou aprender a jogar um jogo jogando contra um adversário [14].

As técnicas de aprendizagem automática são bastante utilizadas em muitos problemas modernos de visão por computador como o reconhecimento de fala, motores de busca, diagnósticos médicos, locomoção de robôs, entre outras.

De entre as várias técnicas de aprendizagem automática existentes, seguem-se algumas das escolhas mais populares:

- SVM [14] – considerada uma técnica simples, mas ainda assim muito utilizada em classificadores. Baseia-se no princípio da maximização da margem, isto é, na constatação da melhor separação entre as classes no espaço. Na sua formulação inicial, SVM seria o único método capaz de encontrar o hiperplano de margem máxima que melhor separa dados em espaços de características. Contudo, esta formulação foi aumentada utilizando o *kernel trick* que permite à SVM encontrar separações de dados não lineares;
- Redes neuronais [14] – consistem em redes de nós em camadas, onde cada nó implementa uma única função paramétrica, um paradigma inspirado no funcionamento do cérebro.
- Redes Bayesianas [14] – modelos gráficos probabilísticos que representam um conjunto de variáveis aleatórias e as suas dependências condicionais através de um grafo acíclico direcionado onde os nós representam as variáveis. Quando uma variável é condicionalmente dependente de uma variável B, estas são interligadas no sentido $B \rightarrow A$. São representações de uma tabela de conjunção de probabilidades no universo do problema. Cada variável tem uma tabela de probabilidades associada, que indica a probabilidade de cada estado dessa variável ocorrer com base nas suas dependências condicionais.

Neste trabalho figuram duas situações distintas: a existência ou ausência de anomalia. Graças a isto, cada fotografia terá um estado binário, onde zero representa ausência de anomalia e um representa a presença da mesma. Desta forma, pode-se classificar cada fotografia da base de dados *a priori* e o modelo de aprendizagem da máquina não precisa de evoluir dinamicamente. Este tipo de simplicidade e a natureza do problema faz com que o foco de interesse seja SVM, por ser simples, direcionada para a classificação e por ter provas dadas na literatura [20]. Para além disso pode-se repetir este processo para outros 3 estados binários: a presença de uma pessoa, a presença de uma mala e a presença de uma porta aberta.

4.2. MÁQUINA DE VETORES DE SUPORTE

A ideia base de SVM é criar um hiperplano que separa conjuntos de dados. O hiperplano é escolhido para fornecer a máxima distância possível dos vetores das diferentes classes que se encontram mais próximas dele (vetores de suporte). Com essa separação, é possível classificar dados não conhecidos dependendo do lado do hiperplano em que eles caem. Este conceito simples tem feito da SVM um poderoso classificador e um dos mais utilizados nos últimos anos [48].

- **SVM Linear**

Assumindo que se tem o seguinte conjunto n de dados de treino $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$, onde \vec{x}_i é um vetor de dimensão p , e y_i a etiqueta correspondente ao vetor \vec{x}_i . De notar que a SVM lida com classificação binária, i.e., classificação entre duas classes (positiva e negativa).

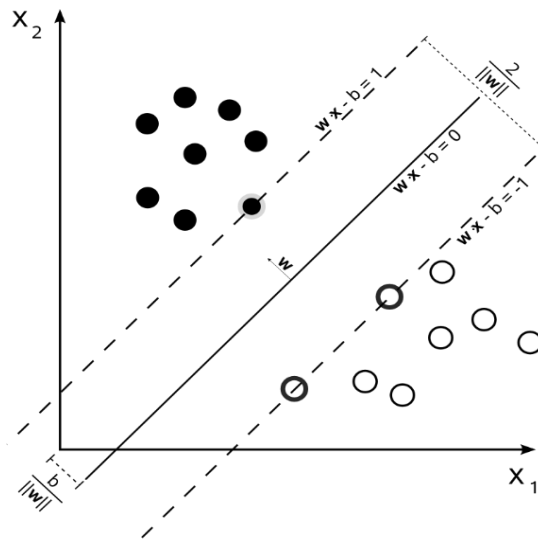


Figura 4.1 – Visualização da separação entre duas classes de dados recorrendo a um hiperplano. Fonte: Wikipédia.

Como é possível observar pela figura 3.15 o objetivo da SVM é encontrar a margem máxima do hiperplano que separa as duas classes. As amostras localizadas na margem são chamadas vetores de suporte. A formulação seguinte foi baseada em [33] e [48].

O hiperplano pode ser descrito como:

$$\vec{w} \cdot \vec{x} - b = 0$$

onde \vec{w} é o vetor normal (não necessariamente normalizado) ao hiperplano e $\frac{b}{\|\vec{w}\|}$ representa o desvio do hiperplano em relação à origem.

Supondo que os dados de treino são linearmente separáveis, podem-se selecionar duas retas paralelas que são as fronteiras do hiperplano que separam as duas classes com a máxima distância possível entre elas. A região entre essas fronteiras é chamada de margem. As fronteiras podem ser definidas como:

$$\vec{w} \cdot \vec{x} - b = 1 \quad \vec{w} \cdot \vec{x} - b = -1$$

Para prevenir que pontos caiam dentro da margem é necessário criar restrições:

$$\vec{w} \cdot \vec{x} - b \geq 1, \text{ para pontos pertencentes à classe positiva, } y_i = 1;$$

$$\vec{w} \cdot \vec{x} - b \leq -1, \text{ para pontos pertencentes à classe negativa, } y_i = -1.$$

O objetivo é maximizar a distância entre essas duas fronteiras. Essa distância é dada por:

$$largura = (\vec{x}_+ - \vec{x}_-) \frac{\vec{w}}{\|\vec{w}\|}$$

onde \vec{x}_+ é uma amostra positiva na fronteira, \vec{x}_- é uma amostra negativa na fronteira e $\frac{\vec{w}}{\|\vec{w}\|}$ é o vetor normal normalizado ao hiperplano. A partir de:

$$\vec{w} \cdot \vec{x} = 1 + b \quad \vec{w} \cdot \vec{x} = -1 + b$$

obtém-se,

$$largura = (\vec{w} \cdot \vec{x}_+ - \vec{w} \cdot \vec{x}_-) \frac{1}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|}$$

sendo necessário minimizar $\|\vec{w}\|$ para maximizar a largura. Uma vez que o problema de minimização envolve a norma, é necessário determinar a raiz quadrada. Recorre-se assim ao problema de otimização:

$$\underset{w,b}{\text{Minimizar}} \frac{1}{2} \|\vec{w}\|^2$$

O problema de otimização obtido é quadrático, cuja solução possui uma ampla e estabelecida teoria matemática. Como a função objetivo sendo minimizada é convexa, esse problema possui um único mínimo global. Problemas deste tipo podem ser solucionados com a introdução de uma

função Lagrangiana, que engloba as restrições associadas a parâmetros denominados multiplicadores de Lagrange α_i :

$$L(w, b, \alpha) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i (\vec{w} \cdot \vec{x}_i - b) - 1] \quad \text{Eq. 4.1}$$

Uma vez que o objetivo é encontrar um extremo da função, é necessário encontrar os zeros das derivadas:

$$\frac{\partial L}{\partial b} = 0 \quad e \quad \frac{\partial L}{\partial \vec{w}} = 0$$

A resolução destas equações leva a:

$$\begin{aligned} \frac{\partial L}{\partial \vec{w}} &= \vec{w} - \sum_{i=1}^n \alpha_i y_i \vec{x}_i \Rightarrow \vec{w} = \sum_{i=1}^n \alpha_i y_i \vec{x}_i \\ \frac{\partial L}{\partial b} &= - \sum_{i=1}^n \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

Substituindo os valores anteriores na eq. 4.1 obtém-se:

$$L = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j)$$

Esta forma permite a representação do problema em termos de produtos internos entre dados. É interessante observar que o problema é formulado utilizando apenas os dados de treino e as suas etiquetas. Assim que o vetor solução $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)$ do problema de maximização for encontrado, o hiperplano ótimo é dado por:

$$w^* = \sum_{i=1}^n \alpha_i^* y_i x_i$$

$$b^* = \frac{1}{2} (w^* \cdot x_r + x_s)$$

onde x_r e x_s são quaisquer vetores de suporte de cada classe, satisfazendo $\alpha_r, \alpha_s > 0$ e $y_r = -1$, $y_s = 1$.

Neste trabalho é utilizada a implementação de [7] que se baseia em [16].

4.3. HISTOGRAMAS DE ORIENTAÇÕES DE GRADIENTES

Antes de classificar qualquer tipo de objeto numa imagem (ou classificar uma como fazendo parte de uma determinada classe) é necessário criar uma representação do objeto/imagem. Uma boa representação deve ser discriminatória o suficiente para ser capaz de distinguir diferentes classes. Esta deve ser robusta ao ruído e a mudanças de iluminação, e também concisa, de modo a que o custo de computação de várias amostras não seja demasiado alto [48].

Neste trabalho utiliza-se o descritor HoG para capturar a informação global da imagem, extraíndo um simples descritor como proposto em [20]. Usar gradientes de intensidades locais é uma ótima maneira de caracterizar a forma local de um objeto. A ideia básica por detrás do descritor é essa mesma [8], a informação de formas locais geralmente é bem descrita pela distribuição dos gradientes de intensidade ou pelas direções das arestas, mesmo não tendo informação exata sobre a localização das arestas em si. Numa imagem, o gradiente é um vetor que indica o sentido e a direção na qual, por deslocamento a partir do ponto especificado, se obtém o maior aumento nos valores escalares na vizinhança.

Considerando uma imagem I , o vetor de gradiente num ponto (x,y) é dado por:

$$\nabla I(x, y) = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]$$

A magnitude do gradiente é dada por:

$$|\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

E a orientação por:

$$\theta = \arctan\left(\frac{\frac{\partial I}{\partial y}}{\frac{\partial I}{\partial x}}\right)$$

Para computar as características HoG da imagem, em primeiro lugar divide-se a imagem em pequenas subimagens denominadas células e, dentro destas, é calculado o gradiente de cada píxel formando assim um histograma de gradientes. Para compensar as alterações de iluminação,

sombras, etc., normalizam-se os histogramas, dentro de regiões constituídas pelas células, chamadas blocos. A concatenação destes histogramas normalizados origina o descritor HoG da imagem.

Neste trabalho é utilizada a implementação disponível no OpenCV [9] baseada no algoritmo implementado em [20].

4.4. VISÃO GLOBAL DO SISTEMA

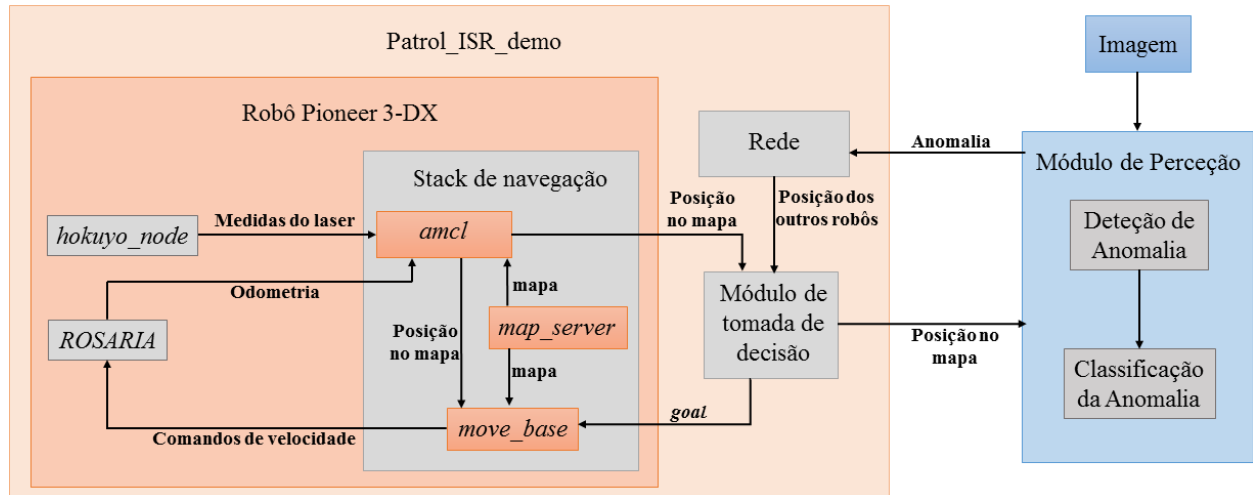


Figura 4.2 – Representação dos módulos do sistema.

Este sistema surge como continuação do protótipo desenvolvido em [44]. No protótipo existente, os robôs são dotados da capacidade de navegar autonomamente e em equipa recorrendo a um grafo representativo no ambiente, parando em pontos de interesse, mas onde não têm nenhuma ação. É no patrulhamento que se foca este trabalho, agregando vigilância ao protótipo já implementado. Na imagem da figura 4.2 estão representados os módulos utilizados no sistema. O módulo de percepção é aquele sobre o qual recai esta dissertação. Está diretamente ligado ao módulo do algoritmo de patrulhamento (coordenação-multiagente), uma vez que os robôs adquirem as fotografias quando param e acedem ao treino do local onde se encontram. Depois, e em caso de anomalia, enviam a informação ao operador humano que se situa na estação base.

Neste sistema o objetivo principal é perceber se existem ou não anomalias em cada local a patrulhar no ambiente, sendo a abordagem proposta para a resolução deste problema o uso de SVM. Posteriormente, classifica-se a anomalia utilizando a mesma técnica.

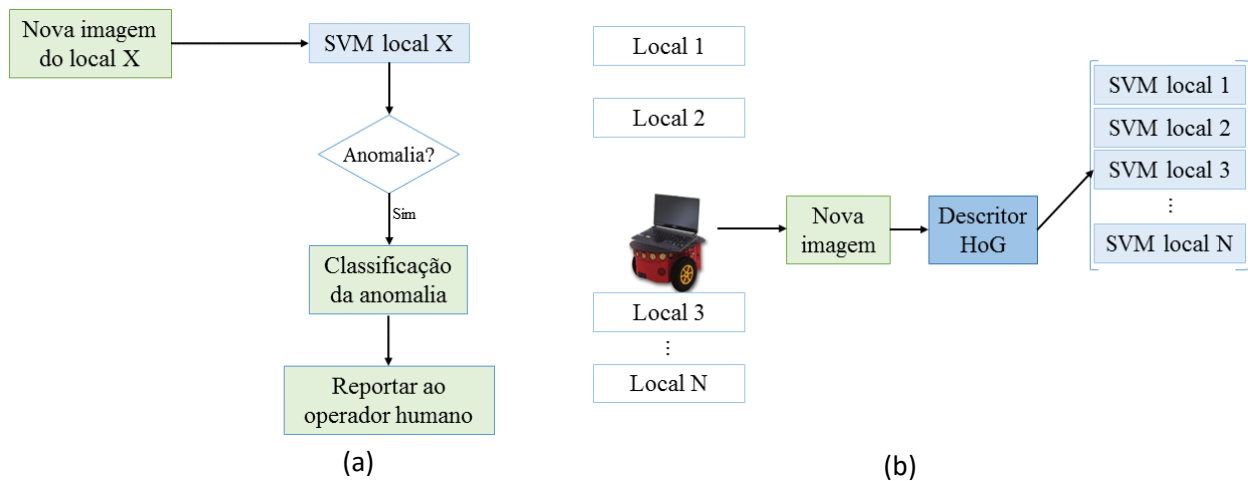


Figura 4.3 – Sistema proposto; (a) Visão global do sistema; (b) Comportamento detalhado num local exemplo (local 3).

Numa situação real de patrulhamento, o robô captura uma fotografia do local onde se encontra, e essa fotografia é comparada a um conjunto de fotografias já conhecidas do mesmo local, tiradas previamente e sobre as quais se aplicou o treino de SVM. Cada treino fica guardado com o nome do respetivo local e, graças a isso, quando o robô se encontra a patrulhar, sabe em que local se encontra e a que treino tem de aceder *a posteriori*. Extraem-se as características da nova imagem adquirida com o descritor HoG e dá-se esse vetor descritor ao treino SVM. Verifica-se se a imagem tem ou não anomalia. Caso não tenha, significa que está tudo em conformidade no ambiente e, portanto, não há necessidade de notificar nada ao operador humano. Caso contrário, classifica-se a anomalia e essa informação é enviada ao operador humano que se encontra na estação base.

Neste sentido, o sistema está dividido em duas partes principais: classificação binária para detetar se há ou não anomalia e classificação da anomalia. Em ambas, o método usado é o mesmo, embora a classificação de anomalia só ocorra quando a deteção de anomalia for positiva. Para além disso, as classes utilizadas em cada treino de SVM são diferentes. De seguida explica-se mais detalhadamente este processo.

- **Treino para cada local para detetar anomalia:**

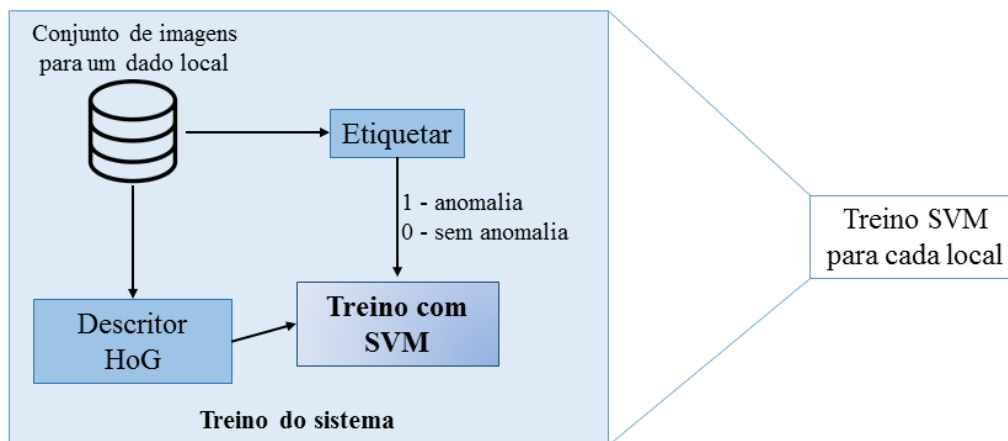


Figura 4.4 – Treino para cada local.

A primeira etapa é treinar o sistema para o conjunto de imagens de cada local. Inicialmente, extraem-se as características das imagens com o descritor HoG, e etiqueta-se todas as imagens com zeros (nas imagens sem anomalia) e uns (nas imagens com anomalia). Essas etiquetas e os vetores descritores são fornecidos à SVM que procede ao treino das imagens.

Nesta etapa é necessário ter em conta quais as anomalias que se pretendem considerar e adquirir um conjunto de imagens em que algumas representem situações anómalas e outras não. A aquisição das imagens é uma tarefa fastidiosa e morosa neste processo, uma vez que quanto maior o número de imagens, melhores são os resultados fornecidos pela SVM. Naturalmente, se o número de imagens não for suficiente, então os resultados não serão satisfatórios, sendo por isso necessário que o número de imagens seja o maior possível e, acima de tudo, que exista um grande número de imagens com as diferentes anomalias a considerar. Após concluído o treino, este será guardado e, para cada local, existirá o seu treino específico.

- **Classificação:**

O sistema proposto tem a capacidade de reconhecer as anomalias para o qual foi treinado. Esta classificação é também ela binária. Ao conjunto de imagens de cada local são atribuídos conjuntos de etiquetas. Para cada local etiqueta-se a imagem a 1 se a anomalia a considerar está presente, caso contrário a etiqueta tem o valor zero. Deste modo, de forma semelhante ao treino efetuado para separar imagens anómalas das restantes, também aqui é feito um treino binário para cada anomalia. Se a anomalia A está presente, então a imagem tem etiqueta 1, caso contrário tem etiqueta 0.

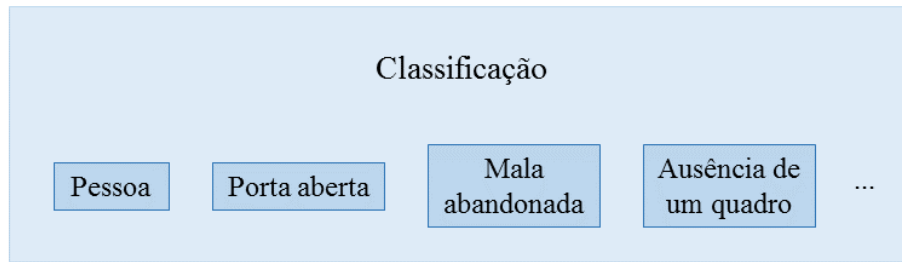


Figura 4.5 – Exemplo de anomalias a considerar.

Toda a vez que o robô tira uma fotografia do local onde se encontra, é verificado se essa imagem é ou não de uma situação anômala. Caso o resultado seja afirmativo, então o vetor característico dessa imagem passa por cada um dos treinos das anomalias até que se encontre qual a classe de anomalias a que pertence. Pode acontecer que uma anomalia não pertença a nenhuma das classes, ou ser classificada como sendo uma anomalia diferente, isto porque o sistema é treinado apenas para algumas anomalias. Esta é, de facto, uma das desvantagens deste método. É impossível ter um conjunto de imagens para todas as anomalias possíveis de aparecerem num determinado ambiente, e desta forma, uma anomalia que apareça pode ser classificada como outra, não ser classificada como anomalia, ou ser classificada como anomalia e não pertencer a nenhuma das classes treinadas.

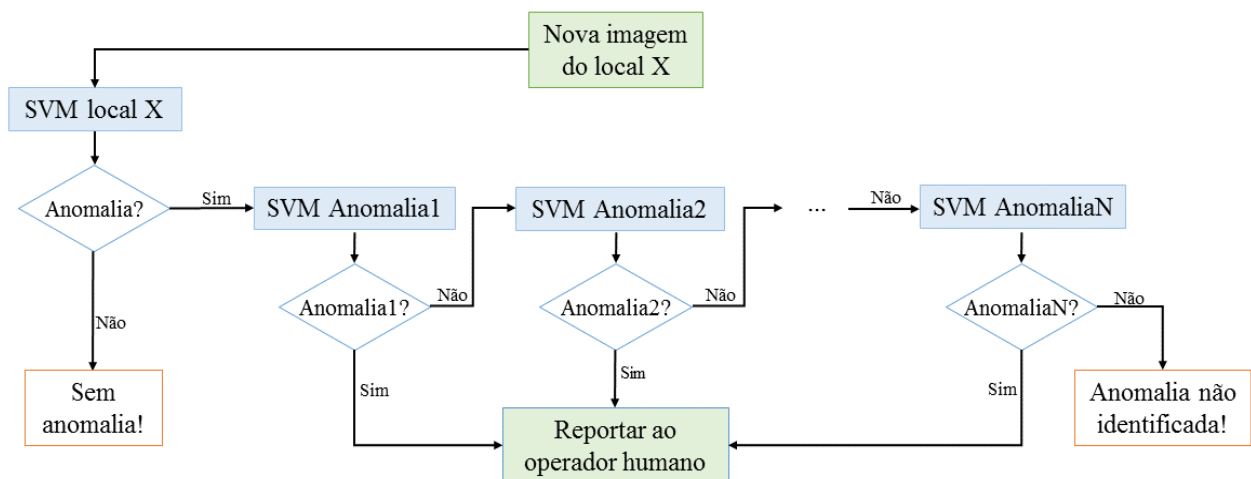


Figura 4.6 – Funcionamento do sistema na classificação das anomalias.

- **Design e princípios de implementação:**

Este projeto foi desenvolvido na linguagem de programação C++, recorrendo ao uso da biblioteca OpenCV. Paralelamente, e uma vez que o protótipo utilizado [44] foi desenvolvido em

Robot Operating System (ROS), houve a necessidade de estudar ROS e de fazer algumas alterações ao código já existente utilizando este sistema. Estas ferramentas utilizadas têm licença de utilização livre, permitindo que qualquer pessoa possa aceder livremente aos seus conteúdos. Desta forma, existe um largo conjunto de exemplos, tutoriais e documentação oficial sobre qualquer função ou pacote que se tenha intenção de utilizar. Permite que qualquer dúvida possa ser pesquisada e facilmente se encontre resposta devido ao elevado número de utilizadores destas ferramentas. Todo o código desenvolvido possui documentação interna e está desenvolvido segundo as práticas de Engenharia de Software para que possa ser utilizado posteriormente. O sistema está modulado, o que permite ao utilizador alterar as anomalias que pretende considerar e com isso os treinos da SVM, sem comprometer o funcionamento global do sistema.

4.5. SUMÁRIO

Neste capítulo foi apresentada a técnica utilizada para resolver o problema proposto, bem como todas as etapas necessárias à sua implementação, sendo possível dar uma visão acerca do funcionamento do sistema final proposto. Na próxima secção são apresentadas todas as experiências realizadas e os resultados obtidos.

5. VALIDAÇÃO EXPERIMENTAL E ANÁLISE DE RESULTADOS

Neste capítulo são demonstradas as experiências realizadas e os resultados obtidos na sua execução, desde as experiências preliminares, importantes para validar esta técnica, às experiências realizadas numa simulação real de patrulhamento. São também abordados alguns pontos fulcrais da execução destas experiências, como a aquisição do *dataset* de imagens, bem como o equipamento utilizado.

5.1. EXPERIÊNCIAS PRELIMINARES

Estas experiências foram realizadas para validar a técnica estudada. Tirou-se fotografias em três locais distintos do ISR - UC onde o deslocamento do robô foi controlado, tirou-se fotografias de uma posição inicial (0°) e simulou-se desvios angulares de 5° até 10° e de -5° até -10° , ou seja, obteve-se imagens de 5 pontos de vista distintos, mas do mesmo local. O robô manteve-se na mesma posição (x,y) mudando apenas a sua orientação. Isto permite controlar o *dataset* e verificar a robustez da técnica aquando das diferenças de ponto de vista. Das imagens capturadas, umas são de situações anómalas e outras não.

Para o local A foram adquiridas 729 fotografias, das quais 142 são sem anomalia. Foram realizados 10 testes, cada um com 50 imagens de teste e 679 imagens de treino. Para o local B foram tiradas 928 fotografias, das quais 119 são sem anomalia. Foi realizado o mesmo número de testes para este local e nas mesmas condições, isto é, foram também utilizadas 50 fotografias de teste para cada teste e 878 fotografias de treino. As fotografias utilizadas em cada teste são escolhidas de forma aleatória, e assim, as imagens de teste e de treino utilizadas em cada ensaio são diferentes.

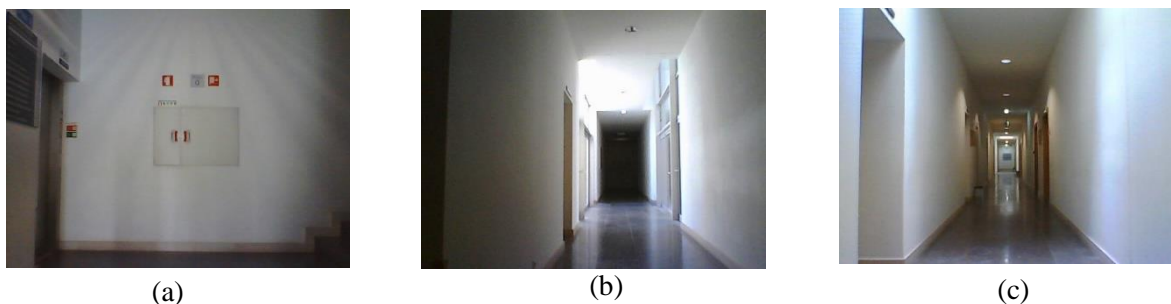


Figura 5.1 – Imagens dos três locais utilizados nesta experiência; (a) imagem do local A; (b) imagem do local B; (c) imagem do local T.

- **Resultados da experiência 1:**

Nesta experiência é utilizado o HoG global com blocos de 8x8 e células de 4x4 para criar um histograma com 10 *bins* de orientação. Para as imagens utilizadas, com tamanho 640x480 obtêm-se 4800 células. Cada teste destes demora cerca de 7 minutos a correr num computador portátil Asus N55SF.

Local A	Anomalia	Não anomalia
Anomalia	0.8933	0.1067
Não anomalia	0.175258	0.824742

Local B	Anomalia	Não anomalia
Anomalia	0.76082	0.23918
Não anomalia	0.491803	0.508197

Tabelas 5.1 e 5.2 – Matrizes de confusão para os locais A e B, respetivamente, referentes à experiência 1.

A exatidão do classificador no local A foi de 88% e no local B de 73%.

Para o local T foram tiradas 422 fotografias, das quais 95 sem anomalia. Neste local, e ao contrário do que aconteceu nos dois anteriores, as fotografias foram capturadas em diferentes dias, com diferentes condições atmosféricas (dias soalheiros e dias de chuva, durante a manhã e durante a tarde) e portanto, com diferentes condições de iluminação. Também foram realizados 10 testes, cada um com 20 imagens de teste e 402 imagens de treino. As imagens de teste e de treino utilizadas em cada teste foram escolhidas de forma aleatória.

Local T	Anomalia	Não anomalia
Anomalia	0.99375	0.00625
Não anomalia	0.175	0.825

Tabela 5.3 – Matriz de confusão para o local T referente à experiência 1.

A exatidão do classificador no local T foi de 96%. Pode concluir-se com base nos resultados que o local B é aquele que apresenta resultados menos satisfatórios. É o local cuja razão entre o número de fotografias sem anomalia e o número total de fotografias é menor e é também o local onde a luz natural incide mais diretamente, sendo que estes podem ser os fatores explicativos dos resultados.

- **Resultados da experiência 2:**

Para os três locais foi realizada a experiência anterior, mas com alteração à janela do HoG. Quanto maior é o tamanho dos blocos utilizados no descritor HoG maior é a área que o descritor descreve, e por isso, só a estrutura mais relevante e global é que é descrita, como as silhuetas. Quando a janela HoG é mais pequena, o descritor é mais detalhado, detetando os detalhes mais pequenos, como texturas. Por outro lado, a nível computacional a utilização de blocos de tamanhos maiores diminui o tempo computacional. Uma vez que os resultados obtidos no local B não foram totalmente satisfatórios, nesta experiência utilizou-se o descritor HoG global com blocos de 32x32 e células de 16x16, para tentar obter uma exatidão superior.

Local A	Anomalia	Não anomalia
Anomalia	0.936709	0.063291
Não anomalia	0.009523	0.990476

Local B	Anomalia	Não anomalia
Anomalia	0.75286	0.24714
Não anomalia	0.571429	0.428571

Local T	Anomalia	Não anomalia
Anomalia	0.993151	0.006849
Não anomalia	0.1111	0.8889

Tabelas 5.4, 5.5 e 5.6 – Matrizes de confusão para os locais A, B e T, respetivamente, referentes à experiência 2.

Para os locais A e T o aumento da janela do HoG mostrou-se vantajoso, provocando uma ligeira subida na exatidão obtida. Para o local A, a exatidão foi de 94.8% e para o local T foi de 96.5%. Ao contrário do que se esperava no local B, o aumento do tamanho de blocos do HoG não resultou num aumento da exatidão, tendo sido obtida uma exatidão de 71.2%. Este resultado pode dever-se ao facto de existir muita luz a incidir diretamente no local. A difusão da luz causa sombras, podendo induzir o algoritmo a detetar objetos onde estes não existem.

Para estes locais de teste, ainda que com movimentos da parte do robô controlados, a exatidão global é de 87.5%. Comparando os resultados obtidos com as técnicas apresentadas no capítulo 3, esta abordagem é mais robusta fornecendo melhores resultados. Para a execução destas

experiências foram utilizados 2 robôs pioneer 3DX com um portátil Asus U36S a bordo. A aquisição de fotografias foi feita a partir das webcams integradas nos portáteis.

5.2. EXPERIÊNCIAS EM SITUAÇÕES REAIS

- **Aquisição de fotografias:**

A aquisição das imagens de treino é uma tarefa fastidiosa e demorada. Tendo por base o número de fotografias tiradas para validar o uso deste algoritmo e tendo em conta os resultados obtidos nas experiências preliminares, optou-se por tirar para cada local cerca de 370 imagens. Como o objetivo trabalho é criar um protótipo funcional no dia-a-dia, foi necessário adquirir fotografias em vários momentos do dia num ambiente real. Tiraram-se fotografias de manhã, de tarde e ao início da noite em diferentes dias, com e sem as diferentes anomalias. Neste caso em concreto as anomalias a considerar são pessoas, portas abertas e malas “abandonadas”, mas claro que são a título de exemplo e num ambiente real terão de ser adaptadas aquilo que é relevante detetar.

Consideraram-se para o efeito 10 pontos do ISR - UC marcados no mapa da figura 5.2. Para a aquisição de fotografias ser o mais realista possível colocou-se o robô a correr o algoritmo de patrulhamento [44] com algumas alterações efetuadas, para que os robôs parem apenas nos pontos a considerar. No decorrer desta tarefa foram encontrados alguns obstáculos: em primeiro lugar, a instabilidade dos ecrãs dos portáteis devido às juntas de dilatação instaladas nos corredores do ISR – UC que faz com que os mesmos se movimentem; em segundo lugar, o valor inicial da tolerância angular do robô que era demasiado elevado e não permitia que este parasse em cada local de patrulhamento com orientações angulares consistentes; e em terceiro lugar, as imagens captadas por um dos portáteis que apareciam invertidas em relação às imagens captadas por outro. Durante a aquisição das fotografias foram escolhidas 6 pessoas para se obter uma maior variedade nas formas das silhuetas que iriam ser treinadas como sendo anomalias do tipo pessoa.

A aquisição de imagens foi feita de forma automática, mas ainda assim não deixa de ser uma tarefa demorada. Este conjunto de imagens é real, com a exceção das imagens de situações anómalas, que embora sendo reais possuem anomalias artificialmente colocadas, e por isso quando utilizado nos testes de simulação fornece resultados reais. Foi necessário dar cerca de 370 voltas ao ISR - UC, o que faz um total de aproximadamente 30 km. Tendo em conta que cada 10 voltas demoram cerca de uma hora foram necessárias aproximadamente 37h para tirar o *dataset* pretendido. Uma vez que o robô não tem uma disponibilidade de 100%, porque fica sem bateria, ou é necessário noutras experiências e uma vez que não se torna possível tirar fotografias durante muitas horas seguidas, esta foi uma tarefa que teve de ser repartida por aproximadamente 5 dias.

As fotografias capturadas encontram-se invertidas horizontalmente. Estes factos constituem uma desvantagem uma vez que, para que o sistema seja o mais robusto possível, o número de fotografias e a variedade das fotografias tem de ser a maior possível.

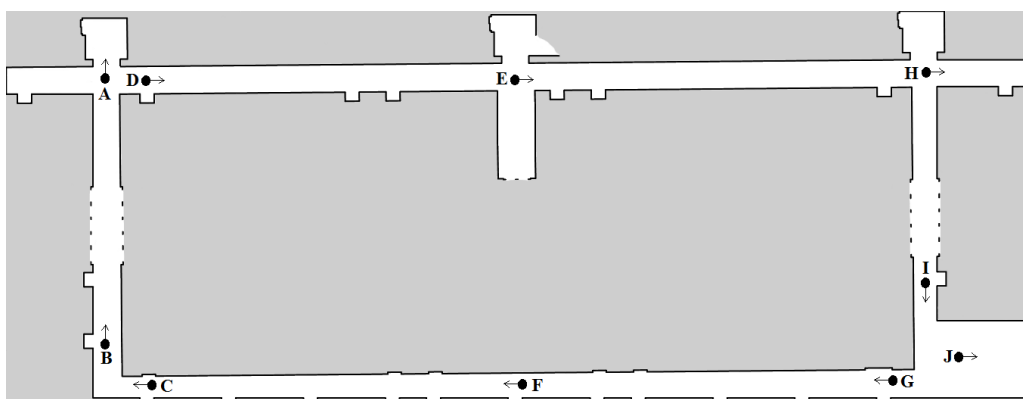


Figura 5.2 – Mapa do ISR - UC onde estão marcados os 10 locais onde cada robô para e as direções em que os robôs param e captam as fotografias.

Começando por exemplo no ponto H os robôs seguem a rota H-I-J-G-F... e tiram fotografias em cada um dos locais assinalados na figura 5.2.



Figura 5.3 – Imagens exemplo de 5 locais distintos do ISR - UC; (a) imagem do local F sem anomalia; (b) imagem do local G com anomalia (pessoa); (c) Imagem do local H com anomalia (porta aberta); (d) imagem do local I com anomalia (mala); (e) imagem do local J com anomalia (pessoa).

	A	B	C	D	E	F	G	H	I	J
Sem anomalia	208	217	221	211	198	193	199	212	221	198
Pessoa	101	99	91	110	108	106	126	99	109	104
Mala	45	42	47	33	42	49	40	37	45	35
Porta aberta	/	/	/	/	/	/	/	33	/	33
TOTAL	354	358	359	354	348	348	365	381	375	370

Tabela 5.7 – Tabela com o número de fotografias tiradas para cada local.

• Resultados da experiência 1:

Nesta primeira experiência com imagens reais é utilizado o descritor HoG global com blocos de 8x8 e células de 4x4 para criar um histograma com 10 bins de orientação. Para as imagens utilizadas, com tamanho 640x480 obtêm-se 4800 células.

Para cada local foram realizados 10 testes, cada um com 20 imagens de teste e as restantes de treino. Para cada teste as imagens foram selecionadas de forma aleatória. No gráfico da figura 5.4 são visíveis as percentagens de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos obtidos com os testes efetuados.

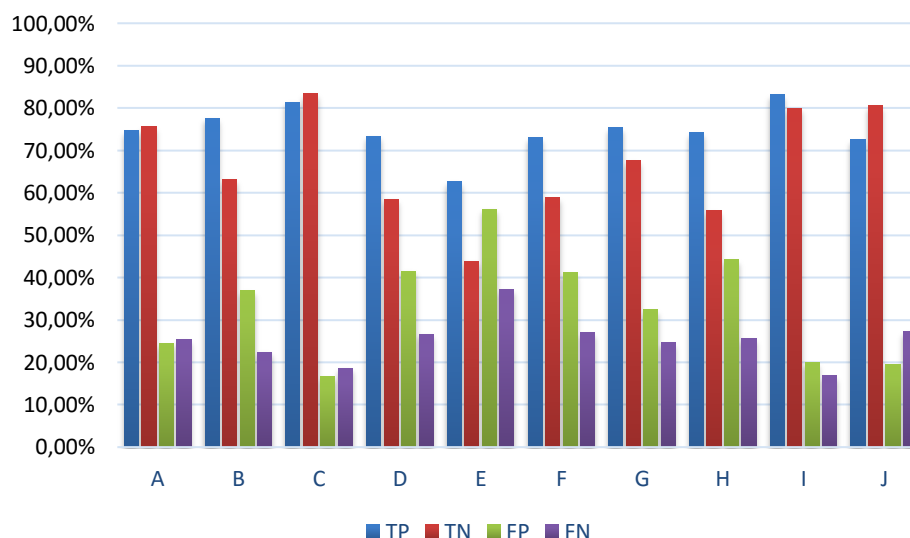


Figura 5.4 – Gráfico com os resultados da experiência 1 utilizando dados reais; apresentação do número de verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN) para cada local.

	A	B	C	D	E	F	G	H	I	J
Exatidão (%)	75	71.5	82	68.5	55	67	72.5	65.5	82	75.5

Tabela 5.8 – Tabela com a exatidão média obtida em cada local.

Dos resultados da figura 4.1 e da tabela 5.8 verifica-se que o local E é o que apresenta piores resultados. Este local não tem muita luz natural a incidir diretamente, mas o conjunto de fotografias tiradas apresenta diferenças acentuadas nos pontos de vista, provocados pelo erro da navegação. Essas diferenças podem induzir em erro, criando muitos falsos positivos. Os resultados obtidos estão um pouco abaixo das expectativas, mas os erros de navegação, as condições de luminosidade e o número de amostras são fatores que influenciam os resultados fornecidos pelo algoritmo. Estes locais têm amostras conseguidas em diferentes momentos do dia, originando uma amostragem com grande variedade de luminosidade. Reunidos todos os fatores, variação de luminosidade, diferentes pontos de vista e diferentes posições das anomalias, o *dataset* conseguido não é totalmente homogêneo, e por isso, os resultados obtidos diferem daqueles que seriam esperados.

Global	Anomalia	Não anomalia
Anomalia	0.7482	0.2518
Não anomalia	0.3328	0.6672

Tabela 5.9 – Matriz de confusão do sistema de detecção de anomalias global.

O sistema global de detecção de anomalias tem uma exatidão média de 71.45%.

- **Resultados da experiência 2:**

Para o mesmo *dataset* foi realizada novamente a experiência anterior, mas utilizando um HoG global com blocos de 32x32 e por isso, células de 16x16, na esperança de se obterem resultados um pouco melhores.

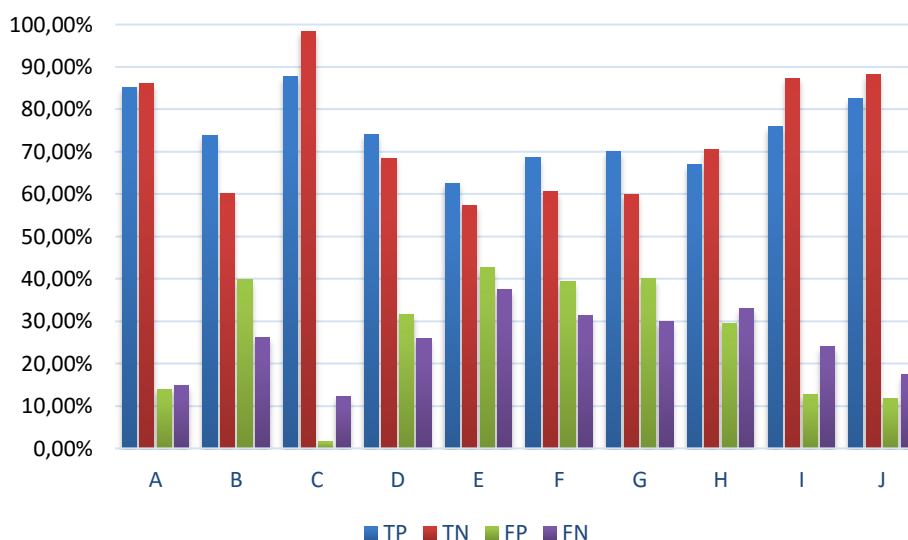


Figura 5.5 – Gráfico com os resultados da experiência 2 utilizando dados reais; apresentação do número de verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN) para cada local.

	A	B	C	D	E	F	G	H	I	J
Exatidão (%)	85.5	68.5	91	72	61	66	66.5	66.5	79.5	85

Tabela 5.10 – Tabela com a exatidão média obtida em cada local.

Global	Anomalia	Não anomalia
Anomalia	0.7473	0.2527
Não anomalia	0.2628	0.7372

Tabela 5.11 – Matriz de confusão do sistema de detecção de anomalias global.

O aumento do tamanho dos blocos utilizados diminuiu em média o número de falsos positivos, aumentando assim a exatidão média de quase todos os locais. Porém nos locais B e G a exatidão média baixa. O local B é um local com muita luz natural incidente e por isso a variação de luminosidade nas amostras pode induzir falsos positivos. Como anteriormente mencionado, quanto maior é o tamanho dos blocos utilizados no descritor HoG maior é a área que o descritor descreve, e por isso, só a estrutura mais relevante e global é que descrita, como silhuetas, e quando

existe variação de luminosidade, esta pode induzir em erros. No local G também se verifica esse problema mais uma vez devido à incidência de luz natural. Porém no local J, onde existe muita luz, isto não acontece. A luz que incide nesse local é menos direta, embora as fotografias sejam capturadas contra ela, existindo menos reflexo nas paredes e no chão, não causando tantas sombras como nos outros locais. Com o aumento do tamanho dos blocos a exatidão média do sistema é 74.35%.

- **Uso da homografia para compensar as diferenças de pontos de vista**

Numa tentativa de eliminar as diferenças de pontos de vista das imagens de uma mesma cena, optou-se por utilizar a homografia. Esta técnica, referida e usada anteriormente forneceu bons resultados para corrigir o mesmo problema (Capítulo 3).

Para tal, foi escolhida uma imagem central em cada local que, ao ser usada como referência, minimiza os desvios existentes nas imagens transformadas em relação a ela. Quando uma imagem é transformada em relação a outra, a imagem resultante possui píxeis a negro que representam a informação descartada. Se a imagem de referência estiver muito deslocada das restantes, o número de píxeis a preto vai ser elevado. Uma vez que neste caso o robô pode parar mais à direita ou mais à esquerda, essas barras aparecem no lado direito de algumas imagens e no lado esquerdo de outras. Ao escolher a imagem mais central possível, tanto as barras à direita como as barras à esquerda vão ser minimizadas.

Infelizmente, a função de homografia disponibilizada no OpenCV não funcionou corretamente para certos locais dada a variação de iluminação presente nos mesmos, o que torna a deteção de pontos característicos comuns entre pares de imagens uma tarefa não fiável. Ainda assim, o *dataset* usado nos locais onde a homografia apresentou bons resultados foi o obtido após as imagens sofrerem transformação em relação à imagem de referência. Nos restantes locais, o treino foi feito com o *dataset* usado nas experiências anteriores. O objetivo desta experiência é perceber o impacto causado pela homografia na exatidão do sistema.

Uma vez que o aumento do tamanho dos blocos de HoG forneceu melhores resultados, usou-se blocos de 32x32 nesta experiência e utilizou-se a homografia nos locais A, C, H e I, locais onde funcionou como esperado.

Global	Anomalia	Não anomalia
Anomalia	0.7473	0.2527
Não anomalia	0.2628	0.7372

Tabela 5.12 – Matriz de confusão do sistema de detecção de anomalias global, após se usar a homografia.

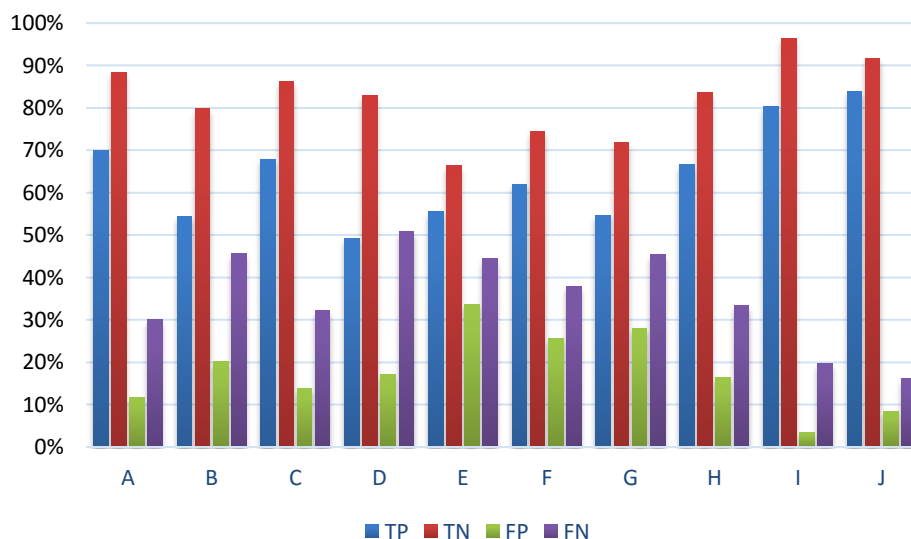


Figura 5.6 – Gráfico com os resultados da experiência 2 utilizando dados reais e homografia; apresentação do número de verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN) para cada local após se usar homografia.

Com o uso da homografia a exatidão do sistema foi 75%. A homografia não forneceu resultados significativamente melhores em relação aos obtidos anteriormente. Seria de esperar que pelo menos nos locais onde a homografia é implementada, a exatidão aumentasse, e isso de facto não se verifica. Esta pode ser uma forma de contornar o problema das diferenças de pontos de vistas das imagens em ambientes controlados. Neste caso, e uma vez que esse deslocamento por vezes é na ordem dos 60 píxeis (ou noutros casos maior), perde-se parte da informação das imagens. A informação que é perdida pode conter anomalias que serão assim descartadas e por isso, o uso da homografia resolve um obstáculo mas introduz outro.

No sistema final os treinos de cada local são realizados com as imagens originais (sem homografia) e com o HoG com blocos de 32x32 tendo por base os resultados da experiência 2 mostrados na figura 5.5 e nas tabelas 5.10 e 5.11.

5.3. CLASSIFICAÇÃO DAS ANOMALIAS

Um objetivo secundário deste trabalho prende-se com a classificação das anomalias detetadas. O objetivo é englobar o classificador das anomalias no sistema, quando o detetor de anomalias der positivo.

Numa fase inicial, foram realizadas experiências para verificar se a técnica funciona como classificador, mesmo em ambientes com muitas variações.

O objetivo desta experiência é treinar 3 classificadores binários para classificar as anomalias. Para cada local foram utilizadas apenas as imagens de situações anómalas, e foram construídos 3 conjuntos de etiquetas, um para cada treino. Para treinar o classificador de anomalia como sendo pessoa, etiquetaram-se as imagens com pessoas a 1 e as restantes a 0. Um procedimento análogo foi utilizado para efetuar os outros treinos. Posteriormente, na aquisição de uma nova imagem, esta é passada ao treino SVM para detetar anomalias e posteriormente passa pelo classificador de pessoas. Se for detetada uma pessoa, o resultado é devolvido. Se não for detetada pessoa, a imagem segue para o treino da classificação de anomalias do tipo mala. Se for mala, o resultado é devolvido, caso contrário a imagem segue para o treino classificação de anomalias do tipo portas abertas. Se for detetada porta aberta o resultado é devolvido. Se nada for detetado, então o sistema não reconhece a anomalia e devolve o resultado como sendo uma anomalia não identificada.

Global	Pessoa	Mala	Porta Aberta	Não Identificada
Pessoa	0.923077	0.0615385	0	0.0153846
Mala	0.407407	0.555556	0	0.037037
Porta Aberta	0.333333	0	0.333333	0.333333

Tabela 5.13 – Matriz de confusão para o classificador de anomalias.

Para cada local foram realizados 10 testes, cada um com 10 imagens de teste, escolhidas aleatoriamente. Foi utilizado o descritor HoG com blocos 32x32 e o classificador foi treinado apenas com as imagens anómalas existentes no *dataset* obtido e as imagens de teste fornecidas ao algoritmo eram também elas anómalas. Através dos resultados obtidos pode concluir-se que o sistema deteta pessoas com alguma exatidão, o que era expectável, mas a deteção de malas não é tão boa quanto o esperado. De notar que em muitas das fotografias as malas estão parcialmente ocultadas e como são objetos pequenos são mais difíceis de distinguir.

Só dois locais é que possuem portas abertas, o local H e o local J. A porta existente no local J é de vidro, como se pode ver na figura 5.3 (e) e por isso quando aberta não causa uma diferença significativa no ambiente. O classificador obteve uma exatidão média de 77%.

Além dos erros de navegação e das diferenças de luminosidade, ocorrem situações como as das figuras 5.7 e 5.8.



Figura 5.7 – Imagem do local C com uma anomalia (mala).

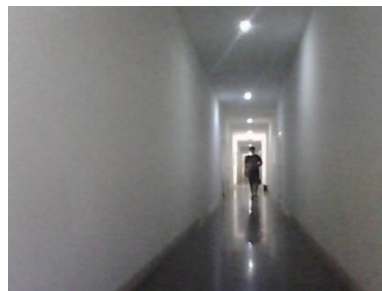


Figura 5.8 – Imagem do local D com uma anomalia (pessoa).

Na figura 5.7 a anomalia está parcialmente ocultada, no entanto a imagem é etiquetada como sendo de uma situação anômala, o que acontece também na figura 5.8 onde a anomalia (pessoa) está muito distante. Este tipo de imagens aparece regularmente no *dataset* podendo induzir o algoritmo em erro, uma vez que a SVM pode não conseguir fazer uma separação dos dados corretamente, e por isso, detetar anomalia quando ela não existe, ou não detetar quando existe.

O sistema proposto, quando em condições relativamente controladas produz os resultados esperados, mostrando-se robusto quando o robô adquire imagens do mesmo local, tiradas na mesma posição (x,y) mas com ligeiras alterações na orientação. Quando os locais têm muita luz natural refletida, criando sombras e reflexos, o algoritmo produz resultados menos exatos. Ao adicionar o movimento do robô, estes resultados ficam aquém do expectável. Ainda assim, com alguns ajustes a nível do *dataset* e do algoritmo de navegação, é expectável que os resultados produzidos sejam mais exatos que os atuais.

- **PROTÓTIPO FINAL**

Atualmente, para cada local, encontram-se treinados e guardados em ficheiros XML os quatro treinos SVM supracitados. Cada vez que o robô pára num determinado sítio adquire uma fotografia, esta é comparada com o primeiro treino SVM, de forma a que o robô possa inferir se existe ou não anomalia. De seguida, e em caso afirmativo, essa fotografia é comparada com cada um dos treinos de classificação de anomalias até que a anomalia presente seja classificada.

Na figura 5.9 está ilustrado o funcionamento esperado do protótipo numa situação de patrulhamento.



(a)



(b)

```
Robot [ID = 0] is Active!
[ INFO] [1473014132.064350249]: All Robots GO!
[ INFO] [1473014137.065108935]: Received message from: /
[ INFO] [1473014137.084913339]: Advertised foreign as /w
with type std_msgs/Int8MultiArray
Robot 0 reached Goal 7.
[0] Robot 0 reached Goal 8.
Robot 0 detected a person at place I!
```

(c)

Figura 5.9 – Ilustração do funcionamento esperado do protótipo numa situação real de patrulhamento.
(a) imagem da cena; (b) imagem capturada pelo robô; (c) informação fornecida à estação base.

Na figura 5.9 está retratado o local I numa situação de patrulhamento com o protótipo a funcionar. O robô adquire uma fotografia do local, verifica se tem ou não anomalia e caso exista, a anomalia é classificada. Em caso de anomalia, a informação é enviada ao operador humano que se encontra na estação base.

5.4 SUMÁRIO

Neste capítulo foram apresentados e discutidos os resultados obtidos com esta técnica e foram explicadas as experiências realizadas para validação do sistema proposto. Foi ainda explicado em que condições é que o *dataset* foi adquirido. Validaram-se as componentes individuais do sistema, nomeadamente as componentes de deteção e classificação, e deste modo foi possível verificar a sua exatidão.

6. CONCLUSÃO

Este foi um trabalho desafiante e motivador. Desde o início o tema mostrou ser apaixonante e com um propósito útil: criar um protótipo de patrulhamento que pudesse ser usado em ambientes reais e dinâmicos, como galerias de arte. Desde cedo se soube que este trabalho teria de passar pela aplicação de um algoritmo de visão por computador para implementar uma solução viável. Contudo, as soluções estudadas inicialmente mostraram-se insuficientes para resolver o problema. Começou-se por estudar subtração de fundo, mas percebeu-se que por causa da navegação, as imagens de um mesmo local tinham pontos de vista diferentes. Posteriormente, estudaram-se técnicas de correspondência de imagens e abordagens para medir a similaridade de duas imagens de modo a encontrar um limiar de separação entre imagens de situações anómalas e imagens sem anomalia. Também estas técnicas mostraram não ser robustas às condições existentes e por isso não forneceram os resultados esperados.

É certo que este não foi de todo um percurso simples e ao longo do desenvolvimento surgiram muitas adversidades e muitos obstáculos para se contornarem. Numa primeira fase foi necessário lidar com o problema do deslocamento, recorrendo à homografia. Depois dessa adversidade estar relativamente controlada foi necessário lidar com o grande problema da visão por computador: variabilidade das condições de iluminação e o seu impacto no funcionamento dos algoritmos de processamento de imagem. Os algoritmos estudados que usam a cor para processar as imagens e para comparar um par delas sofrem com as diferenças de luminosidade e fornecem resultados erróneos.

Este percurso foi de facto atribulado, cheio de avanços e recuos. Quando uma técnica parecia que ia funcionar e era promissora, depois de testada mostrava que mais uma vez não funcionava e que a pesquisa tinha de continuar. No entanto, estes avanços e recuos foram essenciais, uma vez que permitiu aprender muita coisa: aprendeu-se a gerir emoções, a controlar o desgaste de procurar e não encontrar, mas sobretudo trouxe uma grande bagagem de conhecimento e de partilha.

A solução apresentada é o culminar de todas as partilhas, da pesquisa efetuada e das tentativas não tão bem-sucedidas. É certo que os resultados apresentados estão longe de estar perfeitos, carecem de melhorias futuras, como qualquer trabalho científico, embora apresentem resultados melhores que as outras abordagens. A ciência permite isso, que se investigue, que se aprenda, que se aplique e que se melhore, pois está sempre em constante transformação.

Os resultados obtidos com SVM, embora melhores que os anteriores, não são os expectáveis. Em primeiro lugar é necessário um grande número de fotografias, em segundo lugar a

luminosidade provoca sombras que induzem o algoritmo em erro, e em terceiro lugar os erros resultantes da navegação também não permitem que o *dataset* obtido seja homogêneo.

6.1. TRABALHO FUTURO

Como todos os trabalhos científicos, também este pode ser melhorado no futuro. Os resultados obtidos com a solução apresentada não foram os esperados, mas ainda assim, e dadas as adversidades encontradas, conseguiu-se encaminhar o projeto a bom porto. No futuro e, de modo tentar mitigar os erros provocados pela navegação, pode-se alterar o algoritmo de navegação para que as posições onde o robô pára sejam ainda mais exatas. Podem também ser utilizados outros tipos de câmaras, nomeadamente câmaras RGBD (como é o caso da Kinect) que fornecem mais informação que imagens monoculares.

Dadas as limitações temporais, o *dataset* adquirido tinha fotografias com elevada variação. Uma das formas de criar um *dataset* mais heterógeno é adquirindo o maior número de fotografias possível e com isso distribuir as anomalias pela maior área possível dos diferentes locais. A aquisição de um maior número de fotografias pode resolver também o problema da oclusão parcial das anomalias, uma vez que quantas mais amostras nas mesmas condições forem dadas ao algoritmo, mais fácil se torna a aprendizagem.

Podem também ser testados outros classificadores e outros descritores, como o descritor HoG local para tentar aumentar a exatidão do sistema na classificação das anomalias. Para aplicações reais e para que o patrulhamento seja feito em todo o ambiente pode-se optar por trabalhar com vídeo em vez de imagens e assim ter informação atualizada em todos os instantes.

O sistema desenvolvido permite classificar 3 anomalias diferentes, mas para uma aplicação real seria útil abranger o maior número de anomalias, e para isso poder-se-ia utilizar um classificador *multiclass*. Estas são algumas sugestões de melhorias que podem tornar este sistema mais robusto e confiável.

7. BIBLIOGRAFIA

- [1]. Videovigilância.
http://www.img.lx.it.pt/~fp/cav/ano2007_2008/MERC/Trabalho_11/CAV_Grupo11_Sistemas%20de%20Videovigilancia/Video%20Surveillance/Video_Surveillance/One_Nation_Under_CCTV.html. Último acesso: 01/05/2016.
- [2]. Videovigilância. <http://obp.blogs.sapo.pt/14528.html>. Último acesso: 03/05/2016.
- [3]. Ligação para a biblioteca OpenCV: <http://opencv.org>. Último acesso: 06/08/2016.
- [4]. Código online da EMD: <http://robotics.stanford.edu/~rubner/emd/default.htm>. Último acesso: 18/02/2016.
- [5]. PDF com breve explicação do algoritmo de união-por-raiz e de compressão de caminho.
<https://www.cs.princeton.edu/courses/archive/spring13/cos423/lectures/UnionFind.pdf>.
Último acesso: 24/04/2016.
- [6]. Código online da Segmentação proposta por Felzenszwalb e Huttenlocher:
<http://cs.brown.edu/~pff/segment/>. Último acesso: 22/04/2016.
- [7]. SVM no OpenCV:
http://docs.opencv.org/2.4/modules/ml/doc/support_vector_machines.html#cvsvm-train.
Último acesso: 17/08/2016.
- [8]. PDF sobre HoG.
https://www.scss.tcd.ie/Gerard.Lacey/Gerard_Lacey_Homepage/CS4053_Course_files/Lecture%20009%20HOG-HOF.pdf. Último acesso: 16/08/2016.
- [9]. HoG no OpenCV. http://docs.opencv.org/2.4/modules/gpu/doc/object_detection.html#gpu-hogdescriptor. Último acesso: 17/08/2016.
- [10]. Henrik Andreasson, Martin Magnusson and Achim Lilienthal. “Has something changed here? Autonomous difference detection for security patrol robots”. In *IEEE International Conference on Intelligent Robots and Systems*, pages 3429–3435, October – November 2010. doi: 10.1109/IROS.2007.4399381.
- [11]. Carmen Au, Sandra Skaff and James Clark. “Anomaly Detection for Video Surveillance Applications”. In *18th International Conference on Pattern Recognition (ICPR’06)*, 4:18–21, 2006. doi: 10.1109/ICPR.2006.273.
- [12]. H. Bay, A. Ess, T. Tuytelaars and L. Van Gool. “SURF: Speeded up robust features”. In *Computer Vision Image Understanding*, 110(3):346–359, June 2008. doi: 10.1016/j.cviu.2007.09.014.
- [13]. Patrick Beeson, Nicholas Jong and Benjamin Kuipers. “Towards autonomous topological place detection using the extended voronoi graph”. In *International Conference on Robotics and Automation (ICRA ’05)*, pages 4373 – 4379, April 2005.

- [14]. C. M. Bishop. *Pattern Recognition and Machine Learning*, 2006. ISBN: 10: 0-387-31073-8.
- [15]. Varun Chandola, Arindam Banerjee and Vipin Kumar. “Anomaly Detection: A Survey”, University of Minnesota, 2007.
URL: <http://cucis.ece.northwestern.edu/projects/DMS/publications/AnomalyDetection.pdf>
- [16]. Chih-Chung Chang and Chih-Jen Lin. “LIBSVM: A library for support vector machines”. In *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. URL: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
- [17]. Sen-Ching Cheung and Chandrika Kamath. “Robust Techniques for Background Subtraction in Urban Traffic Video”. In *IS&T/SPIE’s Symposium on Electronic Imaging*, pages 881–892, 2004. doi: 10.1117/12.526886.
- [18]. Robert T. Collins, Alan J. Lipton and Takeo Kanade. “Introduction to the special section on video surveillance”. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), August 2000. doi: 10.1109/TPAMI.2000.868676.
- [19]. G. Csurka, C. Dance, L. Fan, J. Willamowski and C. Bray. “Visual Categorization with Bags of Keypoints”. In *ECCV International Workshop on Statistical Learning in Computer Vision*, Prague, 2004.
- [20]. Navneet Dalal and Bill Triggs. “Histograms of Oriented Gradients for Human Detection”. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, 1(3):886–893, 2005. doi: 10.1109/CVPR.2005.177.
- [21]. P. Drews, P. Núñez, R. Rocha, M. Campos and J. Dias. “Novelty detection and 3d shape retrieval using superquadrics and multi-scale sampling for autonomous mobile robots”. In *IEEE International Conference on Robotics and Automation (ICRA’10)*, pages 3635–3640, May 2010. doi: 10.1109/ROBOT.2010.5509405.
- [22]. Elan Dubrofsky. “Homography Estimation”, Master’s essay for the degree of Master of Science in Computer Science, University of British Columbia, Vancouver, Canada, March 2009. doi: 10.1117/1.3364071.
- [23]. Ahmed Elgammal, David Harwood and Larry Davis. “Non-parametric model for background subtraction”. In *Computer Vision - ECCV, 1843:751–767*, Dublin, Ireland, 2000. doi: 10.1007/3-540-45053.
- [24]. Yehuda Elmaliach, Noa Agmon and Gal A. Kaminka. “Multi-robot area patrol under frequency constraints”. In *Annals of Mathematics and Artificial Intelligence*, 57(3):293–320, 2009. doi: 10.1007/s10472-010-9193-y.
- [25]. P. Felzenszwalb and D. Huttenlocher. “Efficient graph-based image segmentation”. In *International Journal of Computer Vision*, 59(2):167–181, September 2004. doi: 10.1023/B:VISI.0000022288.19776.77.
- [26]. Kristen Grauman and Trevor Darrell. “Efficient image matching with distributions of local invariant features”. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, 2:627–634, June 2005. doi: 10.1109/CVPR.2005.138.

- [27]. Hayit Greenspan, Jacob Goldberger and Lenny Ridel. “A Continuous Probabilistic Framework for Image Matching”. In *Computer Vision and Image Understanding*, 84(3):384–406, September 2001. doi: 10.1006/cviu.2001.0946.
- [28]. Maha Hasso and Ruaa Elyas. “Fast Image Registration Based on Features Extraction and Accurate Matching Points for Image Stitching”. In *International Journal of Computer Science Issues (IJCSI)*, 11(5):138–142, September 2004. ISSN: 1694-0814.
- [29]. W. Hu, T. Tan, L. Wang and S. Maybank. “A survey on visual surveillance of object motion and behaviors”. In *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 34(3):334–352, 2004. doi: 10.1109/TSMCC.2004.829274.
- [30]. Boyoon Jung and Gaurav Sukhatme. “Detecting moving objects using a single camera on a mobile robot in an outdoor environment”. In *International Conference on Intelligent Autonomous Systems*, pages 980–987, Amsterdam, Netherlands, March 2004.
- [31]. Paulo Drews Jr and Mario Campos. “Detecção de Mudanças e Recuperação de Forma em Mapas 3D Baseados em Nuvens de Pontos”, Universidade Federal do Minas Gerais, 2010.
- [32]. Jonathan de Lima et al. “Comparação de Histogramas de Imagens Digitais para Determinação de Similaridade em Sementes de Milho”. In *Revista de Engenharia E Tecnologia*, 4(2):106–112, 2012.
- [33]. Ana Lorena & André de Carvalho. “Uma Introdução às Support Vector Machines”. In *Revista de Informática Teórica E Aplicada*, 14(2):43–67, 2007.
- [34]. David G. Lowe. “Object recognition from local scale-invariant features”. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2(8):1150–1157, 1999. doi: 10.1109/ICCV.1999.790410.
- [35]. Markos Markou and Sameer Singh. “Novelty detection: A review - Part 1: Statistical approaches”. In *Signal Processing*, 83(12):2481–2497, July 2003. doi: 10.1016/j.sigpro.2003.07.018.
- [36]. Markos Markou and Sameer Singh. “Novelty detection: A review - Part 2: Neural network based approaches”. In *Signal Processing*, 83(12):2499–2521, July 2003. doi: 10.1016/j.sigpro.2003.07.019.
- [37]. Stafford Michahial et al. “Automatic Image Mosaicing Using Sift, Ransac and Homography”. In *International Journal of Engineering and Innovative Technology (IJEIT)*, 3(10):247–251, April 2014. ISSN: 2277-3754.
- [38]. A. Monnet, A. Mittal, N. Paragios and V.Ramesh. “Background modeling and subtraction of dynamic scenes”. In *IEEE International Conference on Computer Vision*, 2:1305–1312, 2003. doi: 10.1109/ICCV.2003.1238641.
- [39]. Swathi Nair and Thamizharasi A. (2014). “Image Matching Using Invariant Local Features”. In *International Journal on Innovative Research in Science, Engineering and Technology (IC-IASET 2014)*, 3(5):297–302, Kerala, India, July 2014.
- [40]. P. Núñez, P. Drews Jr, A. Bandera, R. Rocha, M. Campos and J. Dias. “Change Detection in 3D Environments Based on Gaussian Mixture Model and Robust Structural Matching for

- Autonomous Robotic Applications”. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10)*, pages 2633–2638, October 2010. doi: 10.1109/iros.2010.5650573.
- [41]. Massimo Piccardi. “Background subtraction techniques: a review”. In *IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, 4:3099–3104, 2004. doi: 10.1109/ICSMC.2004.1400815.
- [42]. Charles Pippin, Henrik Christensen and Lora Weiss. “Performance based task assignment in multi-robot patrolling”. In *Symposium on Applied Computing (SAC'13)*, 70, 2013. doi: 10.1145/2480362.2480378.
- [43]. M. Pontil and A. Verri. “Support Vector Machines for 3D Object Recognition”. In *Pattern Analysis and Machine Intelligence (Pami)*, 20(6):637–646, June 1998. doi: 10.1109/34.683777.
- [44]. David Portugal. “RoboCops: Effective Cooperation and Scalability in Mobile Robot Teams for Automatic Patrolling of Infrastructures”, PhD Thesis, Faculty of Sciences and Technology, University of Coimbra, Portugal, September 2013.
- [45]. David Portugal and Rui P. Rocha. “Multi-Robot Patrol with Bayesian Learning”. In *Autonomous Robots*, 40(5):929–953, Springer, Jun. 2016. doi: 10.1007/s10514-015-9503-7.
- [46]. Richard Radke, Srinivas Andra, Omar Al-Kofahi and B. Roysam. “Image Change Detection Algorithms: A Systematic Survey”. In *IEEE Trans. Image Process*, 14(3):294–307, 2005.
- [47]. Yossi Rubner, Carlo Tomasi and Leonidas Guibas. “A metric for distributions with applications to image databases”. In *International Conference on Computer Vision*, pages 59–66, Bombay, India, 1998. doi: 10.1109/ICCV.1998.710701.
- [48]. António Simões. “Automatic vehicle classification using appearance-based features”, Master’s Degree in Electrical and Computer Engineering Thesis, Faculty of Science and Technology, University of Coimbra, Portugal, September 2015.
- [49]. P. Spagnolo, T. D. Orazio, M. Leo & A. Distanto. “Moving object segmentation by background subtraction and temporal analysis”. In *Image and Vision Computing*, 24(5):411–423, May 2006. doi: 10.1016/j.imavis.2006.01.001
- [50]. Angelo Tzotsos. “A Support Vector Machine Approach for Object Based Image Analysis”. In *International Conference on Object-Based Image Analysis*, 4–5, 2006.
- [51]. S. Sapna Varshney, Navin Rajpal and Ravindar Purwar. “Comparative study of image segmentation techniques and object matching using segmentation”. In *International Conference on Methods and Models in Computer Science (ICM2CS)*, 0:1–6, 2019. doi: 10.1109/ICM2CS.2009.5397985.
- [52]. Xiaohua Wang and Bing Yiang. “Automatic image registration based on natural characteristic points and global homography”. In *International Conference on Computer Science and Software Engineering (CSSE'08)*, 5:1365–1370, December 2008. doi: 10.1109/CSSE.2008.7.

8. ANEXOS

- ANEXO A:

Guia para a construção da estrutura de segurança para os computadores portáteis

Para todo o trabalho que irá ser desenvolvido existe a necessidade de manter a máxima homogeneidade possível entre os dados recolhidos pelos diferentes computadores. Sabe-se que há variáveis, como a distância focal que são intrínsecas a cada câmara e que dificultam essa homogeneidade. Deste modo todos os fatores passíveis de resolver como por exemplo a altura das câmaras são importantes para que a discrepância entre as imagens obtidas seja minimizada. Aproveitando uma estrutura existente nos robôs o objetivo é adaptá-la de modo a colmatar o problema das alturas e acrescentar uma estrutura que proporcione a máxima segurança aos computadores.

Sabendo que existem computadores com alturas diferentes será necessário adaptar cada estrutura já existente de um modo individual. Além das alturas dos computadores também as dimensões das bases são diferentes, o computador Fujitsu Siemens (com maiores dimensões) tem uma base com 26.5x35.5cm, já os computadores Asus (mais pequenos) têm base de dimensões 24x32.5. Deste modo as dimensões do computador maior servem de referência para as dimensões da estrutura de segurança a construir.



(a)



(b)

Figura A.1. (a) Imagem do robô com a estrutura já existente; (b) Estrutura vista de cima (onde irá assentar a estrutura a construir).

De modo a conseguir a máxima segurança para os portáteis é necessário desenvolver uma estrutura que encaixe na já existente. Esta estrutura não pode obstruir a ventilação dos portáteis nem os pode deixar cair. Deve ser fácil de manobrar e deve ser fácil de remover e colocar sobre a estrutura já existente. Decidiu-se fazer uma estrutura com 32x40 cm de dimensões tendo como referência as dimensões do portátil Fujitsu Siemens.

Para atingir os requisitos pretendidos e para ser exequível no tempo disponível optou-se por utilizar madeira para esta estrutura. A madeira é um material fácil de trabalhar, fácil de pintar, não carece de ferramentas demasiado específicas, é resistente e possui elevada durabilidade.

Na figura seguinte está representada a proposta da estrutura:

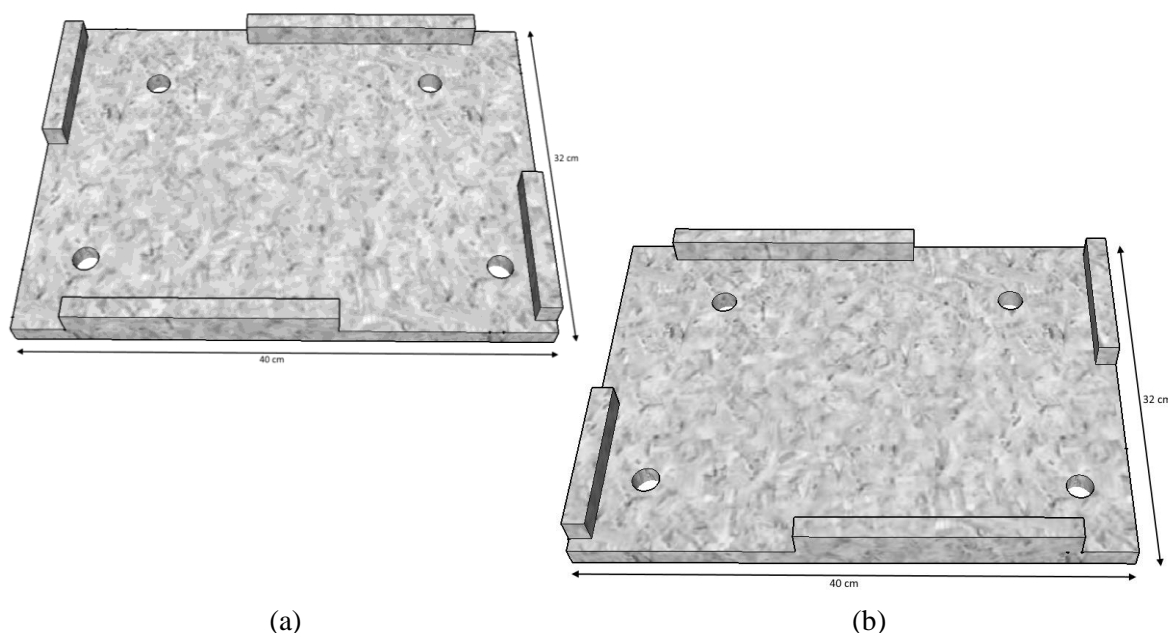


Figura A.2. (a) Imagem da estrutura a construir vista de cima para o portátil Asus;
b) Imagem da estrutura a construir vista de cima para o portátil Fujitsu.




Decidiu-se optar por esta solução uma vez que não obstrui a ventilação dos portáteis e ao mesmo tempo mantém-nos seguros evitando que caiam caso deslizem com o andamento do robô. Nos portáteis Asus a ventilação é feita na parte inferior da lateral esquerda e as portas USB encontram-se um pouco antes da ventilação. No portátil Fujitsu a ventilação é feita na parte superior da lateral esquerda e as portas USB encontram-se na parte de trás do lado direito. Como não foi possível encontrar um compromisso entre as duas bases de modo a serem compatíveis para ambos os portáteis teve de se optar por criar duas plataformas distintas. Cada plataforma tem o espaço da ventilação e das portas USB desimpedido para o respetivo portátil e de forma a tentar uniformizar as plataformas conseguiu-se criar duas, onde uma é espelhada da outra, apresentando cada plataforma uma assimetria estética.

Numa primeira fase irão ser construídas três plataformas e cada uma necessita dos seguintes materiais:

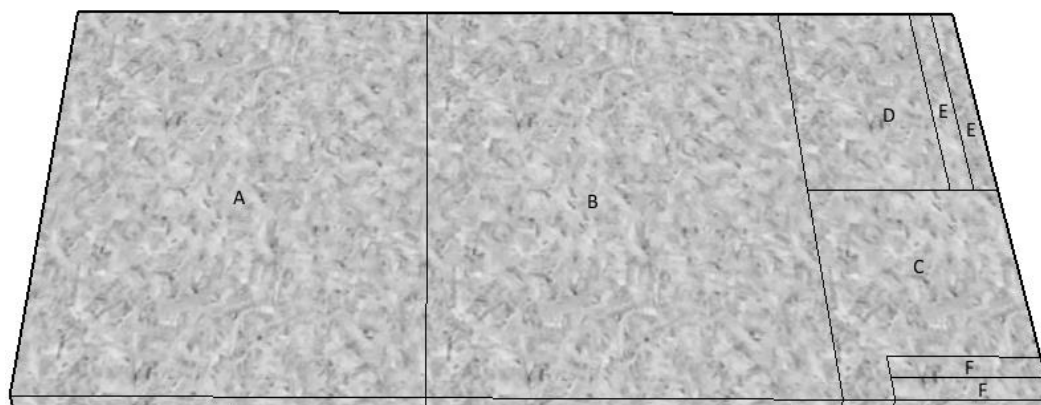
- 1 Tábua de madeira com as dimensões de 32x40 cm;
- Papel Vegetal;
- Lixa para madeira;
- Cola para madeira ou cola de montagem;
- Tinta para madeira;
- 2 Tiras de velcro de 20 cm;

Alguns destes materiais não carecem de aquisição uma vez que se encontram à disposição para serem utilizados. Outros têm de ser adquiridos e optou-se pela loja Leroy Merlin pela variedade de materiais que oferece.

Material a comprar para construir três plataformas (na loja Leroy Merlin):

Descrição	Referência	Dimensões (mm)	Preço (€)	Quantidade
Placa de MDF 	10901513	800x400x16	4,29	2
Cola BR Madeira Axton (250g) 	16715412	_____	2,49	1
Folha de Lixa Básico Madeira Grão 100 	00650720	_____	0,29	2

Procedimento:



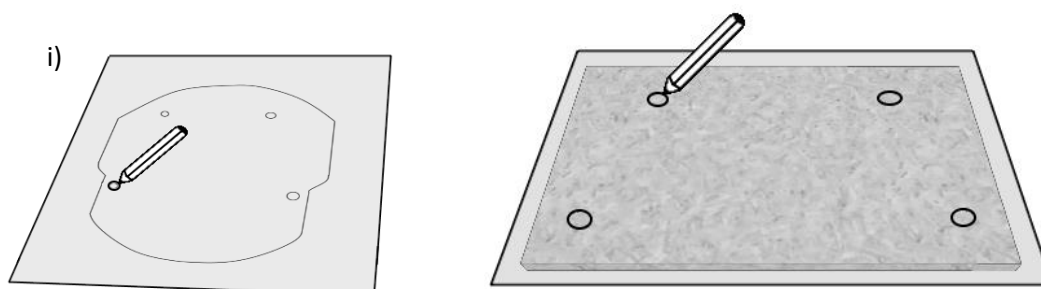
- 1) Cortar uma tábuia de 80x40 cm de modo a obter duas de 32x40 cm (A e B) e uma de 16x40 cm (C + D). Cada tábuia de 32x40 cm dá para duas estruturas;
- 2) Com a tábuia de madeira de 16x40 cm obter duas de 16x20 cm (C e D);

3) Com uma tábua (C) de 16x20 cm fazer duas de 12x2 cm (F) e guardar o restante para a estrutura a fazer com a tábua B;

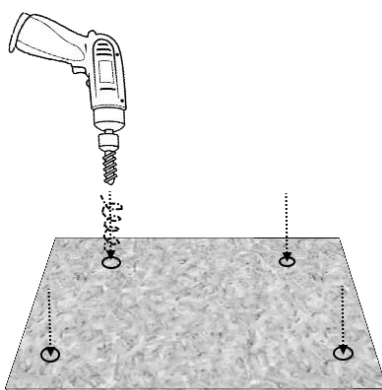
4) Com a outra tábua (D) de 16x20 cm fazer duas de 20x2 cm (E) e guardar o restante para a estrutura a fazer com a tábua B;

5) Lixar as extremidades de todas as tábuas obtidas com a lixa para madeiras;

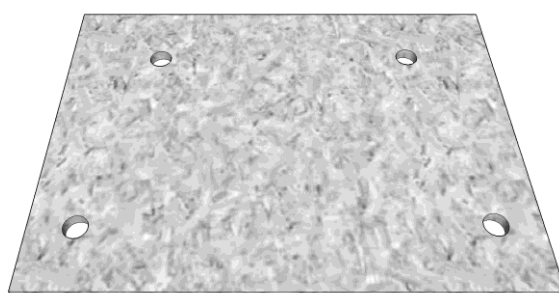
6) Marcar com papel vegetal as porcas da plataforma já existente (i) e de seguida marcá-las na tábua (A) de 32x40 cm (ii);



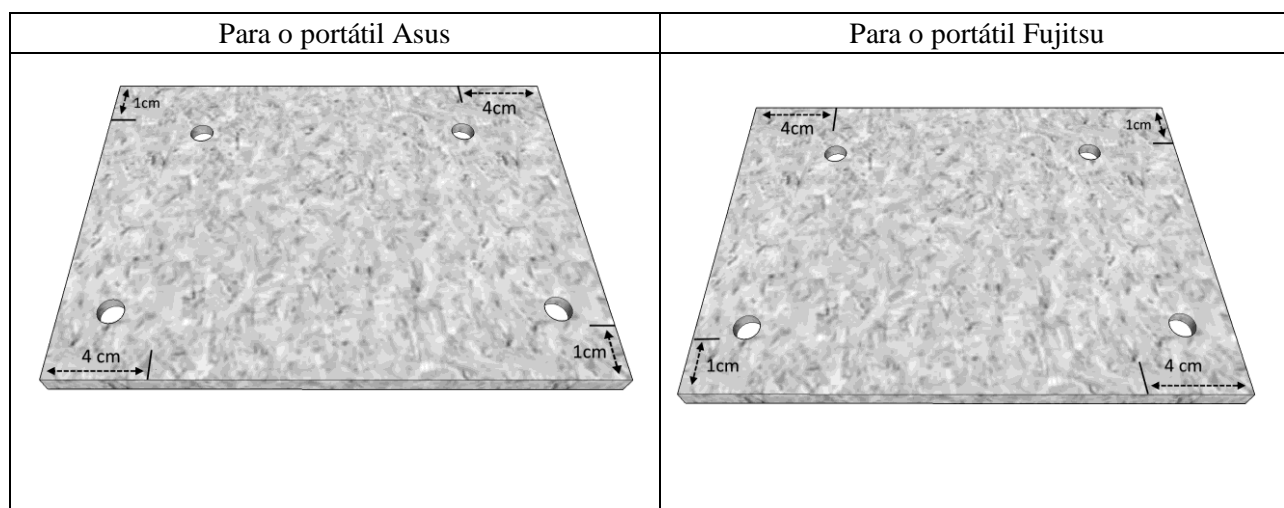
7) Com um berbequim fazer furos nos locais marcados de modo a obter um furo com o diâmetro da porca, para tal utilizar uma broca para madeira com 2.5 cm de diâmetro (nunca inferior);



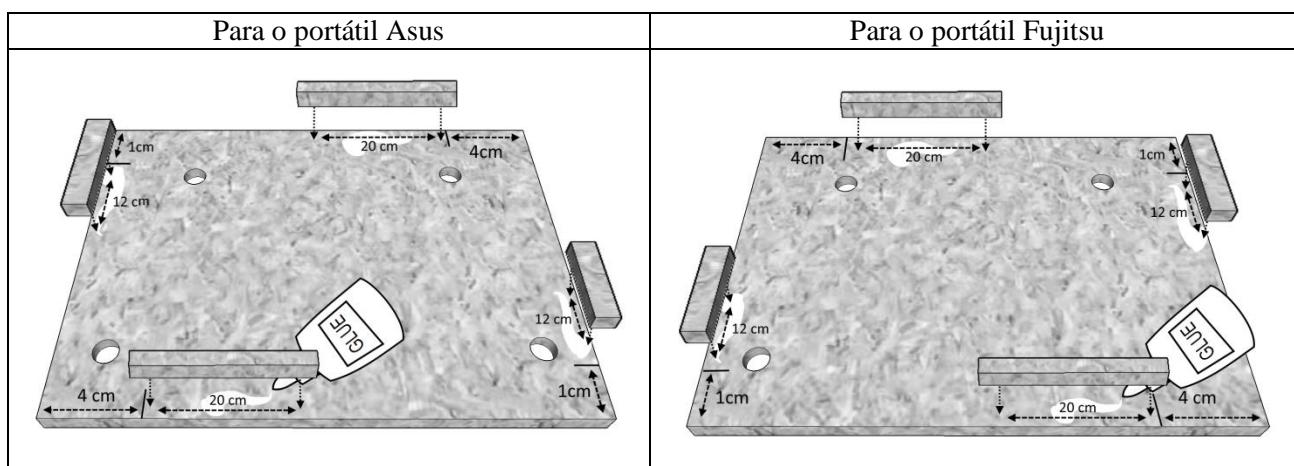
8) Depois dos 4 furos feitos obtemos:



9) Efetuar as medições na tábua da base, como mostra a figura seguinte, dependendo do portátil que nela irá ficar:



10) Colar as duas tábuas E e as duas tábuas F na tábua A como mostra a figura seguinte, dependendo de portátil:



11) No fim de secar pintar a estrutura com tinta branca;

12) Colocar as duas fitas de velcro de 20 cm na estrutura. Um dos lados da fita fica na estrutura e o outro fica no computador.

Este é um trabalho necessário uma vez que a estrutura existente não oferece nem estabilidade nem segurança aos portáteis que serão utilizados. É um trabalho que exige alguma perícia e cuidado e que por isso requer algum tempo, mas é uma mais-valia e um complemento a todo o restante trabalho e poderá ser utilizado em situações futuras.