

Para os eternamente curiosos.

Agradecimentos

Esta dissertação representa o ponto final noutro percurso da minha vida académica, durante o qual aprendi muito a nível académico e a nível pessoal. Deste modo, gostaria de agradecer a todos os que fizeram parte desta aventura e que, de algum modo, contribuíram com perspetiva, ajuda ou sugestões que me permitiram aprender mais ou simplesmente ver o mundo de outro modo.

Assim, começo por agradecer ao meu orientador, o Professor Miguel Barão, por tudo aquilo que me ensinou, por toda a paciência e motivação e por todas as suas sugestões e comentários. Sem a sua contribuição, este trabalho não seria possível.

Gostaria também de agradecer aos Professores do Departamento de Informática por tudo aquilo que me transmitiram nas diversas aulas do curso. A vossa contribuição servir-me-á de base para desenvolver novos projetos e aprender mais sobre os diversos tópicos relacionados com a Informática.

Agradeço também aos meus colegas por tudo o que aprendi convosco e pelos conselhos que gentilmente me concederam ao longo do tempo. Em especial, quero agradecer aos colegas que fazem ou fizeram parte do NEEI, por todas as atividades que desenvolveram e que me permitiram aprender mais sobre tópicos interessantes.

Agradeço também aos meus amigos, não só pela compreensão e pelo tempo dispensado para me ouvirem e aconselharem, mas também por todos os momentos de descontração que partilhámos e por todo o encorajamento e motivação dada.

Finalmente, quero agradecer à minha família pela enorme paciência demonstrada e por todos os conselhos dados durante os momentos bons e, principalmente, durante os menos bons. Em especial, gostaria de agradecer aos meus pais, que sempre me apoiaram ao longo destes anos e que me deram a oportunidade de embarcar nesta aventura. O conhecimento é algo fundamental para compreender melhor o mundo e, por terem criado condições para que eu pudesse aceder ao mesmo, fico-vos eternamente grata.

Conteúdo

Conteúdo	ix
Lista de Figuras	xiii
Lista de Tabelas	xvii
Lista de Acrónimos	xix
Notação	xxi
Sumário	xxiii
Abstract	xxv
1 Introdução	1
1.1 Motivação	2
1.2 Objetivos	2
1.3 Organização do Documento	3
2 Estado da Arte	5
2.1 Métodos de Calibração da Câmara	5
2.2 Algoritmos de Detecção de Elementos em Imagens	6
2.2.1 Algoritmo de Detecção de Contornos: <i>Canny Edge Detector</i>	6
2.2.2 Algoritmos de Detecção de Linhas	7
2.3 Técnicas de Reconstrução 3D	9
3 A Câmara e a Perspetiva	13

3.1	Câmara Buraco de Agulha	13
3.1.1	Enquadramento Histórico	14
3.1.2	O Modelo	15
3.2	Projeção de Perspetiva	15
3.3	Calibração da Câmara	16
3.3.1	Parâmetros Intrínsecos da Câmara	16
3.3.2	Parâmetros Extrínsecos da Câmara	17
3.3.3	Matriz de Projeção	18
4	Deteção de Linhas em Imagens	21
4.1	Transformada de Hough	22
4.2	Transformada de Hough Modificada	23
4.2.1	Transformada de Hough Modificada - Versão 1	24
4.2.2	Transformada de Hough Modificada - Versão 2	26
4.2.3	Transformada de Hough Modificada - Versão 3	28
5	Reconstrução 3D	35
5.1	Correspondência de Linhas	36
5.1.1	O Algoritmo de Correspondência	37
5.2	Reconstrução	40
6	Caso de Estudo	45
6.1	Hardware Usado na Reconstrução	45
6.2	Calibração das Câmaras	47
6.3	Deteção de Linhas	48
6.4	Reconstrução 3D	52
7	Conclusões e Trabalho Futuro	57
7.1	Trabalho Futuro	58
A	Transformada de Hough Modificada: Resultados dos Testes Efetuados	59
A.1	Testes com Imagens Controladas	59
A.1.1	Transformada de Hough Modificada - Versão 1	60
A.1.2	Transformada de Hough Modificada - Versão 2	60
A.1.3	Transformada de Hough Modificada - Versão 3	60
A.2	Testes com Fotografias	64
A.2.1	Transformada de Hough Modificada - Versão 1	64

A.2.2	Transformada de Hough Modificada - Versão 2	64
A.2.3	Transformada de Hough Modificada - Versão 3	73
B	Parametrização das Linhas Detetadas na Forma Standard	79
C	Resultado da Aplicação do Passo de Pré-Processamento das Linhas Detetadas	83
D	Resultados da Aplicação dos Algoritmos de Correspondência	87
Bibliografia		91

Listas de Figuras

2.1	Taxonomia das técnicas óticas de aquisição de estrutura 3D.	10
3.1	Uma Câmara Obscura [All05].	14
3.2	Representação de uma Câmara Buraco de Agulha [Mel05].	15
3.3	Representação esquemática da geometria do Modelo Câmara Buraco de Agulha [HZ03].	16
4.1	Exemplo de um Espaço Hough de uma imagem controlada (quatro linhas pretas sobre fundo branco).	22
4.2	Espaços de parâmetros usados na Transformada de Hough.	23
4.3	Passos usados na deteção de linhas com a Transformada de Hough Modificada.	24
4.4	Exemplo do funcionamento do algoritmo K-Means com centroides iniciais aleatórios. Os quadrados coloridos representam os centroides [Com05].	25
4.5	Linhas detetadas aplicando a Transformada de Hough e a Transformada de Hough Modificada - Versão 1 à mesma imagem controlada. As deteções correspondem às linhas coloridas.	26
4.6	Linhas detetadas aplicando à mesma fotografia a Transformada de Hough e a Versão 1 da Transformada de Hough Modificada. As linhas detetadas são indicadas com linhas coloridas.	27
4.7	Linhas detetadas pelas Transformada de Hough Modificada - Versão 2 e Transformada de Hough, quando aplicadas a uma imagem controlada.	29
4.8	Resultado de um dos testes efetuados aplicando a TH e a Versão 2 da Transformada de Hough Modificada a fotografias.	30
4.9	Representação esquemática do Espaço Hough.	31
4.10	Resultado da aplicação, a imagens controladas, da Transformada de Hough e da THM-V3.	33
4.11	Linhas detetadas pelas Transformada de Hough e Transformada de Hough Modificada - Versão 3, quando aplicadas a fotografias.	34
5.1	Os dois problemas que constituem a reconstrução 3D.	36
5.2	A linha 3D obtida a partir da interseção de dois planos.	38

5.3	Fluxograma do Algoritmo 1, criado para encontrar correspondências entre linhas.	42
5.4	Fluxograma do Algoritmo 2, que permite encontrar correspondências entre linhas.	43
5.5	Fluxograma do algoritmo de reconstrução.	44
6.1	Passos usados desde a obtenção das imagens <i>stereo</i> até à reconstrução 3D.	46
6.2	O objeto a ser reconstruído.	46
6.3	<i>Webcam</i> idêntica às usadas para capturar as imagens <i>stereo</i> [Int15a].	46
6.4	Diagrama e disposição real do <i>hardware</i> usado na reconstrução 3D.	47
6.5	Imagens usadas na calibração das câmaras.	48
6.6	Imagens <i>stereo</i> do cubo a ser reconstruído.	50
6.7	Resultados da aplicação de cada um dos passos de pré-processamento à imagem da obtida com a Câmara 1.	51
6.8	Linhos detetadas pela Transformada de Hough nas imagens <i>stereo</i> da figura 6.6.	53
6.9	Modelo obtido após ser feita a reconstrução 3D, na qual é usado o Algoritmo 1 para encontrar correspondências entre linhas.	55
6.10	Modelo obtido após ser feita a reconstrução 3D, na qual é usado o Algoritmo 2 para encontrar correspondências entre linhas.	56
A.1	Resultados dos testes da Transformada de Hough Modificada - Versão 1, em que são usadas imagens controladas.	61
A.2	Testes da Versão 2 da Transformada de Hough Modificada, nos quais se usam imagens controladas e o mesmo expoente para ρ e θ no passo de <i>clustering</i> do algoritmo.	62
A.3	Testes da Versão 2 da Transformada de Hough Modificada, em que são usadas imagens controladas e expoentes diferentes no passo de <i>clustering</i> do algoritmo.	63
A.4	Resultados dos testes da Transformada de Hough Modificada - Versão 3, em que se usam imagens controladas e o mesmo expoente no passo de <i>clustering</i> do algoritmo.	65
A.5	Resultados dos testes da Versão 3 da Transformada de Hough Modificada, nos quais são usadas imagens controladas e expoentes diferentes no passo de <i>clustering</i> do algoritmo.	66
A.6	Resultados dos testes da Transformada de Hough Modificada - Versão 1, em que são usadas fotografias como <i>input</i>	67
A.7	Resultados dos testes da Transformada de Hough Modificada - Versão 1, em que são usadas fotografias como <i>input</i> (cont.).	68
A.8	Resultados dos testes da Transformada de Hough Modificada - Versão 2, em que são usadas fotografias como <i>input</i> e expoentes iguais no passo de <i>clustering</i> do algoritmo.	69
A.9	Resultados dos testes da Transformada de Hough Modificada - Versão 2, em que são usadas fotografias como <i>input</i> e expoentes iguais no passo de <i>clustering</i> do algoritmo (cont.).	70
A.10	Testes da Versão 2 da Transformada de Hough Modificada, rm que são usadas imagens controladas e expoentes diferentes no passo de <i>clustering</i> do algoritmo.	71
A.11	Testes da Versão 2 da Transformada de Hough Modificada, nos quais são usadas imagens controladas e expoentes diferentes no passo de <i>clustering</i> (cont.).	72

A.12 Resultados dos testes da Transformada de Hough Modificada - Versão 3, em que se usam fotografias como <i>input</i> e expoentes iguais no passo de <i>clustering</i>	74
A.13 Resultados dos testes da Transformada de Hough Modificada - Versão 3, em que se usam fotografias como <i>input</i> e expoentes iguais no passo de <i>clustering</i> (cont.).	75
A.14 Testes da Versão 3 da Transformada de Hough Modificada, nos quais são usadas imagens controladas e expoentes diferentes no passo de <i>clustering</i>	76
A.15 Testes da Versão 3 da Transformada de Hough Modificada, nos quais são usadas imagens controladas e expoentes diferentes no passo de <i>clustering</i> (cont.).	77

Listas de Tabelas

6.1	Pontos de referência extraídos das imagens de calibração.	49
B.1	Parâmetros das linhas detetadas com a Transformada de Hough na figura 6.6-(a), que são apresentadas na figura 6.8-(a).	80
B.2	Parâmetros das linhas detetadas com a Transformada de Hough na figura 6.6-(b), que são apresentadas na figura 6.8-(b).	81
C.1	Grupos de linhas obtidos com pré-processamento das linhas detetadas pela TH, que estão presentes na figura 6.8-(a).	84
C.2	Grupos de linhas obtidos com pré-processamento das linhas detetadas pela TH, que estão presentes na figura 6.8-(b).	85
D.1	Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas verticais, cuja correspondência foi identificada com recurso ao Algoritmo 1.	88
D.2	Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas horizontais, cuja correspondência foi identificada com recurso ao Algoritmo 1.	88
D.3	Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas diagonais, cuja correspondência foi identificada com recurso ao Algoritmo 1.	89
D.4	Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas verticais, cuja correspondência foi identificada com recurso ao Algoritmo 2.	89
D.5	Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas horizontais, cuja correspondência foi identificada com recurso ao Algoritmo 2.	90
D.6	Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas diagonais, cuja correspondência foi identificada com recurso ao Algoritmo 2.	90

Lista de Acrónimos

TR Transformada de Radon

TRI Transformada de Radon Inversa

TH Transformada de Hough

PHT *Probabilistic Hough Transform*

PPHT *Progressive Probabilistic Hough Transform*

RHT *Randomized Hough Transform*

THM Transformada de Hough Modificada

THM-V_i Transformada de Hough Modificada - Versão *i*

*Img*₁ Imagem 1

*Img*₂ Imagem 2

Algo1 Algoritmo 1 usado na correspondência de linhas

Algo2 Algoritmo 2 usado na correspondência de linhas

Notação

- $(u, v, 1)$ - coordenadas de um ponto arbitrário no plano de imagem;
- $(X, Y, Z, 1)$ - coordenadas de um ponto (3D) arbitrário no mundo;
- f - a distância focal da câmara;
- C - centro da câmara;
- p - centro da projeção de perspetiva (*principal point*);
- k_x e k_y - dimensões dos píxeis da imagem;
- u_0 e v_0 - coordenadas do centro da câmara;
- K - matriz que contém os parâmetros intrínsecos da câmara;
- R - a matriz de rotação que alinha os referenciais da câmara e do mundo;
- T - o vetor de translação entre as origens dos referenciais da câmara e do mundo, respetivamente;
- M - a matriz que contém a informação relativa aos parâmetros extrínsecos da câmara;
- P - a matriz de projeção que combina a informação relativa aos parâmetros intrínsecos e aos parâmetros extrínsecos da câmara;
- p_ρ e p_θ - coordenadas ρ e θ de um ponto de um grupo criado pelo K-Means;
- c_ρ e c_θ - coordenadas ρ e θ de um novo centroide calculado no segundo passo do K-Means;
- w_p - número de votos de um ponto do grupo (peso atribuído);
- α - expoente a que se eleva o peso atribuído a um dado ponto do grupo;
- G - grupo de pontos criado pelo K-Means;
- ρ_{max} - valor máximo que ρ pode tomar;
- $[A, B, C]$ - parâmetros que descrevem uma linha na forma *standard*;
- l_i - uma linha na Imagem i ;
- pf_i - ponto focal da câmara i ;
- L - uma linha 3D arbitrária;

- Pl_i - um plano definido por um ponto focal de uma câmara e uma linha 3D;
- $[a, b, c, d]$ ou $[a_i, b_i, c_i, d_i]$ ou w_i - parâmetros de um plano Pl_i ;
- α - ângulo entre dois planos Pl_i ;
- N_i - normal de um plano Pl_i ;
- Q - medida de semelhança entre dois planos Pl_i ;
- Q_{max} - valor máximo da medida de semelhança Q ;
- W - matriz onde constam os parâmetros de um par de planos Pl_i correspondentes;
- β - ângulo entre duas linhas 2D.

Sumário

A reconstrução 3D é uma das várias áreas que estão inseridas no campo da Visão Computacional. A reconstrução 3D com recurso a imagens *stereo* consiste numa técnica ótica. Esta técnica usa como *input* duas ou mais imagens de uma cena ou de um objeto e permite obter um modelo 3D desta cena ou objeto.

Nesta dissertação é apresentada uma metodologia de reconstrução 3D que usa como *input* um par de imagens *stereo*, obtidas com recurso a duas *webcams*. As câmaras usadas para captar estas imagens são calibradas antes de ser iniciada a reconstrução 3D. Com recurso a algoritmos especializados são detetadas as linhas presentes nas imagens captadas. Assim, nesta dissertação apresenta-se também um algoritmo de deteção de linhas baseado na Transformada de Hough. Quando o processo de deteção de linhas termina, são identificadas correspondências entre estas linhas. Os algoritmos criados para este efeito são também apresentados nesta dissertação. Finalmente, é reconstruído o modelo 3D do objeto presente no par de imagens *stereo*.

Palavras chave: Reconstrução 3D, Deteção de Linhas, Transformada de Hough

Abstract

3D Reconstruction From Stereo Images

3D reconstruction is one of several areas that are included in the Computer Vision field. 3D reconstruction from stereo images is an optical technique. This technique uses two or more images of a scene or object as input and outputs a 3D model of this scece or object.

This dissertation introduces a methodology that allows for 3D reconstruction with a pair of stereo images, obtained with two webcams. These cameras are calibrated before the 3D reconstruction starts. The lines present in the pair of stereo images are detected with specialized algorithms. Thus, in this dissertation it is also presented a line detection algorithm based on the Hough Transform. Once the line detection process is completed, the correspondences between the lines detected in the stereo pair are found. The algorithms created to identify these correspondences are also presented in this dissertation. Finally, the 3D model of the object shown in the stereo images is produced.

Keywords: 3D Reconstruction, Line Detection, Hough Transform

1

Introdução

Computers aren't the thing. They're the thing that gets us to the thing.

Joe MacMillan, "Halt and Catch Fire"

A visão é fundamental no nosso dia-a-dia. Com este sentido percecionamos o mundo à nossa volta e conseguimos desempenhar tarefas que nos parecem triviais, como, por exemplo, evitar obstáculos. A visão humana é um tipo de visão binocular, que obtém informação a partir de dois olhos (os "sensores"). Dada a posição dos olhos na cabeça e a distância a que estes se encontram um do outro, o ponto de vista de cada olho confere ligeiras variações às imagens captadas. Estas são então processadas pelo cérebro em estruturas específicas [RRR09], sendo exploradas as diferenças existentes entre estas imagens para percecionar estrutura tridimensional dos objetos do nosso mundo [FP11]. Este processo denomina-se estereopsis [HWW95, TV98].

A percepção de estrutura tridimensional de elementos no mundo, um processo intuitivo para um humano, é uma tarefa complexa para uma máquina. A Visão Computacional pretende usar a informação existente em imagens, transformando-a numa nova representação ou decisão [FBDH⁺13, SS01]. Reconstruir um modelo 3D a partir de imagens ou identificar elementos nestas são dois dos vários tipos de problema estudados nesta área. A longo prazo, a Visão Computacional pretende permitir que máquinas possam ter a possibilidade de observar o mundo que as rodeia, num processo idêntico ao da visão humana [LM11].

A reconstrução 3D, uma área da Visão Computacional, permite reaver a informação relativa à estrutura tridimensional da cena ou de elementos da mesma [FBDH⁺13], que é perdida quando se obtém uma fotografia da referida cena. A Visão Stereo é uma das várias metodologias que podem ser usadas para fazer reconstrução 3D a partir de imagens. Neste processo podem ser usadas como ponto de partida duas ou mais imagens de um objeto ou cena [FBDH⁺13, Fau93]. Para captar estas imagens são usadas câmaras com pontos de vista diferentes da mesma cena ou objeto [MK04]. A Reconstrução 3D com recurso a Visão Stereo pode ser considerada como uma junção de dois problemas: encontrar correspondências entre as imagens obtidas e reconstrução dos objetos ou cena presentes nas mesmas [FP11].

Existem vários contextos em que a reconstrução 3D pode ser aplicada. A obtenção de modelos tridimensionais de peças de arte, artefactos e monumentos são contextos em que a reconstrução 3D é especialmente importante [GRS14]. Podem também obter-se modelos de património que se encontre submerso com recurso a reconstrução 3D [GWYH14] e que não seja de outro modo acessível. Existem também aplicações relacionadas com a reconstrução de cidades ou de zonas destas a partir de grandes coleções de fotografias [AFS⁺11, FFGG⁺10]. As técnicas usadas na reconstrução 3D podem também ser aplicadas a sistemas de deteção de obstáculos de veículos não tripulados [GMM02].

1.1 Motivação

É possível reconstruir uma cena ou um objeto presente num par de imagens *stereo* usando as correspondências entre elementos presentes nestas e triangulações [Sze10]. Porém, apesar desta técnica produzir resultados relativamente fiáveis, a quantidade de informação a processar pode ser elevada [BK08]. Assim, a reconstrução 3D com recurso às linhas detetadas num par de imagens *stereo* pode reduzir a quantidade de informação a processar e, por isso, representa uma alternativa a esta técnica.

Para que seja possível fazer a reconstrução 3D de uma cena ou objeto tendo por base as linhas detetadas nas imagens *stereo*, é necessário usar algoritmos de deteção de linhas que permitam identificar a maioria das linhas presentes nestas imagens e que produzam o menor número de falsos positivos possível. Como nenhum algoritmo de deteção de linhas está imune a erros, a extensão ou modificação destes de modo a reduzir o número de falsos positivos detetados é um problema pertinente. Após serem detetadas as linhas necessárias para fazer a reconstrução, é importante ter ao dispor técnicas que permitam reconstruir as linhas 3D presentes na cena ou num objeto da mesma. Para fazer a reconstrução 3D da cena ou dos seus elementos, é possível usar uma metodologia idêntica à da reconstrução com Visão Stereo, isto é, dividir o problema da reconstrução 3D em encontrar correspondências e reconstruir as linhas detetadas nas imagens. Para que isto seja possível, é necessário encontrar ou desenvolver algoritmos que permitam identificar correspondências entre linhas detetadas em imagens e reconstruir um modelo 3D da cena ou de elementos da mesma com recurso a esta informação. É neste contexto que surgem os algoritmos de deteção de linhas e de reconstrução 3D apresentados neste trabalho.

1.2 Objetivos

Pretende-se fazer a reconstrução 3D de um objeto que conste num par de imagens *stereo* com recurso às linhas detetadas nestas imagens. Para obter o par de imagens *stereo* usam-se duas *webcams*, visto que este *hardware* é relativamente barato e fácil de encontrar. Estão relacionados com o objetivo principal os seguintes objetivos específicos:

- Estudo do estado da arte relacionado com as metodologias usadas neste trabalho, nomeadamente, métodos de calibração das câmaras, deteção de elementos em imagens e técnicas de reconstrução 3D;
- Estudo das componentes teóricas relacionadas com o Modelo Câmara Buraco de Agulha, com a projeção

de perspetiva e com a calibração das câmaras;

- Implementação dos protótipos dos algoritmos que permitem fazer a calibração das câmaras e a reconstrução 3D;
- Teste destes protótipos e análise dos resultados obtidos.

Pretende-se também modificar um algoritmo de deteção de linhas, para que seja possível reduzir o número de falsos positivos detetados pelo mesmo. Relativamente a este objetivo secundário, pretende-se:

- Estudar o estado da arte relacionado com a Transformada de Hough e de outros algoritmos de deteção de linhas idênticos;
- Criação de um protótipo de uma versão modificada da Transformada de Hough, que permita reduzir o número de falsos positivos detetados;
- Teste deste algoritmo protótipo e análise dos resultados obtidos.

1.3 Organização do Documento

O Capítulo 2 fornece uma revisão da literatura relacionada com o tema, bem como diversas técnicas que permitem fazer reconstrução 3D, calibração das câmaras usadas para obter as imagens e deteção de elementos em imagens. O Capítulo 3 apresenta o Modelo Câmara Buraco de Agulha e inclui também informação relativa ao processo de calibração, referenciado o que são os parâmetros da câmara e a matriz de projeção. Neste capítulo, apresenta-se ainda um método que permite estimar a matriz de projeção das câmaras usadas para captar as imagens stereo. A deteção de elementos em imagens, mais concretamente, a deteção de linhas é o tema explorado no Capítulo 4. Neste é apresentada a Transformada de Hough em detalhe, bem como um algoritmo alternativo criado com base na modificação desta Transformada. O Capítulo 5 apresenta a metodologia usada para fazer a Reconstrução 3D usada neste trabalho, sendo apresentados os algoritmos de correspondência e de reconstrução. O Capítulo 6 apresenta os resultados obtidos ao aplicar o método de calibração da câmara, a deteção de linhas e as técnicas de reconstrução 3D apresentadas nos capítulos mencionados anteriormente. Finalmente, o Capítulo 7 apresenta as conclusões obtidas e o trabalho futuro.

São também incluídos vários anexos. O Anexo A apresenta em detalhe os resultados dos testes feitos às várias versões do algoritmo Transformada de Hough Modificada apresentado no Capítulo 4. O Anexo B inclui a parametrização das linhas detetadas na sua forma *standard*. Os resultados da aplicação do passo de pré-processamento das linhas detetadas podem ser consultados no Anexo C. Finalmente, o Anexo D inclui os resultados da aplicação dos algoritmos de correspondência apresentados no Capítulo 5.

2

Estado da Arte

Research is what I'm doing when I don't know what I'm doing.

Wernher von Braun

Este capítulo contém a revisão da literatura relacionada com as várias partes deste trabalho. Para facilitar a leitura, o estado da arte está dividido por categorias. Estas correspondem aos vários tópicos estudados neste trabalho: métodos de calibração da câmara, deteção de elementos em imagens e técnicas de reconstrução 3D.

2.1 Métodos de Calibração da Câmera

A calibração é um passo importante na reconstrução 3D, visto que permite obter informação relativa à câmara (parâmetros intrínsecos) e à sua posição no mundo (parâmetros extrínsecos).¹ Existem diversos métodos que podem ser usados para calibrar uma ou mais câmaras [SAB02, CF98, MFC⁺02].

A calibração de uma câmara pode ser feita com recurso a um objeto 3D. Existem vários tipos de objetos que podem ser usados para o efeito como, por exemplo, um cubo com um padrão xadrez nas suas faces [MK04].

¹Uma descrição mais detalhada dos parâmetros obtidos através da calibração pode ser consultada no Capítulo 3.

Podem ser usados outros objetos de referência na calibração, desde que a geometria destes seja conhecida [MK04]. Aos pontos usados na calibração, obtidos a partir de fotos do objeto de referência, está associado um sistema de coordenadas tridimensionais [MK04]. Hall [HTMS82] apresenta um método de calibração linear, que permite obter a matriz de transformação que relaciona os pontos de referência da imagem com os seus correspondentes num sistema de coordenadas tridimensionais. O método usado para calcular esta matriz é o método dos mínimos quadrados [HTMS82]. A principal vantagem deste tipo de técnica é a rapidez do processo de calibração [SAB02]. Além do método de calibração apresentado por Hall, existem outras técnicas de calibração linear idênticas [FT86].

Roger Tsai apresentou uma técnica de calibração que pode usar planos 2D como objetos de referência [Tsa87]. Neste caso, um tabuleiro de xadrez ou um plano com um padrão axadrezado são objetos de referência viáveis. Estes planos são usados para obter pontos de referência, que são depois usados durante a calibração. A abordagem de Tsai é constituída por duas fases [Tsa87]. A primeira fase consiste em usar a informação dos pontos de referência recolhidos para estimar, usando o método dos mínimos quadrados, o maior número de parâmetros possível [Hor00, Tsa87]. Esta estimativa permite obter os valores iniciais destes parâmetros, pelo que estes não afetam o resultado final [Tsa87]. A segunda fase consiste em usar um método de otimização não linear para obter os parâmetros que não foram estimados [Hor00, Tsa87]. Esta otimização minimiza o erro entre as coordenadas da imagem e as coordenadas estimadas com base nos pontos de referência recolhidos [Tsa87]. Nesta fase, os valores estimados para os parâmetros intrínsecos, extrínsecos e distorções são otimizados [Hor00]. A principal vantagem deste tipo de técnica é a velocidade com que se obtêm os parâmetros da câmara, bem como a precisão dos mesmos [SAB02]. Kang e Medioni [MK04] apresentam uma técnica alternativa ao método de calibração proposto por Tsai. Esta permite fazer a calibração da câmara, sem que seja necessário conhecer o movimento do plano 2D usado como objeto de referência [MK04].

2.2 Algoritmos de Detecção de Elementos em Imagens

Uma imagem contém vários elementos. São exemplos destes: cantos, linhas, curvas ou outro tipo de formas. Toda esta informação presente na imagem pode não ser necessária para o tipo de problema que estamos a resolver, por isso existem algoritmos de deteção de contornos para simplificar a mesma. Caso seja necessário detetar os elementos presentes na imagem, existem algoritmos especializados que permitem fazê-lo. Nesta secção são apresentados alguns algoritmos que permitem detetar linhas e um algoritmo de deteção de contornos.

2.2.1 Algoritmo de Detecção de Contornos: *Canny Edge Detector*

A deteção de contornos é um processo que permite reduzir a quantidade de informação de uma imagem que deve ser processada, preservando a informação estrutural existente [Can86]. Existem vários algoritmos que permitem fazer a deteção de contornos numa imagem [ZT98, PM82, PP11]. Destes destaca-se, devido às suas diversas aplicações, o *Canny Edge Detector*.

Apresentado por John Canny em 1986, o *Canny Edge Detector* é um algoritmo de deteção de contornos que permite simplificar a informação contida numa imagem [Can86]. No primeiro passo do algoritmo, é reduzido o eventual ruído presente na imagem, evitando-se assim situações em que este é confundido com contornos presentes na imagem [Can86]. O segundo passo consiste em encontrar zonas de interesse, ou seja, zonas da imagem em que possam existir contornos. Para tal, medem-se as alterações na intensidade dos píxeis da imagem [SKB13, Can86]. Os contornos estão presentes em zonas da imagem cuja intensidade dos píxeis atingem um valor máximo [SKB13]. Para reduzir a quantidade de informação a ser processada, o terceiro passo do *Canny Edge Detector* usa dois limites² [SKB13]. Estes permitem identificar os pontos de cada zona da imagem

²Um limite máximo e um limite mínimo.

que atingem o valor correspondente à intensidade pretendida [SKB13]. São descartados todos os pontos que estejam abaixo do limite mínimo [Can86]. O último passo consiste em identificar quais os píxeis que devem ser incluídos na imagem final. Esta inclui píxeis cuja intensidade está entre os limites definidos, desde que estes píxeis estejam associados a outros que tenham intensidade acima do limite máximo [Can86, SKB13]. A imagem final inclui também todos os píxeis com intensidades acima do valor do limite máximo [SKB13].

2.2.2 Algoritmos de Detecção de Linhas

Existem vários algoritmos que permitem detetar linhas em imagens. A complexidade, o tipo de método usado para detetar as linhas e as aplicações destes algoritmos variam. A Transformada de Radon, a Transformada de Hough e algumas variantes desta são alguns destes algoritmos.

Transformada de Radon

A Transformada de Radon (TR), ou *Radon Transform*, deve o seu nome ao matemático Johann Radon que, em 1917, conseguiu provar que é possível descrever uma função num espaço usando como base os integrais desta nos planos desse mesmo espaço [Rad05, GLV04]. Este algoritmo pode ser usado para detetar formas em imagens, tendo também outras aplicações mais complexas.

A TR faz uma transformação entre um espaço caracterizado por coordenadas cartesianas (a imagem) e um espaço caracterizado por coordenadas polares (as projeções). A Transformada de Radon de uma imagem descrita pela função $F(x, y)$ é [Dea81, Dea00, Sig99]:

$$R(\rho, \theta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(x, y) \delta(\rho - x \cos(\theta) - y \sin(\theta)) dx dy. \quad (2.1)$$

Na transformação (2.1), ρ corresponde à distância entre a origem do referencial e a linha que se pretende detetar e θ corresponde à inclinação desta em relação ao eixo dos x [Dea00].

A imagem resultante da transformação feita pela Transformada de Radon, $R(\rho, \theta)$, inclui as somas das intensidades dos píxeis da imagem, descrita pela função F , num conjunto de direções (projeções) [Sig99]. As zonas da imagem R em que as projeções têm valores elevados correspondem a formas existentes na imagem original [Sig99]. As coordenadas destes valores no espaço de parâmetros permitem descrever as formas correspondentes que estão presentes na imagem original [Dea00].

É também possível fazer o processo inverso, ou seja, fazer a transformação entre o espaço de parâmetros e o espaço da imagem e obter a informação original presente na mesma [GLV04, Dea00]. Este processo denomina-se Transformada de Radon Inversa (TRI).

A Transformada de Radon tem aplicações em diversas áreas, tais como: imagiologia médica, astronomia, geofísica e ótica, entre outras [Dea00]. Este algoritmo destaca-se especialmente no ramo da Tomografia Computadorizada, mais conhecida por TAC, visto que com este algoritmo é possível obter projeções de um objeto usando Raios-X [GLV04]. A Transformada de Hough pode ser obtida a partir da Transformada de Radon [GLV04].

Transformada de Hough

A Transformada de Hough (TH) [Hou62] é um algoritmo de deteção de linhas. Esta Transformada usa uma parametrização, que permite mapear pontos de uma imagem para pontos num espaço de parâmetros [IK88]. No caso do algoritmo original, a parametrização é feita com recurso a um espaço de parâmetros cartesiano, no qual uma linha é descrita por um declive e uma ordenada na origem [Hou62]. Este tipo de parametrização cria problemas na deteção de linhas verticais, visto que o parâmetro correspondente ao declive das linhas detetadas

pode não ter limite definido [HMSR15]. Porém, esta adversidade deixa de existir quando se usa a generalização da TH apresentada por Duda e Hart [DH72]. No algoritmo proposto por Duda e Hart, uma linha é descrita por um par de parâmetros (ρ, θ) , que estão ambos sujeitos a um limite definido [DH72]. Em ambas as versões da Transformada de Hough é usado um acumulador, no qual é registado, para cada ponto, o número de linhas que podem passar por esse ponto [IK88]. Para identificar situações em que são detetadas linhas, usa-se um parâmetro (*threshold*) [DH72, IK88]. Se existirem pontos do acumulador pelos quais passa um número de linhas acima do valor estabelecido no *threshold*, então considera-se que os parâmetros destes pontos caracterizam linhas na imagem [DH72]. A versão da Transformada de Hough de Duda e Hart é descrita em detalhe no Capítulo 4.

A principal vantagem no uso da Transformada de Hough prende-se com o facto desta conseguir detetar linhas mesmo que estas estejam separadas em segmentos devido a ruído ou oclusões na imagem [HAAg99]. A fiabilidade dos resultados obtidos e a rapidez com que estes são obtidos são outros pontos positivos [Mas09]. A principal desvantagem consiste na não deteção de segmentos que façam parte de uma linha, visto que uma linha detetada pela TH é interpretada como sendo infinita [HAAg99]. Caso existam linhas da imagem que estejam próximas, a Transformada também pode devolver resultados incorretos [HAAg99].

Existem muitas variações da Transformada de Hough. Illingworth e Kittler publicaram um *survey* [IK88], no qual são apresentadas muitas destas variações. Porém, desde a publicação deste *survey* até à atualidade, foram criadas muitas outras variações da Transformada de Hough [HMSR15]. Destas variações destacam-se:

- **Probabilistic Hough Transform (PHT)** - o algoritmo proposto por Kiryati usa como *input* uma amostra aleatória dos pontos de uma imagem [KK00]. Após ser obtida esta amostra, aplica-se a Transformada de Hough à mesma para detetar linhas ou outras formas presentes na imagem [KK00]. Stephens propõe uma PHT alternativa à de Kiryati. Esta PHT usa uma função densidade de probabilidade para medir a verosimilhança da presença de um possível padrão (forma) numa imagem [Ste91]. Dado que a função densidade de probabilidade se adapta a alterações no *input*, esta apresenta um valor máximo no conjunto de parâmetros do padrão mais provável [Ste91];
- **Progressive Probabilistic Hough Transform (PPHT)** - esta Transformada começa por verificar se está a ser usada uma imagem não vazia e, caso não seja esse o caso, obtém pontos aleatórios desta e aplica um processo de votação idêntico ao da TH [MGK98]. O valor de cada ponto do acumulador usado, que é idêntico usado na TH, é comparado com uma função do número de votos para que sejam identificados casos que ultrapassem um dado *threshold* associado a ruído [MGK00]. Se o valor do *threshold* for ultrapassado, estamos perante uma linha detetada [MGK98]. Quando a PPHT deteta uma linha, removem-se da imagem de *input* todos os pontos que pertençam a esta linha, bem como outros pontos da mesma linha que ainda não tenham sido processados pelo algoritmo [MGK00]. A execução da PPHT termina quando todos os pontos da imagem estiverem processados [MGK00].
- **Randomized Hough Transform (RHT)** - a RHT extraí um conjunto de pontos aleatórios da imagem de *input*. Deste extraem-se aleatoriamente dois pontos a cada iteração [KK00]. No caso da deteção de linhas, estes pontos são usados para resolver duas equações na forma $y = mx + b$ [KXO90], em que (x, y) correspondem às coordenadas dos pontos extraídos. A partir destas obtém-se um ponto (m, b) do espaço de parâmetros [KXO90]. A RHT usa uma tolerância para identificar casos em que o valor de (m, b) resulta da média entre as coordenadas novas de (m, b) e as coordenadas antigas do mesmo ponto [XOK90, KXO90]. Os pontos do espaço de parâmetros são guardados numa estrutura dinâmica idêntica a uma árvore, em que cada nó desta estrutura contém as coordenadas do ponto e um contador [McL98]. Se for encontrado um ponto que pertença à linha descrita por (m, b) , o contador correspondente é incrementado [XOK90]. Se ainda não existir nenhum nó da estrutura com o par (m, b) correspondente, um novo nó é criado e o contador deste é incrementado [XOK90]. É usado um *threshold* para identificar pontos do acumulador que correspondem a parâmetros de linhas reais [McL98], ou seja, pontos (m, b) que pertencem a nós do acumulador e cujo contador tem valor acima do *threshold* dado [XOK90]. O processo

repete-se até não existirem pontos na amostra inicial [KXO90].

Além da PHT, PPHT e RHT, existem outras variações da Transformada de Hough que permitem fazer a deteção de formas não arbitrárias [Bal81].

2.3 Técnicas de Reconstrução 3D

A Reconstrução 3D pode ser feita com recurso uma multíitude de técnicas que variam na complexidade e no tipo de informação usada. A aquisição da informação necessária para inferir a estrutura 3D de um objeto pode ser feita de dois modos: com contacto ou sem contacto [Cur00]. As técnicas que adquirem com contacto a informação necessária para a reconstrução fazem-no com recurso a contacto físico com o objeto, podendo danificar o mesmo [ATV08]. Técnicas que adquirem sem contacto a informação necessária para a reconstrução permitem que esta seja obtida de um modo indireto e sem que seja necessário tocar fisicamente no objeto [ATV08]. Este tipo de técnicas usa câmaras ou outros sensores para medir a intensidade do sinal (por exemplo, luz ou laser) refletido pelo objeto [Kum14, ATV08]. Apesar de existirem vários tipos de técnicas de aquisição de estrutura 3D sem contacto, as mais usadas são aquelas que usam informação ótica [REH06].

As técnicas de aquisição de estrutura 3D que usam informação ótica, por sua vez, podem ser divididas em dois grupos, de acordo com o tipo de sistema de aquisição de informação usado [REH06, Cur00, Sei99]. Estes grupos são: técnicas ativas e técnicas passivas [Sei99, Cur00]. As técnicas ativas usam a emissão de um tipo de luz (por exemplo, lasers) e um sensor para medir a informação refletida pela superfície de um objeto e para recuperar a estrutura 3D desse objeto [Sei99]. As técnicas passivas são aquelas que usam a informação contida em imagens de um objeto para adquirir a estrutura 3D desse mesmo objeto [Kum14]. A figura 2.1, criada com recurso à informação disponível em [REH06, Cur00, Sei99, Kum14, ATV08], contém uma taxonomia das técnicas óticas de aquisição de estrutura 3D mais relevantes. No conjunto de técnicas ativas presentes na figura 2.1 estão incluídas:

- **Triangulações** - técnicas que usam a informação relativa à projeção de um raio luminoso e à deteção da reflexão deste por um sensor, após este raio ter atingido um objeto na cena [REH06]. Cada ponto 3D do objeto é estimado com recurso à triangulação entre a linha de visão da fonte luminosa e à posição do sensor que detetou a reflexão do raio luminoso [Cur00]. A estrutura 3D do objeto obtém-se movendo o mesmo e mantendo o raio luminoso estático de modo a que o este passe por todos os pontos do objeto [CL95, Cur00]. Alternativamente, pode mover-se o raio luminoso e manter o objeto estático de modo a que este seja todo processado [CL95, Cur00]. Podem verificar-se erros na aquisição do modelo 3D devido a variações na forma do objeto, que causam ruído, e devido a variações na reflexão do raio luminoso causadas por alterações na superfície do objeto [CL95].
- **Interferometria** - usa-se a projeção de luz a partir de duas fontes e são usadas as interferências criadas por estas projeções para obter a estrutura 3D do objeto [CLWZ08]. Para tal, medem-se as interferências geradas em zonas do objeto iluminadas por luz proveniente das duas fontes luminosas [Che00]. Com recurso a este tipo de técnica obtém-se modelos 3D muito precisos [CLWZ08, REH06].
- **Time of Flight** - este tipo de técnica usa a projeção de luz pulsada ou lasers numa cena [REH06]. Regista-se o momento em que o sinal é emitido e o momento em que a reflexão deste é detetada pelo sensor, sendo estes comparados [CLWZ08, Cur00]. Estes registo permitem obter uma distância entre o sensor e o objeto [Che00, CLWZ08, FBDH⁺13]. Este tipo de técnica produz modelos 3D relativamente completos, mas pode ser necessária informação adicional relativa às texturas dos mesmos [REH06].

Ainda na figura 2.1, incluem-se no grupo de técnicas passivas:

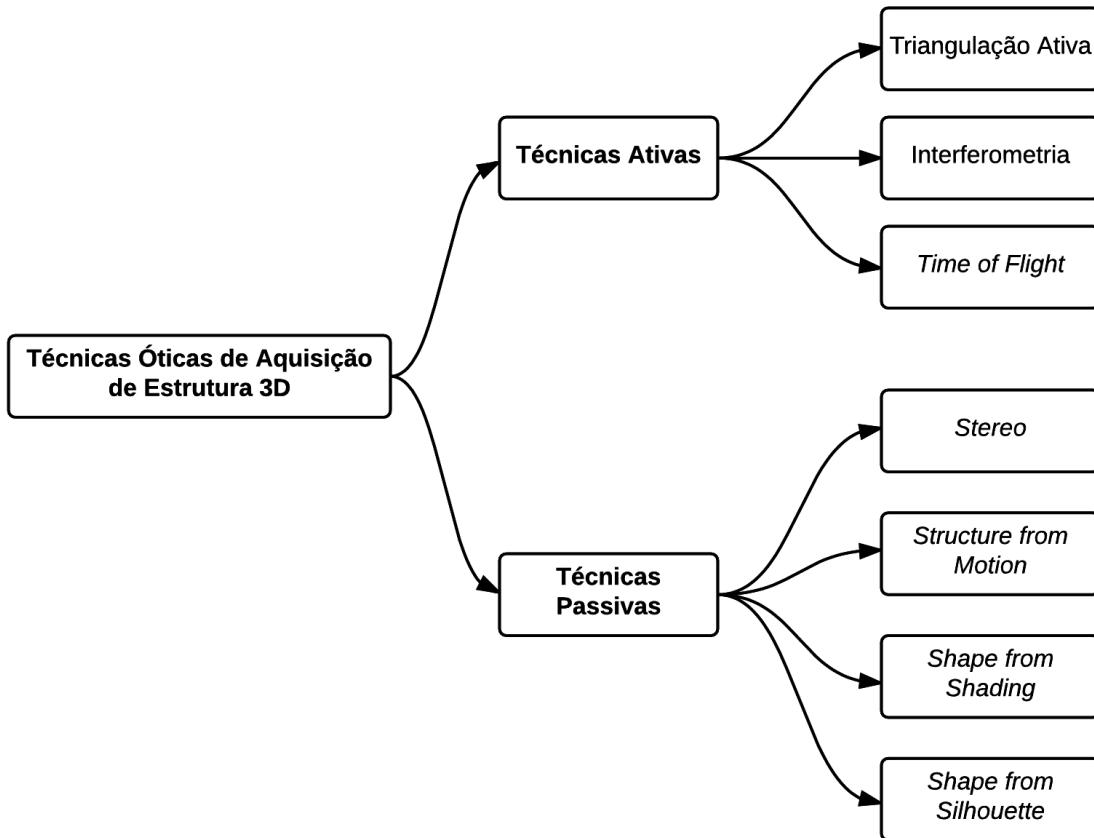


Figura 2.1: Taxonomia das técnicas óticas de aquisição de estrutura 3D.

- **Stereo** - esta técnica permite obter a estrutura da cena, usando duas ou mais imagens. Para poder fazer a reconstrução 3D, é necessário determinar correspondências entre estas imagens [Sei99]. Assim, usam-se algoritmos especializados para o efeito (algoritmos de correspondência). Scharstein apresenta uma taxonomia [SS02], em que são analisados muitos destes algoritmos de correspondência. Após serem estabelecidas as correspondências entre os elementos nestas imagens, são usadas triangulações para obter os modelos 3D dos objetos presentes nas imagens [Kum14]. Seitz [SCD⁺01] apresenta uma avaliação de vários algoritmos que permitem fazer reconstrução 3D usando stereo. A qualidade dos resultados obtidos depende da informação sobre a câmara que se encontra disponível [Sei99]. Portanto, a calibração das câmaras usadas para adquirir as imagens que servem de *input* a este método é um passo muito importante [Sei99]. Stereo tem diversas aplicações, tais como: reconstrução de cidades a partir de fotos [AFS¹¹, FFGG¹⁰], obtenção de modelos de património arqueológico [MTJ12, GY14, GRS14, BGZ⁰⁴], sistemas de visão robótica [SSPY10, GZS11], entre outros.
- **Structure from Motion** - esta técnica usa uma câmara em movimento [Sze10] ou, em alternativa, a cena move-se mas a câmara fica estática [JAP99]. A partir de cada imagem são obtidos elementos (por exemplo, pontos ou linhas) e estabelecidas correspondências entre estes [JAP99]. O modelo 3D é obtido com recurso à informação relacionada com o movimento dos pontos presentes nas imagens obtidas e com a informação relativa ao movimento da câmara [SD99, RC09]. A estrutura 3D da cena e a informação relativa ao movimento da câmara são obtidas em simultâneo [FBDH¹³, RC09]. Uma técnica idêntica a *Structure from Motion* é aplicada no Google Street View [KMR13].
- **Shape from Shading** - esta técnica permite obter um modelo 3D de um objeto opaco tendo por base uma imagem em que se encontra a informação relativa à intensidade da luz refletida pelo mesmo [KvD03].

Para que tal seja possível, são incluídas restrições como, por exemplo, assumir que a fonte de iluminação é sempre a mesma, que a localização desta fonte é conhecida e que as imagens são obtidas com a mesma câmara, entre outras [KvD03]. Independentemente da direção em que a observação é feita, a zona visível do objeto é proporcional à quantidade de luz que esta reflete [HW11]. Com a informação relativa à localização da fonte de luz e à intensidade da luz refletida pelo objeto, é possível obter o modelo 3D do mesmo [HW11]. *Photometric Stereo* é uma técnica idêntica a esta, mas usa mais do que uma imagem para obter a estrutura 3D do objeto [HW11, Sei99].

- **Shape from Silhouette** - com recurso a esta técnica é possível obter o modelo 3D de um objeto a partir de uma sequência de imagens de diferentes pontos de vista [SCMS01, HS02]. Com recurso à informação de calibração e à silhueta do objeto em cada imagem, é possível projetar um cone [Dye01, SCMS01]. Este é então intersecado com os cones provenientes das outras imagens, que foram obtidos da mesma forma [FBDH⁺13, Dye01, SCMS01]. A intersecção destes cones, denominada *visual hull*, contém o objeto reconstruído [FBDH⁺13, HS02, SCMS01, Dye01].

Além das técnicas indicadas na figura 2.1, existem metodologias híbridas, que usam componentes de técnicas ativas e passivas de modo a obter melhores resultados na reconstrução [REH06].

Tal como existem diversas técnicas que podem ser usadas para obter o modelo 3D de um objeto, também existem diversos algoritmos que permitem fazer reconstrução 3D. Lu, Zhang e Wu propõem uma classificação destes algoritmos [LZWL04]. Esta divide os algoritmos de reconstrução em grupos, de acordo com a fase em que a calibração das câmaras usadas é feita [LZWL04]:

- Pré-Calibração - a calibração das câmaras usadas é feita previamente e a informação obtida após este processo é usada pelo algoritmo de reconstrução;
- Calibração *Online* - A calibração da câmara, ou câmaras, é feita enquanto o processo de reconstrução decorre.

No grupo de algoritmos que usam pré-calibração, incluem-se os algoritmos que usam a informação relativa a correspondências presentes nas imagens para fazer a reconstrução (algoritmos baseados na imagem) [LZWL04]. Ainda neste grupo incluem-se também os algoritmos que usam *voxels* para discretizar a cena [SD99, Dye01, SCMS01] e os algoritmos baseados no objeto, que deformam um bloco 3D inicial (por exemplo, um cubo) para obter o modelo 3D deste objeto [FK02, JYS02, JSY03].

No grupo de algoritmos que fazem a calibração *online*, incluem-se algoritmos que usam características da cena e algoritmos que usam objetos geométricos abstratos para determinar os parâmetros da câmara [LZWL04]. Os algoritmos que usam as características da cena estimam os parâmetros intrínsecos e extrínsecos da câmara com recurso a informação que está já presente nas imagens obtidas (por exemplo, pontos de fuga e paralelismos entre linhas) [MK04]. Os algoritmos que usam objetos geométricos abstratos não usam informação relativa à estrutura da cena presente na imagem [LZWL04]. Neste caso, a calibração é feita recurso à imagem da projeção de um objeto geométrico abstrato [MK04].

Hoje em dia, o utilizador comum não necessita de conhecer todas as técnicas que permitem fazer a reconstrução 3D de um objeto. Este utilizador tem à sua disposição uma oferta cada vez maior de *scanners 3D* existentes no mercado e várias aplicações móveis. São exemplos de *scanners 3D*: *Matter Form 3D Scanner*³, *Sense 3D Scanner*⁴ e o *BQ Ciclop 3D Scanner*⁵, entre outros. Das várias aplicações móveis existentes, que permitem levar a Reconstrução 3D para qualquer lugar, destacam-se a *Scann3D* [Sma14], a *123D Catch* [Aut15], a *R-3D-2* [Mex15] e a *Seene* [See15], entre outras. Porém, caso o utilizador prefira criar algo mais adequado às suas

³<https://matterandform.net/scanner>

⁴<http://www.3dsystems.com/shop/sense>

⁵<http://www.bq.com/pt/ciclop>

necessidades, este pode construir a sua própria aplicação de reconstrução 3D com recurso às diversas bibliotecas existentes, tais como a *OpenCV*⁶ [B⁺00], a *SimpleCV*⁷ [DOOS12] e a *Matlab Vision System Toolbox*⁸, entre outras. Para tarefas que dependam do processamento de imagens muito grandes, como imagens de satélite, existem ferramentas especializadas, que incluem funcionalidades para fazer Reconstrução 3D. São exemplos disto o *NASA Vision Workbench* [HBE06] e o *Ames Stereo Workbench* [MBB⁺10]. Estas ferramentas podem ser usadas para criar modelos 3D a partir das imagens *stereo* captadas pelos satélites da NASA ou da Digital Globe [MBB⁺10, HBE06].

⁶opencv.org/

⁷simplecv.org/

⁸www.mathworks.com/products/computer-vision/

3

A Câmara e a Perspetiva

The camera is a save button for the mind's eye.

Rodger Kingston

A câmara é um elemento fundamental na Reconstrução 3D. Dado que é com este dispositivo que se obtêm as imagens do objeto a ser reconstruído, é importante compreender como ocorre o processo de formação de imagem. Além disto, é também importante compreender como se relaciona o sistema de coordenadas da câmara com a sua posição no mundo.

Este capítulo tem início com uma descrição da Câmara Buraco de Agulha e do respetivo modelo matemático. De seguida, é abordada a temática da Calibração da Câmara e dos parâmetros intrínsecos e extrínsecos. Finalmente, é incluída a descrição do que são as matrizes de projeção e de como é possível obtê-las.

3.1 Câmara Buraco de Agulha

A Câmara Buraco de Agulha, um tipo de Câmara Obscura, consiste num objeto com a forma de uma caixa em que existe apenas um orifício numa das suas faces [FP11]. O orifício, ou buraco de agulha, permite que a luz

do exterior entre na caixa, sendo projetada uma imagem na superfície oposta à que tem o buraco sem recorrer a uma lente [CS11]. A Câmara Obscura é estudada há vários séculos [Ren12]. O processo de aquisição de imagem deste tipo de câmara é descrito matematicamente pelo chamado Modelo Câmara Buraco de Agulha. Com recurso a este Modelo, é possível relacionar os pontos do mundo com os pontos de uma projeção desse mundo num plano de imagem [FP11].

3.1.1 Enquadramento Histórico

Imagens obtidas através da passagem da luz por uma pequena abertura têm sido observadas durante séculos. Este tipo de imagens intrigou os seus observadores desde cedo, sendo originárias da China, datando de 400 A.C., as primeiras descrições deste tipo de processo de aquisição de imagem [Ren12]. No Ocidente, as primeiras descrições do género datam de 330 A.C, tendo sido registadas por Aristóteles [Gre96]. Mais tarde, em 1020 D.C., a imagem invertida formada pela luz a atravessar um pequeno orifício foi estudada pelo físico egípcio Ibn al-Haitham [Ren12]. Aquando dos seus estudos relacionados com perspectiva, Leonardo da Vinci descreveu em detalhe pela primeira vez a câmara obscura [Ren12].

Na sua forma primitiva, a Câmara Obscura consistia numa sala, onde uma parede com um orifício permitia

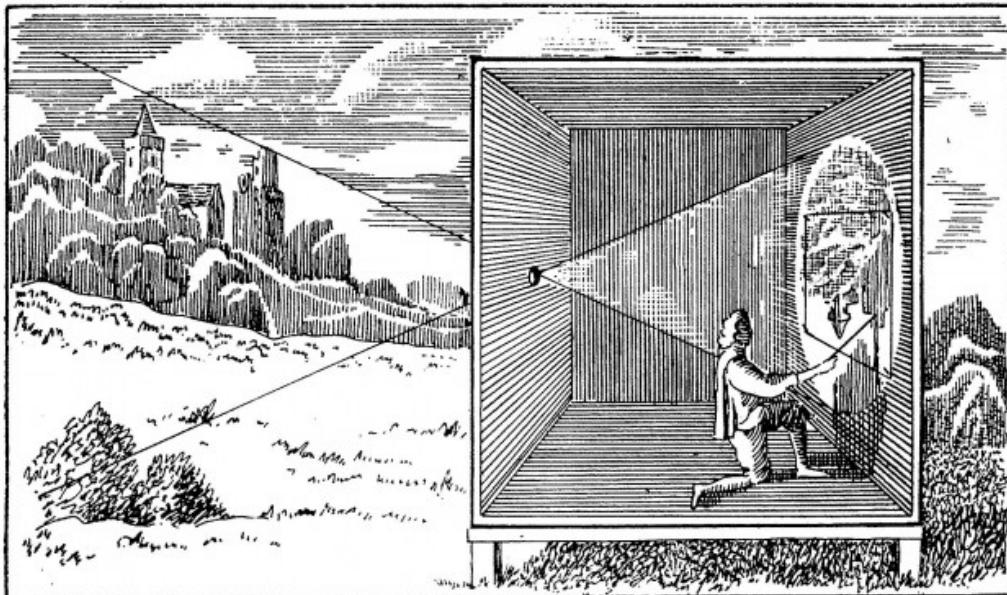


Figura 3.1: Uma Câmara Obscura [All05].

projetar uma imagem do mundo exterior na parede oposta (figura 3.1) [Ren12]. Este tipo de sala era usada tipicamente para estudar as leis da projeção e para observar eclipses solares [Bal01]. Mais tarde este princípio foi adaptado de modo a poderem ser criados objetos que produzissem o mesmo tipo de imagem [Gre96]. Estes objetos evoluíram, sendo usados como base para a construção da primeira câmara fotográfica.

Decorria o ano de 1850 quando Sir David Brewster, um cientista escocês, tirou a primeira fotografia com recurso a uma Câmara Buraco de Agulha [Ren12, Gre96]. O método viria a tornar-se popular durante o século XIX devido às imperfeições das fotos geradas, que contrastavam com as imagens bem definidas obtidas com recurso a lentes [Ren12, Gre96]. A Câmara Buraco de Agulha também provou novamente a sua utilidade em meados do século XX, quando foi usada por cientistas para fotografar Raios- α e Raios- γ [Bal01]. Eventualmente, a Câmara Buraco de Agulha caiu em desuso. Decorria a década de 1960 quando, ao fazer algumas experiências, artistas recuperaram a técnica de aquisição de imagem com recurso à Câmara Buraco de Agulha [Ren12, Bal01]. O interesse por este tipo de câmara foi assim novamente despertado e mantém-se até à atualidade.

3.1.2 O Modelo

Proposto por Brunelleschi no século XV, o Modelo Câmara Buraco de Agulha permite descrever de um modo simplificado o processo formação de imagem [FP11]. Com este Modelo matemático é possível descrever a relação existente entre um ponto no mundo e um ponto numa imagem, sendo descartados os efeitos criados por distorções e pelo uso de lentes [HZ03].

O buraco de agulha denomina-se centro da câmara, isto é, o ponto em que todos os raios de luz se intersetam

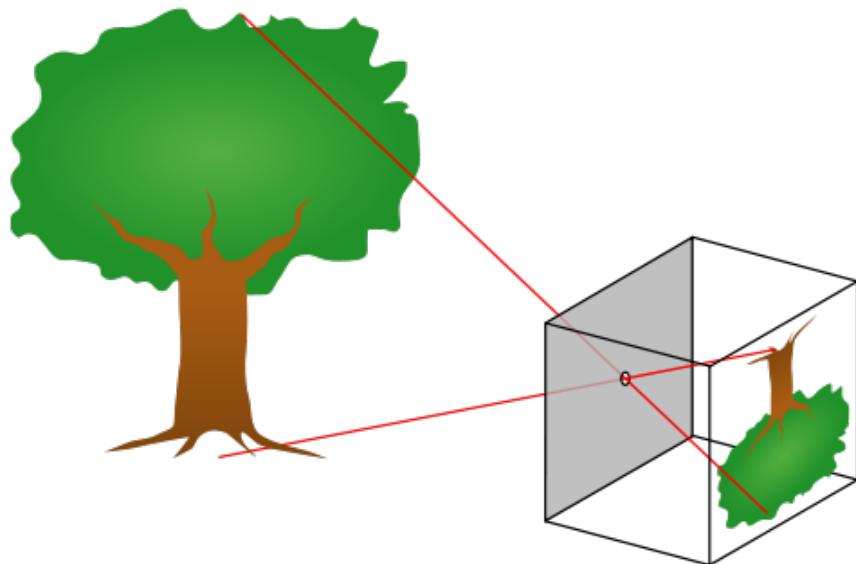


Figura 3.2: Representação de uma Câmara Buraco de Agulha [Mel05].

[HZ03]. Este orifício encontra-se entre o plano de imagem e o mundo 3D exterior (figura 3.2). Assim, um raio de luz que seja emitido ou refletido pelo mundo terá de passar pelo buraco de agulha antes de chegar ao plano de imagem [CS11, HZ03].

O Modelo Câmara Buraco de Agulha considera uma situação ideal, em que cada raio de luz que passa pelo buraco de agulha corresponde a apenas um dos pontos da imagem [FP11, HZ03]. Porém, como não é possível ter um buraco com uma área infinitesimal, um ponto no plano de imagem agrupa a informação proveniente de um cone de raios de luz provenientes do exterior [FP11].

3.2 Projeção de Perspetiva

A perda de uma dimensão, ou seja, a passagem do mundo tridimensional para uma imagem bidimensional denomina-se perspetiva [HZ03, FBDH⁺13]. O Modelo Câmara Buraco de Agulha descreve matematicamente a projeção de um ponto tridimensional do mundo, (X, Y, Z) , num ponto bidimensional da imagem, (u, v) . Este processo de mapeamento denomina-se projeção de perspetiva [BK08]. As coordenadas dos pontos do mundo no plano de imagem são dadas por (3.1) [HZ03, Sze10].

$$u = f \frac{X}{Z} \quad \text{e} \quad v = f \frac{Y}{Z}, \quad (3.1)$$

onde f é a distância focal, que corresponde à distância entre o plano de imagem e o ponto focal (buraco de agulha). A relação entre o ponto 3D do mundo e o ponto 2D da imagem pode ser descrita por:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (3.2)$$

em que P corresponde à matriz de projeção [HZ03, NA12].

Como, no Modelo Câmara Buraco de Agulha, o centro da câmara está entre o plano de imagem e o mundo real, as imagens resultantes da projeção de perspetiva aparecem invertidas verticalmente, isto é, "de cabeça para baixo" [FP11]. Para simplificar a compreensão do modelo, usa-se um esquema em que o plano de imagem aparece em frente do centro da câmara [FP11] (figura 3.3). Isto faz com que a imagem virtual criada apareça com a orientação dita "normal", isto é, sem ser invertida verticalmente. Pode usar-se este tipo de representação, visto que são conservadas as propriedades da imagem original [FP11].

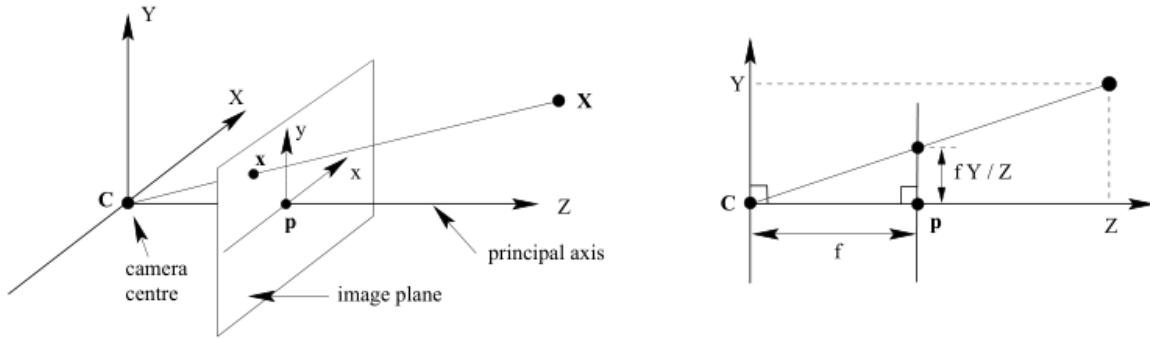


Figura 3.3: Representação esquemática da geometria do Modelo Câmara Buraco de Agulha [HZ03].

3.3 Calibração da Câmera

Para determinar a localização de uma câmara no mundo e a sua orientação interna, é necessário obter os parâmetros intrínsecos e extrínsecos da mesma. Estes obtêm-se através de um processo denominado calibração [Fau93, TV98, FP11, MK04]. Normalmente a calibração é feita apenas uma vez [BW95]. Porém, se a orientação da câmara no mundo for alterada, poderá ser necessário fazer uma nova calibração que reflita a alteração dos parâmetros extrínsecos (nova orientação da câmara) [CS11].

Existem vários métodos de calibração. A escolha do método a ser usado depende do tipo de tarefa para a qual é necessária a informação resultante da calibração [TV98]. No caso do stereo, é suficiente calibrar a câmara para estimar a sua matriz de projeção (sem obter os parâmetros intrínsecos e extrínsecos em separado) [Fau93]. O processo usado para estimar a matriz de projeção neste trabalho é apresentado em (3.3.3) e assemelha-se ao método de calibração apresentado por Hall [HTMS82], que é mencionado no Capítulo 2.

3.3.1 Parâmetros Intrínsecos da Câmera

Os parâmetros intrínsecos da câmara consistem num conjunto de características que permitem fazer corresponder os pontos no sistema de coordenadas da câmara a píxeis na imagem [TV98, FP11]. Trucco e Verri [TV98] definem estes parâmetros como um conjunto de elementos que são importantes para descrever as características da câmara relacionadas com a sua geometria e ótica. A estes parâmetros também podemos chamar

“orientação interna da câmara” [CS11]. Consideram-se parâmetros intrínsecos da câmara:

- **Coordenadas do Centro da Câmara** - Definem a posição correta do centro da câmara, visto que a origem do plano de imagem não está no centro deste [FP11]. Estas coordenadas são medidas em píxeis, porque correspondem a medidas na imagem [TV98];
- **Distância Focal** - A distância (em píxeis) entre o centro da câmara e o plano de imagem [BK08, HZ03];
- **Dimensões dos Píxeis** - Medidas de escala. Definem as dimensões dos píxeis da imagem, que podem ser quadrados ou retangulares [FP11].

Os parâmetros intrínsecos da câmara podem ser representados numa matriz:

$$\begin{bmatrix} k_x & 0 & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.3)$$

Em (3.3), k_x e k_y correspondem às dimensões dos píxeis e u_0 e v_0 correspondem às coordenadas do centro da câmara, respetivamente. A distância focal, f , pode também ser representada sob a forma de uma matriz, como em (3.4).

$$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

A combinação das matrizes (3.3) e (3.4) dá origem a uma terceira matriz, em que se encontram todos os parâmetros intrínsecos da câmara (K). Esta designa-se por matriz de calibração ou matriz da câmara [BW95, HZ03].

$$K = \begin{bmatrix} fk_x & 0 & u_0 \\ 0 & fk_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

3.3.2 Parâmetros Extrínsecos da Câmara

Com a informação relativa à localização da câmara no mundo (ou *camera pose* [FP11]), é possível passar do sistema de coordenadas da câmara para o sistema de coordenadas do mundo (ou cena) [CS11, TV98]. As coordenadas de um ponto no referencial da câmara (a) são obtidas a partir de uma rotação e de uma translação das coordenadas desse mesmo ponto no referencial do mundo (b) [TV98, Fau93, HZ03]. Assim,

$$a = R(b - T). \quad (3.6)$$

A transformação (3.6) pode ser escrita como:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & -RT \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}, \quad (3.7)$$

em que $[X_c, Y_c, Z_c, 1]^T$ é um ponto no referencial da câmara, $[X_w, Y_w, Z_w, 1]^T$ é um ponto no referencial do mundo, R é a matriz de rotação (3.8) e T é o vetor de translação (3.9).

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3.8)$$

$$T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (3.9)$$

Os parâmetros extrínsecos consistem na rotação, R , que alinha os referenciais da câmara e do mundo, e no vetor de translação, T , entre as origens destes mesmos referenciais [Fau93, TV98]. Os parâmetros extrínsecos podem ser escritos na forma matricial, como:

$$M = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -R_1^T T \\ r_{21} & r_{22} & r_{23} & -R_2^T T \\ r_{31} & r_{32} & r_{33} & -R_3^T T \end{bmatrix}, \quad (3.10)$$

na qual R_j , é o vetor correspondente à j-ésima linha da matriz R .

3.3.3 Matriz de Projeção

A matriz de projeção obtém-se a partir das transformações descritas pelos parâmetros intrínsecos e pelos parâmetros extrínsecos [Fau93]. Assim, combinando as matrizes dos parâmetros intrínsecos (3.5) e dos parâmetros extrínsecos (3.10), obtém-se a matriz de projeção P (equação (3.11)) [FP11]. Com recurso a P , é possível fazer a projeção de um ponto no mundo para um ponto na imagem [FP11].

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K_{3 \times 3} M_{3 \times 4} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \Leftrightarrow \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P_{3 \times 4} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.11)$$

A matriz de projeção pode, alternativamente, ser estimada [MK04, TV98]. Para tal, é necessário obter um conjunto de pontos de referência, cujas coordenadas são medidas em píxeis. Estes pontos podem ser obtidos manualmente, a partir de uma fotografia de um objeto de calibração que tenha dimensões conhecidas [MK04]. A cada um dos pontos obtidos a partir da imagem é atribuído um ponto tridimensional nas coordenadas do mundo [CS11, TV98, Fau93]. A relação entre os pontos de referência (u, v) obtidos a partir da imagem e os pontos (X, Y, Z) do mundo é dada por (3.12):

$$u = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}, \quad v = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}. \quad (3.12)$$

As equações lineares (3.12) podem também ser escritas como equações lineares nos coeficientes p_{ij} da matriz de projeção P :

$$\begin{aligned} p_{11}X + p_{12}Y + p_{13}Z + p_{14} &= p_{31}Xu + p_{32}Yu + p_{33}Zu + p_{34}u, \\ p_{21}X + p_{22}Y + p_{23}Z + p_{24} &= p_{31}Xv + p_{32}Yv + p_{33}Zv + p_{34}v. \end{aligned}$$

Para que seja possível estimar a matriz P , são necessários pelo menos seis pontos da imagem ($N \geq 6$) [TV98]. Para cada um dos pontos usados na calibração, existem equações lineares semelhantes a (3.12). A matriz P

estimada é a solução do sistema de equações lineares (3.13) [Fau93].

$$Ax = b. \quad (3.13)$$

Neste sistema, A é uma matriz de dimensões $2N \times 11$ [TV98], que depende das coordenadas dos pares de pontos de referência obtidos para fazer a calibração [Fau93], b é o vetor que contém as coordenadas de cada ponto de referência na imagem e x corresponde ao vetor $[p_{11}, p_{12}, \dots, p_{33}]^T$. O sistema (3.13) pode ser escrito, na forma matricial, como (3.14).

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & -X_1u_1 & -Y_1u_1 & -Z_1u_1 \\ \vdots & & & & & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & -X_nu_n & -Y_nu_n & -Z_nu_n \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -X_1v_1 & -Y_1v_1 & -Z_1v_1 \\ \vdots & & & & & & & & & \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -X_nv_n & -Y_nv_n & -Z_nv_n \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ \vdots \\ p_{33} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_n \\ v_1 \\ \vdots \\ v_n \end{bmatrix}. \quad (3.14)$$

No sistema de equações lineares apresentado em (3.14) existem mais equações (2N equações para 11 incógnitas). Assim, é necessário encontrar uma solução aproximada para este sistema, dado que não existe uma exata [FP11, TV98]. Esta pode ser obtida usando o método dos mínimos quadrados, que permite encontrar o vetor x que minimiza $\|Ax - b\|^2$ [HZ03, FP11, Fau93, TV98], ou seja,

$$\min_x \|Ax - b\|^2.$$

Este problema pode ser resolvido com recurso a equações normais [HZ03]:

$$(A^T A)x = A^T b.$$

Se $A^T A$ for invertível [HZ03], então a solução será dada por:

$$x = [(A^T A)^{-1} A^T] b.$$

O vetor x obtido contém os valores da matriz de projeção P , à exceção de p_{34} .

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \quad (3.15)$$

Para evitar soluções em que $[p_{11} \dots p_{34}] = 0$, pode usar-se uma normalização¹ em que $p_{34} = 1$ [MK04, AAK71]. Existem outras normalizações possíveis [MK04, Fau93, FT87].

¹Esta normalização introduz uma singularidade no caso do valor de p_{34} estar próximo de zero [MK04].

4

Deteção de Linhas em Imagens

Strive for continuous improvement, instead of perfection.

Kim Collins

A deteção de características em imagens, no inglês *feature detection*, é um passo importante na reconstrução 3D. As características (*features*) detetadas são elementos (pontos, formas, linhas, etc) que permitem identificar inequivocamente zonas específicas da imagem, continuando estas a ser passíveis de deteção mesmo que existam ruído ou distorções na imagem [CS11]. Estas características podem ser usadas para identificar correspondências nas imagens *stereo* que constituem o *input* usado na Reconstrução 3D [Sze10]. Existe uma multíitude de algoritmos que permitem obter estas características. Este capítulo foca-se num problema específico da deteção de características em imagens: a deteção de linhas.

As linhas são um dos tipos de características presentes em imagens. Detetá-las é importante, visto que com esta informação é possível descrever zonas específicas destas imagens. Como é referido no Capítulo 2, para este efeito existem muitos algoritmos especializados. Destes, destaca-se a Transformada de Hough, um dos algoritmos de deteção de linhas mais populares em Visão Computacional [Sze10]. A Transformada de Hough, mencionada no Capítulo 2, é descrita em detalhe no presente capítulo. É também apresentado um algoritmo que resulta da modificação da Transformada de Hough. Descrições mais detalhadas de cada uma das versões deste algoritmo modificado, que foram desenvolvidas ao longo deste trabalho, podem também ser consultadas neste capítulo.

4.1 Transformada de Hough

A Transformada de Hough (TH) é um algoritmo de deteção de linhas inventado por Paul Hough [Hou62]. A origem deste algoritmo remonta a 1962 e deveu-se à necessidade de encontrar um método que permitisse detetar, rapidamente e com um elevado grau de confiança, os rastos de partículas que atravessavam uma câmara de bolhas (*bubble chamber*) [Hou62, IK88, NA12]. Os rastos deixados pelas partículas eram fotografados e as linhas que compunham os mesmos eram detetadas pelo algoritmo. Assim, pouparavam-se tempo e recursos, porque a invenção do algoritmo suprimiu a necessidade de ter um observador humano a analisar estas fotos [IK88].

O algoritmo de Paul Hough descreve uma reta usando uma parametrização *slope-intercept* [NA12]. Com esta parametrização, a reta na forma apresentada em (4.1) é descrita pelos parâmetros m e b . Em (4.1), m corresponde ao declive da reta e b corresponde à ordenada na origem [NA12]. Cada ponto na imagem define uma linha no espaço de parâmetros [Hou62, NA12].

$$y = mx + b \quad (4.1)$$

Duda e Hart generalizaram o algoritmo de Paul Hough em 1971 [DH72], sendo esta a versão da TH mais popular no campo da Visão Computacional [IK88]. Nesta generalização, uma reta é descrita por (4.2) [DH72, BK08, IK88].

$$\rho = x \cos(\theta) + y \sin(\theta). \quad (4.2)$$

Esta versão da Transformada de Hough permite mapear cada ponto da imagem para um espaço de parâmetros em coordenadas polares (figura 4.2) [NA12]. Este espaço de parâmetros denomina-se Espaço Hough, ou *Hough Space*, e consiste num acumulador. Um exemplo de um Espaço Hough pode ser consultado na figura 4.1. Pontos no espaço de imagem correspondem a sinusoides no Espaço Hough [NA12, DH72]. Para cada ponto

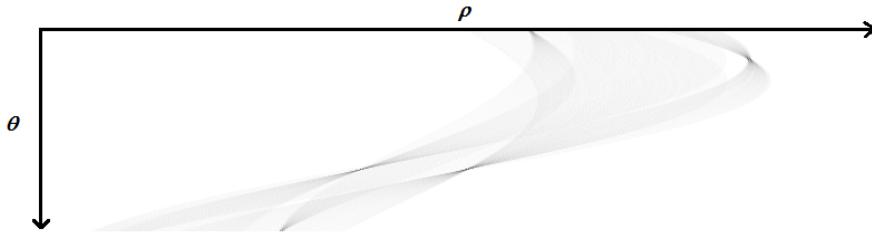


Figura 4.1: Exemplo de um Espaço Hough de uma imagem controlada (quatro linhas pretas sobre fundo branco).

do Espaço Hough é contado o número de linhas que podem passar por este ponto (número de intersecções de sinusoides) [FP11] num processo denominado processo de votação ou, no inglês, *voting process*. Se, no espaço de parâmetros, existe um ponto onde se intersectam vários sinusoides, isto significa que os pontos da imagem que correspondem a estes sinusoides são colineares [DH72].

Quando o processo de votação termina, obtém-se os parâmetros (coordenadas ρ e θ) dos pontos do acumulador que têm um número de votos acima de um valor dado. Este valor corresponde a um limite mínimo de votos a partir do qual se considera que foi encontrada uma linha (*threshold*) [FP11]. As coordenadas (ρ, θ) do ponto do acumulador que tiver o maior número de votos correspondem aos parâmetros que descrevem a linha na imagem [DH72]. Neste caso, ρ corresponde ao comprimento do vetor normal à linha a partir da origem e θ é a inclinação desse mesmo vetor (figura 4.2) [Sze10, DH72, BK08].

A partir deste ponto, quando for mencionado o algoritmo da Transformada de Hough (TH), está a ser referida a generalização criada por Duda e Hart [DH72]. Na implementação deste algoritmo que foi usada, θ varia entre

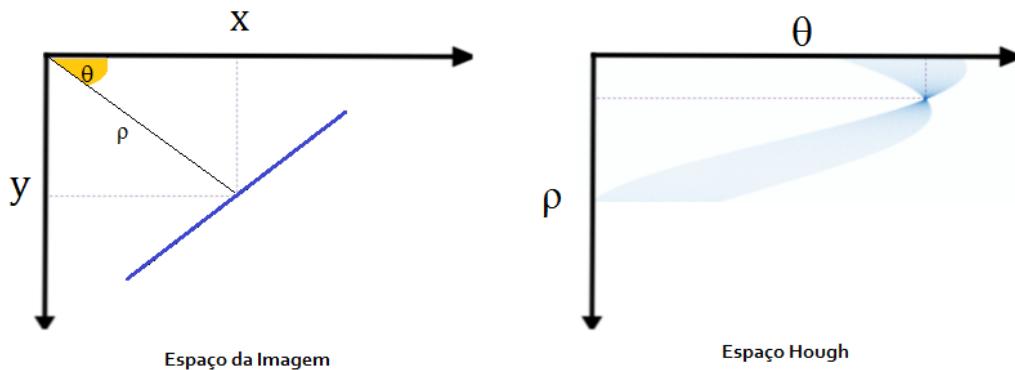


Figura 4.2: Espaços de parâmetros usados na Transformada de Hough.

zero e π e ρ varia entre zero e $\sqrt{N^2 + M^2}$, em que N e M são as dimensões da imagem. Segundo Duda e Hart, com este intervalo de valores, cada linha na imagem corresponde a exatamente um ponto do Espaço Hough [DH72].

4.2 Transformada de Hough Modificada

Apesar da sua precisão, a Transformada de Hough tem algumas características menos boas. Um desses aspectos é a existência de falsos positivos¹ nos resultados da deteção de linhas [GPKZ09]. Para reduzir o número de falsos positivos obtidos após a aplicação da TH, propõe-se uma modificação deste algoritmo.

A alteração da Transformada consiste na substituição do passo de deteção dos máximos locais por um passo de agrupamento (*clustering*). O algoritmo de *clustering* usado neste passo é o *K-Means* [M⁺67]. Além disto, nesta versão modificada do algoritmo da TH também deixa de se usar um *threshold*. Em vez disso, usa-se um parâmetro, k , que corresponde ao número de linhas a detetar². As linhas detetadas por este algoritmo modificado consistem nos centros dos grupos da última iteração do passo de *clustering*. Isto faz com que apenas k linhas sejam detetadas e, consequentemente, sejam reduzidos os falsos positivos.

Antes de ser aplicado o algoritmo modificado, as imagens (reais e controladas) passam por um passo de pré-processamento. Neste passo, as imagens são inicialmente convertidas para uma imagem em tons de cinza. Posteriormente, é aplicado um algoritmo de deteção de contornos. Neste caso, o algoritmo aplicado é o *Canny Edge Detector* [Can86] (mencionado no Capítulo 2). Assim, obtém-se uma imagem que contém apenas os contornos presentes na imagem original. Esta imagem é usada como *input* da Transformada de Hough Modificada (THM) aqui apresentada. O diagrama da figura 4.3 apresenta todos os passos de processamento que ocorrem durante o processo de deteção de linhas em imagens quando é aplicada a Transformada de Hough Modificada.

Algoritmo K-Means

O *K-Means* é um algoritmo de agrupamento (*clustering*), que permite criar grupos de dados de acordo com uma medida de proximidade [HKP11, TSK]. Pretende-se que a semelhança entre os elementos dentro destes grupos seja elevada e que existam diferenças relativamente acentuadas entre os grupos [HKP11].

¹Consideram-se falsos positivos: linhas detetadas que se encontram ligeiramente deslocadas das originais, linhas detetadas em locais da imagem onde não há linhas, linhas detetadas deslocadas da real devido a ruído na imagem original, etc.

² k corresponde também ao número de grupos do passo de *clustering*.



Figura 4.3: Passos usados na deteção de linhas com a Transformada de Hough Modificada.

Pode dizer-se que o *K-Means* é composto por duas fases: uma fase de agrupamento e uma fase de atualização dos centros³ destes grupos [TSK].

Na primeira fase, são obtidos k pontos do conjunto de dados, que são os centroides dos grupos iniciais da fase de agrupamento [TSK]. Estes pontos são obtidos de um modo aleatório. Para criar os grupos de pontos usa-se uma medida de proximidade (usualmente uma distância). Esta medida de proximidade permite determinar o quanto próximo um ponto está de cada um dos centroides existentes [HGP11]. De acordo com o resultado obtido no cálculo da medida de proximidade, o ponto irá pertencer ao grupo do centroide que se encontra mais próximo [M⁺67]. O passo de atualização dos centroides consiste em calcular as coordenadas do novo centroide de cada grupo usando uma média das coordenadas dos pontos pertencentes a cada grupo [TSK, M⁺67].

O *K-Means* vai continuar a sua execução até que uma condição de paragem seja verificada. Existem várias condições de paragem possíveis [MRS⁺08] como, por exemplo, os centroides serem os mesmos da iteração anterior ou os grupos criados serem os mesmos da iteração anterior, entre outras. A figura 4.4 permite observar algumas iterações deste algoritmo de *clustering*.

No caso específico do *K-Means* implementado neste trabalho, a condição de paragem usada consiste em verificar que os centroides não se alteraram (se são iguais aos usados na última iteração do *K-Means*).

4.2.1 Transformada de Hough Modificada - Versão 1

A Transformada de Hough Modificada - Versão 1 (THM-V1) consiste na implementação direta do algoritmo descrito em 4.2. A medida de proximidade usada nesta implementação é a Distância Euclidiana, apresentada em (4.3). Nesta fórmula, $a = (a_\rho, a_\theta)$ e $b = (b_\rho, b_\theta)$ correspondem a dois pontos genéricos do Espaço Hough.

$$d(a, b) = \sqrt{(a_\rho - b_\rho)^2 + (a_\theta - b_\theta)^2} \quad (4.3)$$

Na THM-V1, usa-se uma média ponderada para calcular as coordenadas do novo centroide de cada grupo. Neste cálculo, o número de votos de cada ponto do grupo é usado como um peso (w_p). As fórmulas de cálculo usadas para obter as coordenadas dos novos centroides são (4.4) e (4.5). Em (4.4), p_ρ corresponde à coordenada ρ de cada ponto usado para calcular a média ponderada. c_ρ e c_θ correspondem às coordenadas ρ e θ do novo centroide, respectivamente. Em (4.5), é usada uma notação análoga para o cálculo da coordenada θ (c_θ) do novo centroide, na qual p_θ é a coordenada θ de cada ponto usado para calcular a média.

$$c_\rho = \frac{\sum_{p \in G} p_\rho w_p}{\sum_{p \in G} w_p} \quad (4.4)$$

$$c_\theta = \frac{\sum_{p \in G} p_\theta w_p}{\sum_{p \in G} w_p} \quad (4.5)$$

³Os centros destes grupos de dados denominam-se centroides [TSK].

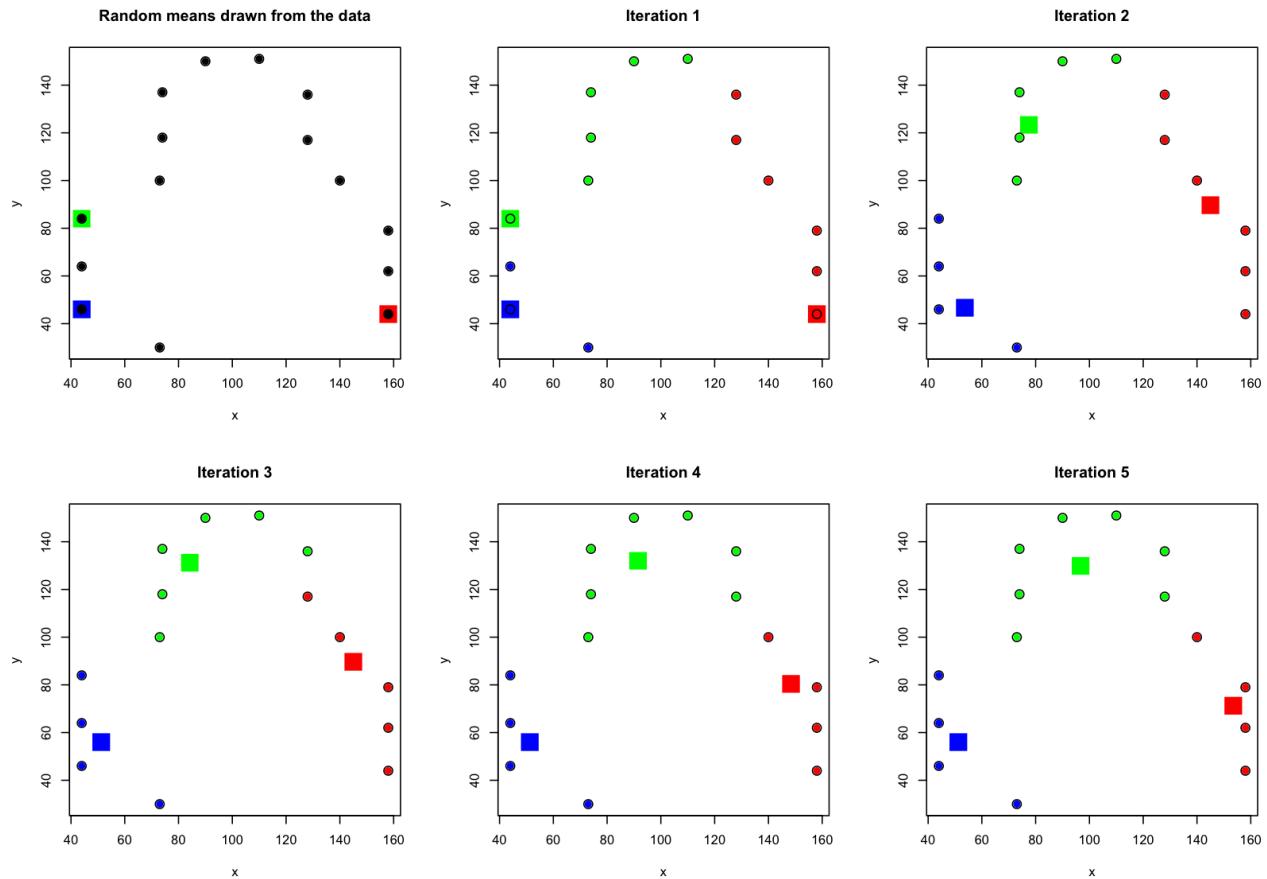


Figura 4.4: Exemplo do funcionamento do algoritmo K-Means com centroides iniciais aleatórios. Os quadrados coloridos representam os centroides [Com05].

Para analisar o desempenho do algoritmo em diferentes contextos, foram feitos testes com imagens controladas⁴ e com fotografias.

Aplicando a Versão 1 da Transformada de Hough Modificada a imagens de teste controladas, verificam-se resultados pouco precisos (figura 4.5-(b)). Como pode ser observado, apesar de serem detetadas quatro linhas, estas aparecem deslocadas das reais. Nos testes em que se usaram fotografias (figura 4.6-(b)) para testar a THM-V1, verifica-se um comportamento semelhante. As retas detectadas, assinaladas a vermelho na figura 4.6, aparecem deslocadas da posição correta e com uma inclinação incorreta. Os resultados detalhados dos testes feitos à THM-V1 podem ser consultados no Anexo A.

A falta de precisão da THM-V1 pode estar relacionada com a forma usada para calcular os novos centroides no passo de *clustering*. Como neste passo é usado o número de votos de cada ponto, estes pesos têm influência no resultado final da deteção. Assim, se existirem muitos pontos do acumulador com um baixo número de votos que estejam próximos de um ponto que tenha um elevado número de votos, o resultado da deteção consistirá em linhas deslocadas das originais. A Transformada de Hough Modificada - Versão 2, apresentada neste Capítulo, pretende resolver este problema.

⁴Neste caso, considera-se uma imagem controlada aquela que tem um fundo branco e linhas pretas.

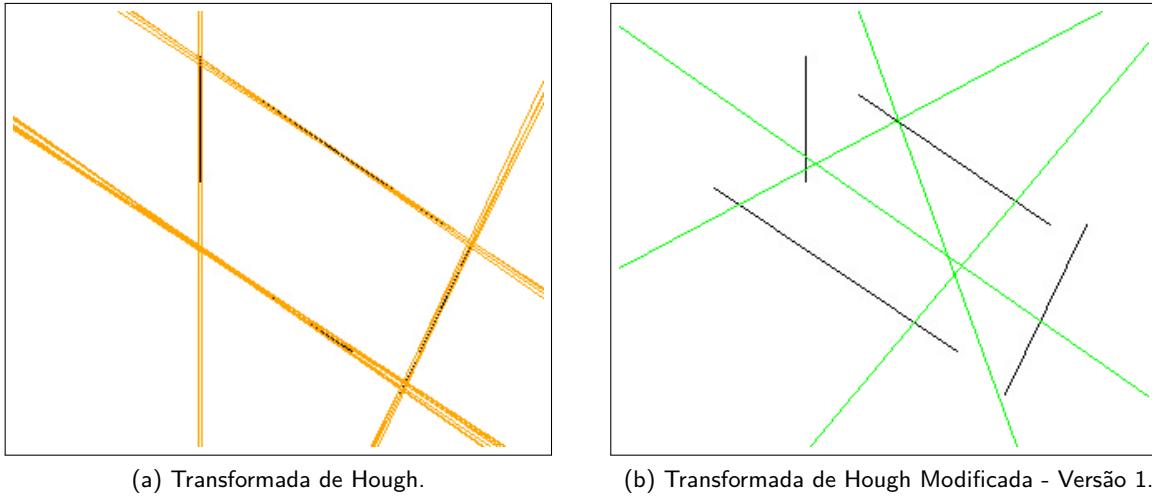


Figura 4.5: Linhas detetadas aplicando a Transformada de Hough e a Transformada de Hough Modificada - Versão 1 à mesma imagem controlada. As deteções correspondem às linhas coloridas.

4.2.2 Transformada de Hough Modificada - Versão 2

Tendo em conta a baixa precisão na deteção de linhas observada usando a THM-V1, foi introduzida uma nova modificação que permita melhorar o desempenho do algoritmo. Esta alteração consiste em elevar a um expoente o acumulador do processo de votação usado na Transformada de Hough Modificada. Isto reduz a influência de pontos que tenham poucos votos no cálculo dos novos centroides no passo de agrupamento do *K-Means*. Por exemplo, se existirem vários pontos com apenas um voto em torno de um ponto que tem mais votos, elevar o número de votos a um expoente fará com que o número de votos do ponto que tem mais votos se destaque dos restantes. A Transformada de Hough - Versão 2 (THM-V2) resulta desta alteração da THM-V1, o que permite usar uma média ponderada ao invés de uma média simples quando são calculados os novos centroides no passo de *clustering*.

Na THM-V2, a medida de proximidade usada no passo de agrupamento do *K-Means* não foi alterada, continuando esta a ser a Distância Euclideana (4.2.1). As coordenadas dos novos centroides dos grupos calculam-se usando as fórmulas (4.6) e (4.7). Nestas, α representa o expoente a que se elevaram os pesos, p_θ e p_ρ são as coordenadas θ e ρ dos pontos usados para calcular a média e w_p corresponde ao número de votos de cada um desses pontos. c_θ e c_ρ são as novas coordenadas θ e ρ dos novos centroides.

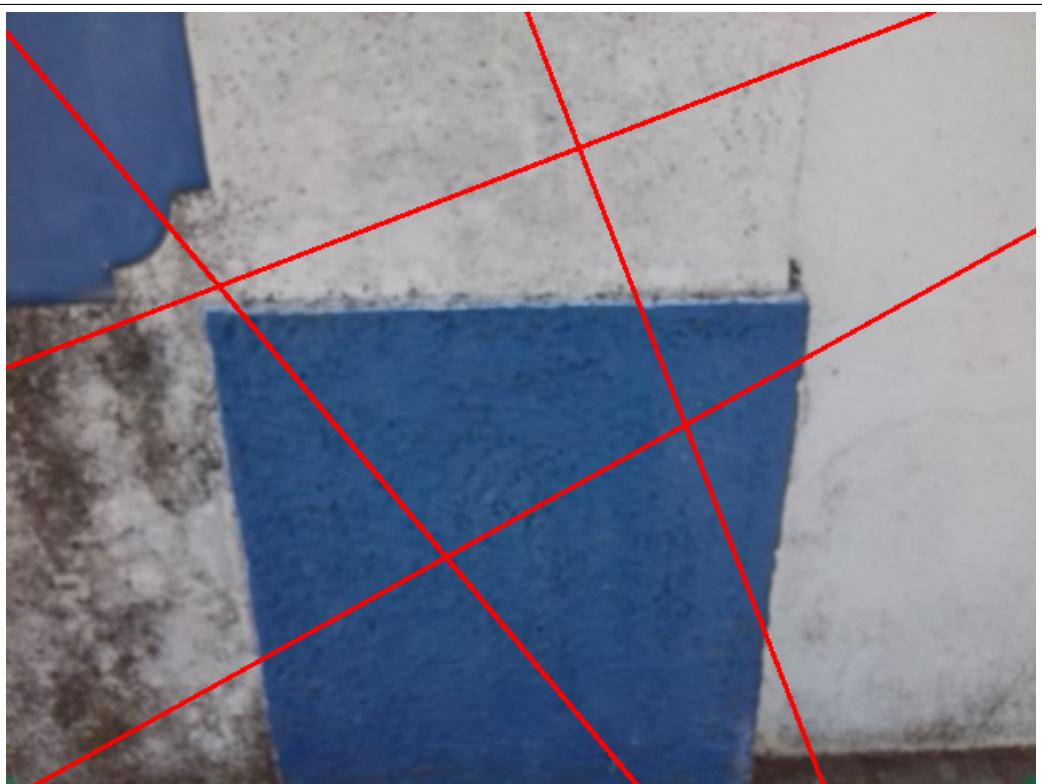
$$c_\rho = \frac{\sum_{p \in G} p_\rho w_p^\alpha}{\sum_{p \in G} w_p^\alpha} \quad (4.6)$$

$$c_\theta = \frac{\sum_{p \in G} p_\theta w_p^\alpha}{\sum_{p \in G} w_p^\alpha} \quad (4.7)$$

À semelhança do que aconteceu em 4.2.1, também a Versão 2 da Transformada de Hough Modificada foi testada com imagens controladas e com fotografias. Neste casos, consideram-se dois casos de teste: usar o mesmo expoente para elevar a coordenada θ e a coordenada ρ e usar expoentes diferentes para elevar cada uma destas coordenadas. Com estes dois casos de teste pretende-se verificar se existem mais vantagens em usar o mesmo expoente para elevar as coordenadas ρ e θ no cálculo da média ponderada ou se existem melhorias quando



(a) Transformada de Hough.



(b) Transformada de Hough Modificada - Versão 1.

Figura 4.6: Linhas detetadas aplicando à mesma fotografia a Transformada de Hough e a Versão 1 da Transformada de Hough Modificada. As linhas detectadas são indicadas com linhas coloridas.

se usam expoentes diferentes no mesmo passo.

O resultado de um dos testes, com imagens controladas, em que foi aplicada a THM-V2 pode ser observado na figura 4.7-(b). Neste teste foi usado o mesmo expoente⁵ para ρ e para θ no passo de *clustering*. As linhas detetadas sobrepõem-se a todas as linhas detetadas às linhas reais presentes na imagem. Em 4.7-(c) pode ser analisado o resultado de um dos testes em que se usaram expoentes diferentes neste mesmo passo do algoritmo⁶. Também na figura 4.7-(c) se verifica que as linhas detetadas se encontram sobrepostas às reais.

A Versão 2 da Transformada de Hough Modificada foi também aplicada ao problema da deteção de linhas em fotografias. Os testes em que se usam fotografias também contemplaram os dois casos de teste mencionados anteriormente (expoentes iguais e expoentes diferentes). Alguns resultados destes testes podem ser consultados na figura 4.8. Para o caso dos expoentes iguais, foi usado um expoente igual a 20. O resultado obtido para este caso de teste (figura 4.8-(b)) mostra três linhas detetadas a delimitarem o quadrado central presente na fotografia e uma linha com uma inclinação incorreta. O resultado do teste na figura 4.8-(c), no qual foram usados expoentes diferentes para ρ e para θ , mostra duas das linhas detetadas sobre as linhas reais e duas com uma inclinação diferente da real. Todos os resultados de todos os testes da THM-V2 podem ser consultados no Anexo A.

Comparando os resultados dos testes da THM-V2 com os resultados obtidos nos testes da THM-V1, verifica-se uma melhoria. Porém, como a medida de proximidade usada na Transformada de Hough Modificada - Versão 2 não é apropriada para o tipo de coordenadas do Espaço Hough, é necessário obter uma nova medida de proximidade para o passo de agrupamento do *K-Means*. Esta alteração origina a Versão 3 da Transformada de Hough Modificada.

4.2.3 Transformada de Hough Modificada - Versão 3

A Transformada de Hough Modificada - Versão 3 (THM-V3) conserva os aspectos positivos das versões que a antecederam. Porém, esta versão da THM usa uma medida de proximidade diferente das usadas nas versões anteriores. Como estamos a trabalhar com um espaço de coordenadas polares, a fórmula da Distância Euclidiana simples, usada nas duas versões anteriores deste algoritmo, não pode ser usada como medida de proximidade. Assim, é necessário encontrar uma fórmula alternativa que funcione neste contexto. Para fazer face a este problema, explorou-se a possibilidade de adaptar a fórmula da Distância Euclidiana (4.3), que funciona corretamente para coordenadas cartesianas, de modo a que esta funcione neste contexto⁷. A figura 4.9 permite observar uma representação esquemática do espaço de coordenadas em que estamos a trabalhar. Assim, é possível verificar que o ângulo (θ_3) entre as linhas que vão desde a origem até aos pontos p_1 e p_2 é dado por

$$\theta_3 = \theta_1 - \theta_2.$$

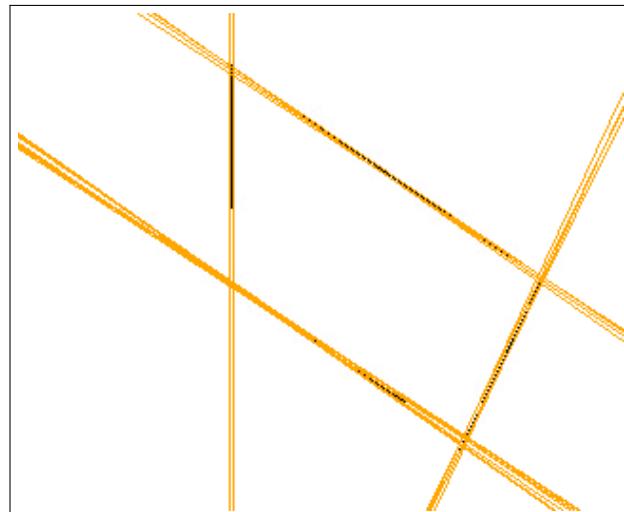
É necessário ter algum cuidado no cálculo desta diferença entre ângulos. Como θ varia entre zero e 180° no espaço em que estamos a trabalhar, é importante garantir que não se obtêm resultados fora deste intervalo quando se calcula a diferença entre os valores das coordenadas θ dos pontos. Assim, para assegurar que não se obtêm valores sem sentido, a fórmula de cálculo usada para este efeito é (4.8).

$$\theta_3 = \begin{cases} |\theta_1 - \theta_2| \bmod 360, & \text{Se } \theta_3 < 180^\circ \\ 360 - |\theta_1 - \theta_2|, & \text{Se } \theta_3 \geq 180^\circ \end{cases} \quad (4.8)$$

⁵Neste caso, o expoente usado tem o valor 20.

⁶Os expoentes usados foram 9 para ρ e 10 para θ , respectivamente.

⁷Testou-se também o caso em que $d = \sqrt{p_\rho^2 + c_\rho^2 - 2p_\rho c_\rho \cos(p_\theta - c_\theta)}$ mas, devido a uma deterioração significativa dos resultados da deteção de linhas, esta ideia foi abandonada.



(a) Transformada de Hough.

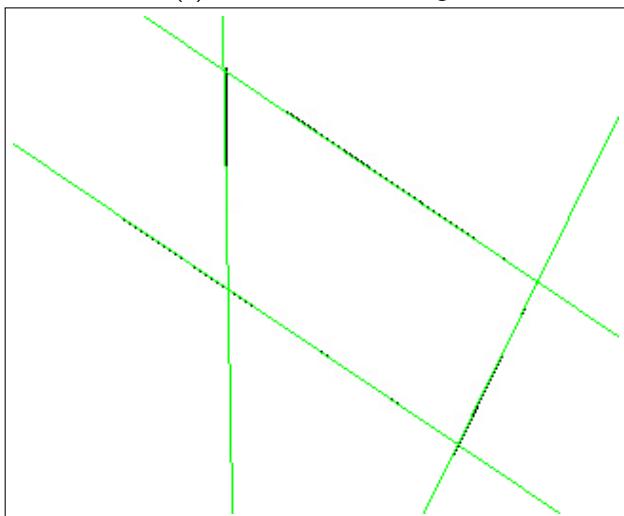
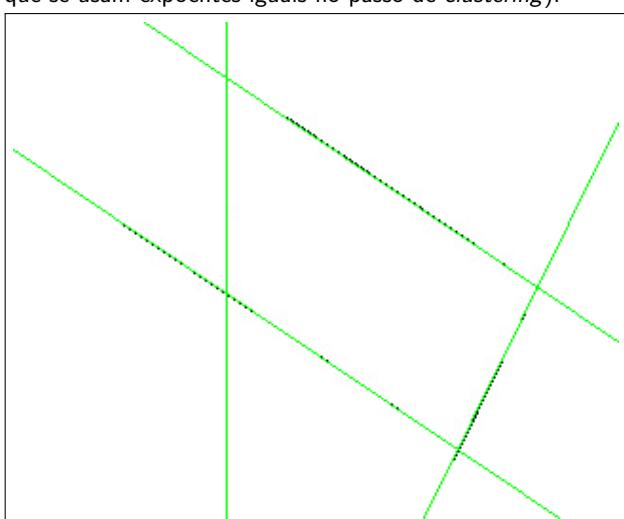
(b) Transformada de Hough Modificada - Versão 2 (Teste em que se usam expoentes iguais no passo de *clustering*).(c) Transformada de Hough Modificada - Versão 2 (Teste em que se usam expoentes diferentes no passo de *clustering*).

Figura 4.7: Linhas detetadas pelas Transformada de Hough Modificada - Versão 2 e Transformada de Hough, quando aplicadas a uma imagem controlada.



(a) Transformada de Hough.

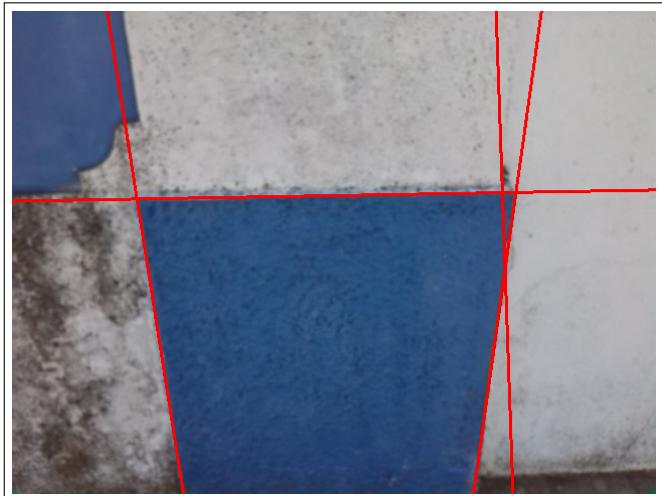
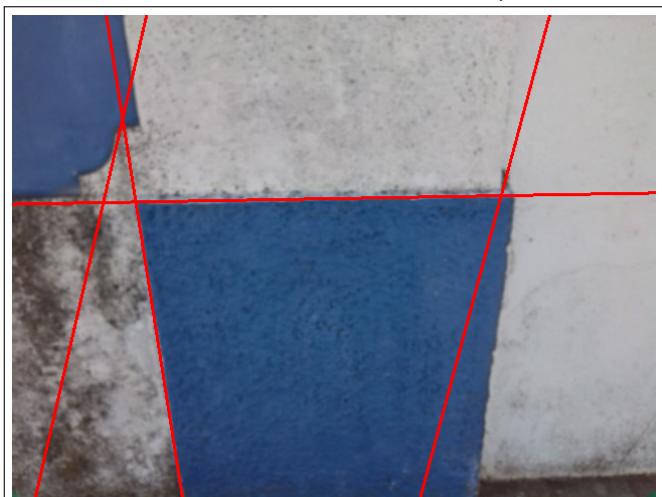
(b) Transformada de Hough Modificada - Versão 2 (Teste em que se usam expoentes iguais no passo de *clustering*).(c) Transformada de Hough Modificada - Versão 2 (Teste em que se usam expoentes diferentes no passo de *clustering*).

Figura 4.8: Resultado de um dos testes efetuados aplicando a TH e a Versão 2 da Transformada de Hough Modificada a fotografias.

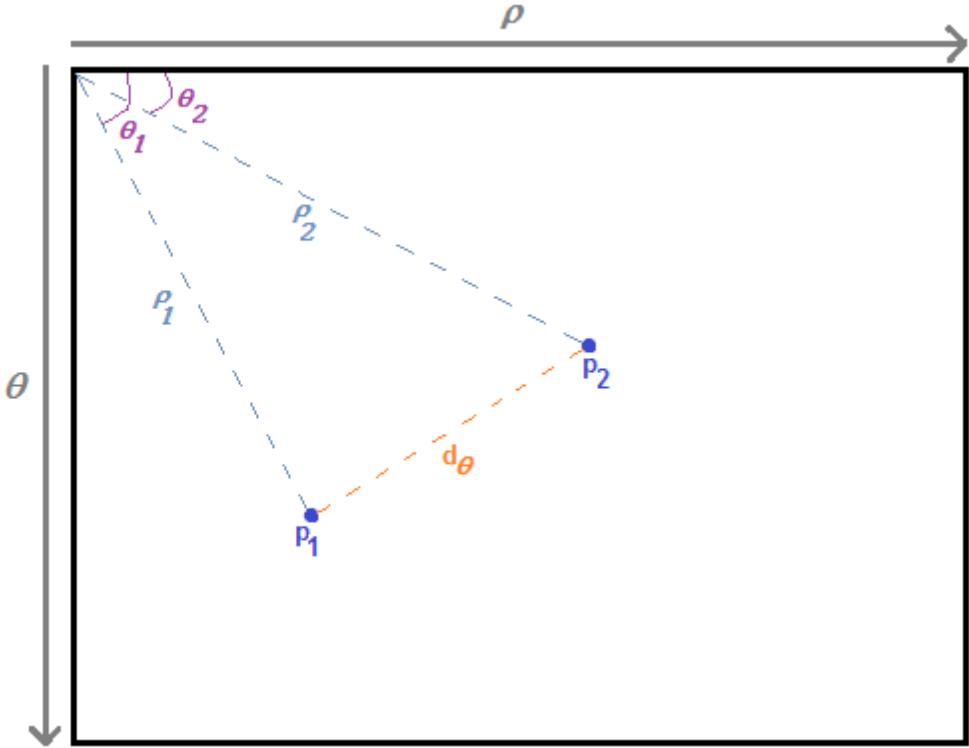


Figura 4.9: Representação esquemática do Espaço Hough.

Assim, a distância, em θ , entre os pontos p_1 e p_2 é dada por

$$d_\theta = \sin(\theta_3).$$

A distância, em ρ , entre os pontos p_1 e p_2 é dada por

$$d_\rho = p_{1\rho} - p_{2\rho}.$$

Tendo em conta d_θ e d_ρ , a fórmula da Distância Euclidiana (4.3), adaptada a este caso, é (4.9).

$$d = \sqrt{d_\rho^2 + d_\theta^2} = \sqrt{(p_\rho - c_\rho)^2 + \sin(p_\theta - c_\theta)^2} \quad (4.9)$$

Porém, é ainda necessário fazer um ajuste a (4.9), de modo a que a fórmula tenha uma sensibilidade tão equilibrada quanto possível a variações em ρ e θ . Isto é necessário porque, caso não seja feito este ajuste, d_ρ vai ter mais influência que d_θ no cálculo da distância entre pontos e centroides. Adaptando a fórmula (4.9) de modo a incluir esta normalização, obtém-se (4.10), em que μ é um coeficiente arbitrário com valor positivo usado para normalizar ρ , (p_ρ, p_θ) correspondem às coordenadas de um ponto do conjunto de dados (o Espaço Hough) e (c_ρ, c_θ) são as coordenadas do centroide.

$$d = \sqrt{\left(\frac{p_\rho - c_\rho}{\mu}\right)^2 + \sin(p_\theta - c_\theta)^2} \quad (4.10)$$

A fórmula (4.10) é a medida de proximidade escolhida para calcular a distância entre pontos e centroides no passo de *clustering* da Transformada de Hough Modificada - Versão 3.

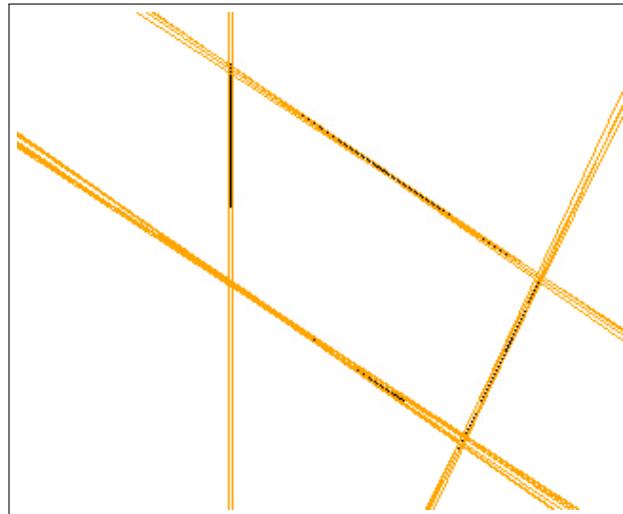
Tal como aconteceu com a THM-V2, a Transformada de Hough Modificada - Versão 3 foi testada com fotografias

e com imagens controladas. Para cada tipo de imagem foram testados dois casos: expoentes iguais e expoentes diferentes para elevar as coordenadas ρ e θ .

A figura 4.10 apresenta um dos vários testes efetuados com recurso a imagens controladas e expoentes iguais. Como pode ser observado na figura 4.10-(b), quando são usados expoentes iguais para as coordenadas ρ e θ no passo de agrupamento, todas as linhas detetadas se sobrepõem às reais. O expoente usado neste caso tem o valor 20 e μ , que corresponde ao coeficiente usado para normalizar ρ , toma o valor 900. Quando são usados expoentes diferentes no mesmo passo (figura 4.10-(c)), uma das linhas detetadas tem uma inclinação ligeiramente diferente da linha real, embora esteja muito próxima da mesma. Os expoentes usados neste passo têm os valores 9 e 10, respectivamente. Quando se usam expoentes diferentes, μ , que é usado para normalizar ρ no cálculo da distância entre pontos e centroides, também deve ser ajustado. Assim, no caso em que os expoentes são diferentes, μ toma o valor 100. Os valores dos expoentes usados nos testes da Versão 3 são idênticos aos dos testes da Versão 2 da Transformada de Hough Modificada. Isto permite estabelecer uma base de comparação entre os resultados dos testes de ambas as versões do algoritmo.

A THM-V3 foi também testada com fotografias, tendo sido aplicadas as mesmas condições de teste dos exemplos com imagens controladas⁸. Analisando a figura 4.11-(b), verifica-se que foram detetadas quatro linhas na fotografia. Apenas uma destas linhas tem uma inclinação diferente da correta. As restantes linhas detetadas estão sobrepostas às reais. No caso de teste em que se usam expoentes diferentes para ρ e para θ (figura 4.11-(c)), foram também detetadas corretamente quatro linhas. Neste caso verifica-se a existência de duas linhas detetadas que têm uma inclinação incorreta. Comparando os resultados da THM-V3 com os das versões deste algoritmo apresentadas anteriormente, é possível concluir que a Versão 3 apresenta resultados idênticos aos da THM-V2. Os resultados de todos os testes feitos à THM-V3 podem ser consultados no Anexo A.

⁸Usam-se os mesmos valores para os expoentes - 9 e 10 para o caso dos expoentes diferentes e 20 para o caso dos expoentes iguais. Também se preservam os valores do coeficiente usado para normalizar ρ : 900 para expoentes iguais e 100 para expoentes diferentes, respectivamente.



(a) Transformada de Hough.

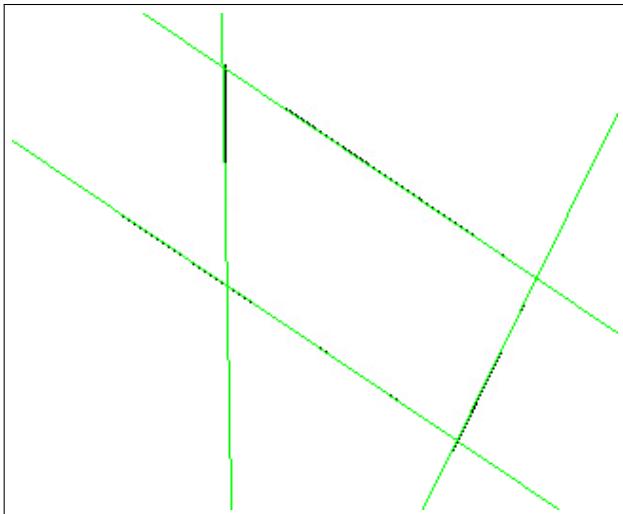
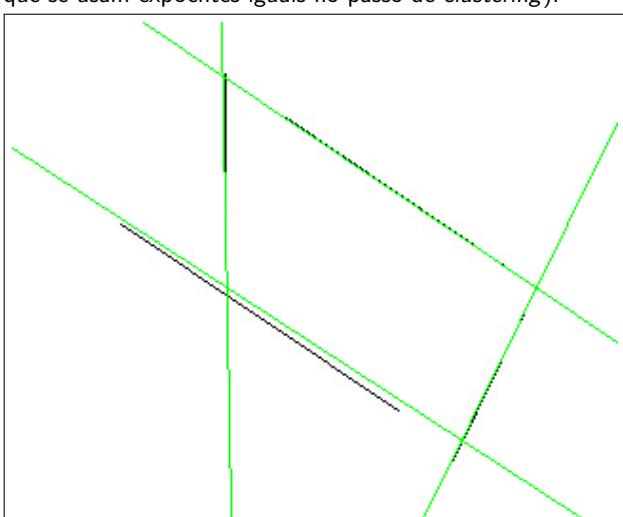
(b) Transformada de Hough Modificada - Versão 3 (Teste em que se usam expoentes iguais no passo de *clustering*).(c) Transformada de Hough Modificada - Versão 3 (Teste em que se usam expoentes diferentes no passo de *clustering*).

Figura 4.10: Resultado da aplicação, a imagens controladas, da Transformada de Hough e da THM-V3.



(a) Transformada de Hough.

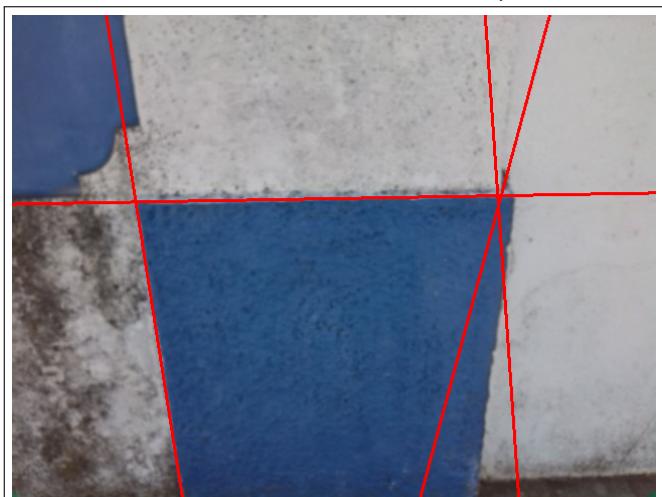
(b) Transformada de Hough Modificada - Versão 3 (Teste em que se usam expoentes iguais no passo de *clustering*).(c) Transformada de Hough Modificada - Versão 3 (Teste em que se usam expoentes iguais no passo de *clustering*).

Figura 4.11: Linhas detetadas pelas Transformada de Hough e Transformada de Hough Modificada - Versão 3, quando aplicadas a fotografias.

5

Reconstrução 3D

It's better to be a pirate than to join the Navy.

Steve Jobs

Neste trabalho usa-se uma técnica ótica passiva (Stereo) para fazer a reconstrução 3D. Assume-se que foi feita previamente a calibração das câmaras usadas, sendo conhecidas as matrizes de projeção das mesmas. A reconstrução é feita com recurso à informação presente nas imagens recolhidas, pelo que pode dizer-se que os algoritmos aqui apresentados são baseados na imagem. O presente capítulo inclui a fundamentação teórica do método usado para fazer reconstrução 3D neste trabalho.

A reconstrução 3D é uma das várias áreas que compõem o campo da Visão Computacional. Especialmente importante quando se pretende simular o processo de percepção tridimensional da visão humana, a reconstrução 3D com Stereo é um problema complexo [BBH]. Porém, de modo a simplificar a compreensão deste problema, é possível subdividir o mesmo (ver figura 5.1) em dois problemas mais simples: encontrar correspondências entre as imagens obtidas e reconstrução [BBH, FP11].

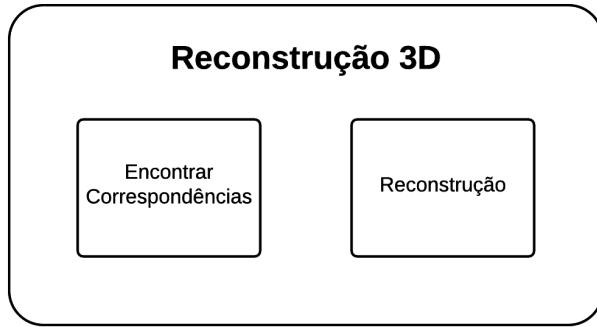


Figura 5.1: Os dois problemas que constituem a reconstrução 3D.

5.1 Correspondência de Linhas

Um passo importante na reconstrução 3D é o estabelecimento de correspondências entre elementos detectados nas imagens *stereo*. Este tipo de correspondência denomina-se *feature-based* e usa elementos da imagem que são preservados perante ruído ou transformações geométricas [CS11]. Para encontrar correspondências usando os elementos das imagens, podem usar-se linhas, pontos ou cantos [CS11].

Neste trabalho, pretendem-se identificar correspondências entre linhas detectadas por um algoritmo especializado. Porém, este processo é significativamente mais complexo que encontrar correspondências entre pontos [Ela99, SZ97, BFV05]. Este aumento na complexidade pode dever-se a vários fatores, tais como:

- Oclusões [SZ97], ou seja, linhas ou partes das mesmas podem não ser visíveis na imagem;
- Problemas na extração das características das linhas [SZ97, Ela99], nomeadamente na obtenção dos pontos que se encontram nos extremos das linhas e perdas da informação relativa às ligações entre estas na imagem;
- Inexistência de uma forma geométrica que permita identificar inequivocamente correspondências corretas a partir de um grupo de possíveis candidatas [SZ97, Ela99];
- Não é possível explorar a geometria epipolar para encontrar correspondências entre linhas do mesmo modo que esta é usada para encontrar correspondências entre pontos [BFV05].

De acordo com Schmid e Zisserman [SZ97], existem dois tipos de abordagem ao problema da correspondência de linhas: encontrar correspondências entre linhas individuais e obter correspondências entre grupos de linhas. A primeira abordagem aposta em encontrar correspondências usando as características geométricas¹ das linhas [SZ97]. A abordagem alternativa consiste em encontrar correspondências entre grupos de linhas que têm características idênticas. Isto permite identificar corretamente mais correspondências entre linhas devido à maior quantidade de informação disponível para descartar ambiguidades² [SZ97].

Neste trabalho, as linhas detectadas na imagem passam por um pré-processamento, onde são agrupadas de acordo com o seu tipo (linhas verticais, horizontais ou diagonais). O algoritmo usado neste trabalho estabelece correspondências entre as linhas de um par de imagens *stereo* com base no tipo de linha. Por exemplo, quando se procura uma linha vertical correspondente a uma linha detectada numa imagem do par, o algoritmo procura a correspondente no grupo de linhas verticais da outra imagem do mesmo par. Assim, evitam-se casos em que linhas de tipos diferentes sejam dadas como correspondentes.

Visto que os algoritmos de correspondência entre linhas podem ser relativamente complexos, optou-se por

¹Por exemplo, o comprimento e a orientação das linhas.

²Casos em que existem várias correspondências possíveis para a mesma linha.

tentar encontrar uma alternativa para obter correspondências entre as linhas detetadas nas imagens. Assim, foram criados os algoritmos de correspondência de linhas descritos em 5.1.1.

5.1.1 O Algoritmo de Correspondência

Considera-se que uma linha, l , numa imagem é descrita por

$$Au + Bv + C = 0.$$

Usando o modelo buraco de agulha, a projeção de l numa imagem é dada por (5.1) [HZ03], em que $[A, B, C]$ são os parâmetros da linha, $(u, v, 1)$ são as coordenadas de um píxel na imagem, $(X, Y, Z, 1)$ são as coordenadas de um ponto no mundo e P é a matriz de projeção associada à câmara com que se obteve a imagem.

$$\lambda \begin{bmatrix} A & B & C \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} A & B & C \end{bmatrix} P_{3 \times 4} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (5.1)$$

Considere-se apenas o membro direito de (5.1). A partir de

$$\begin{bmatrix} A & B & C \end{bmatrix} P_{3 \times 4}$$

é possível obter um vetor $w_{1 \times 4}$:

$$w = [a \ b \ c \ d]. \quad (5.2)$$

Assim, pode dizer-se que

$$w_{1 \times 4} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = [a \ b \ c \ d] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$$

define um plano Pl com parâmetros $[a, b, c, d]$. Este plano é formado pela linha 3D projetada na imagem e pelo ponto focal da câmara que obteve a imagem. Visto que são usadas duas imagens na reconstrução 3D, para cada par de linhas correspondentes existem dois planos, Pl_1 e Pl_2 , cuja interseção consiste na mesma linha 3D projetada nas imagens stereo (figura 5.2). Com algoritmo de correspondência criado, Algoritmo 1 (Algo1), é possível identificar quais as linhas detetadas nas imagens do par *stereo*, que permitem obter planos cuja interseção define a linha 3D projetada nas imagens. O fluxograma deste algoritmo pode ser consultado na figura 5.3. O Algoritmo 1 começa por obter um plano Pl_1 com recurso aos parâmetros de uma linha de uma imagem, e verifica se este é paralelo a algum plano Pl_2 obtido com recurso aos parâmetros de linhas da outra imagem do par *stereo*. Esta verificação faz-se com recurso ao produto interno entre estes planos, que tem o valor 1 se estes forem paralelos [Gel12, Wei15]. Nestas circunstâncias, não existe interseção destes planos e, consequentemente, não existe uma linha 3D que origine as linhas correspondentes projetadas nas imagens. Assim, pretende-se que o algoritmo detete e descarte estes casos.

O passo seguinte consiste em calcular o ângulo, α , formado entre o plano Pl_1 e os planos Pl_2 , obtidos com recurso aos parâmetros das linhas da outra imagem do par, que podem ser seus correspondentes e que não são paralelos a este. α é calculado com recurso à fórmula 5.3 [Gel12, Mey00, Hef06]. Nesta, N_1 e N_2 são as normais dos planos Pl_1 e Pl_2 , respectivamente.

$$\alpha = \arccos \left(\frac{N_1 \cdot N_2}{\|N_1\| \|N_2\|} \right) \quad (5.3)$$

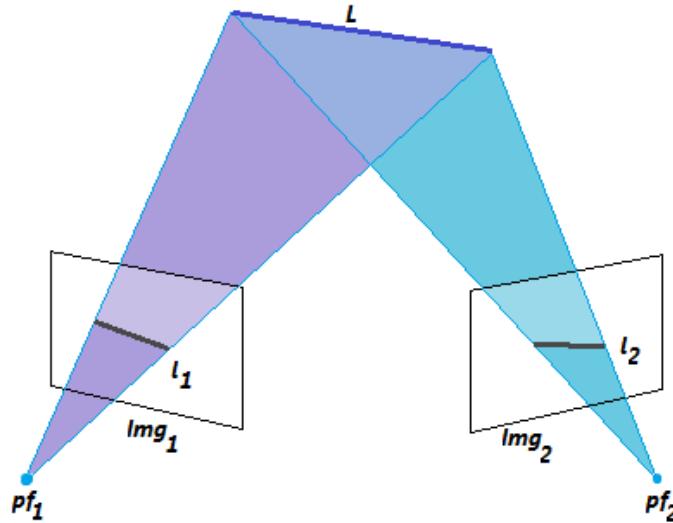


Figura 5.2: A linha 3D obtida a partir da interseção de dois planos.

Os ângulos obtidos no passo anterior são usados para calcular uma medida de semelhança (5.4) entre Pl_1 e os planos Pl_2 , obtidos a partir de linhas da outra imagem do par. Como os valores Q calculados são úteis no passo seguinte do algoritmo, estes são preservados.

$$Q = \cos(4\alpha) + 1 \quad (5.4)$$

O passo final consiste em identificar qual é a melhor correspondência de entre as possíveis. Visto que, neste caso, a melhor correspondência é aquela que tem uma medida de semelhança com o valor mais elevado, procura-se o valor do Q máximo calculado anteriormente. Para obter este valor usa-se a fórmula (5.5).

$$Q_{max} = \max_{i \in I} Q_i, \quad (5.5)$$

em que I é o conjunto das medidas de semelhança calculadas no passo anterior e Q_i é uma medida de semelhança deste conjunto.

Deste processo resulta uma matriz $W_{2 \times 4}$ (5.6), em que constam os parâmetros do par de planos cuja interseção dá origem à linha 3D projetada nas imagens.

$$W \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0 \quad (5.6)$$

Assim, consideram-se correspondentes as linhas que permitem obter os planos Pl_1 e Pl_2 cuja interseção é a linha 3D projetada nas imagens. Após ser encontrada a melhor correspondência, remove-se do grupo de linhas sem correspondente a linha cujos parâmetros permitem obter Pl_2 . Caso existam linhas na imagem que não tenham correspondente, estas são ignoradas. O processo repete-se até que todas as linhas detetadas na imagem tenham sido processadas.

Algoritmo de Correspondência Alternativo

Se a distância entre as câmaras for curta e o objeto a reconstruir se encontrar relativamente distante destas, pode ser dito que a orientação das linhas presentes nas imagens obtidas com estas câmaras é idêntica [ZN94]. Nestas condições, é possível usar uma variante do algoritmo de correspondência anteriormente descrito. Este algoritmo alternativo, Algoritmo 2 (Algo2), tem passos idênticos aos do algoritmo de correspondência apresentado anteriormente, mas a procura por correspondências é feita com recurso às linhas 2D presentes nas imagens stereo. O fluxograma do Algoritmo 2 pode ser consultada na figura 5.4.

Tal como acontece anteriormente, o primeiro passo consiste em obter uma linha de uma imagem do par *stereo* e calcular o ângulo, β , entre esta e as candidatas a correspondente presentes na outra imagem do mesmo par. Considerem-se duas linhas, l_1 e l_2 , descritas por:

$$A_1u + B_1v + C_1 = 0 \text{ e } A_2u + B_2v + C_2 = 0,$$

respectivamente. β é calculado com recurso à fórmula (5.7) [Lar12].

$$\beta = \arctan \left(\pm \frac{A_2B_1 - A_1B_2}{A_1A_2 + B_1B_2} \right) \quad (5.7)$$

O passo seguinte consiste em calcular a medida de semelhança entre a linha e as suas possíveis candidatas. Como nestas condições as linhas correspondentes vão ter uma inclinação idêntica, consideram-se apenas como possíveis correspondentes linhas que formam um ângulo β com valor inferior a $\frac{\pi}{4}$. Linhas que formem ângulos β com valores superiores $\frac{\pi}{4}$ são ignoradas pelo algoritmo. A medida de semelhança entre as linhas que cumpram este critério é calculada com recurso à fórmula (5.8), em que β é o ângulo que as linhas formam entre si.

$$Q = \cos(4\beta) + 1 \quad (5.8)$$

Tal como acontecia com o algoritmo original, os valores Q calculados são também guardados.

Finalmente, é usada a fórmula (5.5) para identificar qual a melhor correspondência de entre as possíveis. Tal como acontece com o Algoritmo 1, também no Algo2 se considera que a correspondência pretendida é aquela que tem a medida de semelhança com o valor mais elevado. Linhas que sejam identificadas como correspondente são removidas do grupo das candidatas a correspondente. São também descartadas linhas que não tenham correspondente quando a execução deste algoritmo termina.

Depois de concluída a procura por correspondências, para cada par de linhas correspondentes, obtém-se os vetores w (5.9) para cada linha do par. Em (5.9), $[A, B, C]$ são os parâmetros de uma das linhas do par, P é a matriz de projeção da câmara que obteve a imagem onde se encontra a linha e $[a, b, c, d]$ são os parâmetros do plano definido pela linha 3D projetada na imagem e pelo ponto focal da câmara que obteve a imagem.

$$w = [A \ B \ C] P_{3 \times 4} = [a \ b \ c \ d] \quad (5.9)$$

Assim, para cada par de linhas correspondentes obtém-se uma matriz $W_{2 \times 4}$ (5.10). As linhas da matriz W correspondem aos parâmetros dos dois planos cuja interseção dá origem à linha 3D projetada nas imagens do par *stereo*.

$$W = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \end{bmatrix} \quad (5.10)$$

5.2 Reconstrução

O último passo deste trabalho é a reconstrução das linhas 3D. Para tal, usam-se as correspondências obtidas com um dos algoritmos de correspondência apresentado em 5.1.1. Para fazer a reconstrução 3D das linhas, foi criado um algoritmo. Este permite obter os pontos da linha 3D a reconstruir, tendo por base os pares de linhas correspondentes obtidos com recurso a qualquer um dos algoritmos de correspondência. O fluxograma deste algoritmo pode ser consultado na figura 5.5.

O primeiro passo do algoritmo de reconstrução consiste em obter dois pontos arbitrários, p_1 e p_2 , da linha 3D a reconstruir. Considere-se a equação (5.11), que representa a interseção de dois planos obtidos com recurso aos parâmetros de um par de linhas correspondentes.

$$\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0_{2 \times 1} \quad (5.11)$$

Reorganizando (5.11) obtém-se o sistema de equações lineares (5.12). Neste, (X, Y, Z) são as coordenadas de um ponto 3D na linha, A é a matriz que tem os parâmetros $[a, b, c]$ dos planos que se intersetam para formar a linha 3D projetada nas imagens e b é o vetor linha com os parâmetros d destes planos.

$$Ax = b \Leftrightarrow \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = - \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \quad (5.12)$$

Para obter as coordenadas de dois pontos da linha 3D, pode atribuir-se um valor arbitrário a uma das coordenadas destes pontos e estimar as restantes com recurso ao método dos mínimos quadrados.

No caso das linhas horizontais paralelas ao eixo dos x , pode atribuir-se à coordenada X do ponto um valor arbitrário, p_x , e estimar as restantes coordenadas deste ponto 3D. Assim, obtém-se:

$$A_{2,3}x = bA_1p_x \Leftrightarrow \begin{bmatrix} b_1 & c_1 \\ b_2 & c_2 \end{bmatrix} \begin{bmatrix} Y \\ Z \end{bmatrix} = - \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} - \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} p_x. \quad (5.13)$$

Aplicando o método dos mínimos quadrados a (5.13), obtém-se um vetor com as coordenadas Y e Z de um ponto da linha 3D:

$$[p_y \ p_z]. \quad (5.14)$$

Adicionando a este vetor a coordenada X , que foi fixada anteriormente, obtém-se as coordenadas do ponto 3D:

$$[p_x \ p_y \ p_z]. \quad (5.15)$$

Caso se pretenda obter um ponto 3D arbitrário pertencente a uma linha horizontal paralela ao eixo dos y , pode fixar-se o valor da coordenada Y do ponto e estimar as restantes coordenadas usando o método dos mínimos quadrados. Para os casos das linhas verticais e diagonais, usa-se um método idêntico para estimar dois pontos das mesmas. Neste caso, a coordenada Z é aquela que tem um valor fixo.

Depois de serem obtidos dois pontos da linha a reconstruir, o algoritmo obtém os restantes pontos da linha 3D. Estes são obtidos com recurso a:

$$p = p_1 + \gamma(p_2 - p_1), \quad (5.16)$$

em que p é o ponto obtido, p_1 e p_2 são os dois pontos da reta 3D obtidos inicialmente e γ é um escalar que corresponde ao número de pontos da linha a obter. Se γ tiver um valor elevado, obtém-se linhas mais longas. Caso contrário, as linhas obtidas são mais curtas. Quando forem obtidos os pontos de todas as linhas, o algoritmo desenha as mesmas num referencial 3D. Após desenhar estas linhas 3D, o algoritmo termina a sua execução. O modelo obtido com recurso a esta técnica é idêntico a um *wireframe*, em que consta a reconstrução das linhas detetadas nas imagens. A restante informação presente na imagem é descartada.

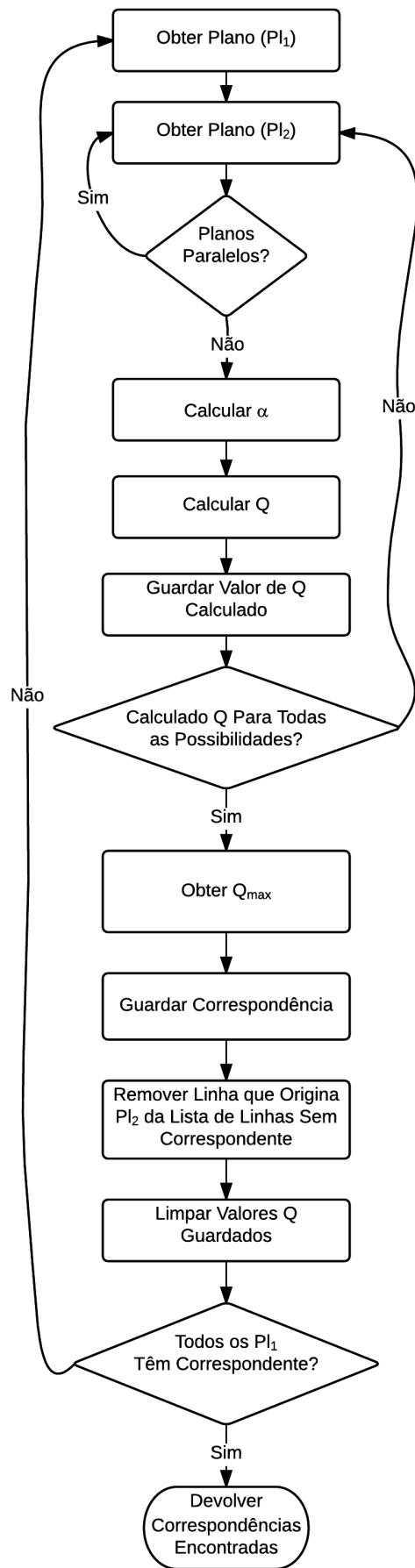


Figura 5.3: Fluxograma do Algoritmo 1, criado para encontrar correspondências entre linhas.

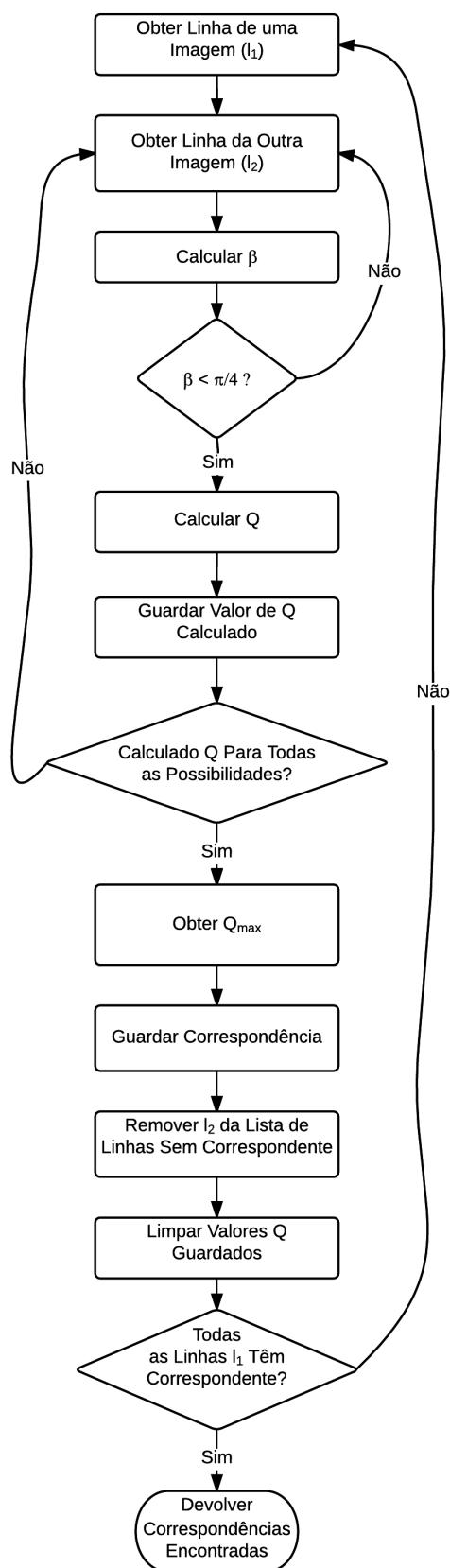


Figura 5.4: Fluxograma do Algoritmo 2, que permite encontrar correspondências entre linhas.

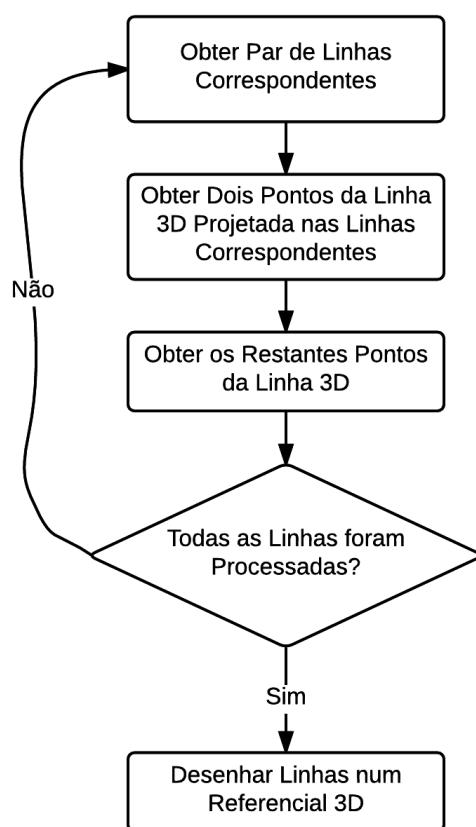


Figura 5.5: Fluxograma do algoritmo de reconstrução.

6

Caso de Estudo

Data is a precious thing and will last longer than the systems themselves.

Tim Berners-Lee

O caso de estudo consiste na reconstrução 3D de um cubo idêntico ao da figura 6.2. Este objeto foi escolhido devido às suas dimensões e características bem definidas. Neste capítulo são apresentados todos os passos usados para fazer a reconstrução 3D. É também apresentado o *hardware* usado e os resultados da reconstrução 3D com recurso aos algoritmos mencionados no Capítulo 5. Os algoritmos usados na reconstrução 3D e na deteção de linhas foram implementados propositadamente para este trabalho. Apenas são usadas bibliotecas de Visão Computacional no pré-processamento das imagens *stereo*, que ocorre antes da deteção de linhas. Neste caso, a biblioteca usada é a *OpenCV*.

6.1 Hardware Usado na Reconstrução

Para capturar as imagens *stereo* necessárias para fazer a reconstrução 3D do objeto, são usadas duas *webcams VGA Trust Exis* [Int15b] idênticas à apresentada na figura 6.3. Cada imagem obtida com recurso a estas câmaras

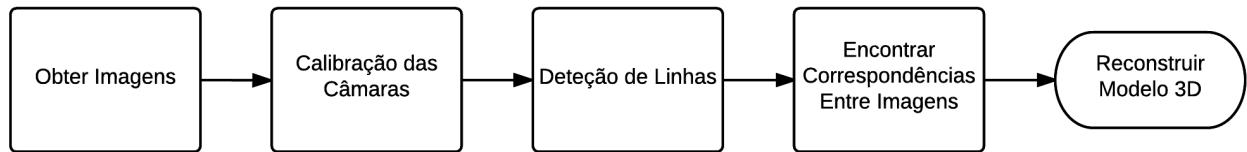


Figura 6.1: Passos usados desde a obtenção das imagens *stereo* até à reconstrução 3D.



Figura 6.2: O objeto a ser reconstruído.

tem dimensões 640 x 480 pixels. Para automatizar o processo de captura das imagens *stereo* necessárias, é usado um *script* Python¹. Este permite capturar, em simultâneo, várias imagens de cada câmara. O computador² em que é feita a reconstrução 3D está ligado às *webcams* via USB. A disposição do *hardware* a apresentada na figura 6.4.



Figura 6.3: Webcam idêntica às usadas para capturar as imagens *stereo* [Int15a].

¹O *script* pode ser consultado em: <https://github.com/moliveira7/ComputerVision>

²Intel i7 @ 2 GHz com 4 GB de RAM.

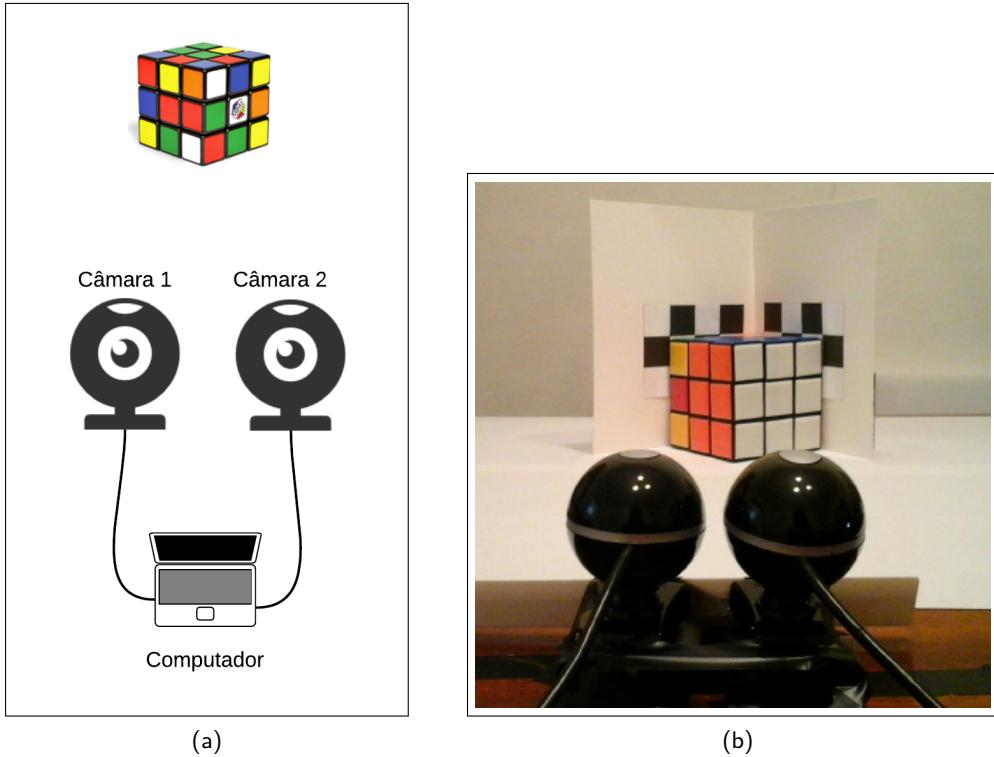


Figura 6.4: Diagrama e disposição real do *hardware* usado na reconstrução 3D.

6.2 Calibração das Câmaras

Após serem obtidas as imagens *stereo* é necessário calibrar as câmeras usadas. Neste processo usa-se um objeto de calibração que, neste caso, consiste em dois planos ortogonais. O método usado para calibrar as câmeras é o descrito em 3.3.3, no qual a calibração consiste apenas em estimar a matriz de projeção P . Como estamos a trabalhar com duas câmeras, é necessário estimar duas matrizes de projeção (uma para cada câmera).

Com recurso às *webcams*, obtêm-se as imagens do objeto de calibração (figura 6.5). A partir destas imagens são então extraídos vários pontos de referência³ para cada câmera, aos quais é associado um sistema de coordenadas tridimensionais. Como foi referido em 3.3.3, são necessários pelo menos seis pontos de referência para que possa ser estimada a matriz de projeção. Neste caso, foram obtidos quarenta pontos de referência para estimar a matriz de projeção correspondente a cada câmera. Os pontos de referência extraídos da figura 6.5-(a) e da figura 6.5-(b) podem ser consultados na tabela 6.1.

Como foi mencionado em 3.3.3, para cada um dos pontos de referência extraídos existem equações lineares na forma:

$$\begin{aligned} p_{11}X + p_{12}Y + p_{13}Z + p_{14} &= p_{31}Xu + p_{32}Yu + p_{33}Zu + p_{34}u, \\ p_{21}X + p_{22}Y + p_{23}Z + p_{24} &= p_{31}Xv + p_{32}Yv + p_{33}Zv + p_{34}v, \end{aligned}$$

nas quais os coeficientes p_{ij} correspondem aos elementos da matriz P , (u, v) são as coordenadas dos pontos de referência na imagem e (X, Y, Z) são as coordenadas dos pontos 3D associados aos pontos de referência.

³Os pontos de referência são extraídos com recurso a um script Python, disponível em: <https://github.com/moliveira7/ComputerVision>

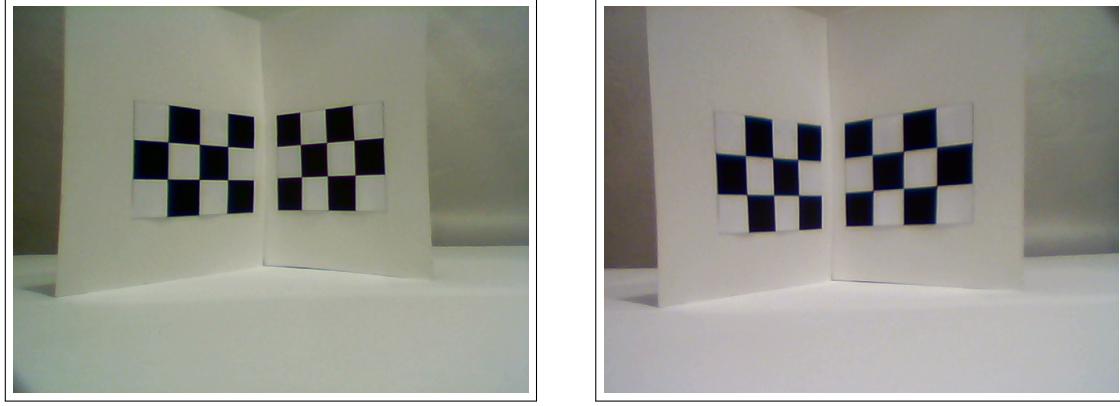


Figura 6.5: Imagens usadas na calibração das câmaras.

Escrevendo estas equações na forma matricial, obtém-se duas matrizes $A_{80 \times 11}$, A_1 e A_2 , idênticas a (6.1).

$$A = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -X_1 u_1 & -Y_1 u_1 & -Z_1 u_1 \\ \vdots & & & & & & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -X_n u_n & -Y_n u_n & -Z_n u_n \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -X_1 v_1 & -Y_1 v_1 & -Z_1 v_1 \\ \vdots & & & & & & & & & & \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -X_n v_n & -Y_n v_n & -Z_n v_n \end{bmatrix}. \quad (6.1)$$

Nestas, (u, v) são as coordenadas dos pontos de referência na imagem e (X, Y, Z) são os pontos do sistema de coordenadas tridimensional atribuído aos pontos de referência.

Visto que é usada a técnica de calibração detalhada no Capítulo 3, a matriz de projeção (P) é estimada com recurso ao método dos mínimos quadrados. Assim, P é dada por

$$x = [(A^T A)^{-1} A^T] b. \quad (6.2)$$

Em (6.2), x corresponde ao vetor com os elementos de P , A é a matriz das equações lineares e b corresponde ao vetor que contém as coordenadas dos pontos de referência extraídos da imagem de calibração. Usando este método, obtém-se as matrizes de projeção (6.3), correspondente à Câmara 1, e (6.4), que corresponde à Câmara 2.

$$P_1 = \begin{bmatrix} -1.6315891344 & 52.1855729180 & -30.7174567049 & 151.8300254322 \\ 46.2431098562 & 11.0362615997 & -22.1814096541 & 122.0839660294 \\ -0.0059571128 & 0.0513731131 & -0.0944570545 & 1.0000000000 \end{bmatrix} \quad (6.3)$$

$$P_2 = \begin{bmatrix} 3.7580119283 & 42.0676722508 & -15.0652354438 & 138.4298191021 \\ 51.6680134893 & 7.6748147089 & -21.3025766197 & 136.7416142351 \\ 0.0107363649 & 0.0328250030 & -0.0757477943 & 1.0000000000 \end{bmatrix} \quad (6.4)$$

6.3 Deteção de Linhas

Após serem obtidas as imagens *stereo* do objeto que se pretende reconstruir (figura 6.6), é possível avançar para a deteção de linhas nestas imagens. Para esta tarefa pode ser escolhido um dos algoritmos de deteção

Tabela 6.1: Pontos de referência extraídos das imagens de calibração.

Pontos de Referência		
Ponto 3D	Ponto na Imagen da Câmara 1	Ponto na Imagen da Câmara 2
(0,0,0)	(151,119)	(138,133)
(0,1,0)	(194,125)	(176,137)
(0,2,0)	(233,130)	(210,143)
(0,3,0)	(268,136)	(242,147)
(0,4,0)	(300,140)	(270,151)
(1,0,0)	(151,169)	(141,185)
(1,1,0)	(194,173)	(177,189)
(1,2,0)	(233,176)	(210,191)
(1,3,0)	(268,179)	(242,194)
(1,4,0)	(299,180)	(270,195)
(2,0,0)	(151,217)	(142,235)
(2,1,0)	(193,218)	(179,237)
(2,2,0)	(232,218)	(213,237)
(2,3,0)	(267,218)	(243,239)
(2,4,0)	(299,217)	(270,238)
(3,0,0)	(149,265)	(143,281)
(3,1,0)	(192,261)	(181,281)
(3,2,0)	(232,260)	(214,277)
(3,3,0)	(266,258)	(243,278)
(3,4,0)	(298,257)	(271,277)
(0,5,0)	(328,139)	(301,148)
(0,6,1)	(357,135)	(334,141)
(0,7,2)	(389,130)	(371,136)
(0,8,3)	(422,127)	(415,129)
(0,9,4)	(459,123)	(459,122)
(1,5,0)	(329,176)	(301,192)
(1,6,1)	(358,177)	(336,187)
(1,7,2)	(390,173)	(374,183)
(1,8,3)	(423,170)	(415,178)
(1,9,4)	(461,166)	(459,172)
(2,5,0)	(329,216)	(301,234)
(2,6,1)	(359,215)	(336,231)
(2,7,2)	(391,214)	(374,229)
(2,8,3)	(426,213)	(415,226)
(2,9,4)	(462,211)	(458,222)
(3,5,0)	(329,255)	(302,273)
(3,6,1)	(359,254)	(335,273)
(3,7,2)	(392,254)	(374,272)
(3,8,3)	(426,254)	(414,271)
(3,9,4)	(463,254)	(456,269)

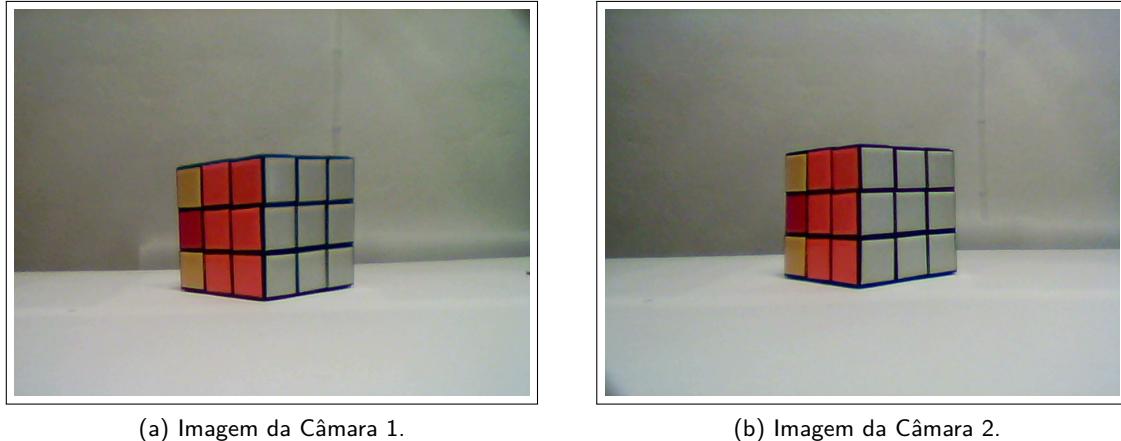


Figura 6.6: Imagens stereo do cubo a ser reconstruído.

de linhas mencionado no Capítulo 4. Porém, como os resultados da deteção de linhas⁴ com a Transformada de Hough Modificada ainda deixam um pouco a desejar, usa-se a Transformada de Hough descrita em 4.1 para detetar as linhas presentes nas imagens.

Como foi referido no Capítulo 4, antes de se proceder à deteção de linhas, as imagens passam por um passo de pré-processamento que tem várias fases. A primeira fase consiste na remoção do fundo das imagens. Na fase seguinte convertem-se as imagens sem fundo em imagens em tons de cinza. Para reduzir a quantidade de informação a processar na deteção de linhas, as imagens passam por uma terceira fase de pré-processamento. Nesta é usado o *Canny Edge Detector*, que permite obter uma imagem em que constam apenas os contornos dos elementos presentes na imagem original. A figura 6.7 ilustra cada uma destas fases de pré-processamento aplicadas à imagem da Câmara 1. Os mesmos passos são usados na fase de pré-processamento da imagem obtida com recurso à Câmara 2.

As imagens resultantes do pré-processamento são então usadas como *input* da Transformada de Hough. Como esta usa um parâmetro (*threshold*) para determinar o número mínimo de votos a partir do qual é detetada uma linha, é necessário escolher um valor adequado para este, de modo a que a maioria das linhas sejam detetadas. Neste caso, o *threshold* usado tem o valor **70** para ambas as imagens em que se pretendem detetar linhas. Os resultados da deteção de linhas podem ser observados na figura 6.8. No total, a Transformada de Hough detetou **34** linhas na imagem da Câmara 1 e **32** linhas na imagem da Câmara 2.

Como foi referido no Capítulo 4, as linhas detetadas pela Transformada de Hough são descritas por (6.5).

$$\rho = u \cos(\theta) + v \sin(\theta) \quad (6.5)$$

O algoritmo de reconstrução usado neste trabalho descreve as linhas a reconstruir na sua forma *standard*. Uma linha na forma *standard* é descrita por:

$$Au + Bv + C = 0. \quad (6.6)$$

Em (6.6), $[A, B, C]$ são parâmetros da reta e (u, v) são as coordenadas dos pontos na imagem. Dada esta diferença na forma de descrever as linhas detetadas, é necessário fazer uma conversão. Esta permite que as retas detetadas pela Transformada de Hough passem a ser descritas na sua forma *standard*. Assim, para representar a reta na forma $Au + Bv + C = 0$, é necessário reorganizar a equação (6.5), obtendo-se:

$$u \cos(\theta) + v \sin(\theta) - \rho = 0. \quad (6.7)$$

De (6.7) obtém-se os parâmetros:

⁴Apresentados no Capítulo 4 e, em detalhe, no Anexo A.

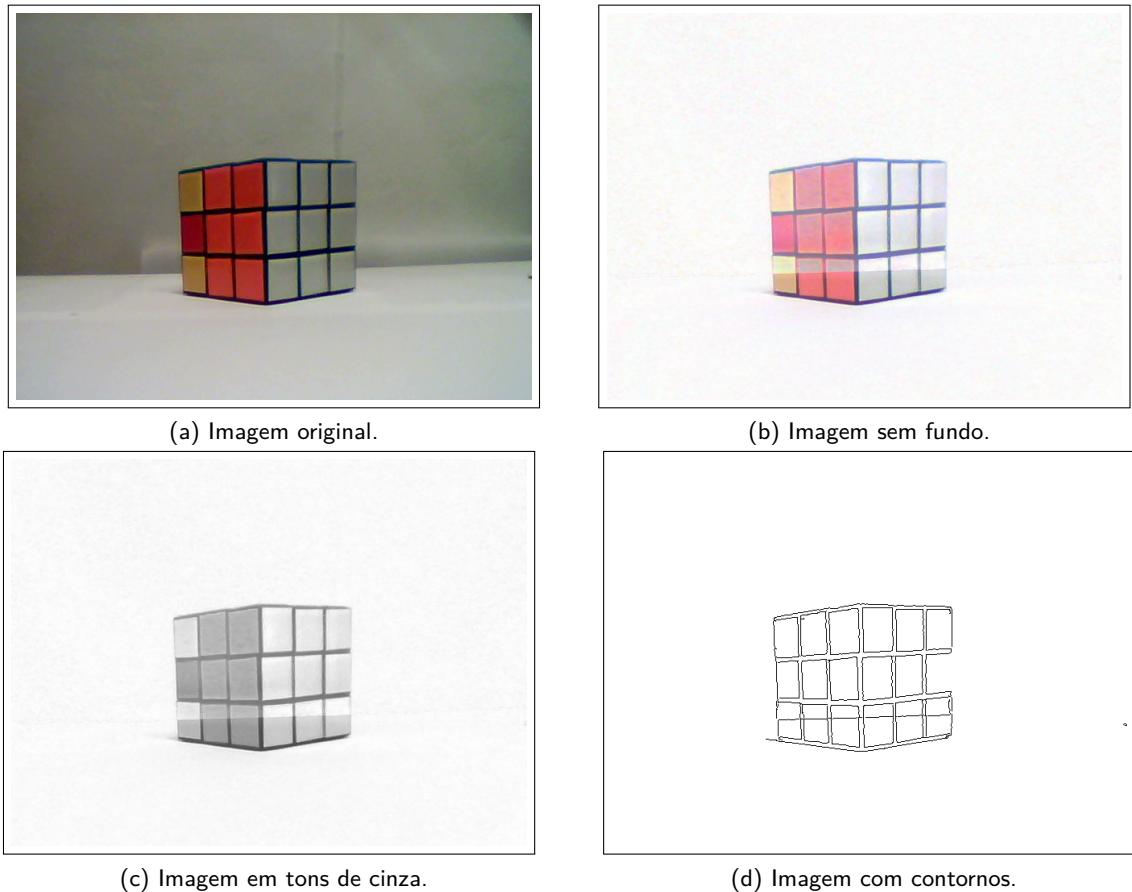


Figura 6.7: Resultados da aplicação de cada um dos passos de pré-processamento à imagem da obtida com a Câmera 1.

- $A = \cos(\theta)$;
- $B = \sin(\theta)$;
- $C = -\rho$.

Os parâmetros das linhas detetadas pela Transformada de Hough podem ser consultados em detalhe no Anexo B.

6.4 Reconstrução 3D

A reconstrução 3D é a fase final do processo. Como foi referido no Capítulo 5, esta é composta por duas partes: encontrar correspondências entre as linhas detetadas nas imagens e reconstrução das linhas 3D.

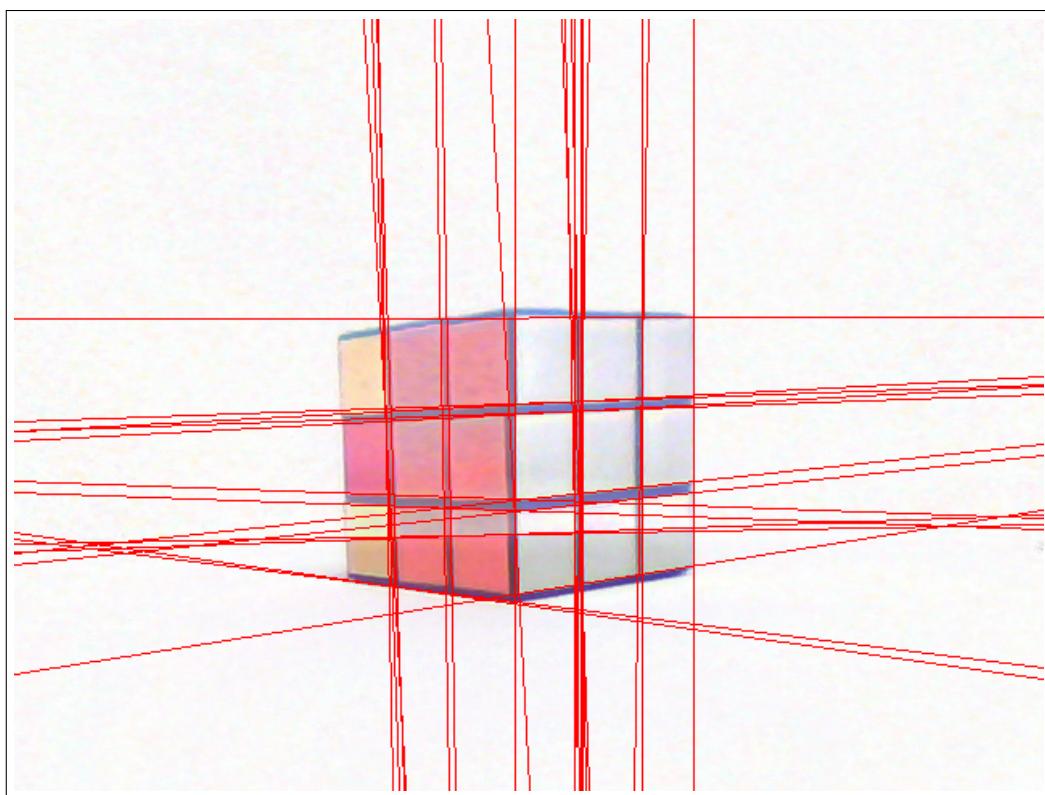
Antes de se iniciar a reconstrução 3D do objeto, é feito o pré-processamento referido no Capítulo 5, que permite distinguir os vários tipos de linhas detetadas (linhas verticais, horizontais e diagonais). Para fazer esta distinção entre os vários tipos de linhas, usa-se um intervalo de possíveis valores do parâmetro θ que caracteriza cada linha. Assim, definem-se como linhas horizontais aquelas que têm θ com valores num intervalo entre zero e 0.175 radianos. Linhas verticais são aquelas que têm um θ com valor entre 1.396 e 1.745 radianos. Consideram-se linhas diagonais aquelas que têm um θ com um valor que exclua a linha dos grupos de linhas horizontais e verticais.

Como foi referido em 6.3, foram detetadas 32 linhas na figura 6.6-(a) e 34 linhas na figura 6.6-(b). Estes conjuntos de linhas detetadas foram submetidos ao passo de pré-processamento aqui referido. Deste resultaram seis grupos de linhas mais pequenos, três para cada imagem, nos quais se encontram separadas as linhas detetadas em cada imagem de acordo com o seu tipo. Com este pré-processamento, foi possível identificar 14 linhas verticais, 8 linhas horizontais e 10 linhas diagonais de entre as 32 linhas detetadas na figura 6.6-(a). De um modo análogo, identificaram-se 17 linhas verticais, 11 linhas horizontais e 6 linhas diagonais na figura 6.6-(b). Os grupos de linhas resultantes deste pré-processamento podem ser consultados no Anexo C.

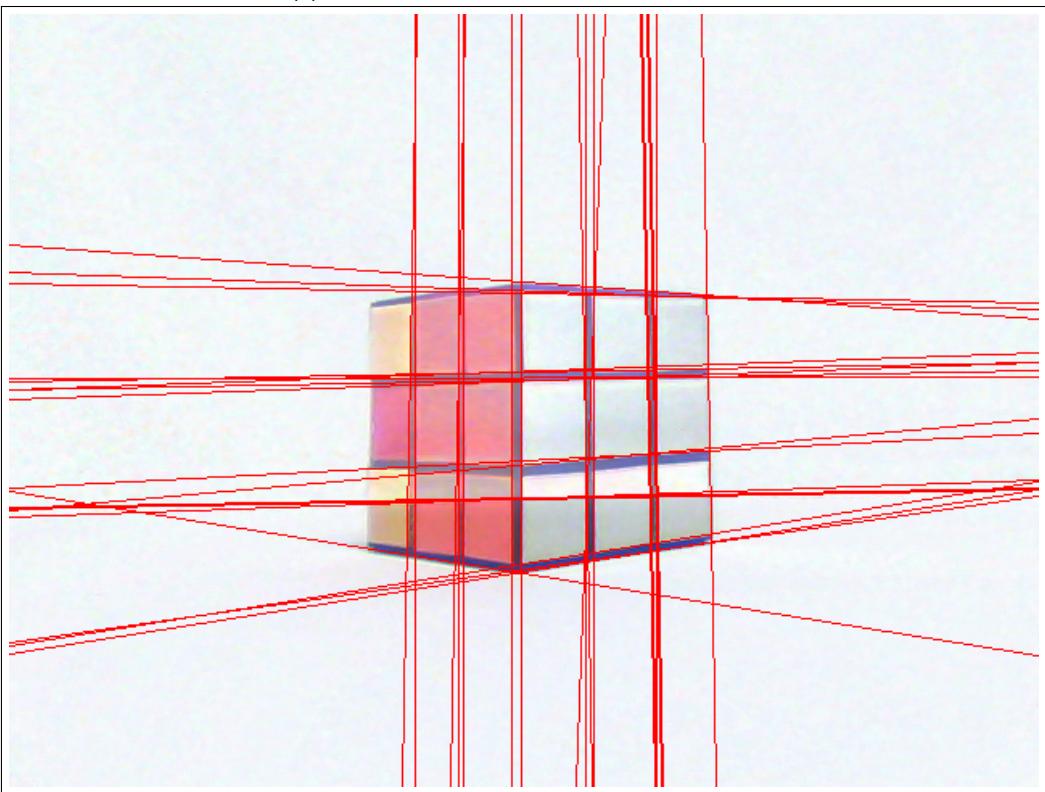
O passo seguinte consiste em encontrar correspondências entre as linhas detetadas em cada imagem do par stereo. Neste passo foram usados os dois algoritmos descritos no Capítulo 5. Ambos os algoritmos encontraram 14 pares de linhas verticais correspondentes, 8 pares de linhas horizontais correspondentes e 6 pares de linhas diagonais correspondentes. Como linhas detetadas sem correspondente são ignoradas pelos algoritmos de correspondência, descartam-se, em ambos os casos, 3 linhas verticais e 3 linhas horizontais detetadas na figura 6.8-(b) e 4 linhas diagonais detetadas na figura 6.8-(a). As correspondências encontradas por estes algoritmos podem ser consultadas no Anexo D.

Finalmente, é possível avançar para a reconstrução. O método usado para reconstruir as linhas detetadas é o descrito no Capítulo 5. É importante ter em mente que o modelo reconstruído depende do que a câmara consegue observar. No caso do cubo, reconstroem-se as linhas que foram detetadas pela Transformada de Hough nas faces deste que são visíveis nas imagens captadas pelas câmaras. Como foram usados dois algoritmos distintos para encontrar correspondências, os resultados da reconstrução com cada um destes algoritmos diferem. O modelo resultante da reconstrução em que se usa o Algoritmo 1 podem ser observados na figura 6.9. Comparando a disposição das linhas reconstruídas (figura 6.9) com a disposição das linhas detetadas nas imagens (figura 6.8), verifica-se que este modelo não é muito fiável. Como pode ser observado na figura 6.9, as linhas verticais são reconstruídas com uma inclinação incorreta, pelo que se assemelham a linhas diagonais. No caso das linhas horizontais, a grande maioria das mesmas encontra-se agrupada numa única zona do modelo, existindo apenas duas destas que são reconstruídas em zonas diferentes do mesmo. As linhas diagonais são, neste caso, reconstruídas como se fossem linhas horizontais e encontram-se sobrepostas às linhas horizontais.

Pode ser consultado na figura 6.10 o modelo resultante da reconstrução em que se usa o Algoritmo 2 para encontrar correspondências entre as linhas detetadas nas imagens do par stereo. Comparando este modelo com



(a) Linhas detetadas na imagem da Câmara 1.



(b) Linhas detetadas na imagem da Câmara 2.

Figura 6.8: Linhas detetadas pela Transformada de Hough nas imagens stereo da figura 6.6.

o apresentado na figura 6.9, verifica-se que este é mais fiável, isto é, as linhas 3D reconstruídas assemelham-se mais às detetadas pela Transformada de Hough (figura 6.8). Como pode ser verificado na figura 6.10, as linhas verticais e horizontais estão dispostas de uma forma idêntica à das linhas detetadas e apresentam uma orientação correta. Como pode ser observado na figura 6.10-(c), a maioria das linhas diagonais é desenhada corretamente. Porém, como o método usado para desenhar estas linhas é idêntico ao usado para as linhas verticais, as linhas diagonais reconstruídas apresentam uma orientação diferente da das linhas diagonais detetadas nas imagens. Contudo, mesmo com a diferença na orientação das linhas diagonais reconstruídas, o modelo obtido tem diversas semelhanças com o cubo que se pretende reconstruir.

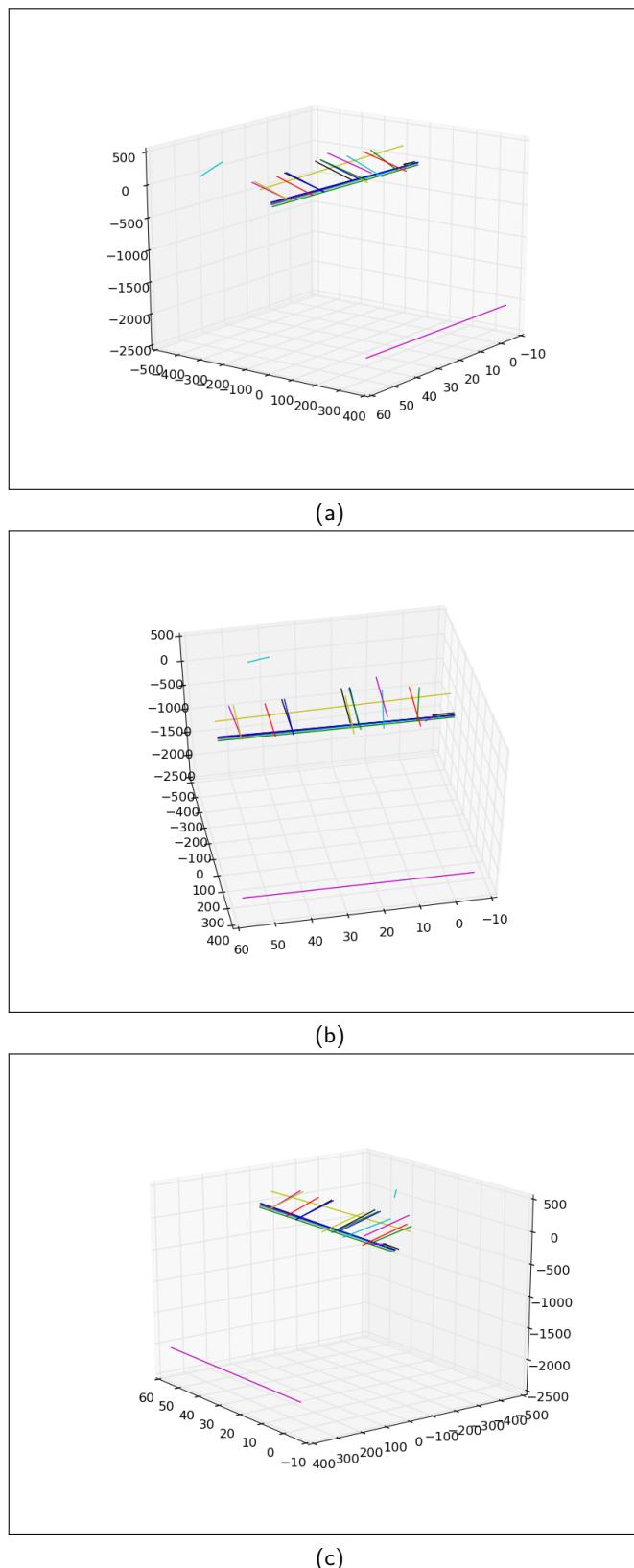


Figura 6.9: Modelo obtido após ser feita a reconstrução 3D, na qual é usado o Algoritmo 1 para encontrar correspondências entre linhas.

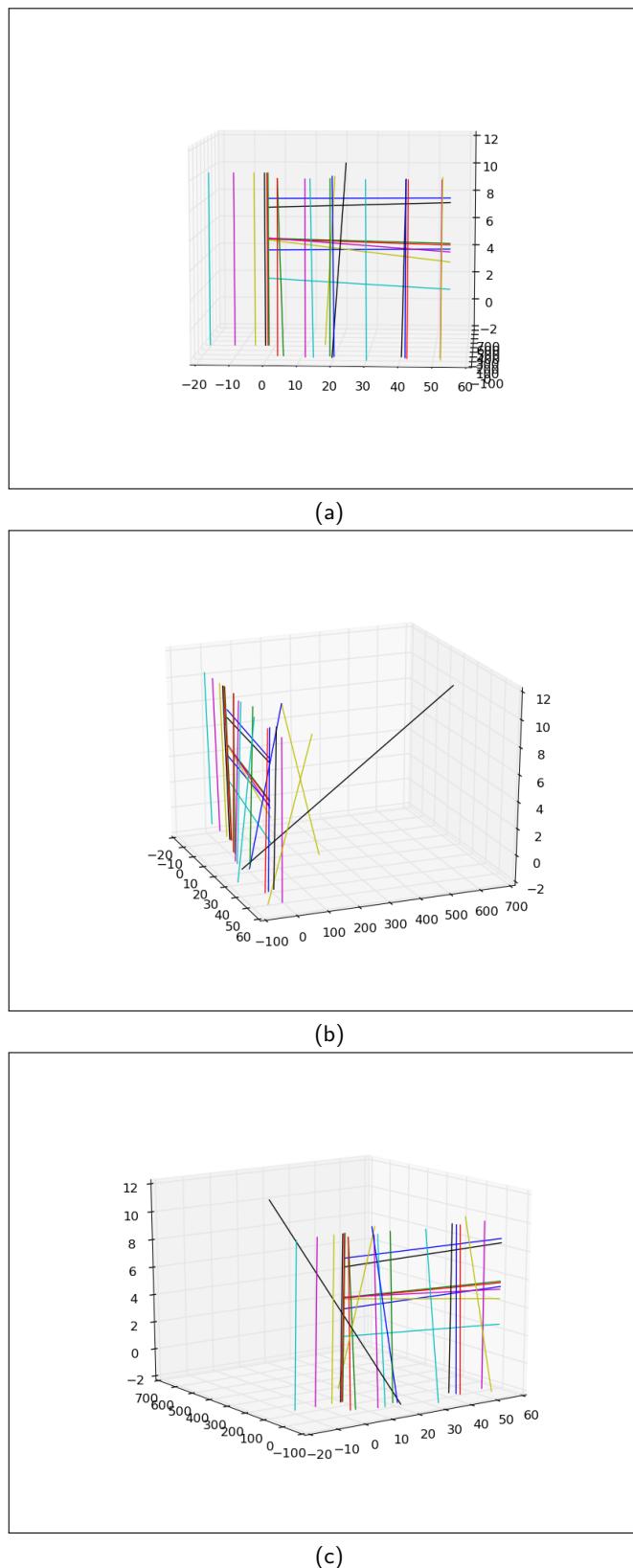


Figura 6.10: Modelo obtido após ser feita a reconstrução 3D, na qual é usado o Algoritmo 2 para encontrar correspondências entre linhas.

7

Conclusões e Trabalho Futuro

You cannot swim for new horizons until you have courage to lose sight of the shore.

William Faulkner

Neste trabalho é apresentado um algoritmo baseado na Transformada de Hough e um método que permite fazer reconstrução 3D a partir de imagens *stereo*. A fundamentação teórica dos algoritmos desenvolvidos e usados é apresentada nos Capítulos 4 e 5, respetivamente. Os resultados dos vários testes feitos a estes algoritmos também podem ser consultados neste trabalho. Além disto, é apresentado também o estado da arte relacionado com os diversos temas abordados neste trabalho (métodos de calibração da câmara, algoritmos de deteção de linhas e técnicas de reconstrução 3D).

O algoritmo de deteção de linhas, que resulta de uma modificação da Transformada de Hough, permite reduzir o número de falsos positivos (linhas detetadas incorretamente) obtidos. No caso específico da Transformada de Hough Modificada - Versão 3, verificaram-se resultados bastante positivos na deteção de linhas em imagens sintéticas, sendo corretamente detetada a maioria das linhas presentes nas imagens de teste. Os resultados dos testes deste algoritmo nos quais são usadas fotografias como *input* são também positivos. Porém, em alguns casos, linhas presentes nas imagens são ignoradas por este algoritmo ou o resultado da deteção de linhas é algo deslocado do real (por exemplo, existem casos em que a linha detetada tem uma inclinação diferente da verificada para a linha real). Em comparação com este algoritmo, as Versões 1 e 2 da Transformada de Hough

Modificada, também mencionadas neste trabalho, têm desempenhos algo inferiores no que respeita à deteção de linhas em imagens. Devido às limitações da Transformada de Hough Modificada - Versão 3, que ainda afetam o seu desempenho, foi escolhida a Transformada de Hough apresentada por Duda e Hart para fazer a deteção de linhas, que antecede a reconstrução 3D feita neste trabalho.

Antes de avançar para a reconstrução 3D, procede-se à calibração das câmaras usadas para obter as imagens *stereo*. Como foi mencionado nos Capítulo 1 e 6, as câmaras usadas são duas *webcams low-cost*. A calibração destas é feita com recurso ao método mencionado em 3.3.3, sendo obtidas as matrizes de projeção destas câmaras. Estas matrizes são estimadas com recurso ao método dos mínimos quadrados.

Os algoritmos desenvolvidos e usados para fazer a reconstrução 3D permitem reconstruir a parte visível de um objeto (um cubo). Neste caso, a reconstrução é feita em dois passos: encontrar correspondências entre as imagens *stereo* obtidas com as *webcams* e reconstrução do modelo 3D. Para encontrar correspondências, são usados os algoritmos de correspondência de linhas cuja fundamentação teórica é apresentada no Capítulo 5. Após serem obtidas as correspondências, é feita a reconstrução propriamente dita. Neste caso, o modelo resultante do processo de reconstrução 3D é idêntico a um *wireframe*, visto que apenas são reconstruídas as linhas detetadas nas imagens. Os resultados da reconstrução 3D feita neste trabalho podem ser consultados no Capítulo 6. Observando os dois modelos obtidos com a reconstrução, é possível verificar que, quando a procura de correspondências é feita com recurso às linhas 2D, o modelo reconstruído apresenta mais semelhanças com o objeto que se pretendia reconstruir.

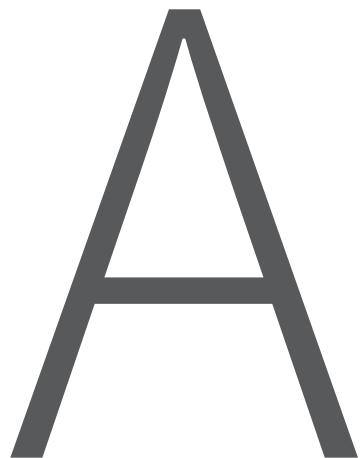
Tendo em conta todo o trabalho desenvolvido e os objetivos apresentados no Capítulo 1 desta dissertação, pode concluir-se que estes objetivos foram cumpridos. Deste modo, pode dizer-se que os principais contributos deste trabalho são a metodologia e os algoritmos desenvolvidos e usados para fazer a reconstrução 3D. A importância destes prende-se com o facto de ser possível fazer a reconstrução 3D com recurso a um par de imagens *stereo* usando as linhas presentes nestas, sendo a reconstrução feita de um modo alternativo ao que é normalmente usado. O algoritmo da Transformada de Hough Modificada, apresentado neste trabalho, é também um contributo relevante, visto que este representa uma alternativa aos algoritmos de deteção de linhas existentes.

7.1 Trabalho Futuro

O trabalho futuro inclui melhorar os algoritmos de deteção de linhas e os algoritmos de reconstrução apresentados neste trabalho. Estas melhorias incluem extensões da funcionalidade destes algoritmos e otimizações que permitam melhorar o desempenho dos mesmos.

No caso dos algoritmos de deteção de linhas, trabalho futuro consiste na correção de eventuais erros que não tenham sido detetados até ao momento e que afetem o desempenho destes algoritmos. No trabalho futuro, incluem-se também otimizações da Transformada de Hough Modificada, que permitam reduzir ainda mais os falsos positivos detetados pelos mesmos. No trabalho futuro relacionado com os algoritmos de deteção de linhas incluem-se também otimizações que permitam agilizar, de modo a que exista uma poupança de tempo e recursos computacionais, o processamento da informação necessária para a deteção de linhas.

No caso dos algoritmos usados para fazer a reconstrução 3D, o trabalho futuro consiste em optimizar estes algoritmos e corrigir eventuais problemas que existam nestes e que não tenham sido detetados até ao momento. Além disto, prevê-se também uma otimização destes de modo a que possa ser poupar tempo e recursos computacionais na execução dos mesmos. A extensão dos algoritmos usados na reconstrução 3D de modo a que estes possam reconstruir objetos mais complexos fica também reservada para trabalho futuro.



Transformada de Hough Modificada: Resultados dos Testes Efetuados

Neste anexo são disponibilizados os resultados dos testes efetuados com as diversas versões da Transformada de Hough Modificada (apresentadas no Capítulo 4). Tal como referido nesse capítulo, foram usados dois tipos de imagens: imagens controladas e fotografias. Nas imagens controladas apenas constam linhas pretas sobre um fundo branco. Para os testes com fotografias, recolheram-se imagens com diversos ambientes retratados. Três destas fotografias (figuras A.6-(a), A.6-(d) e A.7-(a)), usadas nos testes às diferentes versões da Transformada de Hough Modificada, pertencem ao *Middlebury Stereo Dataset* de 2006 [HS07], encontrando-se as mesmas devidamente referenciadas.

A.1 Testes com Imagens Controladas

Os testes em que se usam imagens controladas como *input* têm como propósito identificar situações em que os algoritmos detetam linhas incorretamente ou a deteção de linhas falha (não se detetam quaisquer linhas). Este conjunto de testes serve como um grupo de resultados de controlo. Isto é especialmente útil para avaliar o impacto de alterações nas implementações das várias Versões da Transformada de Hough Modificada. Além disto, estes resultados permitem identificar situações em que os algoritmos estão a ter um desempenho abaixo do esperado.

A.1.1 Transformada de Hough Modificada - Versão 1

Como pode ser observado na figura A.1, a Versão 1 da Transformada de Hough Modificada tem resultados pouco precisos. Em nenhuma das imagens de teste existem linhas detetadas sobrepostas às linhas reais. Porém, nas figuras A.1-(a), A.1-(g) e A.1-(h) existem linhas detetadas que têm a orientação correta, apesar de se encontrarem deslocadas das originais.

A.1.2 Transformada de Hough Modificada - Versão 2

Como foi referido no Capítulo 4, os testes da Versão 2 da Transformada de Hough Modificada contemplaram dois casos: elevar os valores do acumulador da Transformada a um expoente igual para ρ e θ e elevar estes valores a expoentes diferentes para as referidas coordenadas. Os resultados dos testes da THM-V2, nos quais se usaram imagens controladas, são aqui apresentados.

Na figura A.2 constam os resultados dos testes em que foi usado o mesmo expoente para as coordenadas ρ e θ no passo de *clustering* da Versão 2 da Transformada de Hough Modificada. Neste caso, o valor escolhido para ambos os expoentes foi 20. Como pode ser observado nesta figura, as linhas detetadas correspondem, na maioria dos casos, às linhas reais. Apenas se registam erros graves na figura A.2-(c), em que uma das linhas reais não é detetada. Na A.2-(d), uma das linhas detetadas tem uma orientação ligeiramente diferente da real, apesar desta se sobrepor em parte à linha real correta.

A Transformada de Hough Modificada - Versão 2 também foi submetida a testes em que o expoente usado é diferente para ρ e θ no passo de *clustering* do algoritmo. Neste caso, os expoentes usados foram 9 para ρ e 10 para θ . Os resultados destes testes são apresentados na figura A.3. Na maioria dos casos de teste, verifica-se que o algoritmo consegue detetar corretamente as linhas existentes nas imagens. Porém, na figura A.3-(c) existe uma linha que é ignorada pelo algoritmo. Na figura A.3-(d) ocorre outro tipo de problema: a linha ao centro é interpretada como se ali existissem duas linhas.

A.1.3 Transformada de Hough Modificada - Versão 3

As condições de teste da Transformada de Hough Modificada - Versão 3 são idênticas às da Versão 2. Também nos testes deste algoritmo foram contemplados os casos: usar expoentes iguais para ρ e θ no passo de *clustering* e usar expoentes diferentes para ρ e θ no mesmo passo deste algoritmo. Além disto, também é ajustado μ , o coeficiente usado para normalizar ρ , no cálculo da distância entre pontos e centroides para cada um dos casos de teste. No caso em que os expoentes são iguais, μ toma o valor 900 e, no caso em que os expoentes são diferentes, este toma o valor 100. As figuras A.4 e A.5 incluem os resultados dos testes da Transformada de Hough - Versão 3 em que são usadas imagens controladas como *input*.

Como pode ser verificado na figura A.4, na qual estão presentes os resultados dos testes do algoritmo em que se usam expoentes iguais, a maioria das linhas reais são corretamente detetadas. O teste da figura A.4-(c) é especialmente positivo, visto que as três linhas reais presentes na imagem nunca foram corretamente detetadas pelas outras versões da Transformada de Hough Modificada. Porém, na figura A.4-(g), uma das linhas reais é ignorada pelo algoritmo, apesar das restantes linhas reais da imagem serem corretamente detetadas. Também se verifica um resultado menos bom na figura A.4-(d), em que uma linha detetada tem orientação ligeiramente diferente da inclinação da linha que se pretende detetar.

Os resultados dos testes em que se usam expoentes diferentes estão presentes na figura A.5. Observando esta figura, é possível verificar que a maioria dos resultados dos testes são positivos. Verifica-se um declínio da qualidade dos resultados dos testes das figuras A.5-(d), A.5-(e) e A.5-(f). Na figura A.5-(d), além de ainda se observar uma linha detetada com uma inclinação ligeiramente diferente da original, verifica-se também que uma outra linha detetada está ligeiramente deslocada da linha que se pretende detetar. No teste da figura A.5-(e) uma das

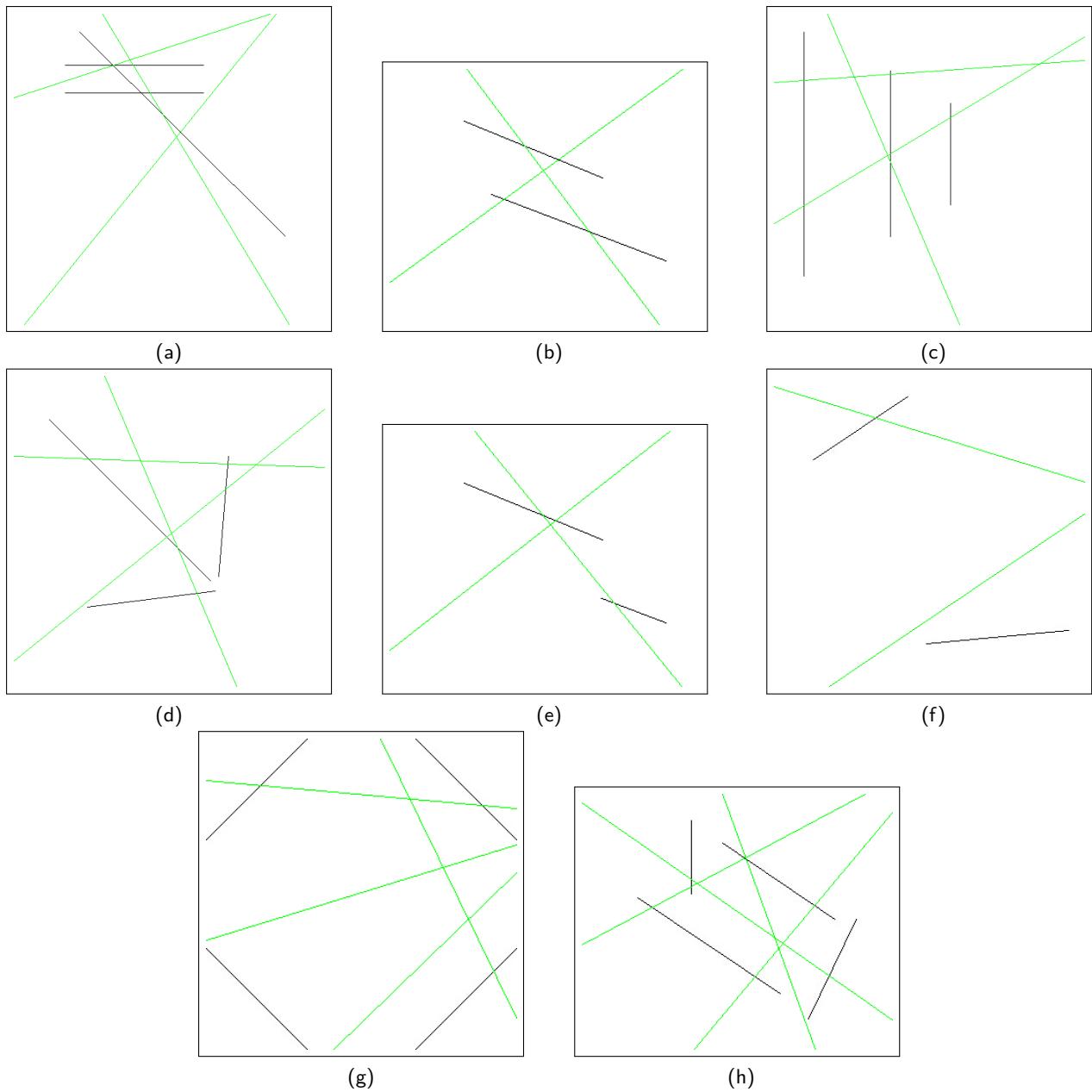


Figura A.1: Resultados dos testes da Transformada de Hough Modificada - Versão 1, em que são usadas imagens controladas.

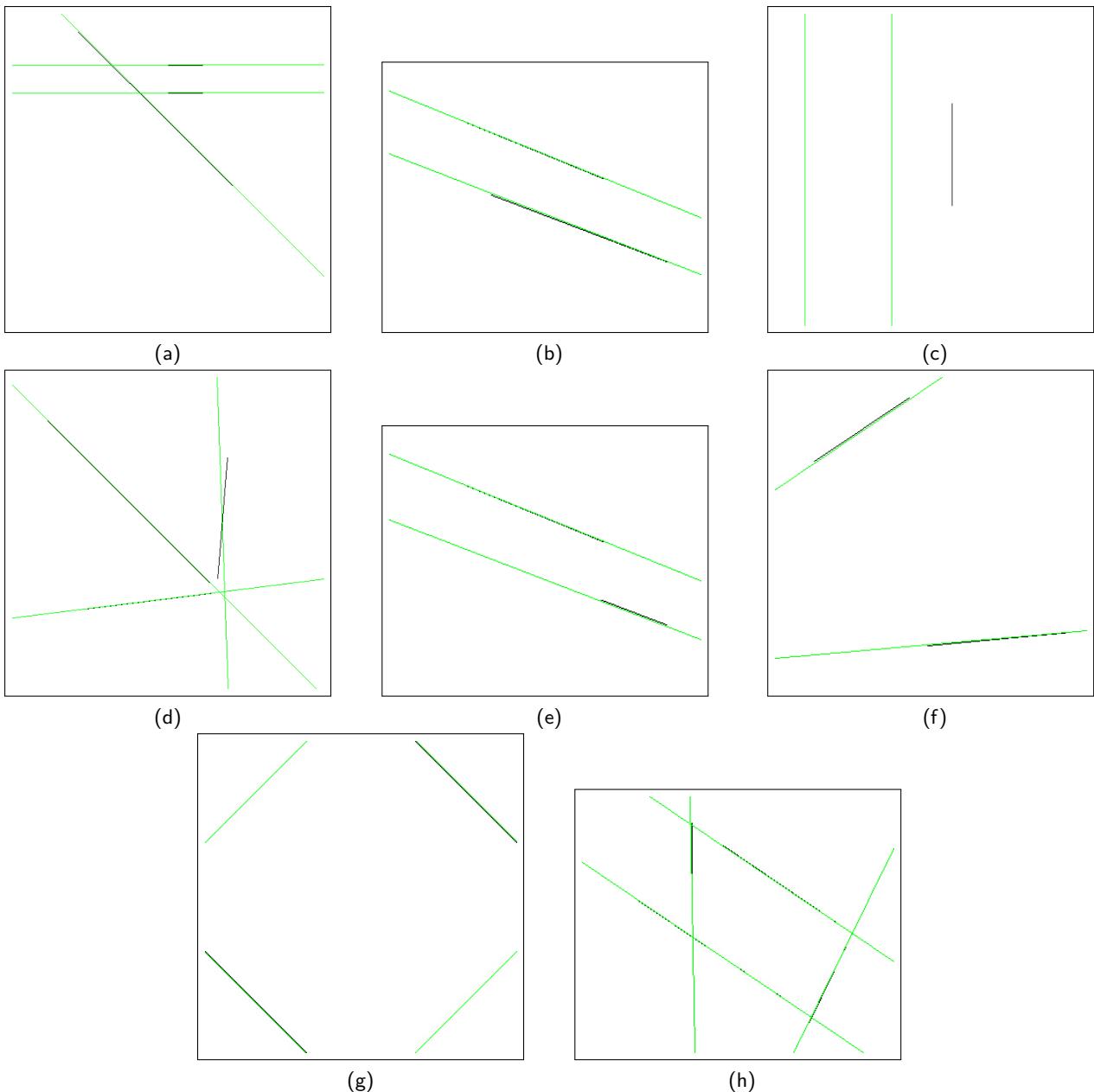


Figura A.2: Testes da Versão 2 da Transformada de Hough Modificada, nos quais se usam imagens controladas e o mesmo expoente para ρ e θ no passo de *clustering* do algoritmo.

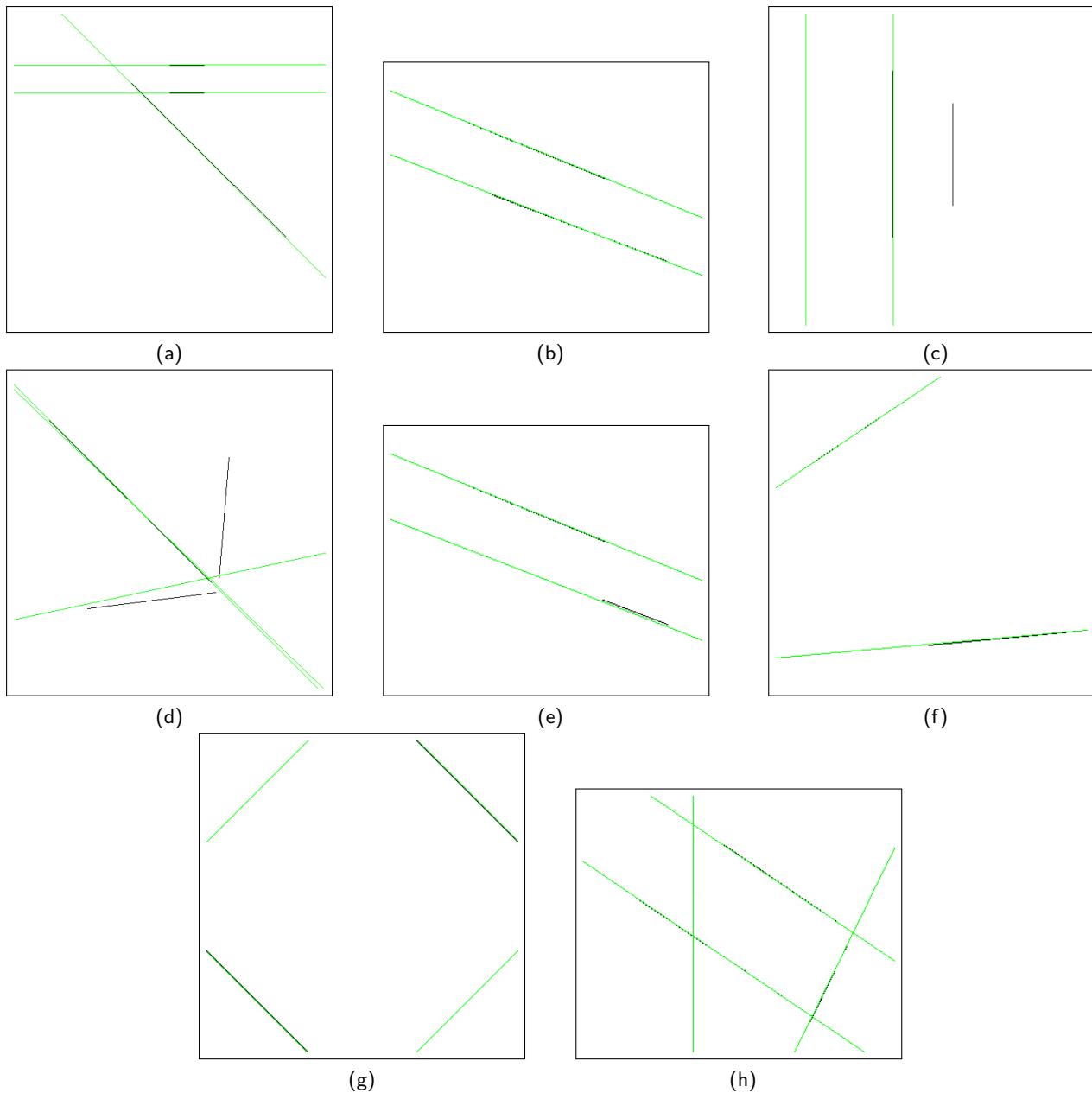


Figura A.3: Testes da Versão 2 da Transformada de Hough Modificada, em que são usadas imagens controladas e expoentes diferentes no passo de *clustering* do algoritmo.

linhas que se pretende detetar é ignorada pelo algoritmo, algo que não se verifica nos testes em que se usam expoentes iguais (figura A.4). Dos testes com imagens controladas, aquele em que se verifica pior desempenho, em termos de deteção de linhas, é o da figura A.5-(f). Neste teste ambas as linhas detetadas estão incorretas, visto que uma delas está bastante deslocada da linha que se pretende detetar e a outra tem uma inclinação completamente incorreta.

A.2 Testes com Fotografias

Para determinar se os algoritmos de deteção de linhas implementados funcionam em situações reais, foram feitos testes em que se usam fotografias como *input*. À semelhança do que aconteceu nos testes com imagens controladas, também neste caso se pretende identificar situações em que os algoritmos implementados falham. Ao contrário do que acontecia com as imagens anteriores, as fotografias foram obtidas em ambientes com condições de luminosidade variáveis.

A.2.1 Transformada de Hough Modificada - Versão 1

Tal como acontece nos testes deste algoritmo quando são usadas imagens controladas, também se verificam resultados pouco precisos quando é testada a Versão 1 da Transformada de Hough Modificada usando fotografias como *input*. Os resultados destes testes podem ser consultados nas figuras A.6 e A.7. Apesar dos resultados menos bons generalizados, nas figuras A.6-(a) e A.7-(a) existem linhas horizontais que têm a orientação correta. Porém, estas ainda se encontram deslocadas das linhas reais. Na figura A.6-(c), a linha horizontal tem a orientação correta e encontra-se muito próxima da linha real. Na figura A.7-(b), as linhas detetadas têm a configuração correta, mas apresentam uma orientação incorreta.

A.2.2 Transformada de Hough Modificada - Versão 2

As condições de teste da Transformada de Hough Modificada - Versão 2 são idênticas às aplicadas em A.1.2. Assim, são testados os casos em que se usam expoentes iguais e expoentes diferentes para elevar as coordenadas ρ e θ no passo de *clustering* do algoritmo.

Os resultados dos testes em que se usou o mesmo expoente podem ser consultados nas figuras A.8 e A.9. Neste caso de teste, os expoentes de ρ e θ têm o valor 20. Como pode ser observado, em diversos casos são obtidos resultados positivos. São exemplos disto as figuras A.8-(a), A.8-(b), A.8-(d), A.9-(a), A.9-(b) e A.9-(c). Nestas imagens, apesar de existirem algumas linhas detetadas com inclinações incorretas, a maioria das deteções está correta. Porém, ainda existem casos em que se verificam erros. São exemplos disto as figuras A.8-(c) e A.9-(d). Nestas figuras verifica-se o oposto: a maioria das linhas detetadas está incorreta, existindo poucas deteções corretas (ou muito próximas disso).

Nas figuras A.10 e A.11 podem ser analisados os resultados dos testes em que são usados expoentes diferentes no passo de *clustering* da THM-V1. Neste caso, o expoente de ρ tem o valor 9 e o expoente de θ tem o valor 10. Tal como se verifica para o caso em que são usados expoentes iguais, também para o presente caso de teste se verificam resultados positivos. As figuras A.10-(a), A.10-(b), A.10-(d), A.11-(a) e A.11-(b) são exemplos em que muitas das linhas detetadas correspondem a linhas reais. Nestas figuras existem também algumas deteções que têm a inclinação errada, apesar destas se encontrarem próximas das linhas reais. As figuras A.10-(c), A.11-(c) e A.11-(d) são exemplos de casos em que existe uma maioria deteções incorretas.

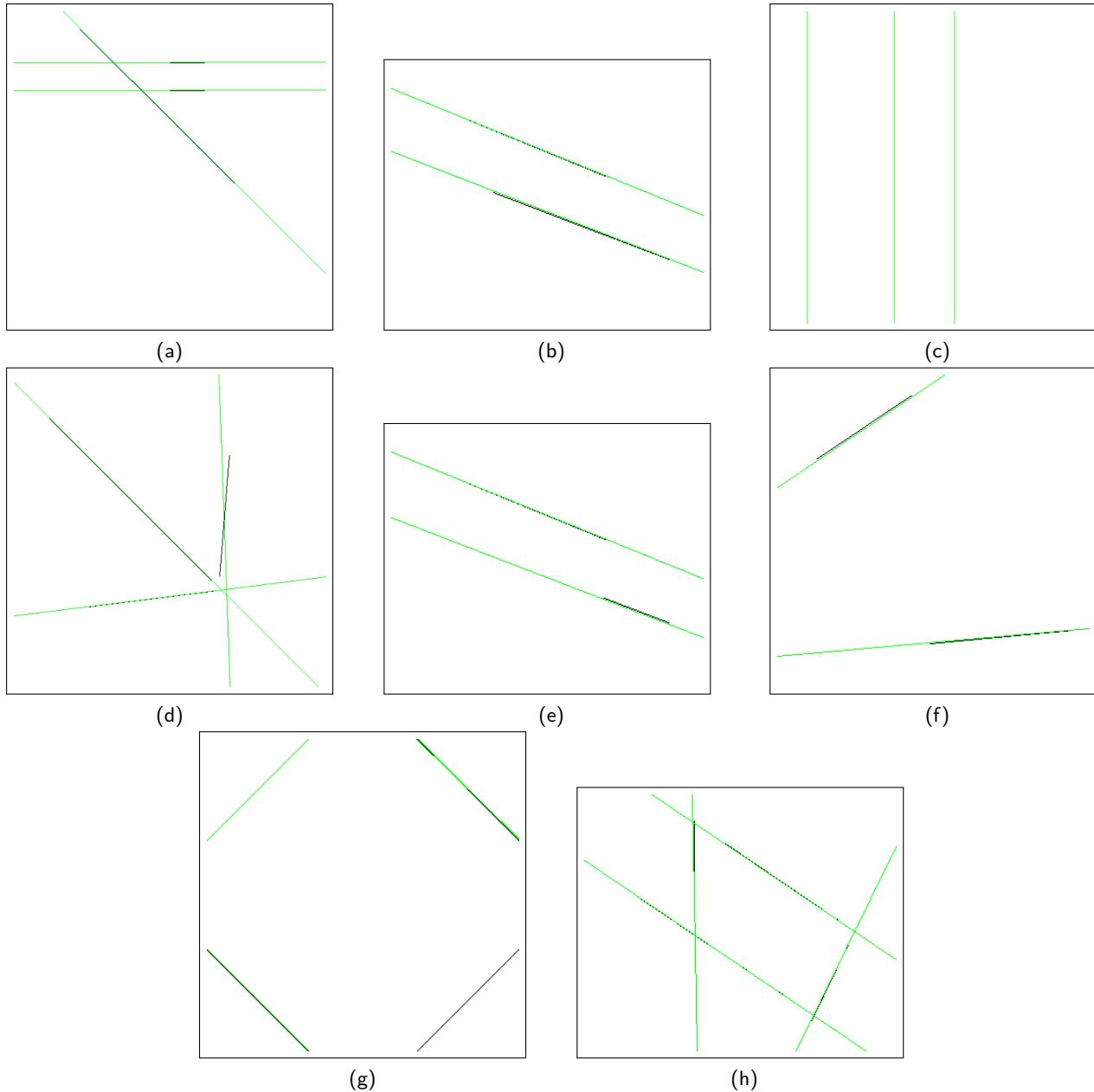


Figura A.4: Resultados dos testes da Transformada de Hough Modificada - Versão 3, em que se usam imagens controladas e o mesmo expoente no passo de *clustering* do algoritmo.

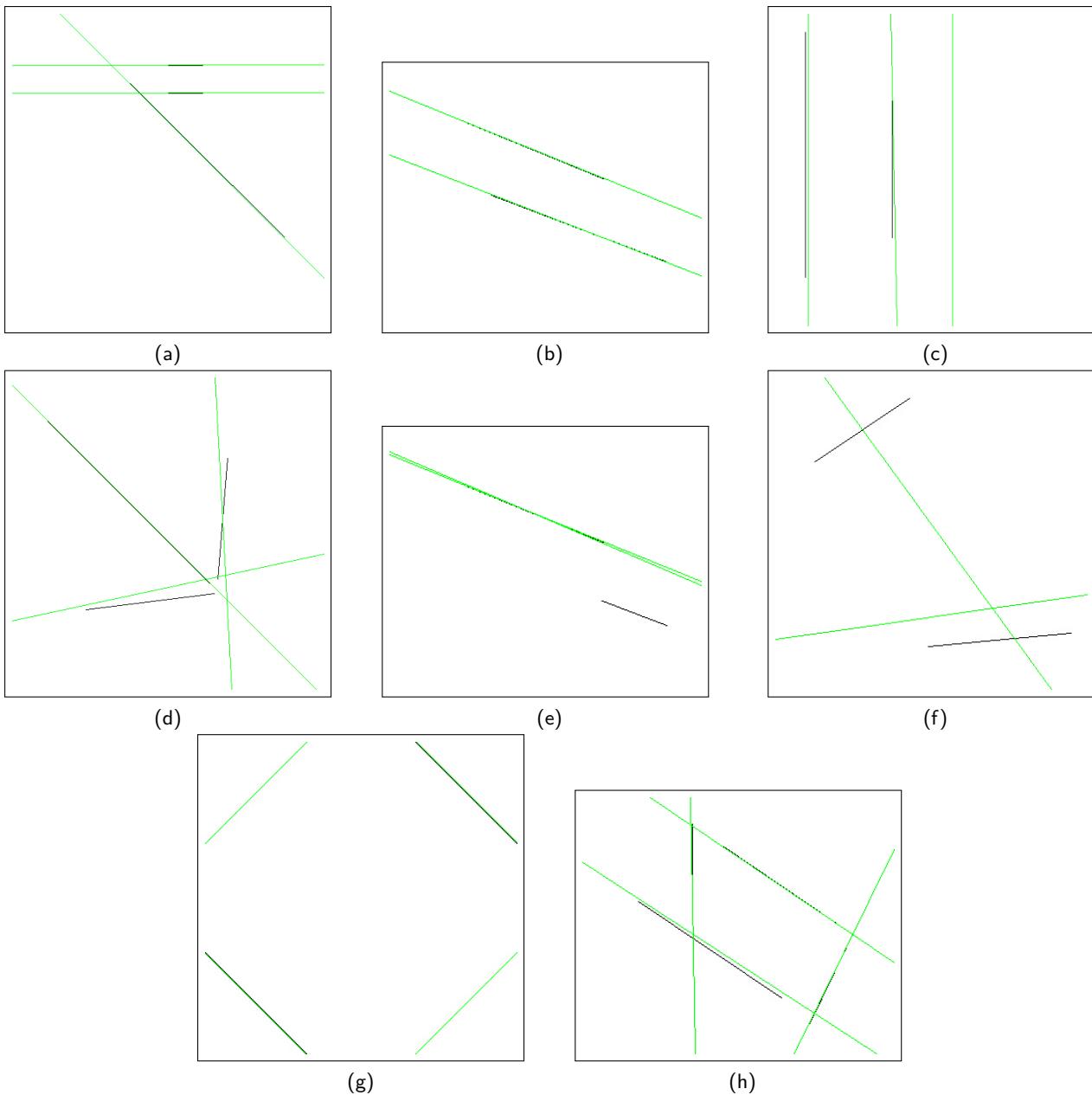
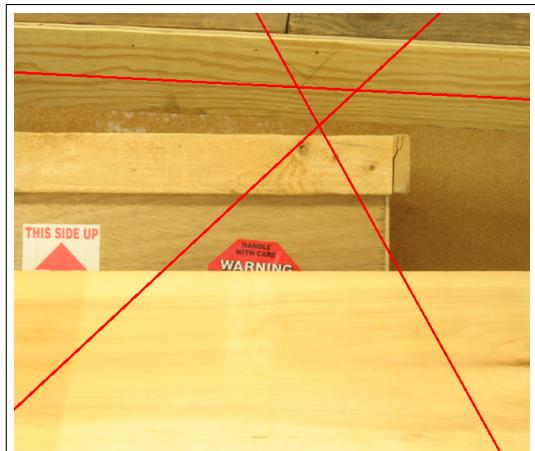
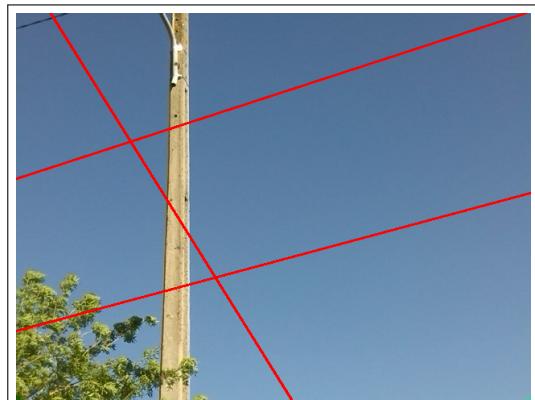


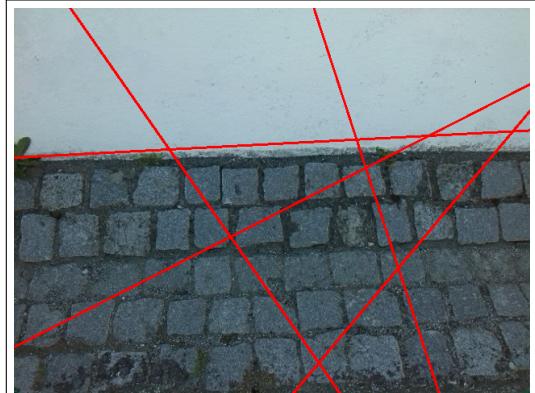
Figura A.5: Resultados dos testes da Versão 3 da Transformada de Hough Modificada, nos quais são usadas imagens controladas e expoentes diferentes no passo de *clustering* do algoritmo.



(a) [HS07]



(b)

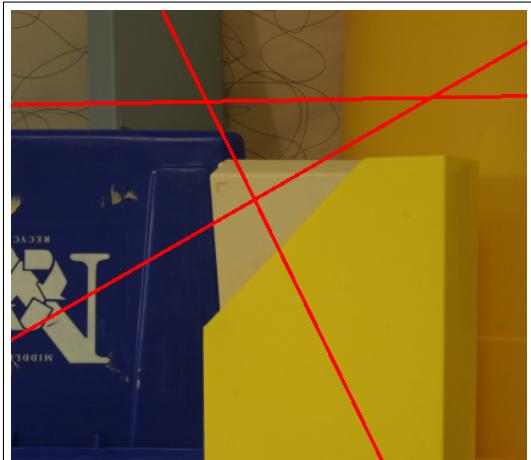


(c)



(d) [HS07]

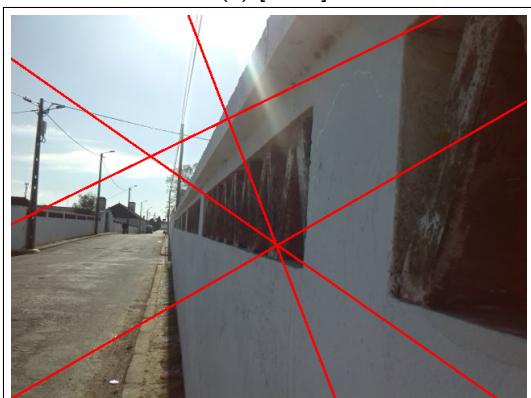
Figura A.6: Resultados dos testes da Transformada de Hough Modificada - Versão 1, em que são usadas fotografias como *input*.



(a) [HS07]



(b)



(c)

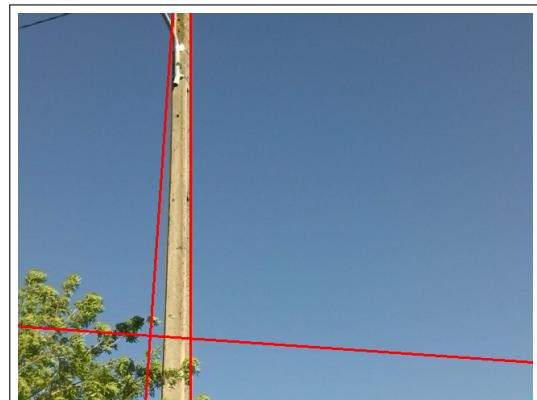


(d)

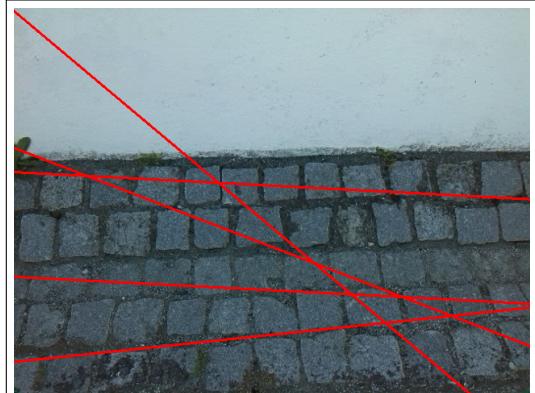
Figura A.7: Resultados dos testes da Transformada de Hough Modificada - Versão 1, em que são usadas fotografias como *input* (cont.).



(a) [HS07]



(b)



(c)



(d) [HS07]

Figura A.8: Resultados dos testes da Transformada de Hough Modificada - Versão 2, em que são usadas fotografias como *input* e expoentes iguais no passo de *clustering* do algoritmo.

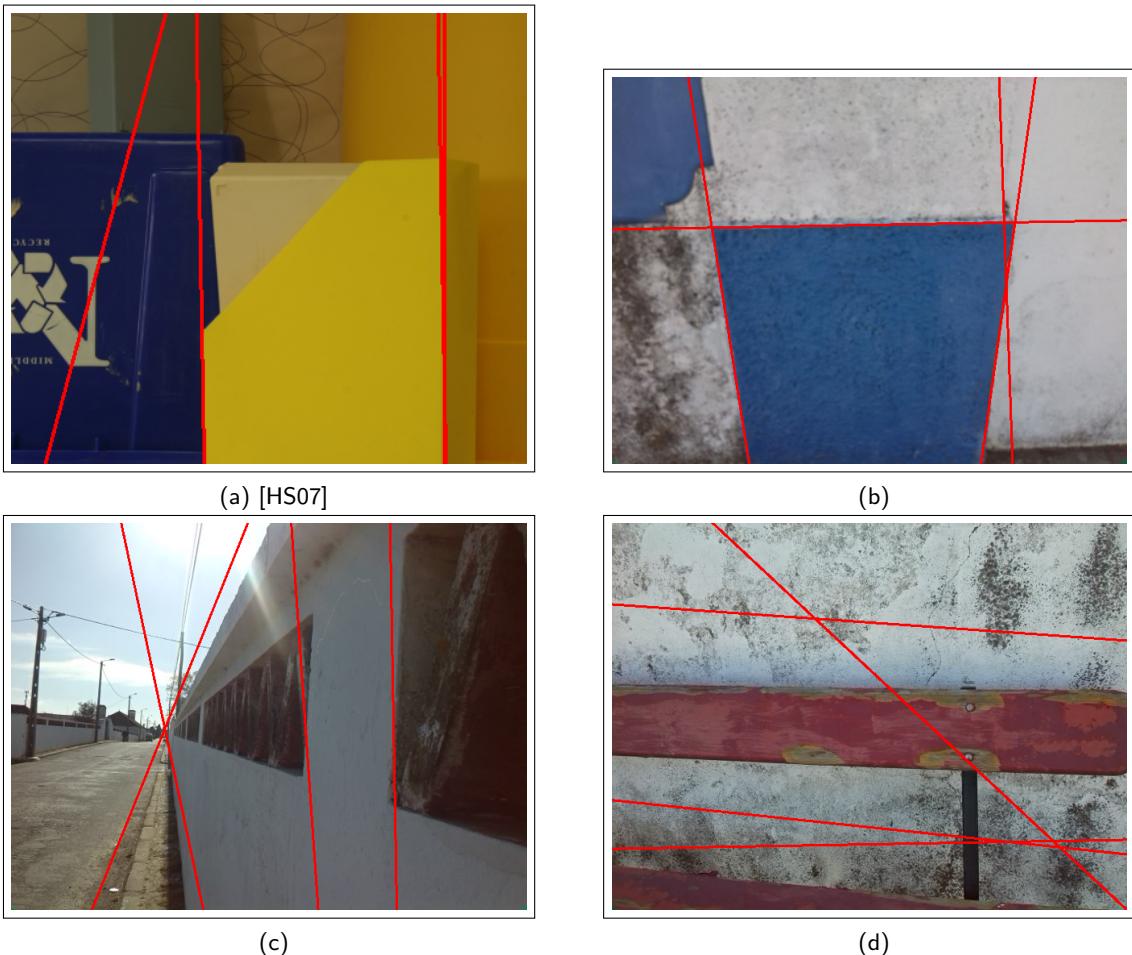


Figura A.9: Resultados dos testes da Transformada de Hough Modificada - Versão 2, em que são usadas fotografias como *input* e expoentes iguais no passo de *clustering* do algoritmo (cont.).



(a) [HS07]



(b)

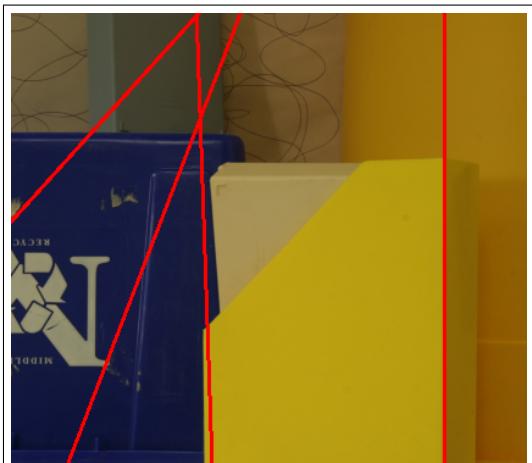


(c)

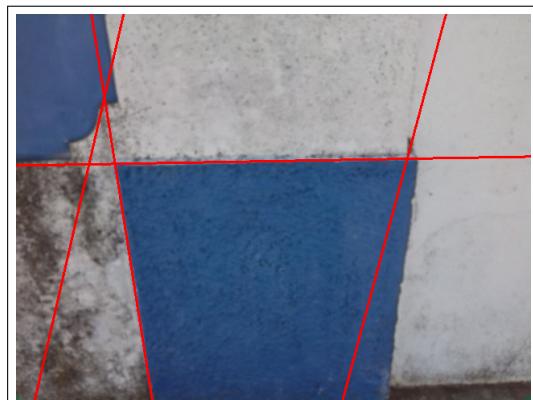


(d) [HS07]

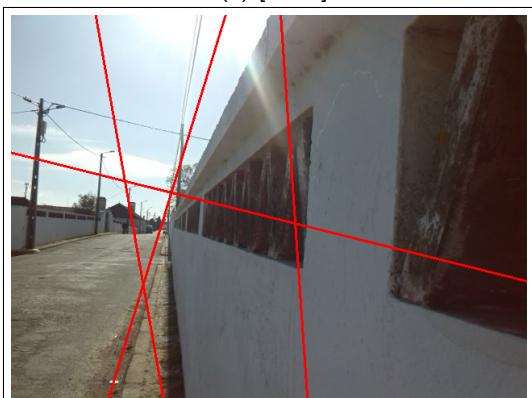
Figura A.10: Testes da Versão 2 da Transformada de Hough Modificada, rm que são usadas imagens controladas e expoentes diferentes no passo de *clustering* do algoritmo.



(a) [HS07]



(b)



(c)



(d)

Figura A.11: Testes da Versão 2 da Transformada de Hough Modificada, nos quais são usadas imagens controladas e expoentes diferentes no passo de *clustering* (cont.).

A.2.3 Transformada de Hough Modificada - Versão 3

Também quando se usam fotografias como *input* as condições de teste da THM-V3 são idênticas às descritas em A.1.3. Tal como aconteceu em A.1.3, também aqui foram testados os casos em que se usam expoentes iguais e expoentes diferentes no passo de *clustering*. São usados os mesmos valores para expoentes iguais e expoentes diferentes. Preservam-se também os valores do coeficiente usado na normalização da distância em ρ no passo em que se calcula a distância entre pontos e centroides.

Os resultados dos testes da Transformada de Hough Modificada - Versão 3, em que se usam expoentes iguais, podem ser consultados nas figuras A.12 e A.13. Observando as referidas figuras, é possível verificar que os resultados, no geral, são positivos. Na figura A.12-(a) todas as linhas detetadas se sobrepõem às reais, pelo que o resultado deste teste é bastante positivo. Nas figuras A.12-(b), A.12-(d), A.13-(a), A.13-(b) e A.13-(c) também se verificam resultados positivos, visto que a maioria das linhas detetadas corresponde a linhas reais. Apesar disto, nestas figuras, existem ainda linhas detetadas que apresentam inclinações incorretas ou que se encontram deslocadas das linhas reais presentes nas fotografias. Nas figuras A.12-(c) e A.13-(d), os resultados dos testes ao algoritmo são pouco satisfatórios, visto que existem mais linhas detetadas incorretas que corretas. Apesar disto, em A.12-(c), existem duas retas que correspondem a linhas reais na imagem.

Os resultados dos testes do mesmo algoritmo, correspondentes ao caso em que se usam expoentes diferentes, podem ser observados nas figuras A.14 e A.15. À semelhança do que se verificou nos testes com imagens controladas, quando se usam expoentes diferentes no passo de *clustering* da Transformada de Hough Modificada - Versão 3, verifica-se um ligeiro declínio na qualidade das deteções de linhas efetuada por este algoritmo. De notar que, na figura A.14-(a), uma das linhas detetadas tem uma inclinação incorreta. Na figura A.14-(b) apenas duas das três retas são detetadas (a outra é ignorada) e na figura A.14-(d) existem duas linhas detetadas com inclinações ligeiramente diferentes das reais e duas deteções incorretas. Nas figuras A.15-(a), A.15-(b) e A.15-(c) existem duas retas detetadas corretamente e duas outras retas com inclinações diferentes das reais. Nas figuras A.14-(c) e A.15-(d) mantém-se os resultados menos bons verificados anteriormente.

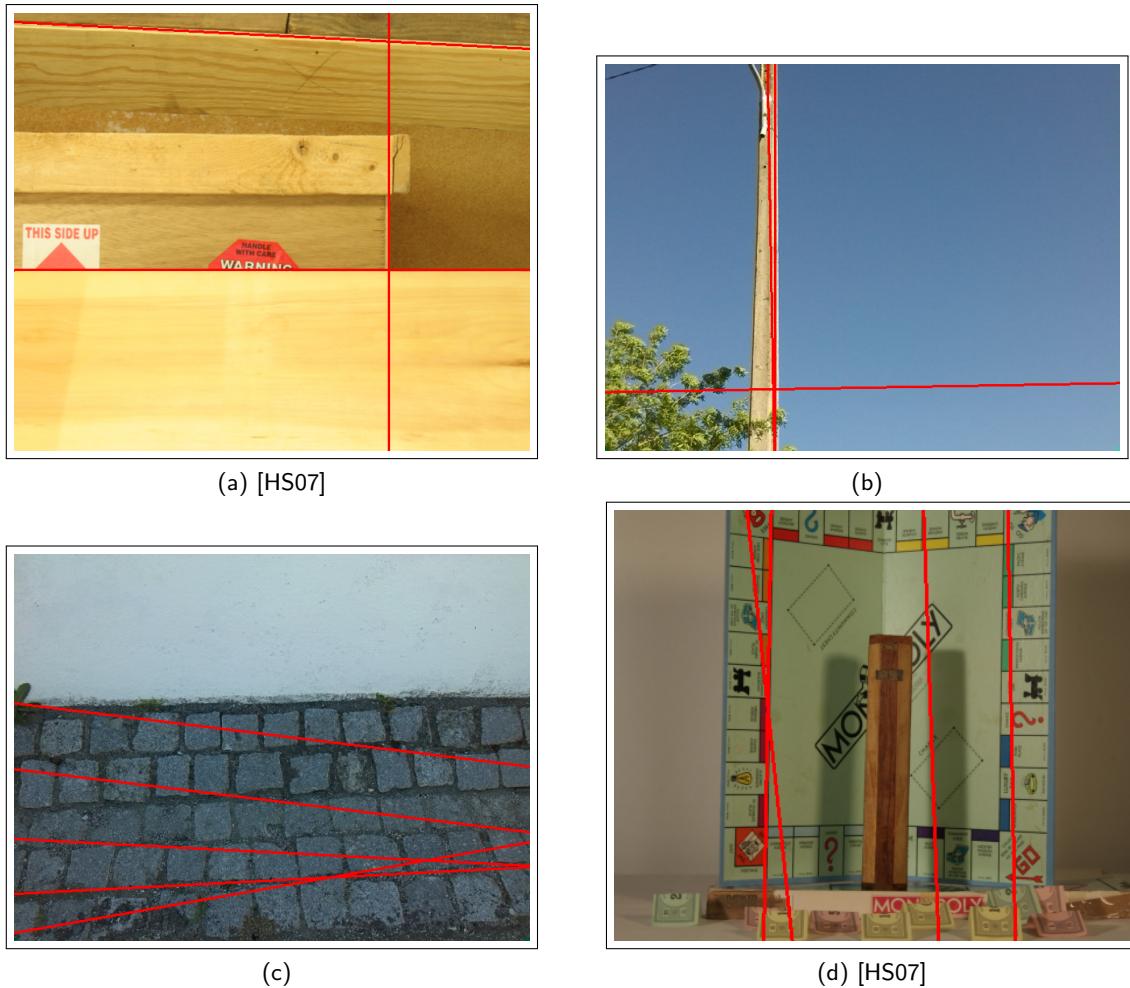


Figura A.12: Resultados dos testes da Transformada de Hough Modificada - Versão 3, em que se usam fotografias como *input* e expoentes iguais no passo de *clustering*.

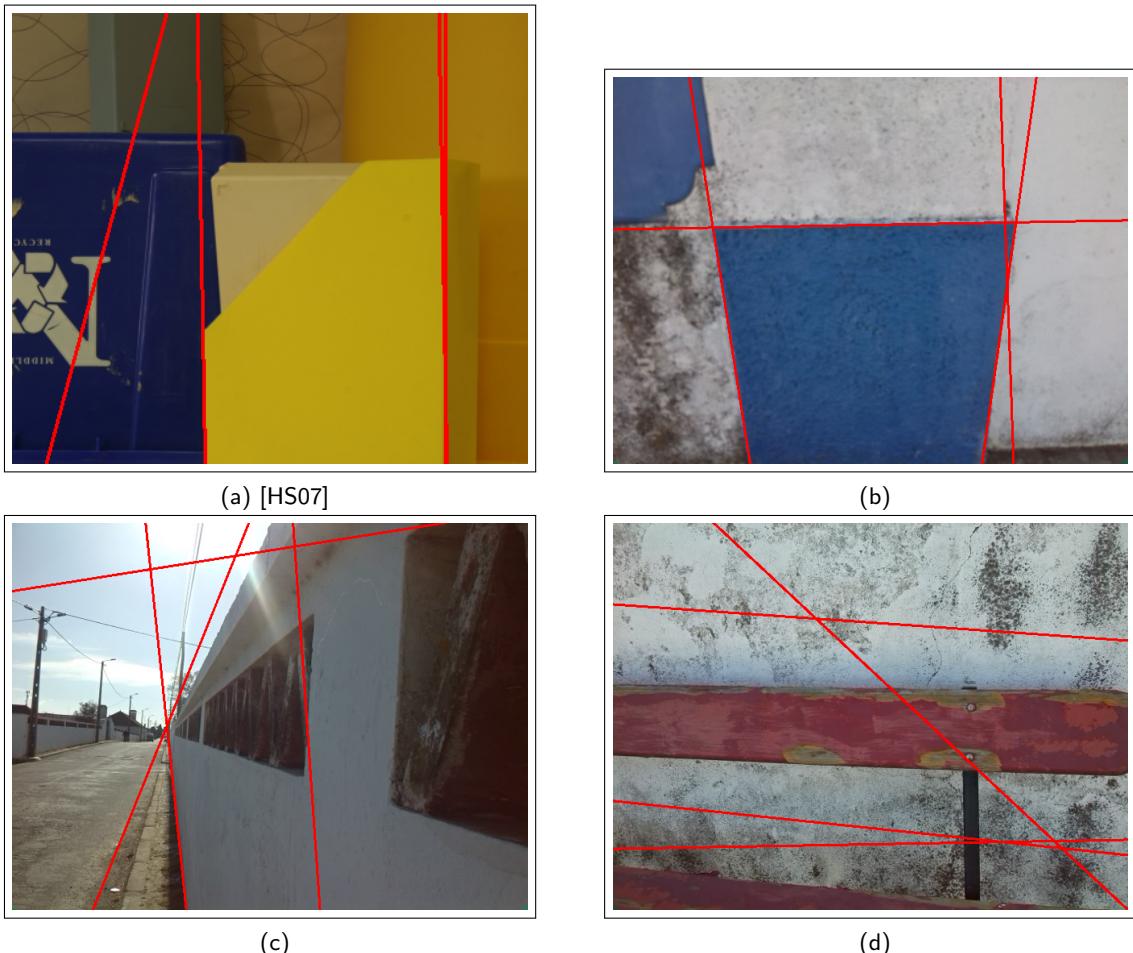


Figura A.13: Resultados dos testes da Transformada de Hough Modificada - Versão 3, em que se usam fotografias como *input* e expoentes iguais no passo de *clustering* (cont.).

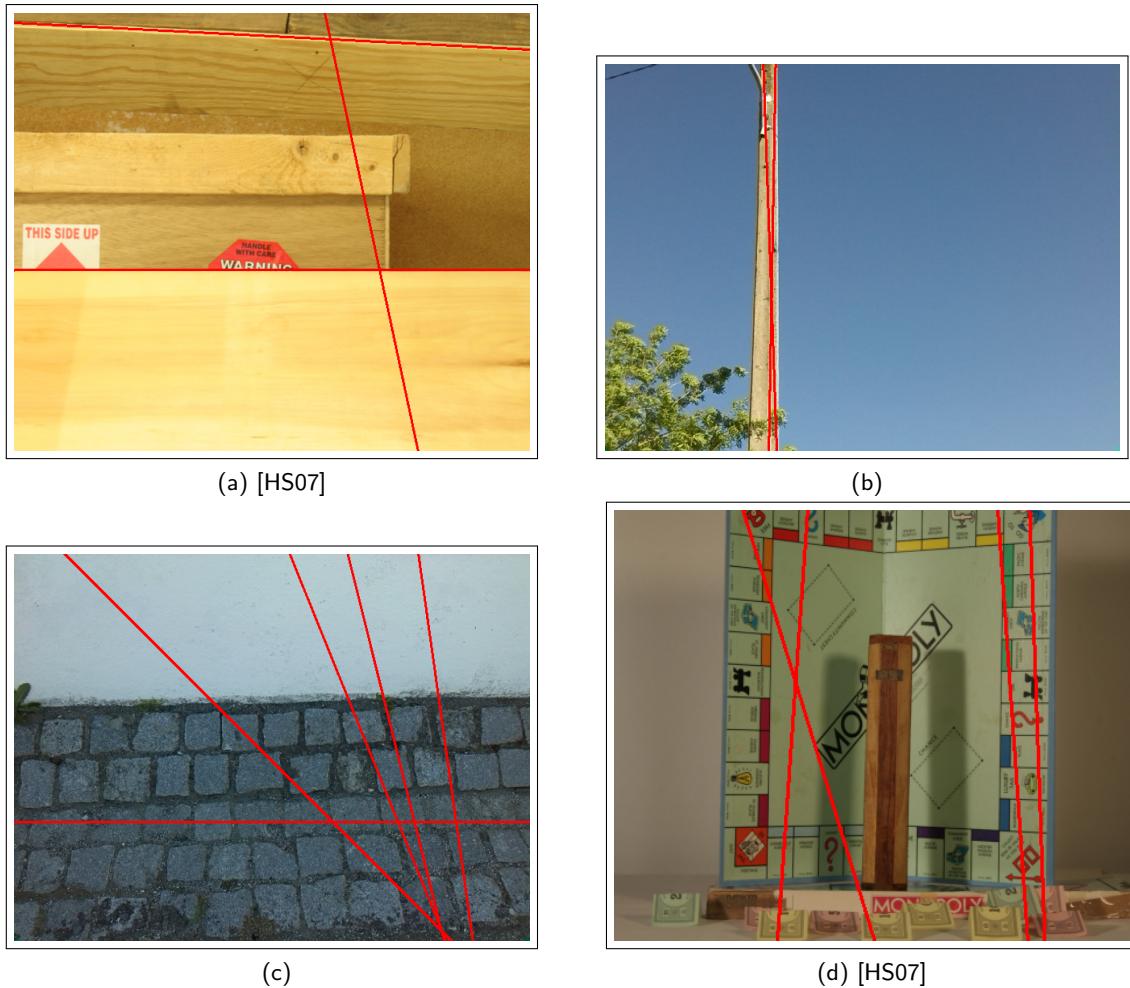


Figura A.14: Testes da Versão 3 da Transformada de Hough Modificada, nos quais são usadas imagens controladas e expoentes diferentes no passo de *clustering*.

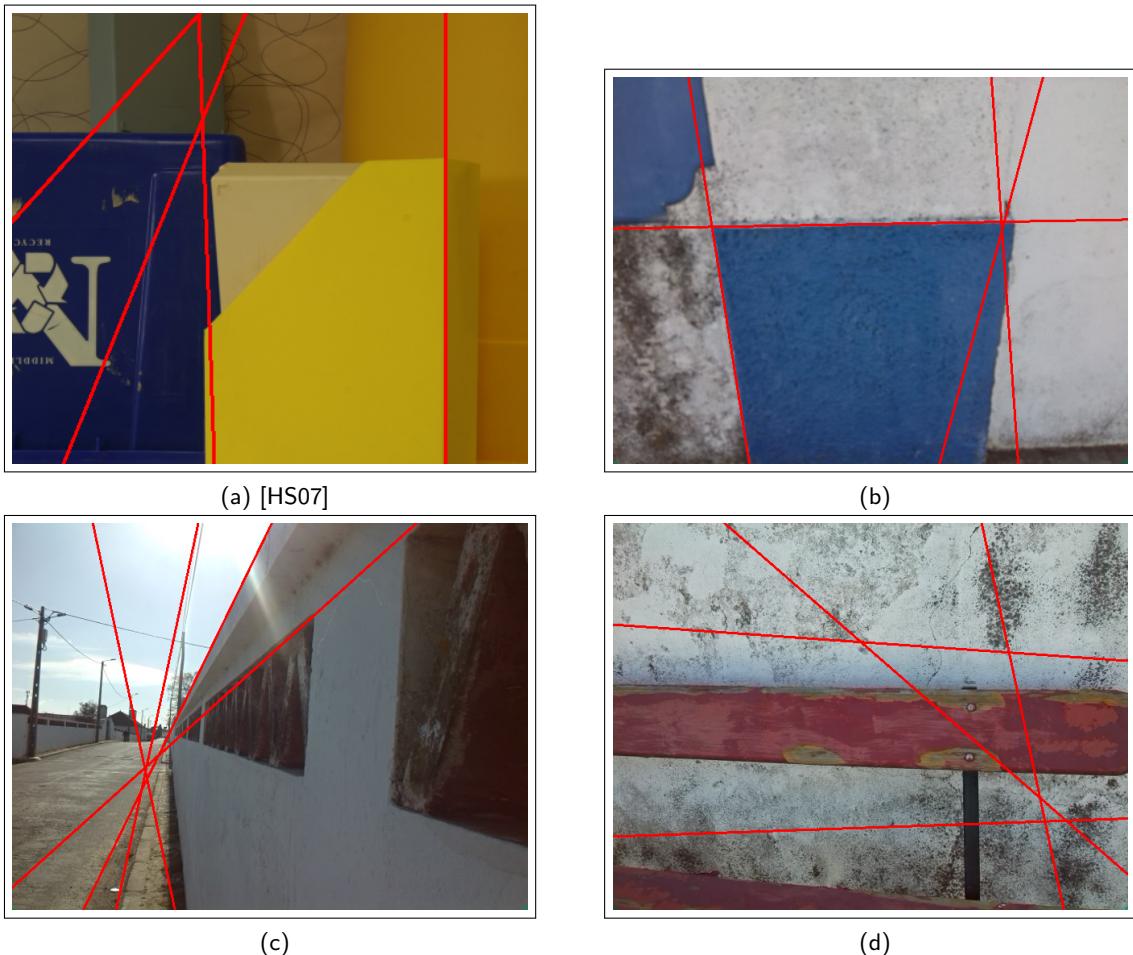


Figura A.15: Testes da Versão 3 da Transformada de Hough Modificada, nos quais são usadas imagens controladas e expoentes diferentes no passo de *clustering* (cont.).

B

Parametrização das Linhas Detetadas na Forma Standard

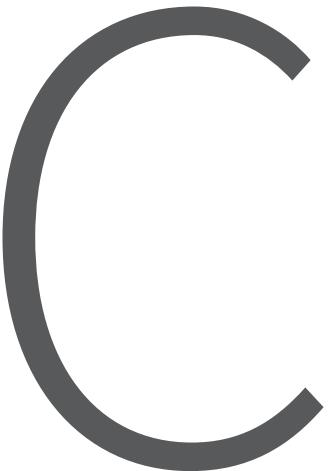
Este anexo inclui os parâmetros das linhas detetadas pela Transformada de Hough nas imagens *stereo* da figura 6.6. Cada linha detetada pela TH é descrita de dois modos: na sua forma *standard* e por dois parâmetros (ρ, θ) . Esta informação pode ser consultada nas tabelas B.1 e B.2.

Tabela B.1: Parâmetros das linhas detetadas com a Transformada de Hough na figura 6.6-(a), que são apresentadas na figura 6.8-(a).

Parâmetros da Linha Descrita Por (ρ, θ)		Parâmetros da Linha na Forma Standard		
ρ	θ	A	B	C
311.0000	0.0000	1.0000	0.0000	-311.0000
348.0000	0.0000	1.0000	0.0000	-348.0000
352.0000	0.0000	1.0000	0.0000	-352.0000
353.0000	0.0000	1.0000	0.0000	-353.0000
390.0000	0.0000	1.0000	0.0000	-390.0000
422.0000	0.0000	1.0000	0.0000	-422.0000
357.0000	0.0175	0.9998	0.0175	-357.0000
394.0000	0.0175	0.9998	0.0175	-394.0000
402.0000	1.4137	0.1564	0.9877	-402.0000
330.0000	1.4661	0.1045	0.9945	-330.0000
337.0000	1.4661	0.1045	0.9945	-337.0000
256.0000	1.5184	0.0523	0.9986	-256.0000
262.0000	1.5184	0.0523	0.9986	-262.0000
250.0000	1.5359	0.0349	0.9994	-250.0000
256.0000	1.5359	0.0349	0.9994	-256.0000
331.0000	1.5359	0.0349	0.9994	-331.0000
326.0000	1.5533	0.0175	0.9998	-326.0000
186.0000	1.5708	0.0000	1.0000	-186.0000
288.0000	1.6057	-0.0349	0.9994	-288.0000
295.0000	1.6057	-0.0349	0.9994	-295.0000
322.0000	1.6930	-0.1219	0.9925	-322.0000
317.0000	1.7104	-0.1392	0.9903	-317.0000
-295.0000	3.0892	-0.9986	0.0523	295.0000
-217.0000	3.0892	-0.9986	0.0523	217.0000
-341.0000	3.1067	-0.9994	0.0349	341.0000
-226.0000	3.1067	-0.9994	0.0349	226.0000
-223.0000	3.1067	-0.9994	0.0349	223.0000
-347.0000	3.1241	-0.9998	0.0175	347.0000
-343.0000	3.1241	-0.9998	0.0175	343.0000
-266.0000	3.1241	-0.9998	0.0175	266.0000
-262.0000	3.1241	-0.9998	0.0175	262.0000
-227.0000	3.1241	-0.9998	0.0175	227.0000

Tabela B.2: Parâmetros das linhas detetadas com a Transformada de Hough na figura 6.6-(b), que são apresentadas na figura 6.8-(b).

Parâmetros da Linha Descripta Por (ρ, θ)		Parâmetros da Linha na Forma Standard		
ρ	θ	A	B	C
252.0000	0.0000	1.0000	0.0000	-252.0000
279.0000	0.0000	1.0000	0.0000	-279.0000
282.0000	0.0000	1.0000	0.0000	-282.0000
312.0000	0.0000	1.0000	0.0000	-312.0000
318.0000	0.0000	1.0000	0.0000	-318.0000
358.0000	0.0000	1.0000	0.0000	-358.0000
363.0000	0.0000	1.0000	0.0000	-363.0000
402.0000	0.0000	1.0000	0.0000	-402.0000
253.0000	0.0175	0.9998	0.0175	-253.0000
283.0000	0.0175	0.9998	0.0175	-283.0000
370.0000	0.0349	0.9994	0.0349	-370.0000
387.0000	1.4137	0.1564	0.9877	-387.0000
392.0000	1.4137	0.1564	0.9877	-392.0000
386.0000	1.4312	0.1392	0.9903	-386.0000
307.0000	1.4835	0.0872	0.9962	-307.0000
294.0000	1.5184	0.0523	0.9986	-294.0000
233.0000	1.5359	0.0349	0.9994	-233.0000
239.0000	1.5359	0.0349	0.9994	-239.0000
312.0000	1.5359	0.0349	0.9994	-312.0000
228.0000	1.5533	0.0175	0.9998	-228.0000
233.0000	1.5533	0.0175	0.9998	-233.0000
306.0000	1.5533	0.0175	0.9998	-306.0000
307.0000	1.5533	0.0175	0.9998	-307.0000
226.0000	1.5708	0.0000	1.0000	-226.0000
168.0000	1.5882	-0.0175	0.9998	-168.0000
161.0000	1.6057	-0.0349	0.9994	-161.0000
144.0000	1.6406	-0.0698	0.9976	-144.0000
294.0000	1.7279	-0.1564	0.9877	-294.0000
-431.0000	3.1241	-0.9998	0.0175	431.0000
-398.0000	3.1241	-0.9998	0.0175	398.0000
-397.0000	3.1241	-0.9998	0.0175	397.0000
-394.0000	3.1241	-0.9998	0.0175	394.0000
-393.0000	3.1241	-0.9998	0.0175	393.0000
-354.0000	3.1241	-0.9998	0.0175	354.0000



Resultado da Aplicação do Passo de Pré-Processamento das Linhas Detetadas

Neste anexo é possível consultar os grupos de linhas verticais, horizontais e diagonais obtidos no passo de pré-processamento, que antecede a procura por correspondências entre linhas, referido nos Capítulos 5 e 6. As linhas incluídas nestes grupos são detetadas pela Transformada de Hough nas imagens stereo da figura 6.6 do Capítulo 6, sendo os resultados da aplicação deste algoritmo apresentados na figura 6.8. Os grupos de linhas detetadas presentes na figura 6.8-(a) podem ser consultados na tabela C.1. Os grupos constituídos por linhas detetadas presentes na figura 6.8-(b) podem ser consultados na tabela C.2.

Tabela C.1: Grupos de linhas obtidos com pré-processamento das linhas detetadas pela TH, que estão presentes na figura 6.8-(a).

Grupo	Parâmetros das Linhas na Figura 6.8-(a)		
	A	B	C
Linhas Verticais	0.1564	0.9877	-402.0000
	0.1045	0.9945	-330.0000
	0.1045	0.9945	-337.0000
	0.0523	0.9986	-256.0000
	0.0523	0.9986	-262.0000
	0.0349	0.9994	-250.0000
	0.0349	0.9994	-256.0000
	0.0349	0.9994	-331.0000
	0.0175	0.9998	-326.0000
	0.0000	1.0000	-186.0000
	-0.0349	0.9994	-288.0000
	-0.0349	0.9994	-295.0000
	-0.1219	0.9925	-322.0000
	-0.1392	0.9903	-317.0000
Linhas Horizontais	1.0000	0.0000	-311.0000
	1.0000	0.0000	-348.0000
	1.0000	0.0000	-352.0000
	1.0000	0.0000	-353.0000
	1.0000	0.0000	-390.0000
	1.0000	0.0000	-422.0000
	0.9998	0.0175	-357.0000
	0.9998	0.0175	-394.0000
Linhas Diagonais	-0.9986	0.0523	295.0000
	-0.9986	0.0523	217.0000
	-0.9994	0.0349	341.0000
	-0.9994	0.0349	226.0000
	-0.9994	0.0349	223.0000
	-0.9998	0.0175	347.0000
	-0.9998	0.0175	343.0000
	-0.9998	0.0175	266.0000
	-0.9998	0.0175	262.0000
	-0.9998	0.0175	227.0000

Tabela C.2: Grupos de linhas obtidos com pré-processamento das linhas detetadas pela TH, que estão presentes na figura 6.8-(b).

Grupo	Parâmetros das Linhas na Figura 6.8-(b)		
	A	B	C
Linhas Verticais	0.1564	0.9877	-387.0000
	0.1564	0.9877	-392.0000
	0.1392	0.9903	-386.0000
	0.0872	0.9962	-307.0000
	0.0523	0.9986	-294.0000
	0.0349	0.9994	-233.0000
	0.0349	0.9994	-239.0000
	0.0349	0.9994	-312.0000
	0.0175	0.9998	-228.0000
	0.0175	0.9998	-233.0000
	0.0175	0.9998	-306.0000
	0.0175	0.9998	-307.0000
	0.0000	1.0000	-226.0000
	-0.0175	0.9998	-168.0000
Linhas Horizontais	-0.0349	0.9994	-161.0000
	-0.0698	0.9976	-144.0000
	-0.1564	0.9877	-294.0000
	1.0000	0.0000	-252.0000
	1.0000	0.0000	-279.0000
	1.0000	0.0000	-282.0000
	1.0000	0.0000	-312.0000
	1.0000	0.0000	-318.0000
	1.0000	0.0000	-358.0000
	1.0000	0.0000	-363.0000
Linhas Diagonais	1.0000	0.0000	-402.0000
	0.9998	0.0175	-253.0000
	0.9998	0.0175	-283.0000
	0.9994	0.0349	-370.0000
	-0.9998	0.0175	431.0000
	-0.9998	0.0175	398.0000

D

Resultados da Aplicação dos Algoritmos de Correspondência

Neste anexo incluem-se os resultados da aplicação dos algoritmos de correspondência descritos no Capítulo 5. As linhas usadas para estabelecer correspondências entre as imagens do par *stereo*, figura 6.6, são as detetadas pela Transformada de Hough. Estas podem ser consultadas na figura 6.8 do Capítulo 6. Como foi mencionado no Capítulo 5, estes algoritmos identificam correspondências entre grupos de linhas do mesmo tipo. Assim, existem três tabelas para cada algoritmo, nas quais podem ser consultadas as correspondências identificadas pelos mesmos. As correspondências encontradas pelo Algoritmo 1 podem ser consultadas nas tabelas D.1, D.2 e D.3. As tabelas D.4, D.5 e D.6 contêm as correspondências encontradas pelo Algoritmo 2.

Tabela D.1: Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas verticais, cuja correspondência foi identificada com recurso ao Algoritmo 1.

Parâmetros dos Planos da Imagem da Câmara 1 (Pl_1)				Parâmetros dos Planos da Imagem da Câmara 2 (Pl_2)			
A	B	C	D	A	B	C	D
47.8133	-1.5880	11.2581	-257.6676	47.5440	0.7844	6.0467	-231.3234
47.7851	-0.5224	5.9001	-192.7143	47.4111	1.2938	6.2961	-235.2867
47.8268	-0.8821	6.5613	-199.7143	47.4648	1.4579	5.9174	-230.2867
47.6194	0.6008	0.4224	-126.1372	48.6374	0.2154	0.2080	-150.2009
47.6551	0.2926	0.9891	-132.1372	48.5029	1.2348	0.7200	-158.7138
47.6473	0.0075	0.3743	-122.6916	48.4179	-1.1031	1.8179	-170.5106
47.6830	-0.3007	0.9411	-128.6916	48.4404	-1.6366	1.6166	-166.8633
48.1298	-4.1537	8.0254	-203.6916	47.2875	-8.6511	3.5863	-180.5971
48.1496	-4.8023	8.0789	-201.2848	48.4297	-1.6694	1.6923	-167.8633
47.3511	1.4809	-4.6124	-63.9160	49.2661	1.4901	-4.1661	-91.5106
47.9875	-5.5872	6.1078	-171.2892	49.2416	0.2564	-4.1836	-89.2584
48.0292	-5.9468	6.7690	-178.2892	49.2242	0.7596	-3.9130	-93.8633
48.0155	-11.9480	12.1426	-219.3295	49.2017	1.2931	-3.7116	-97.5106
47.9086	-12.6192	12.2524	-217.2348	49.2778	0.9237	-4.2918	-88.8633

Tabela D.2: Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas horizontais, cuja correspondência foi identificada com recurso ao Algoritmo 1.

Parâmetros dos Planos da Imagem da Câmara 1 (Pl_1)				Parâmetros dos Planos da Imagem da Câmara 2 (Pl_2)			
A	B	C	D	A	B	C	D
0.2211	36.2085	-1.3413	-159.1700	1.9429	33.8905	3.7295	-112.2048
0.4415	34.3077	2.1536	-196.1700	1.0524	33.7958	4.0232	-113.5702
0.4653	34.1022	2.5314	-200.1700	0.7626	32.9095	6.0684	-140.5702
0.4713	34.0509	2.6259	-201.1700	1.6208	32.9057	6.0019	-142.2048
0.6917	32.1501	6.1208	-238.1700	0.7304	32.8110	6.2956	-143.5702
0.8823	30.5061	9.1434	-270.1700	0.3438	31.6293	9.0226	-179.5702
1.3024	34.0300	2.6213	-203.0624	0.4083	31.8263	8.5681	-173.5702
1.5228	32.1292	6.1162	-240.0624	-0.0856	30.3163	12.0525	-219.5702

Tabela D.3: Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas diagonais, cuja correspondência foi identificada com recurso ao Algoritmo 1.

Parâmetros dos Planos da Imagem da Câmara 1 (Pl_1)				Parâmetros dos Planos da Imagem da Câmara 2 (Pl_2)			
A	B	C	D	A	B	C	D
0.3713	-34.1585	-2.4509	197.3238	1.7717	-27.7797	-17.9561	294.9777
1.2131	-34.2504	-2.2852	193.5231	1.4174	-28.8630	-15.4565	261.9777
0.3951	-34.3640	-2.0731	193.3238	1.4066	-28.8958	-15.3807	260.9777
2.2922	-36.3814	1.6496	149.7674	1.3744	-28.9943	-15.1535	257.9777
0.8538	-38.3198	5.2001	116.3238	1.3637	-29.0271	-15.0777	256.9777
0.8776	-38.5253	5.5779	112.3238	0.9450	-30.3073	-12.1236	217.9777

Tabela D.4: Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas verticais, cuja correspondência foi identificada com recurso ao Algoritmo 2.

Parâmetros dos Planos da Imagem da Câmara 1 (Pl_1)				Parâmetros dos Planos da Imagem da Câmara 2 (Pl_2)			
A	B	C	D	A	B	C	D
47.7239	-0.8174	9.8413	-242.6676	47.3038	0.9655	7.0536	-245.2867
47.7538	-0.2290	4.2241	-172.1477	48.2348	1.1978	2.2361	-179.5376
47.8654	-1.6383	10.2198	-243.9735	48.1596	0.9680	2.7664	-186.5376
47.8457	-1.3514	4.0117	-164.1372	49.0454	1.4627	-2.6704	-112.2009
47.2310	0.9372	-5.8883	-44.2892	48.9810	1.2658	-2.2159	-118.2009
47.5460	0.8809	-1.2314	-105.6916	49.0836	0.9320	-2.8784	-108.5106
47.5817	0.5726	-0.6647	-111.6916	49.0192	0.7351	-2.4239	-114.5106
48.0166	-3.1776	6.2307	-184.6916	48.2140	-1.7268	3.2572	-189.5106
47.5658	0.2323	-1.1779	-103.2848	48.2257	-2.2931	3.1315	-186.8633
47.7537	-1.0743	10.3136	-247.6676	49.6710	1.5694	-7.2135	-49.2584
47.5956	-0.0246	-0.7056	-108.2848	48.4133	-3.2516	1.0515	-156.1728
48.0305	-3.7748	6.1897	-181.2848	48.3382	-3.4814	1.5818	-163.1728
47.6804	-12.3669	10.6673	-197.1705	47.3678	-8.0788	5.0830	-203.1480
47.1021	-0.0286	-6.3828	-32.8046	47.2387	-8.6601	5.0135	-200.8549

Tabela D.5: Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas horizontais, cuja correspondência foi identificada com recurso ao Algoritmo 2.

Parâmetros dos Planos da Imagem da Câmara 1 (Pl_1)				Parâmetros dos Planos da Imagem da Câmara 2 (Pl_2)			
A	B	C	D	A	B	C	D
0.0483	37.6984	-4.0806	-130.1700	0.4190	31.8591	8.4923	-172.5702
0.2270	36.1572	-1.2469	-160.1700	0.0218	30.6446	11.2950	-209.5702
0.2628	35.8489	-0.6801	-166.1700	-0.0212	30.5133	11.5980	-213.5702
0.5011	33.7940	3.0982	-206.1700	-0.0319	30.4804	11.6737	-214.5702
0.5308	33.5371	3.5705	-211.1700	-0.4292	29.2659	14.4764	-251.5702
0.7632	31.5336	7.2543	-250.1700	-0.7727	28.2155	16.9003	-283.5702
-0.1304	39.2395	-6.9143	-100.1700	0.8263	30.4767	11.6072	-216.2048
0.0304	37.8525	-4.3639	-127.1700	0.4290	29.2622	14.4099	-253.2048

Tabela D.6: Parâmetros dos planos Pl_1 e Pl_2 , obtidos a partir de linhas diagonais, cuja correspondência foi identificada com recurso ao Algoritmo 2.

Parâmetros dos Planos da Imagem da Câmara 1 (Pl_1)				Parâmetros dos Planos da Imagem da Câmara 2 (Pl_2)			
A	B	C	D	A	B	C	D
-0.1291	-29.8432	-10.3853	281.3238	0.8698	-30.5370	-11.5933	210.9777
0.0675	-31.5385	-7.2682	248.3238	0.8269	-30.6683	-11.2903	206.9777
0.0734	-31.5899	-7.1738	247.3238	0.0002	-33.1959	-5.4578	129.9777
0.0913	-31.7440	-6.8904	244.3238	-0.0428	-33.3272	-5.1548	125.9777
0.0972	-31.7954	-6.7960	243.3238	-0.4186	-34.4760	-2.5036	90.9777
0.3296	-33.7989	-3.1121	204.3238	1.7086	-30.5809	-11.5174	207.4267

Bibliografia

- [AAK71] YI Abel-Aziz and HM Karara. Direct linear transformation from comparator coordinates into object space coordinates. *Urbana, IL: American Society of Photogrammetry*, pages 1–18, 1971.
- [AFS⁺11] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.
- [All05] William Allen. Big portable room camera obscura - diagram. <https://www.flickr.com/photos/woodpainter/16156163997>, 2005. Consultado em: 20-07-2015.
- [ATV08] Teresa Azevedo, João Manuel RS Tavares, and Mário Vaz. Volumetric object reconstruction using generalized voxel coloring. In *International Workshop on Combinatorial Image Analysis*, 2008.
- [Aut15] Autodesk. 123d catch. <http://www.123dapp.com/howto/catch>, 2015. Consultado em: 01-10-2015.
- [B⁺00] Gary Bradski et al. The opencv library. *Doctor Dobbs Journal*, 25(11):120–126, 2000.
- [Bal81] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.
- [Bal01] David Balihar. What is a pinhole camera? <http://www.pinhole.cz/en/pinholecameras/whatis.html>, 2001. Consultado em: 12-07-2015.
- [BBH] M Z Brown, D Burschka, and G D Hager. ADVANCES IN COMPUTATIONAL STEREO. pages 1–50.
- [BE08] Michael J Broxton and Larry J Edwards. The ames stereo pipeline: Automated 3d surface reconstruction from orbital imagery. In *Lunar and Planetary Science Conference*, volume 39, page 2419, 2008.
- [BFV05] Herbert Bay, Vittorio Ferrari, and Luc Van Gool. Wide-baseline stereo matching with line segments. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:329–336, 2005.
- [BGZ⁺04] Y Bastanlar, N Grammalidisb, X Zabulisc, E Yilmaz, Y Yardimci, and G Triantafyllidisb. 3D Reconstruction for a Cultural Heritage Virtual Tour System. *Archives*, pages 3–11, 2004.

- [Bir98] Stan Birchfield. An introduction to projective geometry (for computer vision). *Unpublished note, Stanford university*, 1998.
- [BK08] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* "O'Reilly Media, Inc.", 2008.
- [BW95] Serge Belongie and Markus Weber. Recognising spoken words from lip movement. *CNS248 Report*, 1995.
- [Can86] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.
- [CF98] Ta Clarke and Jg Fryer. The development of camera calibration methods and models. *The Photogrammetric Record*, 16(91):51–66, 1998.
- [Che00] Frank Chen. Overview of three-dimensional shape measurement using optical methods. 2000.
- [CL95] B Curless and M Levoy. Better optical triangulation through spacetime analysis. , *Fifth International Conference on Computer Vision, 1995. Proceedings*, pages 987–994, 1995.
- [CLWZ08] Shengyong Chen, YF Li, Wanliang Wang, and Jianwei Zhang. *Active sensor planning for multiview vision tasks*, volume 1. Springer, 2008.
- [Com05] Wikimedia Commons. K-means. <http://compprag.christopherpotts.net/figures/swda/kmeans-wikipedia-ex.png>, 2005. Consultado em: 25-07-2015.
- [CS11] Boguslaw Cyganek and J Paul Siebert. *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons, 2011.
- [Cur00] B Curless. Overview of Active Vision Techniques. *Siggraph' 99*, pages 1–24, 2000.
- [Dea81] Stanley R. Deans. Hough Transform from the Radon Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(2):185–188, 1981.
- [Dea00] Stanley Deans. Radon and abel transforms. 2000.
- [DH72] Richard O Duda and Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [Dia03] Paulo Dias. Tsai Camera Calibration. , 2003. Consultado em: 01-11-2015.
- [DOOS12] Kurt Demaagd, Anthony Oliver, Nathan Oostendorp, and Katherine Scott. *Practical Computer Vision with SimpleCV: The Simple Way to Make Technology See.* "O'Reilly Media, Inc.", 2012.
- [Dye01] C.R. Dyer. Volumetric scene reconstruction from multiple views. *Foundations of Image Understanding*, (1):1–20, 2001.
- [Ela99] A. F. Elaksher. Multi image matching of straight lines with geometric constraints. *Computer*, 1999.
- [Fau93] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993.
- [FBDH⁺13] Robert B Fisher, Toby P Breckon, Kenneth Dawson-Howe, Andrew Fitzgibbon, Craig Robertson, Emanuele Trucco, and Christopher KI Williams. *Dictionary of computer vision and image processing*. John Wiley & Sons, 2013.

- [FFGG⁺10] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, et al. Building rome on a cloudless day. In *Computer Vision–ECCV 2010*, pages 368–381. Springer, 2010.
- [FK02] Olivier Faugeras and Renaud Keriven. *Variational principles, surface evolution, pde's, level set methods and the stereo problem*. IEEE, 2002.
- [FP11] David A Forsyth and Jean Ponce. *Computer Vision: A Modern Approach (2nd Edition)*. Pearson, 2011.
- [FT86] Olivier D Faugeras and Giorgio Toscani. The calibration problem for stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 86, pages 15–20, 1986.
- [FT87] OD Faugeras and Giorgio Toscani. Camera calibration for 3d computer vision. In *Proceedings of International Workshop on Machine Vision and Machine Intelligence, Tokyo, Japan*, 1987.
- [Gel12] Walter Gellert. *The VNR concise encyclopedia of mathematics*. Springer Science & Business Media, 2012.
- [GLV04] M Van Ginkel, C. L. Luengo Hendriks, and L J Van Vliet. A short introduction to the Radon and Hough transforms and how they relate to each other. *Reading*, pages 1–11, 2004.
- [GMM02] Steven B. Goldberg, Mark W. Maimone, and Lany Matthies. Stereo vision and rover navigation software for planetary exploration. *IEEE Aerospace Conference Proceedings*, 5(March):2025–2036, 2002.
- [GPKZ09] Siyu Guo, Tony Pridmore, Yaguang Kong, and Xufang Zhang. An improved hough transform voting scheme utilizing surround suppression. *Pattern Recognition Letters*, 30(13):1241–1252, 2009.
- [Gre96] Jon Grepstad. Pinhole photography. <http://photo.net/pinhole/pinhole>, 1996. Consultado em: 12-07-2015.
- [GRS14] Leonardo Gomes, Olga Regina Pereira Bellon, and Luciano Silva. 3D reconstruction methods for digital preservation of cultural heritage: A survey. *Pattern Recognition Letters*, 50:3–14, 2014.
- [GWYH14] Lin Gui, Zhiqiang Wei, Bo Yin, and Lei Huang. Reconstruction with underwater stereo vision. In *International Conference on Logistics Engineering, Management and Computer Science (LEMCS 2014)*. Atlantis Press, 2014.
- [GY14] Lin Gui and Bo Yin. Reconstruction with Underwater Stereo Vision. (Lemcs), 2014.
- [GZS11] Andreas Geiger, Julius Ziegler, and Christoph Stiller. StereoScan: Dense 3d reconstruction in real-time. *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968, jun 2011.
- [HAAg99] Ghassan Hamarneh, Karin Althoff, and Rafeef Abu-gharbieh. Automatic Line Detection. (September), 1999.
- [HBE06] MD Hancher, MJ Broxton, and LJ Edwards. A user's guide to the nasa vision workbench. *Intelligent Systems Division, NASA Ames Research Center*, 2006.
- [Hef06] Jim Hefferon. Linear algebra. <http://joshua.smcvt.edu/linearalgebra/book.pdf>, 2006. Consultado em: 12-12-2015.
- [HKP11] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.
- [HMSR15] Allam Shehata Hassanein, Sherien Mohammad, Mohamed Sameer, and Mohammad Ehab Ragab. A Survey on Hough Transform, Theory, Techniques and Applications. page 18, 2015.

- [Hor00] Berthold KP Horn. Tsai's camera calibration method revisited. *Online: http://people.csail.mit.edu/bkph/articles/Tsai_Revisited.pdf*, 2000.
- [Hou62] Paul VC Hough. Method and means for recognizing complex patterns. Technical report, 1962.
- [HS02] Carlos Hernández Esteban and Francis Schmitt. Multi-stereo 3D object reconstruction. In *3DPVT - 1st International Symposium on 3D Data Processing Visualization and Transmission*, pages 159–166, 2002.
- [HS07] Heiko Hirschmüller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.
- [HTMS82] Ernest L. Hall, James B K Tio, Charles a. McPherson, and Firooz a. Sadjadi. Measuring Curved Surfaces for Robot Vision. *Computer*, 15(12):42–54, 1982.
- [HW11] Steffen Herbort and Christian Wöhler. An introduction to image-based 3D surface reconstruction and a survey of photometric stereo methods. *3D Research*, 2(3):4, 2011.
- [HWW95] David H Hubel, Janice Wensveen, and Bruce Wick. *Eye, brain, and vision*. Scientific American Library New York, 1995.
- [HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [IK88] John Illingworth and Josef Kittler. A survey of the hough transform. *Computer vision, graphics, and image processing*, 44(1):87–116, 1988.
- [Int15a] Trust International. EXIS WEBCAM - BLACK/SILVER. <http://www.trust.com/pt/product/17003-exis-webcam-black-silver>, 2015. Ficheiro: 16006183.jpg; Consultado em: 15-09-2015.
- [Int15b] Trust International. EXIS WEBCAM - BLACK/SILVER. <http://www.trust.com/pt/product/17003-exis-webcam-black-silver>, 2015. Consultado em: 15-09-2015.
- [JAP99] Tony Jebara, Ali Azarbayejani, and Alex Pentland. 3D structure from 2D motion. *IEEE Signal Processing Magazine*, 16(3):66–84, 1999.
- [JSY03] Hailin Jin, Stefano Soatto, and Anthony J Yezzi. Multi-view stereo beyond lambert. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–171. IEEE, 2003.
- [JYS02] Hailin Jin, A.J. Yezzi, and S. Soatto. Variational multiframe stereo in the presence of specular reflections. In *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pages 626–630, 2002.
- [KKA00] Nahum Kiryati, Heikki Kälviäinen, and Satu Alaoutinen. Randomized or probabilistic Hough transform: Unified performance evaluation. *Pattern Recognition Letters*, 21(13-14):1157–1164, 2000.
- [KMR13] Bryan Klingner, David Martin, and James Roseborough. Street view motion-from-structure-from-motion. *Proceedings of the IEEE International Conference on Computer Vision*, pages 953–960, 2013.
- [Kum14] Praveen Kumar. *3D surface reconstruction and features extraction technique for underwater images*. PhD thesis, 2014.
- [KvD03] Jan J Koenderink and Andrea J van Doorn. Shape and shading. *The visual neurosciences*, pages 1090–1105, 2003.

- [KXO90] P Kultanen, L Xu, and E Oja. Randomized Hough transform (RHT). *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, i:631–635 vol.1, 1990.
- [Lar12] Ron Larson. *Precalculus: a concise course*. Cengage Learning, 2012.
- [LF06] Anthony Lobay and D. A. Forsyth. Shape from texture without boundaries. *International Journal of Computer Vision*, 67(1):71–91, 2006.
- [LM11] Erik G Learned-Miller. Introduction to computer vision, 2011.
- [LZWL04] Y Lu, J Z Zhang, Q M J Wu, and Z.-N. Li. A survey of motion parallax based {3-D} reconstruction algorithms. *Part C, Applications and Reviews*, 34(4):532–548, 2004.
- [M⁺67] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.
- [Mas09] Frank Masci. Line detection by hough transformation. http://web.ipac.caltech.edu/staff/fmasci/home/astro_refs/HoughTrans_lines_09.pdf, 2009. Consultado em: 10-10-2015.
- [MBB⁺10] ZM Moratto, MJ Broxton, RA Beyer, M Lundy, and K Husmann. Ames stereo pipeline, nasa’s open source automated stereogrammetry software. In *Lunar and Planetary Science Conference*, volume 41, page 2364, 2010.
- [McL98] Robert a. McLaughlin. Randomized Hough Transform: Improved ellipse detection with comparison. *Pattern Recognition Letters*, 19(3-4):299–305, 1998.
- [Mel05] Bob Mellish. Pinhole camera. <https://commons.wikimedia.org/wiki/File:Pinhole-camera.png>, 2005. Ficheiro: Pinhole-camera.png; Consultado em: 15-07-2015.
- [Mex15] Ardit Mexico. R-3d-2. <http://r-3d-2.com>, 2015. Consultado em: 01-10-2015.
- [Mey00] Carl D Meyer. *Matrix analysis and applied linear algebra*, volume 2. Siam, 2000.
- [MFC⁺02] Ryuji Matsuoka, Kiyonari Fukue, Kohei Cho, Haruhisa Shimoda, Yoshiaki Matsumae, Kenji Hongo, and Seiju Fujiwara. A study on calibration of digital camera. *Photogrammetric Computer Vision PCV*, 2., pages 1–5, 2002.
- [MGK98] J Matas, C Galambos, and J Kittler. Progressive Probabilistic Hough Transform. *British Machine Vision Conference*, 24(4):303–316, 1998.
- [MGK00] J. Matas, C. Galambos, and J. Kittler. Robust Detection of Lines Using the Progressive Probabilistic Hough Transform. *Computer Vision and Image Understanding*, 78:119–137, 2000.
- [MK04] Gerard Medioni and Sing Bing Kang. *Emerging topics in computer vision*. Prentice Hall PTR, 2004.
- [MRS⁺08] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.
- [MTJ12] Arnaud Meline, Jean Triboulet, and Bruno Jouvencel. Comparative study of two 3D reconstruction methods for underwater archaeology. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 740–745, oct 2012.
- [NA12] Mark S Nixon and Alberto S Aguado. *Feature extraction & image processing for computer vision*. Academic Press, 2012.

- [PM82] Tamar Peli and David Malah. A study of edge detection algorithms. *Computer Graphics and Image Processing*, 20(1):1–21, 1982.
- [PP11] Giuseppe Papari and Nicolai Petkov. Edge and line oriented contour detection: State of the art. *Image and Vision Computing*, 29(2-3):79–103, 2011.
- [Rad05] Johann Radon. 1.1 über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Classic papers in modern diagnostic radiology*, 5, 2005.
- [RC09] DP Robertson and R Cipolla. Practical image processing and computer vision, 2009.
- [REH06] F Remondino and S F El-Hakim. Image-Based 3D Modeling : A Review. *Photogrammetric Record*, 21(115):269–291, 2006.
- [Ren12] Eric Renner. *Pinhole photography: from historic technique to digital application*. CRC Press, 2012.
- [RRR09] Vilayanur S. Ramachandran and Diane Rogers-Ramachandran. Two Eyes, Two Views: Your Brain and Depth Perception. <http://www.scientificamerican.com/article/two-eyes-two-views/>, 2009. Consultado em: 10-10-2014.
- [SAB02] Joaquim Salvi, Xavier Armangué, and Joan Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7):1617–1635, 2002.
- [SCD⁺01] S M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Volume 1 CVPR06*, 1:519–528, 2001.
- [SCMS01] Greg Slabaugh, Bruce Culbertson, Tom Malzbender, and Ron Schafer. A Survey of Methos for Volumetric Scene Reconstruction from Photographs. *International Workshop on Volume Graphics*, 2001.
- [SD99] Steven M. Seitz and Charles R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.
- [See15] Seene. Seene. <http://seene.co>, 2015. Consultado em: 01-10-2015.
- [Sei99] S Seitz. An Overview of Passive Vision Techniques. *Proc. of SIGGRAPH*, pages 1–3, 1999.
- [SHB14] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.
- [Sig99] Sigurdur Helgason. *Radon Transform Second Edition*. 1999.
- [SKB13] Shubham Saini, Bhavesh Kasliwal, and Shraey Bhatia. Comparative Study Of Image Edge Detection Algorithms. 2013.
- [Sma14] SmartMobileVision. Scann3d. <http://scann3d.smartmobilevision.com/>, 2014. Consultado em: 01-10-2015.
- [SS01] Linda Shapiro and George C Stockman. Computer vision. 2001. ed: Prentice Hall, 2001.
- [SS02] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002.
- [SSPY10] Danail Stoyanov, Marco Visentini Scarzanella, Philip Pratt, and Guang-Zhong Yang. Real-time stereo reconstruction in robotically assisted minimally invasive surgery. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 13:275–82, 2010.

- [Ste91] RS Stephens. Probabilistic approach to the Hough transform. *Image and Vision Computing*, 9(1):66–71, 1991.
- [SZ97] Cordelia Schmid and Andrew Zisserman. Automatic line matching across views. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–6, 1997.
- [Sze10] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [Tsa87] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, aug 1987.
- [TSK] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Pearson Education India.
- [TV98] Emanuele Trucco and Alessandro Verri. *Introductory techniques for 3-D computer vision*, volume 201. Prentice Hall Englewood Cliffs, 1998.
- [Wei15] Eric W Weisstein. Parallel planes. <http://mathworld.wolfram.com/ParallelPlanes.html>, 2015. Consultado em: 12-11-2015.
- [XOK90] Lei Xu, Erkki Oja, and Pekka Kultanen. A new curve detection method: Randomized Hough transform (RHT). *Pattern Recognition Letters*, 11(5):331–338, 1990.
- [ZN94] Li Ze-Nian. Stereo correspondence based on line matching in Hough space using dynamic programming. *Systems, Man and Cybernetics, IEEE Transactions on*, 24(1):144–152, 1994.
- [ZT98] Djemel Ziou and Salvatore Tabbone. Edge detection techniques - An overview. *International Journal of Pattern Recognition and Image Analysis*, 8(4):1–41, 1998.



UNIVERSIDADE DE ÉVORA
INSTITUTO DE INVESTIGAÇÃO
E FORMAÇÃO AVANÇADA

Contactos:

Universidade de Évora

Instituto de Investigação e Formação Avançada — IIFA
Palácio do Vimioso | Largo Marquês de Marialva, Apart. 94

7002 - 554 Évora | Portugal

Tel: (+351) 266 706 581

Fax: (+351) 266 744 677

email: iifa@uevora.pt