FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

Sistema de Localização Remota de Robôs Móveis Baseado em Câmaras de Vigilância

Filipe Monteiro

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Orientador: Paulo José Cerqueira Gomes da Costa

Co-orientador: Héber Miguel Plácido Sobreira

Junho de 2011

Resumo

Ultimamente tem havido um crescente interesse e desenvolvimento nos robôs móveis de serviço. Estes são criados com o intuito de melhorar e simplificar a vida dos seres humanos, desde tarefas domésticas às de segurança e vigia.

Um dos principais temas relacionados com o desenvolvimento dos robôs móveis é a sua navegação. Para executar as tarefas autonomamente, o robô necessita de conhecer a sua posição e orientação. É com isto em mente que várias técnicas de localização têm vindo a ser desenvolvidas.

Neste trabalho foi criado um sistema de localização remota usando câmaras de vigilância. Foi usado um método de detecção de marcadores passivos com a forma triangular, baseado na detecção de cantos e orlas. Foi realizado também a calibração da câmara para poder obter correspondências entre os píxeis das imagens e as coordenadas do mundo. O sistema desenvolvido é capaz de obter a posição e orientação do robô móvel com precisão.

Esta dissertação insere-se no projecto ROBOT VIGILANTE, que tem como objectivo desenvolver um robot de serviço vigilante que funcionará como extensão aos operacionais humanos no terreno e tem como principais destinatários as grandes superfícies onde já se pressupõe existir um sistema de vídeo-vigilância.

Abstract

Lately there has been a growing interest and development in the mobile service robots. These are created in order to improve and simplify the lives of human beings, from housekeeping to security and vigilance tasks.

One of the main issues related to the development of mobile robots is navigation. To perform tasks autonomously, the robot needs to know its position and orientation. It is with this in mind that several localization techniques have been developed.

In this work it was created a system of remote localization using surveillance cameras. It was used a method for detecting passive markers with the triangular shape, based on the detection of corners and edges. It was also performed the calibration of the camera in order to obtain correspondences between the pixels of the images and the coordinates of the world. The developed system is able to get the position and orientation of the mobile robot with precision.

This dissertation is part of the ROBOT VIGILANTE project, which aims to develop a vigilant service robot who will work as an extension to the human operating in the field and is primarily aimed at the large stores where it is presumed there is a video surveillance system.

Agradecimentos

Agradeço aos meus orientadores, Paulo Costa e Héber Sobreira, por toda a disponibilidade e apoio prestado durante a execução deste trabalho.

Aos meus colegas, em especial a João Pinto, Rui Barbosa, André Moreira, Hugo Pinto e Luís Pinto pelos bons momentos vividos ao longo desta etapa académica.

Agradeço especialmente à minha família, amigos e namorada pelo apoio e motivação dada.

Filipe Monteiro

"Believe you can
and you're halfway there"

Theodore Roosevelt

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

DIND	Distributed Intelligent Networked Device
FEUP	Faculdade De Engenharia da Universidade do Porto
GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
iGPS	indoor Global Positioning System
LED	Light-Emitting Diode
MR	Mobile Robot
SLAM	Simultaneous Localization And Mapping
XML	Extensible Markup Language

Capítulo 1

Introdução

Neste capítulo será feita uma introdução da tese. Será descrito o enquadramento e motivação da proposta na secção 1.1, será feita uma caracterização dos problemas a tratar na secção 1.2 e o modo como o documento está estruturado na secção 1.3.

1.1 Enquadramento e Motivação

Tem-se vindo a notar um crescente interesse e consequente desenvolvimento e pesquisa na área de robótica nos últimos tempos. Para além dos robôs industriais, estão também em foco os robôs utilizados para entretenimento, como o AIBO ou o Mindstorms da Lego, e os robôs de serviço. Estes últimos têm como objectivo o de realizar actividades para o auxílio e bem-estar do ser humano. Tais actividades podem ir desde a execução de tarefas domésticas (como o Roomba por exemplo) até à realização de operações de vigilância e segurança.

Um dos principais obstáculos dos robôs móveis é o de navegação, mais concretamente o de conhecer a sua localização no ambiente onde se encontra. Tendo isto em conta, é necessário o desenvolvimento ou adaptação de novas técnicas que possam tornar possíveis dotar o robô do conhecimento da sua posição e orientação.

A presente proposta insere-se no projecto ROBOT VIGILANTE em desenvolvimento com a colaboração da FEUP. Este projecto tem o objectivo ambicioso de desenvolver um dos primeiros robôs vigilantes 'inteligente' no mundo, que será usado na área da segurança e que funcionará como extensão dos operacionais humanos no terreno. Tendo como principais destinatários as grandes superfícies (armazéns, centros comerciais, escritórios,...), com necessidades características, o robô representará um acréscimo de segurança na medida em que se caracteriza pela mobilidade, autonomia e pela capacidade de cooperar com outros robôs e com operacionais humanos. Uma solução atractiva ao nível da localização de robôs móveis seria utilizar as infra-estruturas já existentes onde o robô vai operar, mais concretamente o sistema de vídeo-vigilância existente. Tendo isto em conta, neste trabalho pretende-se implementar um sistema de localização remota de

robôs móveis baseado em câmaras de vigilância. Este sistema terá acesso à imagem captada pelas diferentes câmaras, e através de algoritmos de processamento de imagem, um robô será detectado e a sua posição é calculada.

1.2 Caracterização detalhada dos problemas a tratar

Os problemas a tratar durante este projecto podem ser divididos nos seguintes pontos:

- **Detecção e identificação de marcadores activos ou passivos presentes no robô através de processamento de imagens**

  O primeiro objectivo deste trabalho será o de localizar o robô nas imagens capturadas pela câmara. Se o robô estiver ao alcance da visão da câmara, este será apresentado nas imagens capturadas. O objectivo será conseguir detectar com certeza a localização do robô na imagem. para tal, é comum a utilização de marcadores passivos ou activos. A escolha deste marcador depende das condições do ambiente onde nos encontramos e do tempo de processamento desejado. O marcador vai ser conhecido e colocado no topo do robô. Através de algoritmos de processamento de imagem este marcador deverá ser detectado e corresponderá à localização do robô na imagem.

- **Calibração das câmaras**

  Uma imagem quando é capturada apresenta apenas 2 dimensões num plano projectado com coordenadas medidas por pixéis. Para a localização do robô, é necessário que se conheça as coordenadas reais da posição deste. Esta posição poderá ser encontrada realizando uma transformação do referencial da câmara para o referencial desejado do ambiente. Para realizar esta correspondência é necessário encontrar os parâmetros intrínsecos da câmara (distância focal, ponto central da imagem e possíveis distorções geradas pela lente da câmara) e os parâmetros extrínsecos que são compostos por uma rotação e translação do referencial. O cálculo destes parâmetros é realizado através de uma técnica denominada por *Calibração da câmara*. Torna-se então possível realizar uma correspondência entre a localização do robô na imagem e a localização do mesmo no mundo.

1.3 Estrutura da Dissertação

No capítulo 1 é realizada uma introdução da tese, é apresentado o seu enquadramento e detalhados os objectivos principais.

No capítulo 2, é descrito o estado da arte relacionado com os pontos principais do trabalho e são apresentados alguns projectos e trabalhos já realizados.

O capítulo 3 apresenta o procedimento efectuado para a detecção do marcador. É escolhido um marcador adequado, e descrito por passos, os métodos de processamento de imagem utilizados que levam à sua detecção.

No capítulo 4 é apresentada a calibração de câmara efectuada. São descritos os diversos passos efectuados assim como a utilidade da calibração no trabalho proposto.

No capítulo 5 são apresentados e comentados os resultados experimentais obtidos.

No capítulo 6 são apresentadas as conclusões finais do trabalho realizado, tais como, satisfação de objectivos e são apresentados possíveis melhoramentos em aplicações relacionadas no futuro.

Capítulo 2

Estado da Arte

Neste capítulo é realizado um estudo sobre alguns trabalhos e sistemas já criados que aprofundem os mesmos tópicos em que vou tocar nesta dissertação. São eles: Localização Remota, Detecção de um robô e calibração do referencial da câmara.

2.1 Introdução

Uma das características essenciais necessárias de saber para realizar a navegação de um robô é a sua localização. Existem já vários procedimentos que permitem determinar a posição e orientação de um robô, que podem ser divididos em métodos de auto-localização e de localização remota. A diferença entre estes é que nos segundos é necessário haver comunicação entre o robô e um sistema externo a este para troca de informação ou controlos. Como o robô a ser criado terá como principais destinatários as grandes superfícies, onde se pressupõe existir um sistema de câmaras de vigilância já implementado anteriormente, iremos apenas aprofundar os métodos de localização remota. Visto que este tipo de localização depende de um sistema que consiga detectar e identificar o robô, assim como determinar as suas coordenadas, serão também estudados vários métodos de detecção do robô através de processamento de imagem e algumas técnicas utilizadas para a calibração do referencial da câmara.

2.2 Localização Remota

Nesta secção serão estudados alguns sistemas já existentes que têm como objectivo fornecer a um robô as coordenadas da sua localização. Iremos aprofundar:

- O sistema iSpace em 2.2.1.
- A utilização de Pseudolites em 2.2.2.
- E o sistema iGPS em 2.2.3.

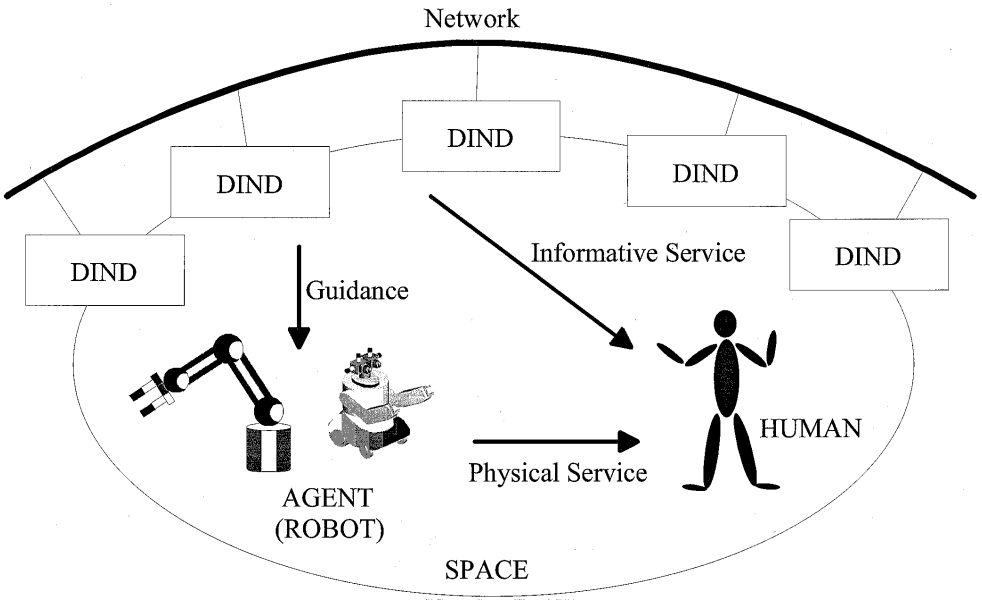

Figura 2.1: Arquitectura do iSpace. [2]

2.2.1 iSpace

Um sistema *"Intelligent Space"* é um espaço que possui uma rede composta por DINDs (*Distributed Intelligent Networked Device*). Estes recolhem informação do ambiente, que é processada e que pode ser consultada pelo MR (*Mobile Robot*) ou enviada pelo sistema para o MR em forma de instruções de controlo ou de sinais de aviso. [1] Portanto, o DIND é constituído por 3 elementos básicos: Sensores (monitorizar a dinâmica do ambiente), processador (processa a informação adquirida e toma decisões) e dispositivos de comunicação (troca de informação entre DINDs e MRs). [1]

C. Christo e C. Cardeira (2007) [1], afirmam que o crescimento de Web Services, abre-se um leque de novas oportunidades para a integração empresarial de diversos tipos de recursos, como os MRs.

Os Web Services são uma excelente solução para aplicações distribuidas. Baseiam-se em normas como o XML ou HTTP e tornam a comunicação possível mesmo que os serviços estejam a correr em diferentes sistemas operativos. [1]

No seu artigo, C. Christo e C. Cardeira (2007) [1], apresentam um exemplo prático que utiliza Web Services para comunicação (MR pede a actualização da sua posição) e iSpace para localização do MR (baseada em câmaras).

Já Drazen Brseie e Hideki Hashimoto (2009) [3], apresentam uma implementação de iSpace que não só localiza o MR como também gere os seus movimentos. Para o mapeamento do espaço e localização do MR utilizaram sensores Laser Range Finders como DINDs do iSpace. O algoritmo desta implementação está representado na Figura 2.2.

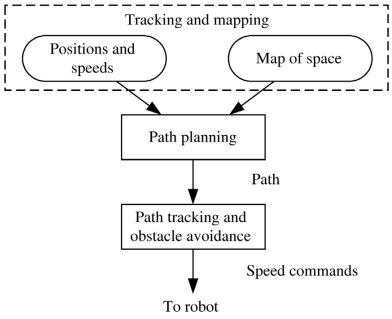

Figura 2.2: Algoritmo de Controlo. [3]

Eles sustentam que a utilização de sensores externos dispostos no ambiente, permite uma localização e mapeamento independentes (contrariamente, ao caso SLAM) e em tempo real. Quando comparado com as soluções que utilizam apenas, os sensores embutidos, o conceito iSpace alivia a necessidade de recorrer ao mapa do ambiente durante a estimação da localização, e possibilita a detecção de obstáculos, sem que o robô se encontre por perto.

Lee [2], apresenta um sistema iSpace numa fábrica, em que cada área é abrangida por um DIND diferente. Os DINDs são capazes de detectar humanos, MRs e obstáculos presentes no espaço, de maneira a conseguir exercer um controlo eficaz e sem colisões de cada MR. Dispõem também dos testes e resultados alcançados.

2.2.2 Pseudolites

Os pseudolites são transmissores colocados ao nível do solo e que são utilizados para complementar a navegação baseada na informação dos satélites (GPS). Os pseudolites sao particularmente utilizados no auxilio da navegação em ambientes internos, onde os sinais dos satélites GPS tem dificuldade em chegar. As pseudolites tem assim vindo a ganhar mais atenção nesta área de localização indoor, chegando a apresentar resultados com precisões ao nível do centímetro. [4] Podemos observar na Figura 2.3 a arquitectura de funcionamento dos pseudolites.

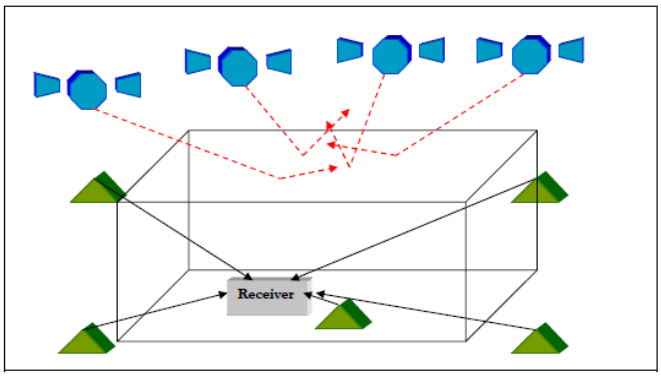

Figura 2.3: Arquitectura dos Pseudolites. [4]

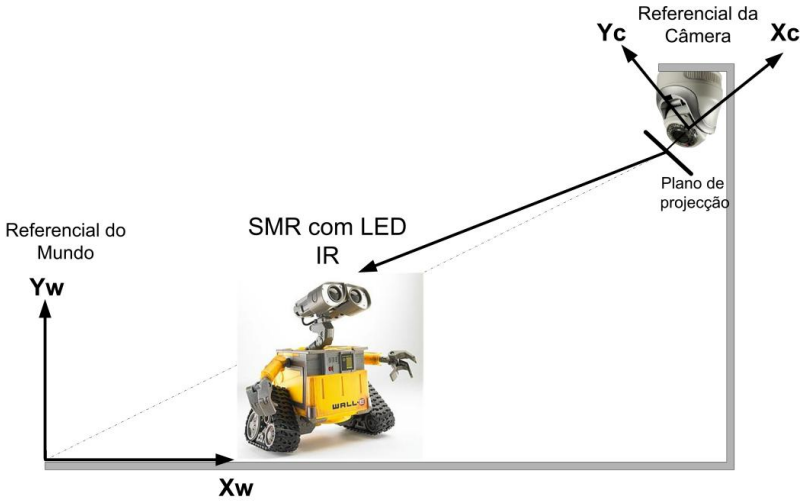

Figura 2.4: Arquitectura do iGPS.[7]

2.2.3 iGPS

Outro método de localização remota de um MR é o iGPS. Este tem a vantagem de apenas utilizar equipamento simples (ao contrário das pseudolites), muito utilizado na robótica como câmaras e marcadores LEDs.

Y.Hada e K.Takase (2001) [5], desenvolveram uma infra-estrutura de controlo e localização de robôs móveis em ambientes interiores, com recurso ao iGPS. Afirmam que o sistema é capaz de determinar a posição e orientação de múltiplos robôs, em simultâneo. O método iGPS analisa o ambiente e no caso de detectar um conjunto de três LEDs de infravermelhos não colineares acoplados no MR, determina a sua localização global, assim como a sua orientação, devido à não colinearidade dos LEDs. [5]

Y. Hada et al. (2003)[6], sugeriu um controlo de trajectória utilizando iGPS e a Odometria, com vista em minimizar o erro gerado pelo atraso entre a aquisição de imagem da câmara e a recepção da informação da posição por parte do MR. Cada imagem capturada pelo iGPS tem um timestamp associado, possibilitando o cálculo do tempo que demorou todo o processo (desde a captura da imagem à recepção da posição por parte do robô). De maneira a calcular este tempo com precisão, o MR tem de estar sincronizado com o iGPS.[6]

2.3 Detecção de um MR

2.3.1 Detecção baseada em Marcadores Activos

Yoshi Hada e Kunikatsu (2001)[5], montam em cada MR um conjunto de 3 LED's de infravermelhos (dispostos em triângulo), que ao movimentar-se, estes serão captados por um conjunto de câmaras instaladas no ambiente. De seguida, o sistema realiza um processamento de imagens,

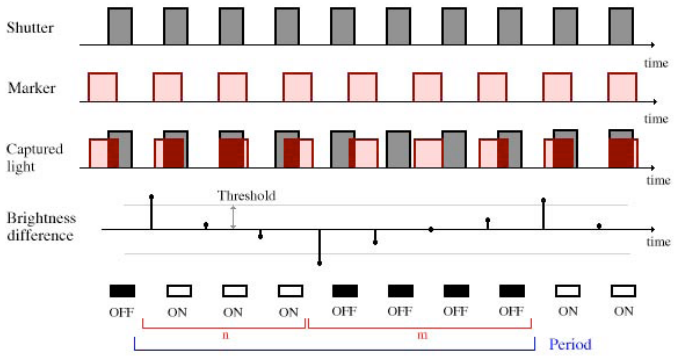

Figura 2.5: Luminosidade Captada e Padrão Gerado.[8]

de forma a retirar as coordenadas de cada LED e obter a pose do robô (LED's colocados não colinearmente de modo a poder obter a orientação do MR).

Ricardo Cassinis et al. (2005)[8], sugeriram o desenvolvimento de um projecto que deveria detectar um MR quer em ambientes internos, quer em exteriores. Após descartarem alguns tipos de marcadores, optaram por utilizar um marcador que emitisse flashes a uma frequência fixa, cujo período fosse muito parecido com o do tempo de aquisição de imagens por parte da câmara. O resultado foi que nas imagens captadas pela camara, o marcador apresentava uma luminosidade que oscilava periodicamente seguindo um formato de uma onda quasi-triangular cuja frequência é a diferença entre as frequências do obturador da câmara e dos flashes do marcador.

A luminosidade do marcador nas imagens obtidas pela câmara seguia assim um padrão que pode ser facilmente observado (ON-ON-ON-OFF-OFF-OFF-OFF). De modo a detectar o marcador (objectivo inicial), bastava então realizar um processamento de imagens que testava a existência desse padrão em todos os pixéis de imagens seguidas. Os pixéis que mais se aproximassem dessa sequência de intensidades seriam os pixéis onde o MR estaria. O marcador utilizado foi um painel preto com um circulo de LED's de alta luminosidade.[8]
O marcador pode ser observado na Figura 2.6 (esquerda).

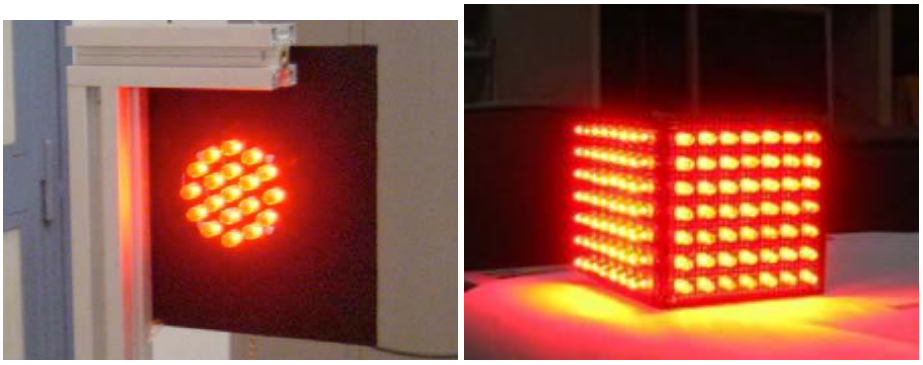

Figura 2.6: Marcador utilizado em [8](esquerda) e em [9](direita)

Figura 2.7: Marcador passivo utilizado em [11]

DongJu Kim et al. (2010)[9], utilizam um marcador activo em forma de cubo, com as 4 faces laterais cobertas com 49 LED's de alto brilho cada uma. Os LED's dispostos em cada face lateral possuem diferentes frequências, de maneira a ser possível distingui-las com base no padrão de luminosidade em sequências de imagens, e com base nisto, conhecer a orientação do MR. Os LED's têm também um espaçamento fixo entre eles, para cálculo de distâncias entre MRs. (ver Figura 2.6 (direita)).

2.3.2 Detecção baseada em Marcadores Passivos

Owen, Xiao e Middlia (2002)[10] especificaram um conjunto de critérios que definem um bom marcador passivo. Estes são:

- Um marcador ideal deve suportar a determinação inequívoca da posição e orientação relativamente a uma câmara calibrada.
- O marcador não deve favorecer algumas orientações.
- O marcador deve pertencer a um conjunto de imagens que são improváveis de se confundir, tal que, um grande espaço ou um conjunto de objectos possam ser marcados distintivamente.
- O marcador deve ser fácil de localizar e identificar usando algoritmos rápidos e simples.
- O marcador deve funcionar sobre uma elevada gama de intervalo de captura da câmara.

Alan Mutka et al. (2008)[11] propuseram uma solução em que a localização do MR é baseada em marcadores passivos dispostos no tecto. O MR possui uma camara na parte superior a apontar para cima, e vai conhecendo a sua posição e orientação com a detecção desses marcadores através dum processamento de imagem.

Alan Mutka et al. (2008)[11] defende que os marcadores circulares permitem uma localização mais robusta que os de forma quadrada. Além disso, os algoritmos de processamento de imagem

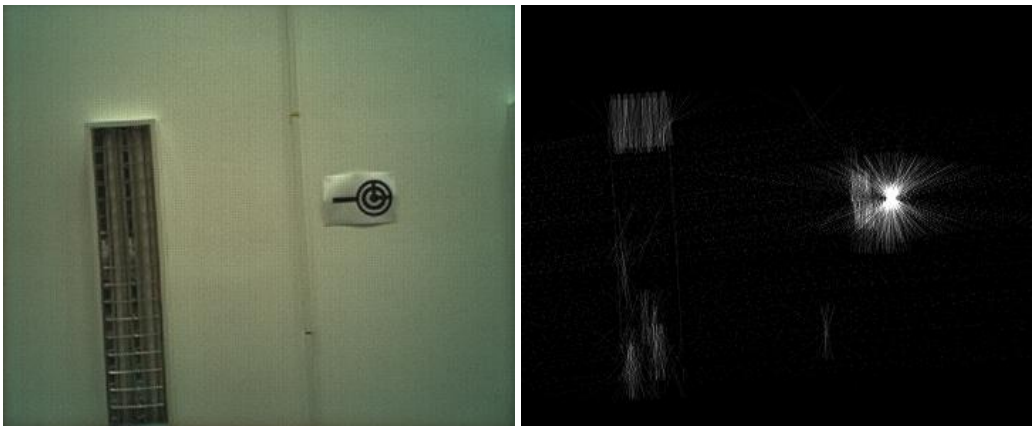

Figura 2.8: Exemplo da aplicação da transformação de Hough no marcador produzido em [11]

que lidam com marcadores circulares são mais simples que os que lidam com marcadores quadrados.

Os marcadores utilizados em [11] contém 2 circulos, um dentro do outro, de maneira a melhorar a detecção do marcador. Contém também uma linha para fora do círculo exterior, de maneira a ser possível descobrir a orientação do MR. Este marcador pode ser observado na Figura 2.7.

O algoritmo de processamento utilizado foi a transformação de Hough. Este método é utilizado para isolar componentes de uma determinada forma dentro da imagem. É muito usado para detecção de curvas, como linhas, circulos, elipses etc. Então Alan Mutka et al. (2008)[11] utilizaram este método para a detecção de círculos. Um exemplo da aplicação deste algoritmo pode ser observado na Figura 2.8.

Vários autores ([10], [12], [13], [14]) defendem o uso de um marcador quadrado com borda preta e interior branco, ou com algum código de identificação. As margens rectas do quadrado podem ser usadas para calcular as linhas dos lados do quadrado permitindo encontrar os cantos com uma grande precisão.[10]
Alguns exemplos destes marcadores podem ser observados na Figura 2.9

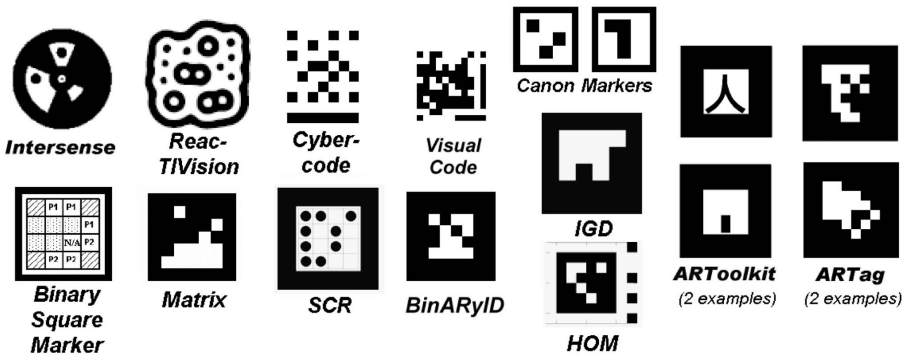

Figura 2.9: Vários exemplos de marcadores utilizados. Retirado de [13]

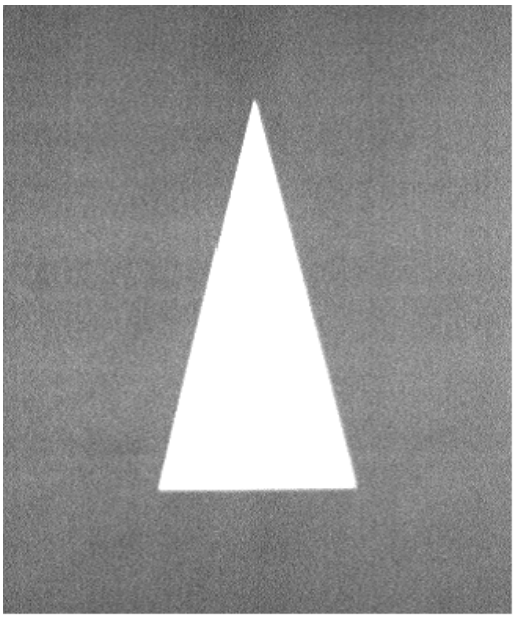

Figura 2.10: Marcador utilizado por David Lima [15]

Já David Lima (2010)[15] optou por escolher um marcador em forma de triângulo isósceles (ver Figura 2.10) colocado no pavimento, o que permite definir a posição cartesiana x, y do robô, bem como a sua orientação.

2.4 Calibração do Referencial da Câmara

A calibração da câmara é uma técnica muito importante e necessária em computação visual, especialmente na reconstrução 3D (extrair informação métrica 3D a partir de imagens 2D) [16], mas também para corrigir alguns problemas da imagem capturada, como a sua distorção.

Para realizar uma correcta calibração, é necessário calcular os parâmetros externos (posição e orientação relativamente a um sistema de coordenadas do mundo e os parâmetros intrínsecos da câmara (centro da imagem, distância focal e os coeficientes de distorção).[17]

De acordo com Berthold Horn (2000)[18], para calibrar uma câmara é necessário capturar uma imagem de um objecto de geometria conhecida. Em seguida são feitas correspondências entre pontos estratégicos do objecto e as suas imagens (fazer corresponder as coordenadas no mundo com as coordenadas na imagem). Estas correspondências formam a informação base utilizada para realizar a calibração da câmara.

Yoshi Hada e Kunikatsu (2001)[5], para realizar a calibração da câmara realizaram o seguinte procedimento:

- determinaram as coordenadas de localização do centro da câmara no sistema de coordenadas do “mundo”;

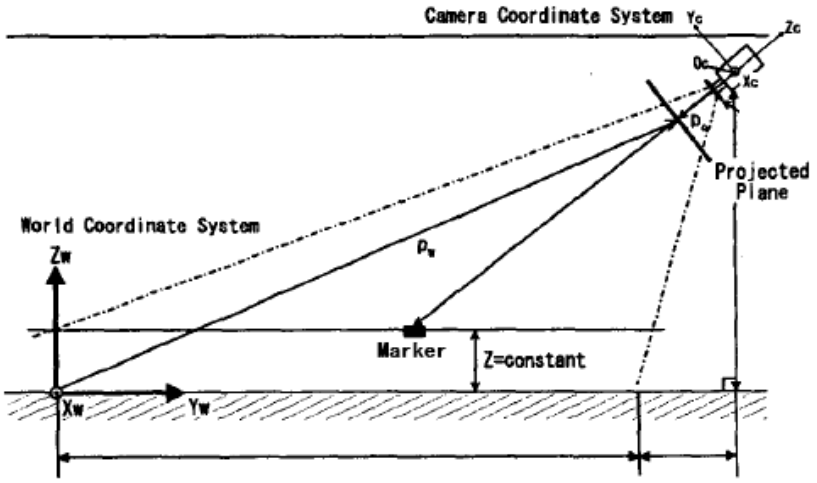

Figura 2.11: Representação do problema. Retirado de [16].

- determinaram uma matriz de transformação afim (A) que transforma a imagem óptica projectada num plano (2D), numa mesma imagem mas com o sistema de coordenadas da câmara (3D).
- determinaram uma matriz rotação R, que transforma o sistema de coordenadas do mundo no sistema de coordenadas da câmara.

Na Figura 2.11 podemos observar uma representação do problema.

X. Meng and Z. Hu (2003)[19], aplicam um padrão de calibração que consiste numa circunferência com um conjunto de linhas que se cruzam no centro do círculo. Y. Wu et al. (2004)[20] apresentam um método de calibração a partir da invariância quase-afim de dois círculos paralelos. Já Zijian Zhao et al. (2008)[16], propõem uma calibração baseada em apenas 3 pontos não colineares.

Vários autores ([21], [22], [23]), utilizam um tabuleiro de xadrez para a calibração. Com o auxílio de varias fotos do tabuleiro em posições e orientações diferentes, são extraídos todos os cantos formados no tabuleiro, e sabendo as suas correspondentes coordenadas em 3D, ou a largura de cada quadrado do xadrez, torna-se possível obter os parâmetros necessários à calibração da câmara. Existem já várias aplicações que utilizam este procedimento, entre eles destacam-se a Camera Caibration toolbox da Matlab e a biblioteca de funções relacionadas com computação visual em C, o OpenCV.

Capítulo 3

Detecção do Robô

Neste capítulo vai ser descrito o procedimento realizado na detecção do robô móvel. De maneira a detectar o robô através de imagens de uma câmara, é necessário que este possua um marcador. Este marcador terá de ser criado de maneira a que a sua identificação possa ser o mais exacta possível. Desta forma será apresentado na secção 3.1 o formato e características do marcador escolhido. Na secção 3.2 é descrito o algoritmo de detecção do marcador escolhido e na secção 3.3 serão apresentados alguns procedimentos que melhoraram o desempenho do programa desenvolvido, tanto na detecção do marcador como na diminuição do tempo de processamento.

3.1 Marcador escolhido

Existem dois grandes tipos de marcadores, os activos e os passivos. Os marcadores activos são distinguidos dos passivos pela necessidade de alimentação eléctrica. Os primeiros são, normalmente, constituídos por um ou mais LED's, enquanto que os passivos, são formados por combinações de cores e/ou formas geométricas conhecidas.

A escolha do marcador depende do ambiente em que este será utilizado e do nível de complexidade do processamento desejado. Como este trabalho destina-se a grandes superfícies, o marcador utilizado será passivo. Isto permite que a detecção do robô seja mais robusta, por exemplo, não é tão afectada com uma possível iluminação não uniforme.

Owen, Xiao e Middlia (2002)[10] especificaram um conjunto de critérios que definem um bom marcador passivo. Estes são:

- Um marcador ideal deve suportar a determinação inequívoca da posição e orientação relativamente a uma câmara calibrada.
- O marcador não deve favorecer algumas orientações.
- O marcador deve pertencer a um conjunto de imagens que são improváveis de se confundir, tal que, um grande espaço ou um conjunto de objectos possam ser marcados distintamente.

- O marcador deve ser fácil de localizar e identificar usando algoritmos rápidos e simples.
- O marcador deve funcionar sobre uma elevada gama de intervalo de captura da câmara.

Tendo em conta este conjunto de critérios, o marcador escolhido foi um triângulo isósceles, de interior branco e exterior preto. Pode ser observado na figura 3.1 o marcador utilizado.

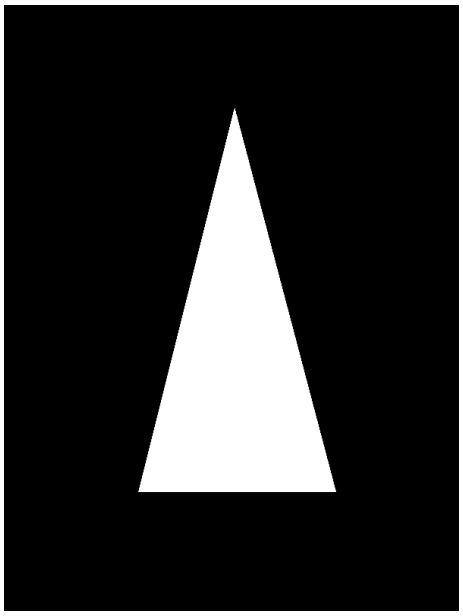

Figura 3.1: Marcador escolhido

Foi escolhido um triângulo, porque é uma forma geométrica conhecida e de relativa facilidade de identificação usando simples algoritmos. Este triângulo apresenta o lado mais pequeno com metade do comprimento dos lados maiores. Esta propriedade torna possível a determinação da orientação do marcador, considerando que a orientação do robô móvel é a mesma da do eixo de simetria do triângulo.

As cores utilizadas foram apenas o preto e o branco, porque a fronteira entre branco e preto forma um ponto de máxima variação de luminosidade, permitindo uma melhor distinção dos pontos de borda do triângulo. A não utilização de cor deve-se também ao facto de que uma cor sujeita a diferentes níveis de iluminação apresenta uma gama de valores possíveis muito alargada, o que pode dificultar a detecção do marcador ou gerar a detecção de falsos marcadores.

3.2 Detecção do Marcador

Visto que o marcador tem uma forma geométrica triangular, o procedimento natural é tentar encontrar três rectas que se intersectem e formem um triângulo. Para tal, numa primeira abordagem foi utilizada a transformada de Hough. Esta técnica é muito usada em aplicações de processamento de imagem, e tem como objectivo detectar *formas geométricas* simples, como rectas, círculos ou elipses. Esta abordagem foi rapidamente rejeitada, visto que, apesar do enorme gasto

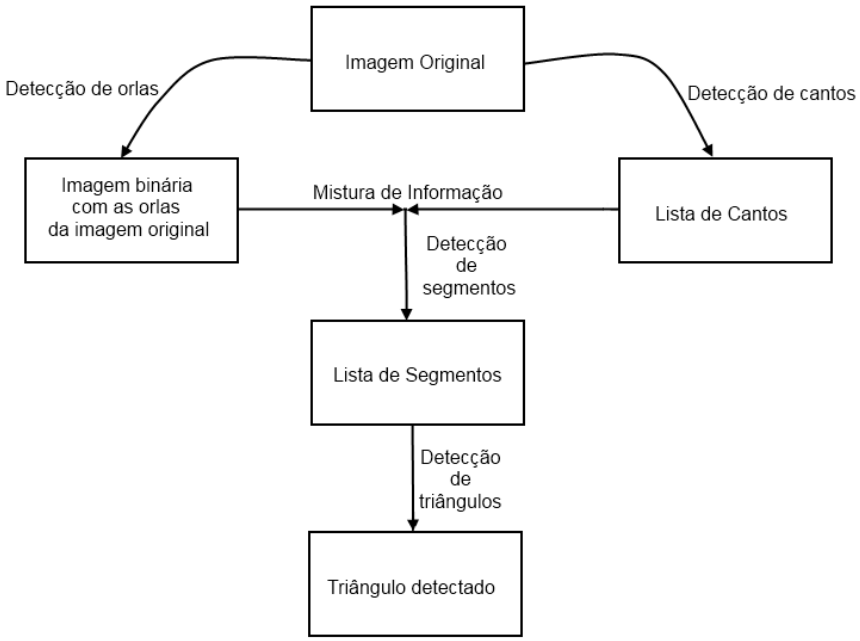

Figura 3.2: Algoritmo de detecção do marcador utilizado

de processamento, ela também não apresenta os melhores resultados em ambientes em que seja necessário detectar o marcador quer seja visível em grandes ou pequenas dimensões, ou seja, quer este se encontre próximo ou afastado da câmara. Este facto pode ser entendido, tendo em conta que a transformada de Hough funciona por um método de votação, ou seja, é percorrida a imagem inteira, e para cada píxel que seja considerado ponto de orla, todas as rectas que passem por este acumulam um valor de voto. Ou seja, uma recta é avaliada pelo número de pontos de orla que possuir no seu caminho. Se o marcador estiver muito afastado da câmara, as rectas correspondentes aos lados deste que desejamos encontrar, vão ser consideradas muito *fracas* e pode tornar-se difícil a localização do marcador.

A segunda abordagem consiste num conjunto de passos bem definidos de processamento de imagem, que permitem avaliar a possível existência de um marcador na imagem capturada pela câmara. Esta abordagem é descrita nesta secção, começando com a detecção de orlas (ver subsecção 3.2.1) e cantos (ver subsecção 3.2.2) na imagem, seguido da detecção de segmentos de recta (ver subsecção 3.2.3) que juntos poderão formar um triângulo (ver subsecção 3.2.4). Serão também apresentadas algumas técnicas de eliminação de falsos positivos em (ver subsecção 3.2.5). O algoritmo de detecção do marcador pode ser observado na figura 3.2.

3.2.1 Detecção de orlas

Uma orla pode ser definida como o limite exterior de um objecto. Em imagens digitais, uma orla pode ser visto como uma variação local da intensidade de cor.

A detecção de orlas é muito usada em aplicações de visão computacional. O seu objectivo é distinguir objectos na imagem e identificar as suas formas. É identificado um local de orla quando ocorre uma transição de intensidade sobre um conjunto de pixeis. Um ponto de borda perfeito é quando existe uma transição do preto para branco em apenas um píxel. [24]

Na figura 3.3 pode ser observado essa situação, bem como outra transição mas sobre mais que um píxel.

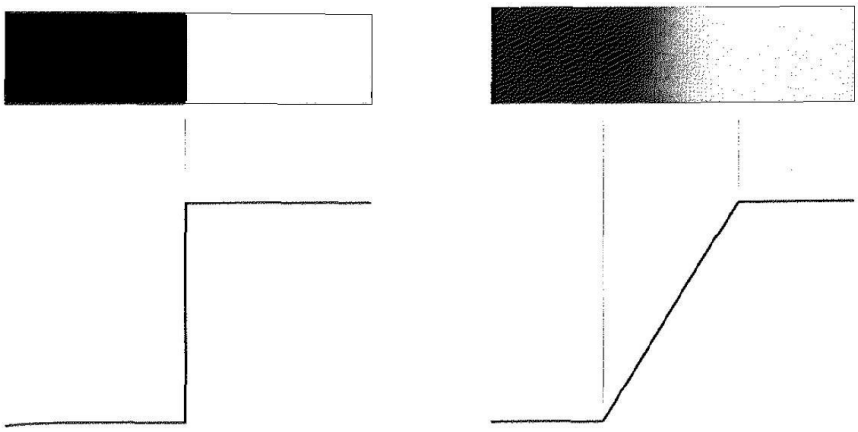

Figura 3.3: Transição em um píxel e transição sobre mais que um píxel. Retirado de [24]

Como foi referido na secção 3.1 o marcador será um triângulo com interior branco e borda preta, de forma a criar um contraste máximo nas fronteiras do marcador. Assim, os lados do triângulo poderão ser detectados através de um processamento de imagem que detecte orlas.

Foram testados dois operadores locais que aproximam o gradiente da imagem: Prewitt e Sobel. Ambos possuem duas máscaras para filtrar a imagem, uma que aproxima a derivada na direcção X e outra para a direcção Y.

Nas equações 3.2.1 podem ser observadas as máscaras utilizadas pelos filtros Prewitt e Sobel em ambas as direcções.

$$
\begin{aligned}
\textit{Prewitt} \quad \rightarrow \delta x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \delta y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \\
\textit{Sobel} \quad \rightarrow \delta x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \delta y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{3.1}
$$

A aplicação destes métodos é realizada através da convolução das máscaras com a imagem. Isto é, a imagem será toda percorrida e centrando a máscara em cada píxel, o valor da intensidade deste será substituído pela soma da multiplicação de cada valor da máscara pelos valores de intensidades dos correspondentes píxeis vizinhos. (Ver anexo A)

Após a aplicação completa do filtro, ficaremos com as componentes G_x e G_y distintas. A magnitude e orientação finais do gradiente podem ser obtidas da seguinte maneira:

$$|G| \approx \sqrt{G_x^2 + G_y^2}$$

$$\left|\theta_{grad}\right| \approx arctan2(\frac{G_y}{G_x}) \tag{3.2}$$

A magnitude do gradiente em cada píxel representa neste momento uma imagem de cinzentos em que os pontos de orla apresentam uma cor próxima do branco e os pontos e áreas uniformes apresentam uma cor próxima do preto, chamaremos de ImgEdges. Necessitamos agora de escolher quais os píxeis que vão ser usados para detecção dos segmentos, para tal, realizou-se uma binarização da imagem com um threshold adequado. Considerou-se que o nível de threshold deveria variar conforme a luminosidade da imagem captada, portanto calculou-se a média do brilho da imagem e foi binarizada com um limiar de *mediaImg+100*. A detecção de orlas pode então ser efectuado num ambiente muito ou pouco iluminado. A operação de binarização é realizada da seguinte maneira:

$$ImgEdges[u,v] = \begin{cases} 0 & \text{se } ImgEdges[u,v] < threshold \\ 1 & \text{se } ImgEdges[u,v] \geq threshold \end{cases}$$

Em que ImgEdges agora apresenta uma imagem binarizada com os pontos de orla a apresentarem valor 1 (branco) e tudo o resto 0 (preto). O threshold neste caso vale então o limiar escolhido de *médiaImg+100*. O resultado e comparação da aplicação dos filtros e da binarização pode ser observado na figura 3.4.

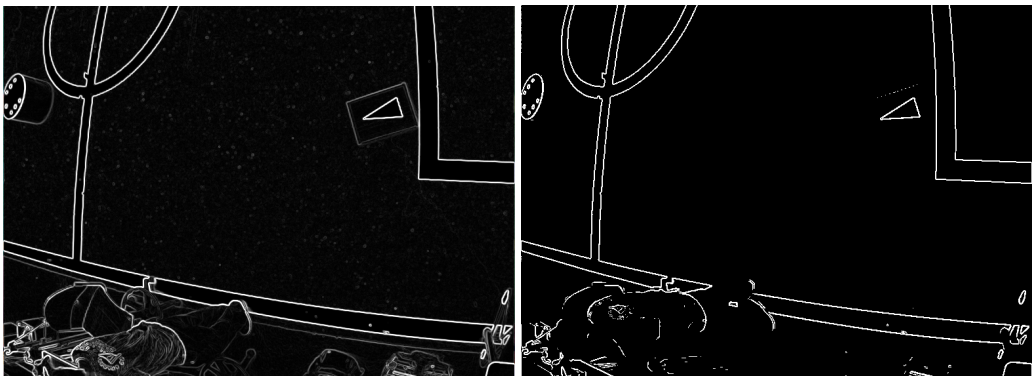

Figura 3.4: Resultado da aplicação do filtro sobel (esquerda) e a sua binarização (direita)

A única diferença entre o Sobel e o Prewitt é no peso que atribuem à linha/coluna do meio no filtro vertical ou horizontal respectivamente. Sobel utiliza um peso de 2/-2, enquanto que o Prewitt usa 1/-1. Isto resulta numa diferença de suavização visto que no Sobel é dada mais importância ao ponto central. [24]

3.2.2 Detecção de cantos

Um canto é um ponto onde duas linhas de inclinações diferentes se cruzam e formam um ângulo entre as duas. É um ponto facilmente reconhecível à vista humana, e importante para computação visual, pois também se revela relativamente fácil de detectar, sendo por isso a detecção de cantos uma técnica tão utilizada em trabalhos nesta área. Tem um papel muito importante nas aplicações relacionadas com reconhecimento facial ou de objectos, correspondência de imagens, alinhamento de imagens, seguimento de movimentos, realidade aumentada entre outras.

Um canto é um ponto na imagem que apresenta uma grande variação de intensidade em ambas as direcções X e Y, como pode ser observado na figura 3.5. [25]

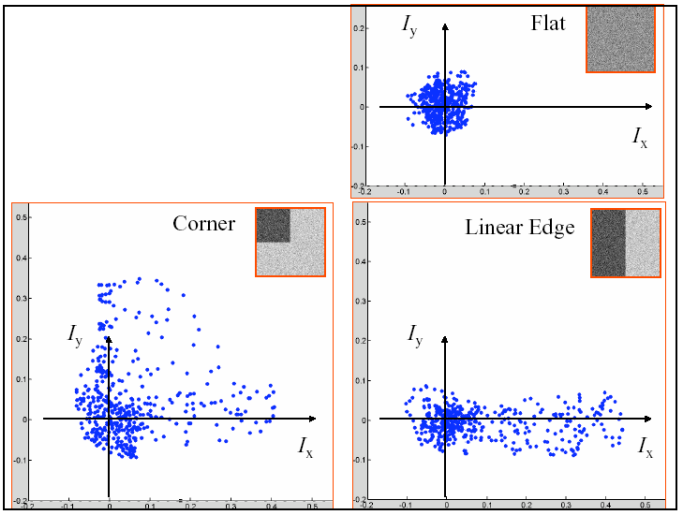

Figura 3.5: Variação de intensidades nos diferentes tipos de pontos. Retirado de [26]

Nas figuras 3.6 e 3.5 é possível observar as diferentes características apresentadas por um canto, um ponto numa orla e uma região uniforme. Colocando uma janela em cima de um ponto de orla, e movendo-a nas várias direcções, verifica-se que ela não apresenta variações de intensidade ao longo da direcção da orla. Enquanto que num canto, movendo a janela, obtém-se grandes variações de intensidade em qualquer direcção. [25]

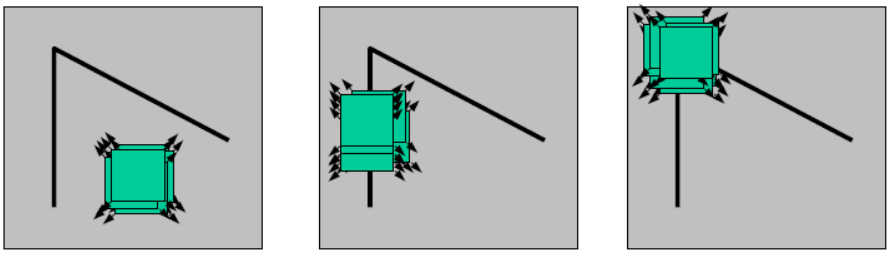

Figura 3.6: Caracterização de um canto

Existem várias técnicas de detecção de cantos, neste trabalho irá ser estudado e aplicado o algoritmo de Harris. Este baseia-se na detecção de pontos que apresentem as características já

referidas comuns nos cantos.

A variação infinitesimal $(\delta x, \delta y)$ de intensidade num ponto $(x,\ y)$ em qualquer direcção pode ser definida como:

$$c_(x,y;\delta x,\delta y) = \sum_{u,v} w_{u,v}[I(u+\delta x, v+\delta y) - I(u,v)]^2 \tag{3.3}$$

Em que $w_{u,v}$ corresponde a um ponto de uma janela centrada em (x,y) e que representa uma suavização gaussiana. $w_{u,v} = e^{\frac{-(u-x)^2-(v-y)^2}{2\sigma^2}}$

Como δx e δy são pequenos (variação infinitesimal), pelo teorema de Taylor:

$$I(u+\delta x, v+\delta y) \approx I(u,v) + I_x\delta x + I_y\delta y \tag{3.4}$$

Em que $I_x = \frac{\partial I}{\partial x}$ e $I_y = \frac{\partial I}{\partial y}$ são as derivadas parciais da intensidade nas direcções X e Y, respectivamente. Substituindo a equação 3.4 na equação 3.3, obtemos:

$$\begin{aligned} c_(x,y;\delta x,\delta y) &\approx \sum_{u,v} w_{u,v}[I(u,v) + I_x\delta x + I_y\delta y - I(u,v)]^2 \\ &= \sum_{u,v} w_{u,v}[I_x\delta x + I_y\delta y]^2 \\ &= \begin{bmatrix} \delta x & \delta y \end{bmatrix} C_{struct}(x,y) \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} \end{aligned} \tag{3.5}$$

$$C_{struct}(x,y) = \sum_{u,v} w_{u,v} \begin{bmatrix} I_x(x,y)^2 & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y(x,y)^2 \end{bmatrix} \tag{3.6}$$

[27]

Neste momento, possuímos então a matriz C_{struct} que caracteriza a variação da intensidade num certo ponto numa certa direcção. Definiu-se agora $\lambda 1$ e $\lambda 2$ como os valores próprios da matriz. Analisando estes valores, pode ser retirada informação acerca da natureza do ponto em questão, como pode ser observado na figura 3.7.

Existem então três situações possíveis:

- se $\lambda 1 \approx 0$ e $\lambda 2 \approx 0$, então o ponto não tem interesse;
- se um dos valores próprios tiver um valor grande e o outro for ≈ 0 então estamos na presença de uma orla;
- se ambos os valores próprios apresentarem valores suficientemente grandes, então um canto foi encontrado.

O método de Harris utiliza uma definição de força de um canto, que é descrita por:

$$H(x,y) = detC_{struct} - \alpha(traceC_{struct})^2 \tag{3.7}$$

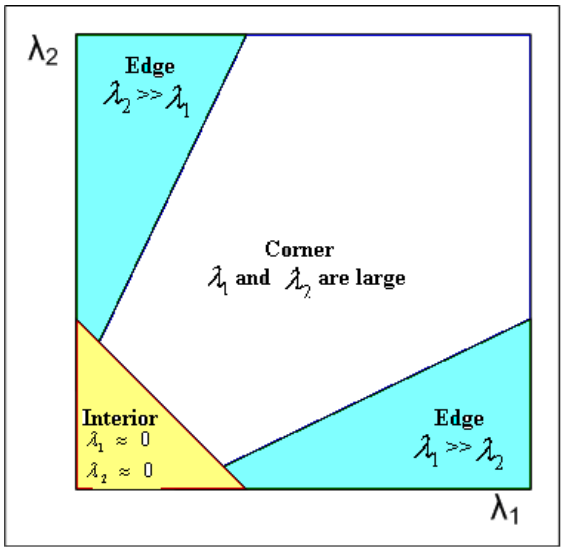

Figura 3.7: Classificação do ponto. Retirado de [28]

Em que α é um parâmetro que pode variar entre os 0.04 e 0.25, e depende da sensibilidade com que desejamos que o nosso detector de cantos funcione. O canto é então detectado quando a sua *força* (ver equação 3.7) é maior que um valor H_{thr}. Este valor também é variável, mas normalmente fixa-se este e varia-se apenas α. [29]

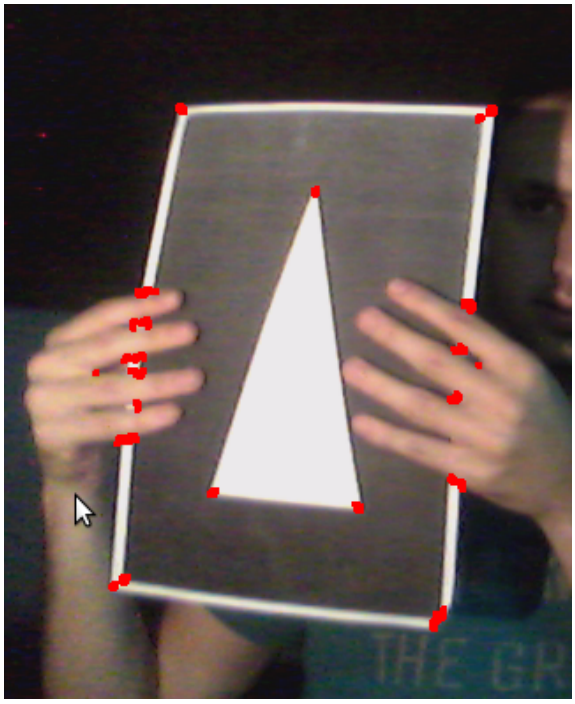

Figura 3.8: Fenómeno de *cluster* de cantos

Como os píxeis à volta de um canto, apresentam todos variações de intensidade parecidas a este, muitos também são detectados e admitidos pelo filtro do threshold de força do canto,

formando um aglomerado de "cantos"detectados em redor do canto original. Na figura 3.8 tal facto pode ser observado, estando os píxeis admitidos como cantos pintados a cor vermelha.

Surge então a necessidade de avaliar a vizinhança de cada canto, e apenas manter o mais forte. Desta maneira, apenas o canto original em cada aglomerado será guardado.

Para realizar este filtro, todos os cantos detectados são colocados numa lista, e ordenados por ordem decrescente de *força*. Essa lista é percorrida e cada canto é testado se pertence à *vizinhança*[1] dos cantos anteriores (mais fortes), se pertencer, é eliminado da lista.

No fim deste procedimento, ficamos com uma lista de pontos que representam os cantos da imagem (*ListCorners*).

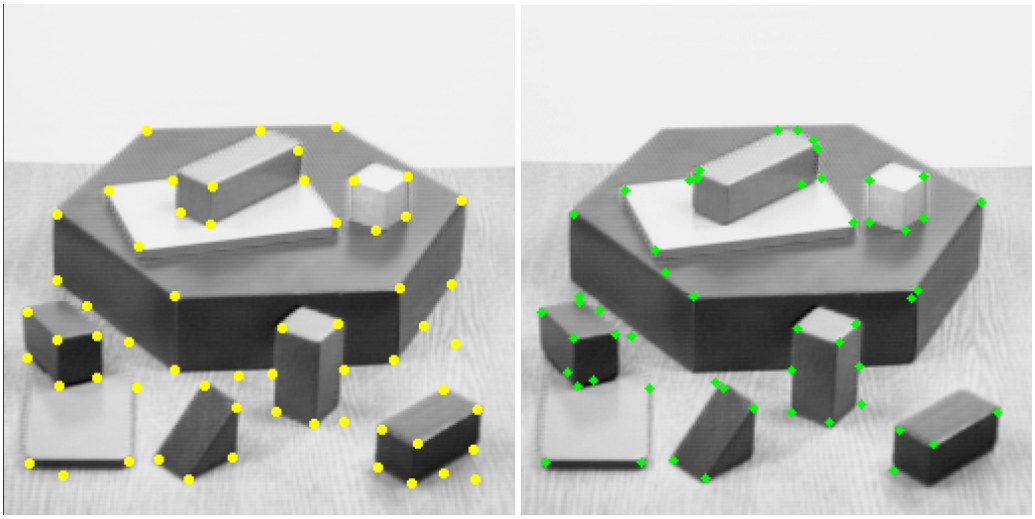

Figura 3.9: Comparação entre KLT(à esquerda) e Harris(à direita). Retirado de [29]

Existem vários outros métodos de detecção de cantos, que são apenas pequenas modificações do algoritmo de Harris. Neste trabalho, foi utilizado o algoritmo KLT, que é uma interpretação diferente do algoritmo original de Harris. A diferença entre estes é q o KLT não utiliza a mesma definição de *força de um canto* que é utilizada pelo Harris. Como um canto é identificado por ter os dois valores próprios suficientemente grandes (ver figura 3.7), o algoritmo KLT, apenas coloca um threshold em $\lambda 2$ ($\lambda 1 \geq \lambda 2 \geq 0$). Se o menor valor próprio for superior a uma constante definida, então esse ponto pode ser considerado um canto. A partir daqui o procedimento é o mesmo, estes cantos serão colocados numa lista e ordenados por ordem decrescente de $\lambda 2$. Em seguida é percorrida a lista de cima a baixo, e são eliminados os cantos que pertençam à vizinhança de um canto mais forte. Na figura 3.10, pode ser observada a aplicação do algoritmo KLT no programa desenvolvido.

[1] Neste trabalho foi considerado vizinhança um conjunto de pontos que distam menos de 10 pixéis do central

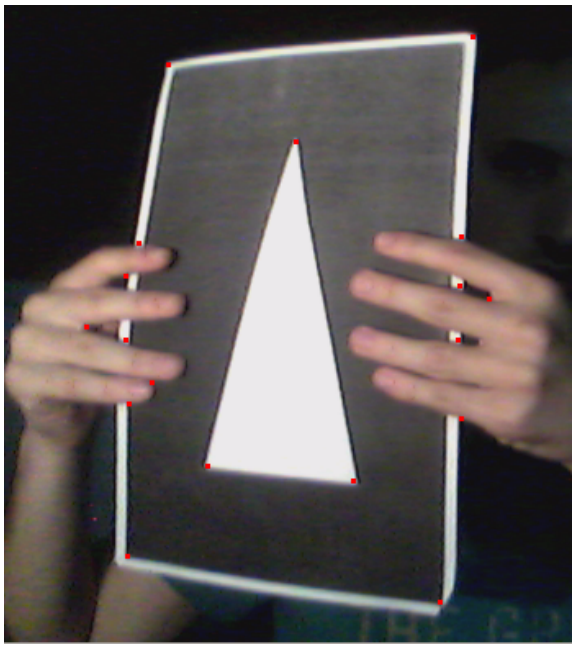

Figura 3.10: Aplicação do algoritmo de KLT. Cantos estão marcados a vermelho.

3.2.3 Detecção de segmentos

Um segmento de recta é descrito como um pedaço de uma recta limitado por dois pontos. Posto isto, pode ser verificado que um triângulo é uma forma geométrica que possui três segmentos de recta que são unidos pelos três cantos do triângulo.

Neste momento, já possuímos duas imagens que nos vão auxiliar na detecção de segmentos:

- ImgEdges, que contém informação sobre as orlas da imagem principal, e
- ListCorners, que contém em ordem decrescente de força, todos os cantos que podem ser utilizados na procura do triângulo, bem como a sua localização na imagem.

Para encontrar esses segmentos de recta é verificado se existem conjuntos de dois cantos da ListCorners que são unidos na ImgEdges por pontos de orla. Para isto, partindo dum píxel que represente o primeiro canto, e seguindo na direcção da recta até ao segundo canto, é verificado se pelo *caminho*[2] vão ser encontrados pontos de orla da ImgEdges.

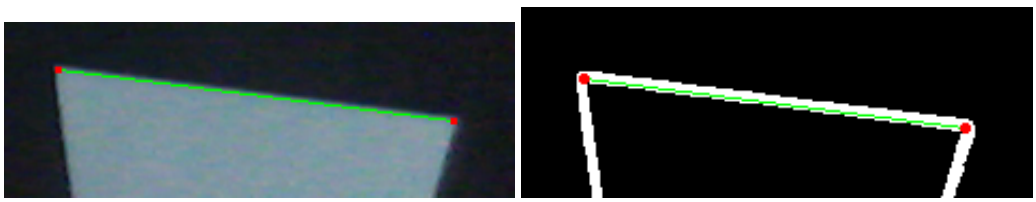

Figura 3.11: Representação de detecção de segmento

[2]Por caminho é considerado o píxel atravessado pelo segmento de recta mais um píxel de margem para cada lado

O número total de píxeis no caminho entre os dois cantos é denominado de *TotPix*, enquanto que o número de píxeis de orla entre os dois cantos chamaremos de *EdgePix*. Definindo como *força de um segmento* a razão entre os píxeis de orla encontrados e o número total de píxeis do segmento ($\frac{TorPix}{EdgePix}$) pode ser realizada uma operação de threshold para escolher os segmentos de recta que realmente vão ser utilizados na pesquisa do triângulo. A força do segmento vai então apresentar valores entre 0 e 1, sendo que o threshold escolhido é de 0.5. Posto isto, todos os segmentos de recta com força superior a 0.5 serão aproveitados e colocados numa lista (*ListSeg*). Com este método é permitido detectar segmentos de recta mesmo que estes sofram algumas oclusões.

3.2.4 Detecção do triângulo

Neste momento, para detectar o triângulo, temos a lista de segmentos de recta obtida na subsecção 3.2.3. Como se sabe, um triângulo tem três lados, portanto irá ser percorrida a ListSeg com o objectivo de encontrar três segmentos de recta que estejam ligados entre si.

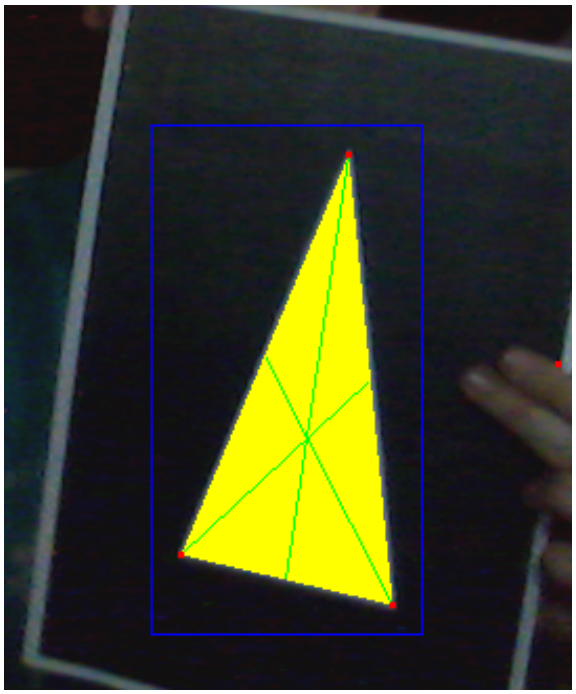

Figura 3.12: Representação de detecção de triângulo

Após encontrados três segmentos de recta ligados será calculada a *força do triângulo* que é a multiplicação das forças de cada segmento constituinte (ver equação 3.8).

$$forca = seg1.forca \times seg2.forca \times seg3.forca \tag{3.8}$$

É necessário agora verificar a força do triângulo e escolher um limite inferior ao qual o triângulo tem de obedecer. Este valor é definido empiricamente, por testes à aplicação e permite detectar o triângulo mesmo que este sofra pequenas oclusões.

Neste momento podemos ter vários triângulos detectados, incluindo alguns indesejados (falsos triângulos). Portanto o próximo passo é o filtro dos falsos positivos, com o objectivo ideal de restarem apenas os verdadeiros marcadores.

3.2.5 Eliminação de falsos positivos

A eliminação de falsos positivos é uma parte muito importante no desenvolvimento de aplicações de visão computacional relacionadas com a identificação de marcadores. O ideal seria conseguir igualar a percepção da visão computacional à do ser humano, mas como sabemos que isso ainda é muito difícil, o objectivo é então tentar dotar a aplicação de conhecimentos que possam rejeitar o número máximo de falsos positivos.

Surge então a necessidade da realização desta tarefa, porque mesmo com as condições estabelecidas para detecção de um objecto, o programa pode ser facilmente *enganado* por outros objectos com algumas características parecidas ao desejado. No trabalho desenvolvido, um falso positivo é quando o programa detecta um triângulo onde ele não está.

Nesta subsecção irá então ser apresentada os métodos utilizados para conseguir obter uma boa detecção do triângulo verdadeiro, rejeitando os falsos positivos. Eles são:

- Teste aos ângulos do triângulo;
- Teste à intensidade de brilho no interior do triângulo;
- Teste ao comprimento dos lados do triângulo.

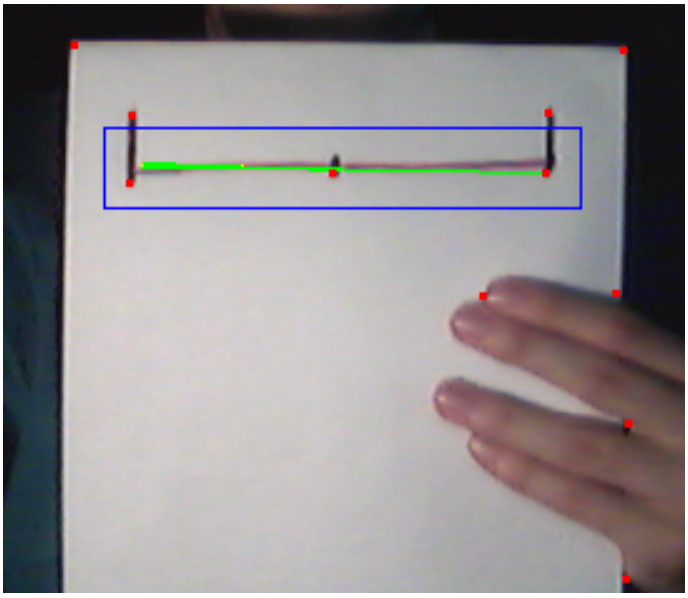

Figura 3.13: Representação de detecção de um falso positivo

O primeiro teste realizado é aos ângulos do triângulo, surgiu a necessidade deste teste, porque sem ele o programa detectava triângulos falsos sobre uma só orla que possuísse três cantos colineares. Na figura 3.13 pode ser observado um exemplo de um falso triângulo detectado nas

condições referidas. o triângulo encontrado apresenta dois ângulos com o valor de 0^o e um com 180^o. Após a imposição da condição de ângulos com valores mínimos, estes "triângulos"deixam de ser detectados.

Em seguida é testada a cor do interior do triângulo detectado. É sabido que o marcador escolhido tem interior branco, portanto é calculada a média do brilho dos píxeis no interior do triângulo, se for superior à média de brilho da imagem toda, então é considerado triângulo, se não, é considerado falso e consequentemente rejeitado.

Como já foi referido, o marcador escolhido é um triângulo isósceles, em que os lados maiores medem aproximadamente o dobro do lado menor. Através da calibração da câmara (abordada no capítulo 4), é possível obter o verdadeiro comprimento dos lados, e com isto testar se o triângulo detectado corresponde ao desejado, medindo e comparando os lados. O teste realizado é o seguinte:

SE

$$\begin{aligned} kb1 \leq dist\_maior1 &\leq kb2 \\ kb1 \leq dist\_maior2 &\leq kb2 \\ ks1 \leq dist\_menor &\leq ks2 \\ |dist\_maior1 - dist\_maior2| &\leq k1 \\ |dist\_maior1 + dist\_maior2 - 4 \times dist\_menor| &\leq k2 \end{aligned} \tag{3.9}$$

ENTÃO é aceite, SENÃO é rejeitado.

dist_maior1 e dist_maior2 correspondem aos lados maiores do triângulo, e dist_menor ao menor lado.

Esta condição serve para realizar um teste as dimensões do triângulo detectado. É imposto um limite inferior e superior para cada lado do triângulo, é testado se os lados maiores apresentam pouca diferença entre si e é testado se os lados maiores medem o dobro do menor, característica do triângulo referida na secção 3.1.

3.3 Melhoramentos no programa desenvolvido

Após realizar alguns testes de detecção do marcador com o programa desenvolvido até aqui, concluiu-se que este é capaz de encontrar o triângulo, mas que possuía ainda algumas limitações. Se um dos cantos deixasse de ser detectado ou fosse encoberto, o programa já não detectava o triângulo, por exemplo.

Foi introduzido no programa um método de seguimento do robô, ou seja, se o robô for detectado num frame, o programa guarda a sua posição e no frame a seguir, se o marcador deixar de ser detectado por alguma razão, ele testa a sua existência na posição detectada anteriormente. Mais detalhadamente, o método funciona da seguinte maneira: quando um marcador é detectado num frame, a posição dos três cantos é guardada. Se no frame a seguir não for encontrado um

triângulo, é colocada uma janela quadrada de 31 píxeis de lado à volta de cada canto anteriormente detectado. Agora, são testadas combinações de 3 cantos dentro das janelas que formem o triângulo mais forte (ver subsecção 3.2.4) mantendo as características conhecidas desejadas do marcador apresentadas nas equações 3.9. Se as condições se verificarem, então estamos na presença do marcador escolhido, e este continuará detectado. Ou seja, este método testa a existência do triângulo anteriormente detectado, e assim, torna possível a detecção do triângulo mesmo que este apresente um ou mais cantos obstruídos, assim como um ou mais lados, embora não aceite um lado completamente obstruído.

Verificou-se também que o processamento do algoritmo era bastante pesado, e o programa era lento, devido à grande quantidade de píxeis que necessitavam de ser processados (resolução da câmara: 780x582). Como melhoramento, o processamento da detecção dos cantos da imagem deixou de ser feita em todos os píxeis da imagem. Antes da detecção de cantos é realizada a detecção de orlas como foi dito anteriormente. Como sabemos que um canto tem de estar localizado no cruzamento de duas orlas, apenas é testada a existência de cantos nos píxeis resultantes da detecção de bordas, diminuindo assim o número de píxeis a serem processados. Deste modo é melhorada a fluidez do programa e com isso torna possível uma localização mais rápida e robusta.

3.4 Resumo

Neste capítulo foi descrito o procedimento efectuado para a detecção do robô. Inicialmente é escolhido o marcador utilizado para ser colocado no topo do robô. Foi escolhido o triângulo isósceles de interior branco e fundo preto, porque é uma figura fácil de detectar e que também permite a determinação da orientação do robô. São identificados os segmentos de recta existentes na imagem através do cruzamento de informação entre os cantos e as orlas encontradas na imagem. De seguida é testado se existem três segmentos de recta que formem o triângulo desejado. Este triângulo terá de obedecer ao conjunto de condições impostas. Algumas das condições foram criadas de maneira à detecção do triângulo suportar algumas oclusões dos lados. Outras condições advém da calibração da câmara realizada no capítulo 4 que nos vai permitir obter a localização real dos cantos no referencial gerado no laboratório de robótica. Com isto poderemos obter os comprimentos reais dos lados do triângulo e testar se o triângulo detectado realmente corresponde ao marcador criado.

Ou seja, para um triângulo ser detectado, os cantos têm de ser detectados, e devem existir pontos de orla entre estes. Caso um ou mais cantos do triângulo não seja detectado num frame, mas tenha sido no frame anterior, então entra em cena o método de seguimento implementado, que testa se o triângulo ainda se encontra presente, se os cantos tivessem perto do mesmo sitio anteriormente detectados.

Capítulo 4

Calibração da Câmara

A calibração de câmara é uma técnica muito utilizada em visão computacional. Esta tem como função determinar os parâmetros intrínsecos e extrínsecos da câmara com o objectivo de tornar possível a correspondência de coordenadas da imagem para as do mundo e vice-versa, e também de corrigir as distorções causadas pela lente da câmara. Neste capítulo será então descrito o funcionamento e o procedimento da calibração utilizada. Na secção 4.1 será descrito o modelo geral de uma câmara, bem como a caracterização dos parâmetros intrínsecos e extrínsecos da câmara e é apresentada a correspondência entre pontos do mundo e da imagem e vice-versa. De seguida, na secção 4.2 é apresentada a estratégia geral (ver subsecção 4.2.1) e procedimento realizado (ver subsecção 4.2.2) para a calibração de câmara.

Este capítulo do trabalho foi realizada com a ajuda da *calibration toolbox* da aplicação matlab e da sua documentação. [30]

4.1 Modelo da câmara

Para a compreensão do funcionamento da câmara normalmente utiliza-se o modelo pinhole. É apresentada uma representação deste modelo na figura 4.1.

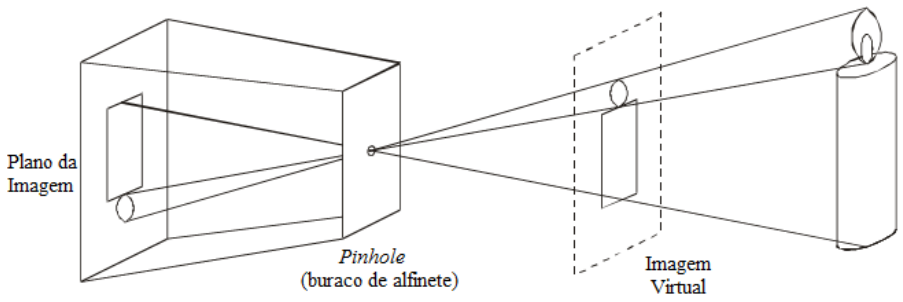

Figura 4.1: Modelo pinhole. Retirado de [15]

Neste modelo a câmara é vista como uma caixa com um orifício idealmente considerado como um ponto (pinhole) onde passa toda a luz situada à frente da câmara que é projectada no plano de imagem. [15] Na realidade é utilizada uma lente que serve para focar a imagem e também alterar a magnitude da imagem registada. O efeito da lente pode ser observado na figura 4.2. [31]

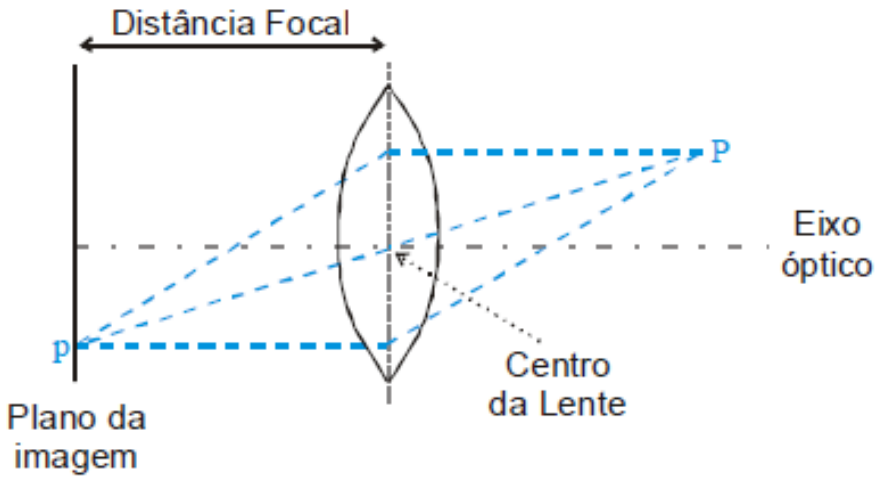

Figura 4.2: Efeito da lente. Retirado de [31]

Inicialmente é necessário compreender os sistemas de eixos utilizados no modelo. Estes podem ser observados na figura 4.3. O referencial $O_iX_iY_iZ_i$ corresponde ao da câmara e $O_wX_wY_wZ_w$ ao do mundo, as coordenadas de ambos estes referenciais são medidas pelo sistema métrico. O plano da imagem é definido pelas coordenadas u e v que são medidas em píxeis.

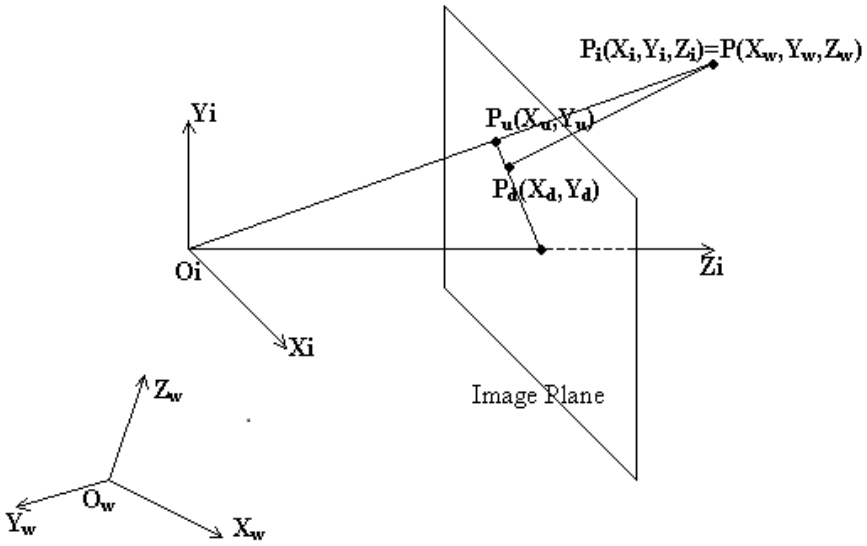

Figura 4.3: Sistema de coordenadas. Retirado de [17]

Existindo um ponto genérico nas coordenadas do mundo, este pode ser representado nas coordenadas da câmara através de uma transformação homogénea constituída por uma rotação e uma translação. Os parâmetros desta transformação podem ser calculados e correspondem aos parâmetros extrínsecos da calibração.

$$R = \begin{bmatrix} r1 & r2 & r3 \\ r4 & r5 & r6 \\ r7 & r8 & r7 \end{bmatrix}; T = \begin{bmatrix} t1 \\ t2 \\ t3 \end{bmatrix} \tag{4.1}$$

Obtidas as coordenadas do ponto no referencial da câmara, é ainda necessário calcular a sua representação no plano da imagem. Aqui já vai ser necessário conhecer os parâmetros intrínsecos da câmara, que são compostos por:

- f (distância focal) - distância entre origem do referencial da câmara e o plano da imagem.
- c (ponto central) - ponto central do plano da imagem (nem sempre igual ao centro dos limites da imagem).
- k's (distorções) - distorções da imagem produzidas pela lente, podem ser radiais ou tangenciais.
- skew - representa o ângulo entre os eixos x e y de um píxel (nem sempre 90^o).

4.2 Calibração

Já apresentado o modelo da câmara e descritos os parâmetros que necessitamos de conhecer, procedemos então para a sua determinação. Vai ser apresentada a estratégia geral utilizada na calibração de câmara bem como o procedimento verdadeiramente realizado neste trabalho.

4.2.1 Estratégia geral

Para determinar os parâmetros da calibração foi realizado um procedimento muito utilizado nas aplicações deste género. É necessário reunir a informação necessária para realizar a determinação dos parâmetros. Este conjunto de dados é composto por correspondências entre pontos da imagem e a sua correspondente localização no sistema de coordenadas do mundo. Com estas correspondências entre pontos, é possível através de um método numérico obter todos os parâmetros da câmara. Este método numérico tenta minimizar uma função de custo que compara os resultados obtidos através do algoritmo referente à correspondência de pontos entre a imagem e o mundo (ver secção 4.3) com os dados fornecidos.

$$C = \sum_i (|x_i - \hat{x}_i| + |y_i - \hat{y}_i|) \tag{4.2}$$

Em que
x_i e y_i são as coordenadas reais retiradas dos dados fornecidos e
$\hat{x}_i$ e $\hat{y}_i$ são os resultados obtidos pelo algoritmo

São realizadas várias iterações até a função de custo convergir para um mínimo global. Para que isto seja alcançado, os valores iniciais do algoritmo (parâmetros intrínsecos) devem ser colocados de forma cuidada. [31]

4.2.2 Procedimento realizado

Posto isto, para proceder à determinação dos parâmetros necessários para a calibração da câmara, foi primeiro necessário obter o conjunto de correspondências entre os pontos da imagem e do mundo. Foi utilizado um tabuleiro de xadrez para criar o conjunto de dados, visto que contém vários pontos facilmente extraíveis que são os cantos dos quadrados. Como o número de quadrados existentes é previamente conhecido, assim como as suas dimensões, torna-se possível conhecer a distância real entre os pontos extraídos.

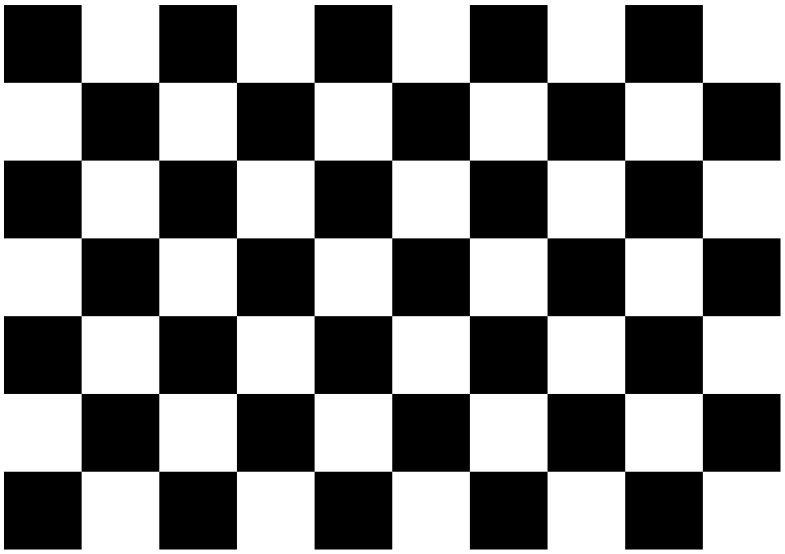

Figura 4.4: Padrão de um tabuleiro de xadrez

Para obter estes dados é realizado o procedimento a seguir descrito. É necessário extrair os cantos do tabuleiro de xadrez, através de um algoritmo de detecção de cantos com precisão até ao sub-píxel. Tendo o conjunto de cantos da imagem, é agora necessário proceder à selecção e ordenação dos cantos a serem utilizados. Posto isto, basta agora fazer corresponder os cantos extraídos de acordo com a sua linha e coluna no padrão de xadrez utilizado.

Realizado isto, obtém-se o conjunto de dados necessários para a calibração dos parâmetros da câmara.

Neste trabalho, este procedimento foi realizado com o auxílio da *caixa de ferramentas* de calibração do matlab. Foram tiradas várias fotos, com a câmara a ser calibrada, do tabuleiro de xadrez em várias orientações diferentes. Depois de obtido o conjunto de fotos, este foi inserido e processado no matlab. Na *Calibration Toolbox* pegou-se no conjunto de imagens e uma-a-uma identificaram-se os quatro cantos que limitam o padrão de xadrez utilizado, é definido o comprimento do lado dos quadrados do tabuleiro de xadrez, e o número de quadrados existentes nas duas direcções (X e Y). O programa cria uma grelha por cima do tabuleiro, e tenta encontrar todos os cantos existentes. Após realizar o procedimento de calibração, os parâmetros intrínsecos são calculados e apresentados. De seguida necessitamos de calcular os parâmetros extrínsecos, que incluem a rotação e translação do referencial, então é escolhida uma foto que contenha o tabuleiro de xadrez numa posição que represente o referencial desejado. Realizando o mesmo procedimento anteriormente descrito são então calculados os parâmetros extrínsecos desejados.

4.3 Aplicações no trabalho realizado

Já identificados e calculados os parâmetros intrínsecos e extrínsecos da câmara, é apresentado nesta secção como são utilizados para converter um ponto nas coordenadas do mundo para um píxel da imagem e vice-versa, e também como remover a distorção da imagem.

4.3.1 Correspondência de pontos Mundo-Imagem

Suponhamos então que desejamos representar um ponto, que só conhecemos as coordenadas do mundo ($O_w X_w Y_w Z_w$), na imagem através das coordenadas (u,v) em píxeis. Descrevendo o problema e resolução por passos, temos:

1. Transformação do ponto no referencial do mundo para o da câmara. Como já vimos anteriormente, esta transformação é realizada através de uma rotação e translação de referencial. Ou seja,

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \tag{4.3}$$

   em que R, e T são os parâmetros definidos nas equações 4.1.

2. Transformar das coordenadas 3D do referencial da câmara para 2D da imagem, mas sem contar ainda com a distorção da lente. Chamaremos de (X_u, Y_u) a estas coordenadas.

$$X_u = \frac{X_i}{Z_i}$$

$$Y_u = \frac{Y_i}{Z_i} \tag{4.4}$$

3. Após incluir a distorção da lente, o novo valor de coordenadas (X_d, Y_d) vai ser definido como:

$$\begin{aligned} X_d &= [1 + k(1)r^2 + k(2)r^4]X_u + 2k(3)X_uY_u + k(4)(r^2 + 2X_u^2) \\ Y_d &= [1 + k(1)r^2 + k(2)r^4]Y_u + 2k(4)X_uY_u + k(3)(r^2 + 2Y_u^2) \end{aligned} \tag{4.5}$$

em que:
$r^2 = X_u^2 + Y_u^2$
k(1) e k(2) representam a distorção radial e
k(3) e k(4) representam a distorção tangencial.

4. E finalmente o píxel final terá então o valor (u,v):

$$\begin{aligned} u &= f_x(X_d + skew * Y_d) + c_x \\ v &= f_y Y_d + c_y \end{aligned} \tag{4.6}$$

em que:
f_x e f_y representam a distância focal em x e em y respectivamente (geralmente iguais ou muito parecidas)
c_x e c_y representam as coordenadas do ponto central da imagem e
skew representa o ângulo entre os eixos x e y de um píxel da imagem.

4.3.2 Correspondência de pontos Imagem-Mundo

Para obter a correspondência de pontos Imagem-Mundo não é possível inverter o procedimento realizado na subsecção 4.3.1, porque como é facilmente perceptível um ponto no mundo corresponde a um píxel na imagem, mas um ponto na imagem corresponde a uma infinidade de pontos no mundo, formam uma linha de pontos desde o píxel na câmara até ao ponto no mundo. Para ser possível obter a correspondência desejada, necessitamos de, além das coordenadas do píxel na imagem, uma coordenada do ponto no mundo, normalmente considera-se conhecida a posição Z do ponto. Outro problema desta correspondência inversa é não ser possível aplicar o modelo de distorção directamente, visto que o modelo é de grau elevado. Para o aplicar é utilizado um método iterativo que vai ser descrito nesta subsecção.

Vai ser descrito por passos o procedimento realizado para obter a correspondência entre um ponto na imagem e as suas coordenadas no mundo:

1. Invertendo o realizado nas equações 4.6:

$$\begin{aligned} X_d &= \frac{u - c_x}{f_x} \\ Y_d &= \frac{v - c_y}{f_y} \\ X_d &= X_d - skew * Y_d \end{aligned} \tag{4.7}$$

2. Neste momento é preciso remover a distorção do ponto da imagem, para isto é realizado um método iterativo de acordo com [30] apresentado a seguir:

   Iterando 20 vezes o seguinte conjunto de equações, e definindo como estimativa inicial $X_u = X_d$ e $Y_u = Y_d$

$$\begin{aligned} r^2 &= X_u^2 + Y_u^2 \\ X_u &= \frac{X_d - 2k(3)X_uY_u + k(4)(r^2 + 2X_u^2)}{1 + k(1)r^2 + k(2)r^4} \\ Y_u &= \frac{Y_d - 2k(4)X_uY_u + k(3)(r^2 + 2Y_u^2)}{1 + k(1)r^2 + k(2)r^4} \end{aligned} \tag{4.8}$$

   Depois de realizado estas equações no ciclo de 20 vezes, conseguimos obter uma boa aproximação das coordenadas do ponto depois de remover a distorção.

3. Neste momento, seguindo a ordem inversa de operações realizada na subsecção 4.3.1, é necessário obter X_i, Y_i e Z_i, coordenadas do ponto no referencial da câmara. Pela equação 4.4 é possível obter que:

$$\begin{aligned} X_i &= X_uZ_i \\ Y_i &= Y_uZ_i \end{aligned} \tag{4.9}$$

   Se conhecermos o valor de uma das coordenadas, é possível ficar a conhecer o valor das restantes. Nesta situação é usual definir a posição Z_w do ponto constante. Conhecendo este valor, é possível ficar a conhecer o valor de Z_i. A partir de 4.3 e 4.9, pode-se obter:

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = R^{-1} \begin{bmatrix} X_uZ_i - T_x \\ Y_uZ_i - T_y \\ Z_i - T_z \end{bmatrix} \tag{4.10}$$

   É conhecido Z_w, $R^{-1} = R^t$ (porque R é uma matriz ortonormal), X_u, Y_u e T. Portanto da equação acima, resolvendo apenas a equação relativa a Z_w, é possível obter Z_i. Utilizando as equações 4.9, obtém-se de seguida X_i e Y_i.

4. Uma vez conhecidas as coordenadas do ponto no referencial da câmara, basta agora aplicar a rotação e translação inversamente como foi realizado no ponto 3, mas agora para determinar as restantes coordenadas X_w e Y_w.

4.3.3 Remover Distorção

Uma das aplicações dos parâmetros de calibração da câmara mais usadas, é a de remover a distorção de uma imagem capturada. Para tal, para cada píxel da nova imagem sem distorção, vai ser calculado qual a localização correspondente na imagem da câmara. Em seguida, através de uma interpolação bilinear de intensidades da vizinhança do ponto na imagem original, é calculada a cor do píxel sem distorção.

Para calcular o valor da localização do píxel sem distorção na imagem original é realizado o seguinte procedimento:

1. É calculado o valor das coordenadas X_u e Y_u através do píxel da imagem (u,v).

$$\begin{aligned} X_u &= \frac{(u-c_x)}{f_x} \\ Y_u &= \frac{(v-c_y)}{f_y} \end{aligned} \tag{4.11}$$

2. Em seguida é aplicada a distorção, são calculados X_d e Y_d:

$$\begin{aligned} X_d &= [1+k(1)r^2+k(2)r^4]X_u + 2k(3)X_uY_u + k(4)(r^2+2X_u^2) \\ Y_d &= [1+k(1)r^2+k(2)r^4]Y_u + 2k(4)X_uY_u + k(3)(r^2+2Y_u^2) \end{aligned} \tag{4.12}$$

3. E finalmente o píxel da nova imagem sem distorção terá a correspondente localização na imagem original de:

$$\begin{aligned} u' &= f_x(X_d + skew * Y_d) + c_x \\ v' &= f_y Y_d + c_y \end{aligned} \tag{4.13}$$

Depois de calculados (u',v'), será realizada uma interpolação bilinear com os valores dos píxeis vizinhos a este. O resultado desta operação será colocado no píxel (u,v) da nova imagem.

Denominando $(u1',v1')$, $(u2',v1')$, $(u2',v2')$ e $(u1',v2')$ como as coordenadas dos quatro píxeis da vizinhança de (u',v'), de acordo com a figura 4.5.

O valor do píxel da nova imagem sem distorção será:

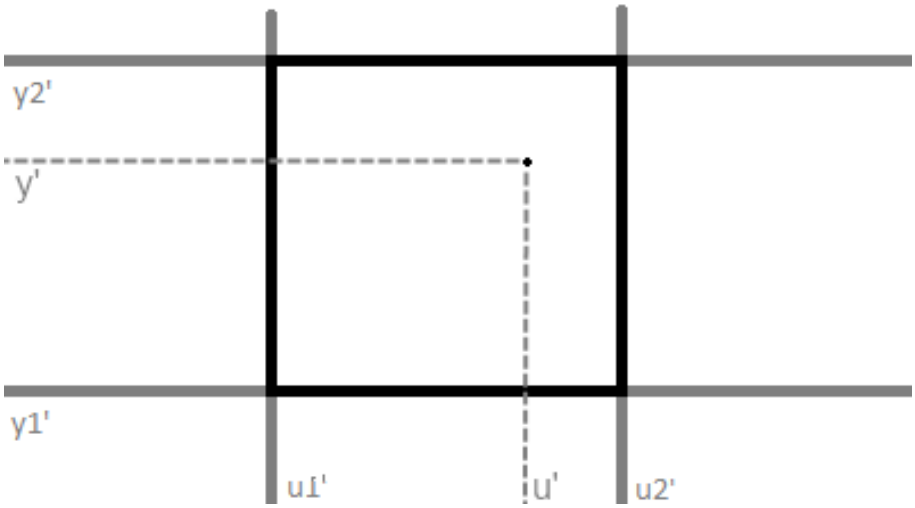

Figura 4.5: Representação auxiliar para a interpolação bilinear apresentada.

$$\begin{aligned} f(u,v) &= f(u1',v1')*(u2'-u')(v2'-v') \\ &+ f(u2',v1')*(u'-u1')(v2'-v') \\ &+ f(u2',v2')*(u'-u1')(v'-v1') \\ &+ f(u1',v2')*(u2'-u')(v'-v1') \end{aligned} \tag{4.14}$$

4.4 Resumo

Neste capítulo foi apresentado o conceito de calibração de uma câmara. Foi estudado o modelo da câmara, assim como os seus parâmetros intrínsecos. No trabalho realizado é necessário fazer corresponder pontos 2D em 3D e vice-versa, portanto parâmetros extrínsecos também são apresentados e calculados. Outra das aplicações da calibração é a remoção da distorção numa imagem, que apenas utiliza os parâmetros intrínsecos calculados da câmara.

Capítulo 5

Resultados e verificações experimentais

Neste capítulo serão apresentados os resultados dos testes realizados deste trabalho. Vão ser apresentados exemplos da detecção do marcador escolhido e testes à robustez do sistema de detecção. Em seguida vão ser apresentados os resultados obtidos da calibração da câmara: os parâmetros calculados, o referencial utilizado e testes à certeza da posição e orientação.

Os testes foram realizados no laboratório de robótica, utilizando uma câmara colocada no tecto.

5.1 Detecção do robô

Neste capítulo vão ser apresentados os resultados referentes ao processamento de imagem efectuado para detecção do marcador escolhido. O marcador foi impresso e colado numa placa de madeira de 1 cm de espessura, para evitar algumas dobras da folha que levariam à não detecção do mesmo. As suas dimensões escolhidas estão apresentadas na tabela 5.1.

Tabela 5.1: Medidas do Marcador

Lado menor	13.4
Lado maior	26.8

5.1.1 Detecção do marcador

Nas imagens quando o marcador apresenta um rectângulo azul à sua volta significa que esta a ser detectado. Foram então realizados alguns testes de detecção do triângulo. A altura do referencial foi elevada de uma constante de 1 cm, porque o marcador encontra-se a 1 cm do chão, correspondente à espessura da placa onde está colado. Portanto, se o marcador for colocado num robô, é necessário actualizar o valor da elevação dependente da altura a que este for colocado. O marcador não será então detectado se se encontrar inclinado, pois os comprimentos dos lados do

triângulo calculadas apresentarão valores diferentes dos reais. Três exemplos de detecção do triângulo podem ser observadas nas figuras 5.1. Na primeira é apresentada a detecção de um marcador, nas restantes detecta-se dois marcadores.

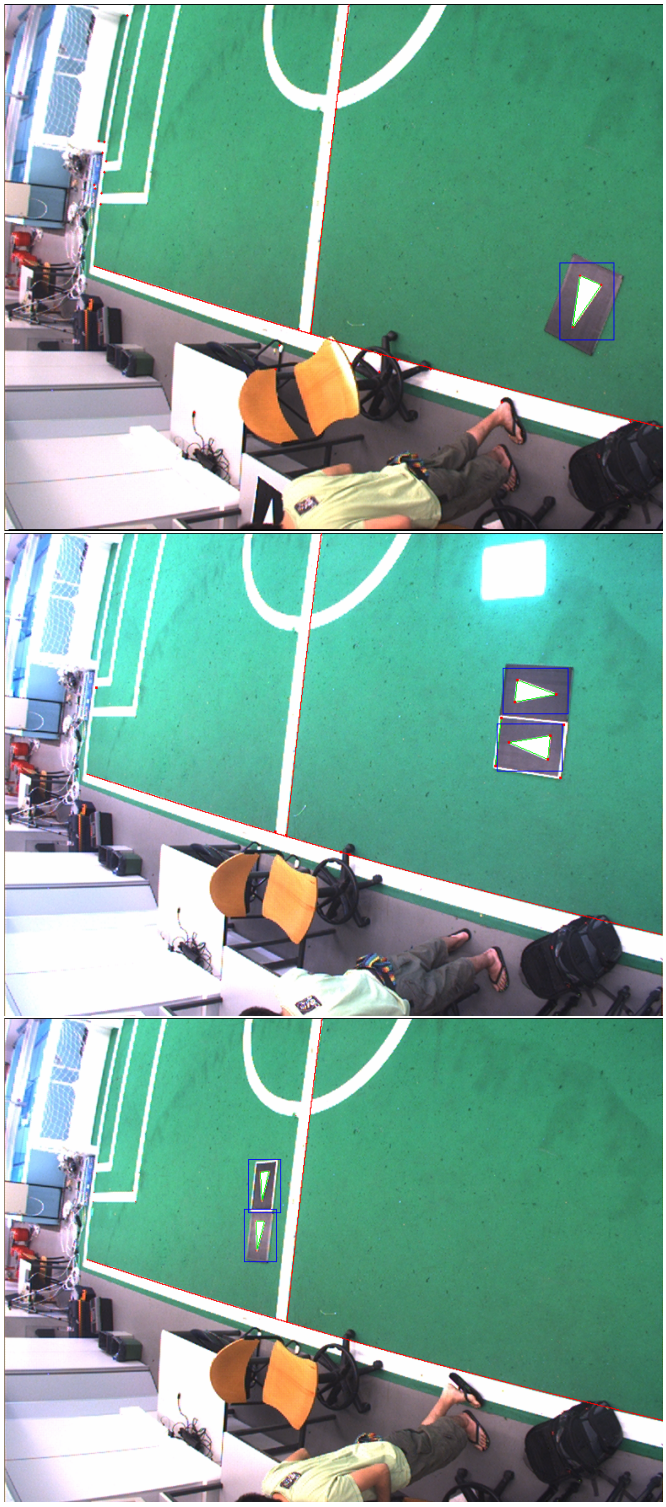

Figura 5.1: Detecção do marcador(1)

Vão agora ser apresentadas mais duas imagens 5.2, mas desta vez com os triângulos mais afastados da câmara, o que provoca uma maior inclinação no marcador. É possível observar que os marcadores continuam a ser detectados mesmo com a visualização dos triângulos deformados.

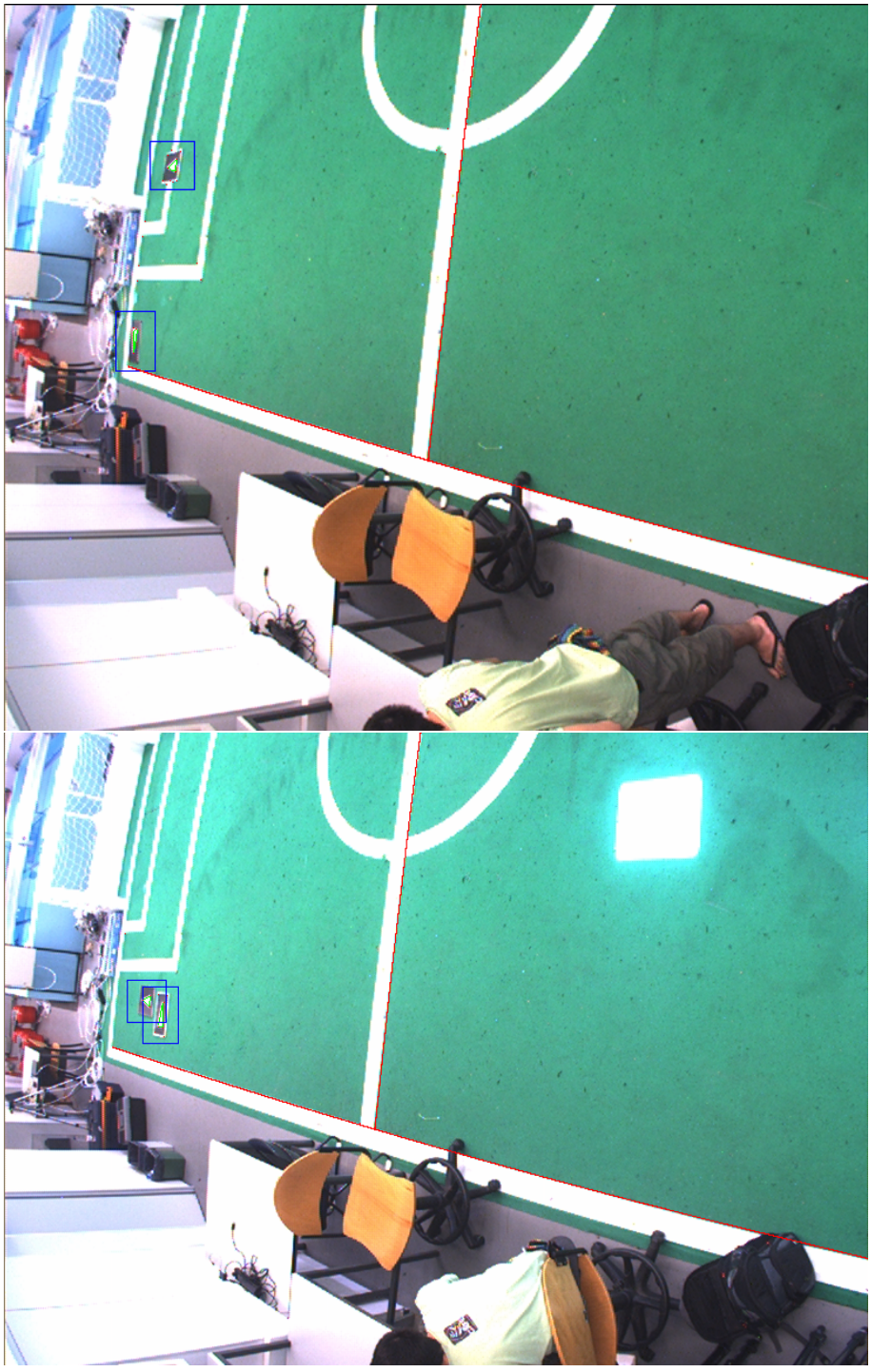

Figura 5.2: Detecção do marcador(2)

Nas figuras 5.3, pode ser observada a simulação da colocação do marcador no robô. Na primeira imagem, a elevação do referencial para a detecção do marcador está colocada em 1cm, é por isso que dos dois triângulos, apenas o que se encontra a 1cm do chão é detectado. Na figura seguinte, e simulando a altura do robô com a cadeira, é imposta a detecção de marcadores a uma altura de 60cms. Verifica-se então, que apenas o marcador situado a essa altura é detectado.

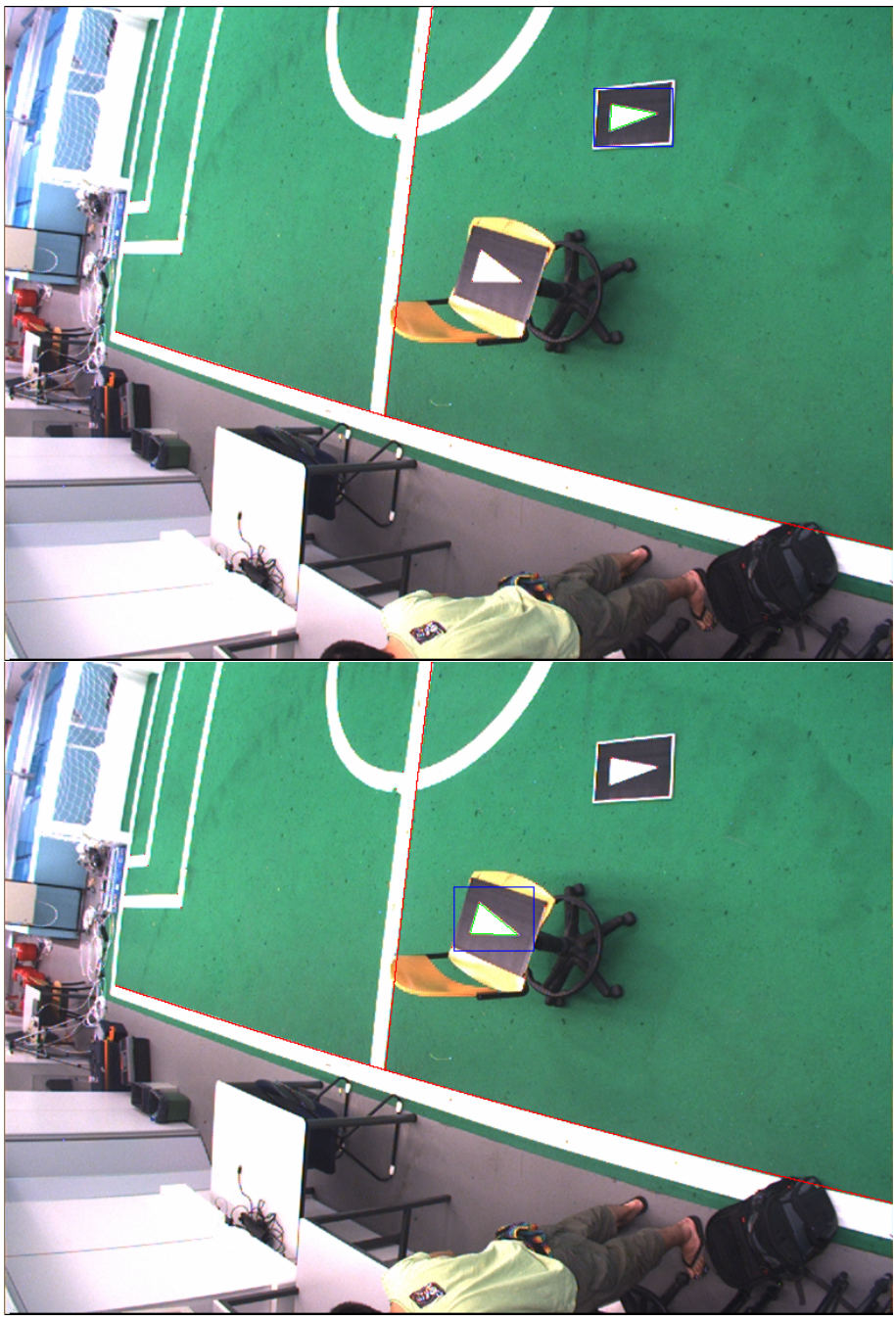

Figura 5.3: Simulação com elevação da posição do marcador

Por último, na imagem 5.4 pode ser observada a detecção de vários marcadores com diversas orientações espalhados pelo campo robótico.

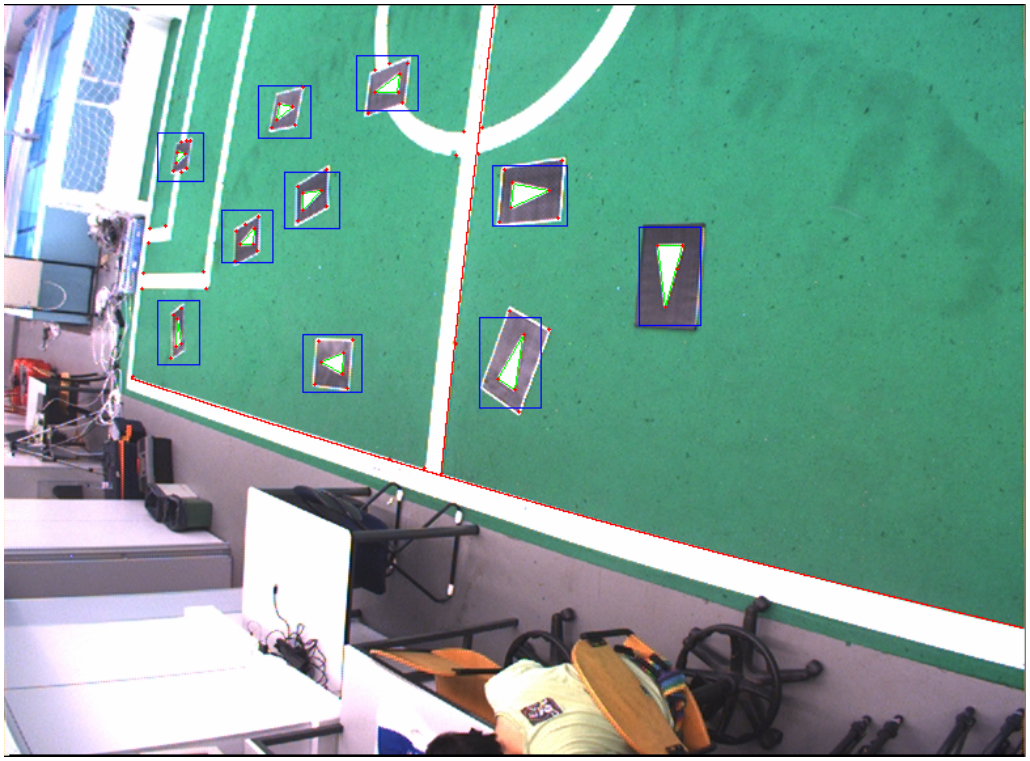

Figura 5.4: Detecção do triângulo(3)

5.1.2 Detecção com oclusão

Foi agora testado a robustez da detecção do marcador. Os testes incluirão a oclusão de várias partes do marcador. Neste capítulo serão apresentadas figuras que demonstram os testes realizados.

Na figura 5.5, pode ser observada uma detecção de marcador com dois dos lados obstruídos. O triângulo é detectado porque os cantos são correctamente detectados, e os segmentos entre cantos são aceites como validos, visto que se tivessem completos possuiriam comprimento real mínimo imposto pela condição de detecção (ver equações 3.9) e a força do segmento é superior ao threshold imposto (0.5, ver subsecção 3.2.3).

Na figura 5.6, pode ser observada outra oclusão de dois lados do triângulo. Desta vez a oclusão de cada um dos lados torna cada um dos segmentos mais fracos que o limite imposto. Portanto esta detecção só se tornou possível devido à técnica de seguimento utilizada referida na secção 3.3. Ou seja, o triângulo neste frame não foi encontrado, devido à obstrução de mais de metade de pelo menos um dos lados, mas como existia um triângulo anteriormente detectado, o método de seguimento testou a ainda existência do mesmo triângulo com uma condição de detecção mais

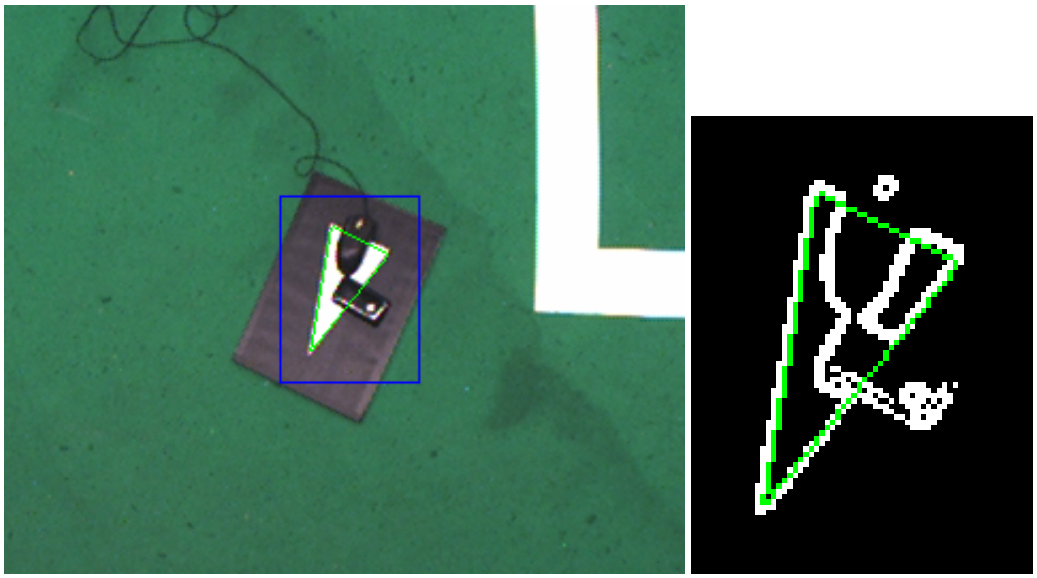

Figura 5.5: Oclusão de dois lados(1)

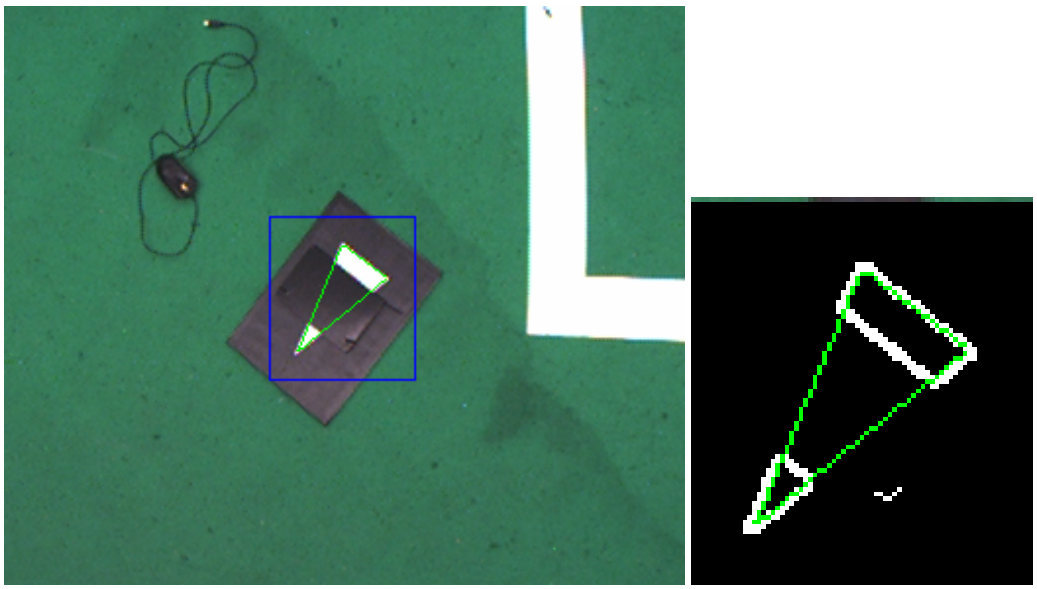

Figura 5.6: Oclusão de dois lados(2)

leve. Assim sendo, verificou-se a detecção do triângulo com dois lados muito obstruídos, desde que este já tenha sido detectado anteriormente.

Na figura 5.7 é apresentada a detecção do triângulo com um canto tapado. Mais uma vez esta detecção só é possível com a técnica de seguimento implementada. Foi tapado um canto ao triângulo previamente detectado, o método de seguimento, neste caso, sabe qual é o canto perdido, por isso mantém os restantes e testa se, se o canto tapado ainda existisse numa posição próxima do detectado anteriormente, o triângulo ainda está presente. Obviamente, como se depara com os segmentos ainda parcialmente existentes até ao canto simulado, o triângulo continua a ser

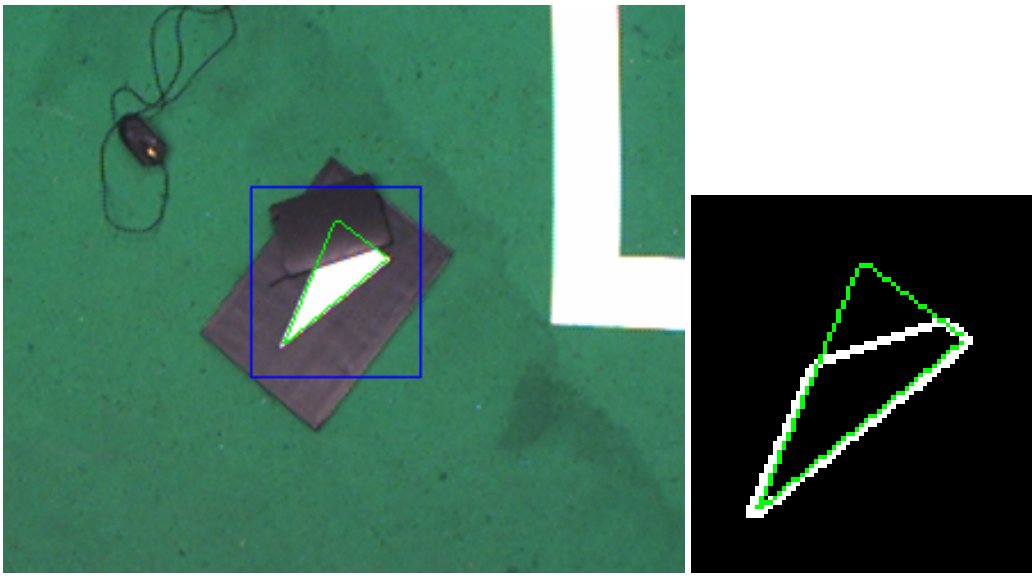

Figura 5.7: Oclusão de um canto

detectado.

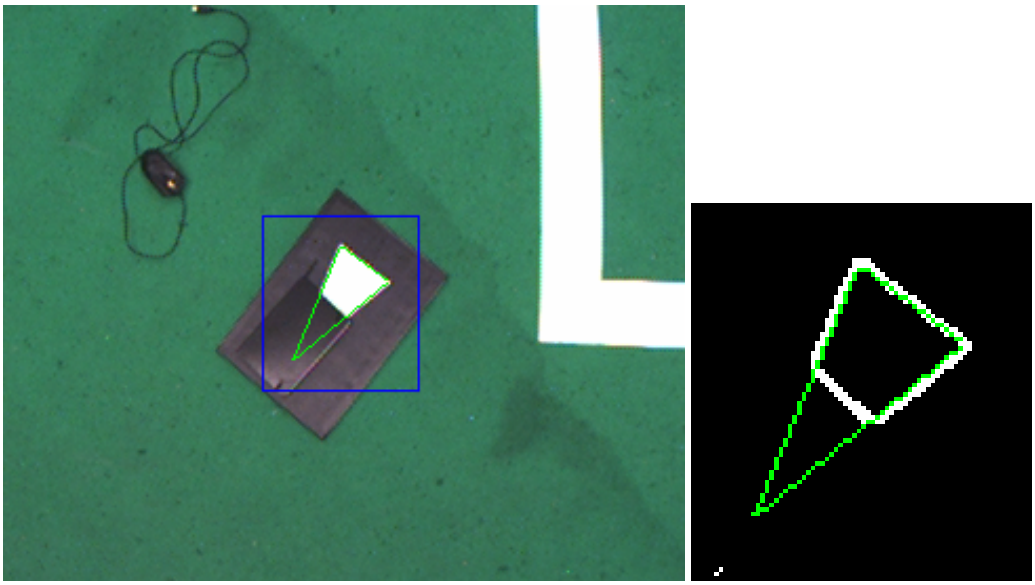

Figura 5.8: Oclusão de outro canto

Na figura 5.8 é apresentada uma detecção semelhante à anterior, mas com um canto diferente obstruído. A explicação da detecção é a mesma.

Na figura 5.9 é observado mais uma vez o resultado da detecção do triângulo através da técnica de seguimento implementada. Desta vez é verificado que este método continua a ter bons resultados mesmo quando tem de simular os três cantos simultaneamente.

Na figura 5.10 vemos o triângulo com um lado completamente obstruído. Neste caso a detecção do triângulo não foi possível, porque o programa "vê" um triângulo menor do que o que está à procura. É relembrado que com a calibração da câmara é possível obter e testar as dimensões reais do triângulo, tais testes são realizados de acordo com as equações 3.9. É fácil de verificar que o triângulo encontrado não verifica as condições de dimensão necessárias, e portanto o triângulo não pode ser detectado. O método de seguimento ainda tenta simular os dois cantos perdidos, mas necessita da presença de todos os segmentos, nem que seja apenas parcialmente, porque uma das condições de aceitação do triângulo, é dependente da força deste (ver equação 3.8), e como um dos segmentos tem força 0, o triângulo também vai ter força 0, rejeitando-o.

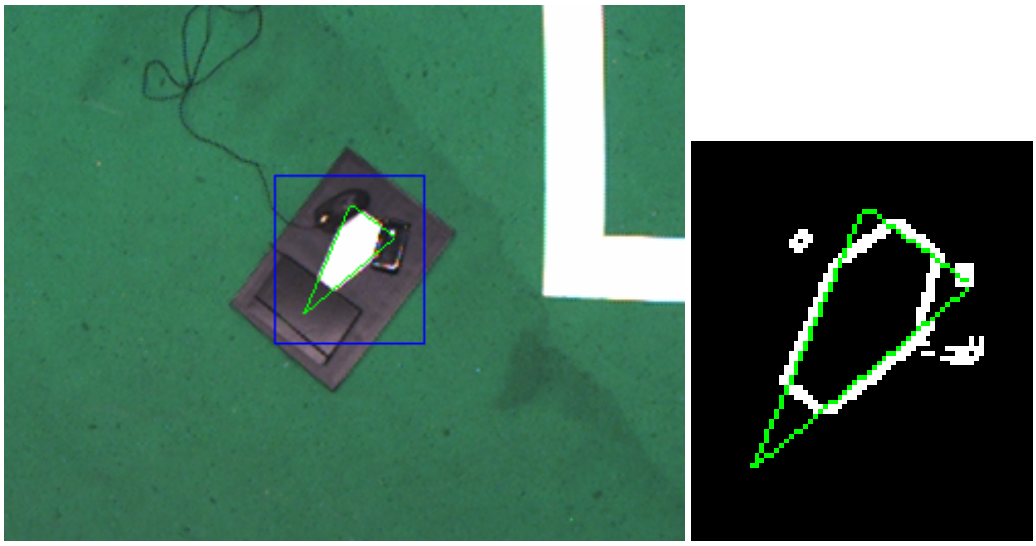

Figura 5.9: Oclusão dos três cantos do triângulo

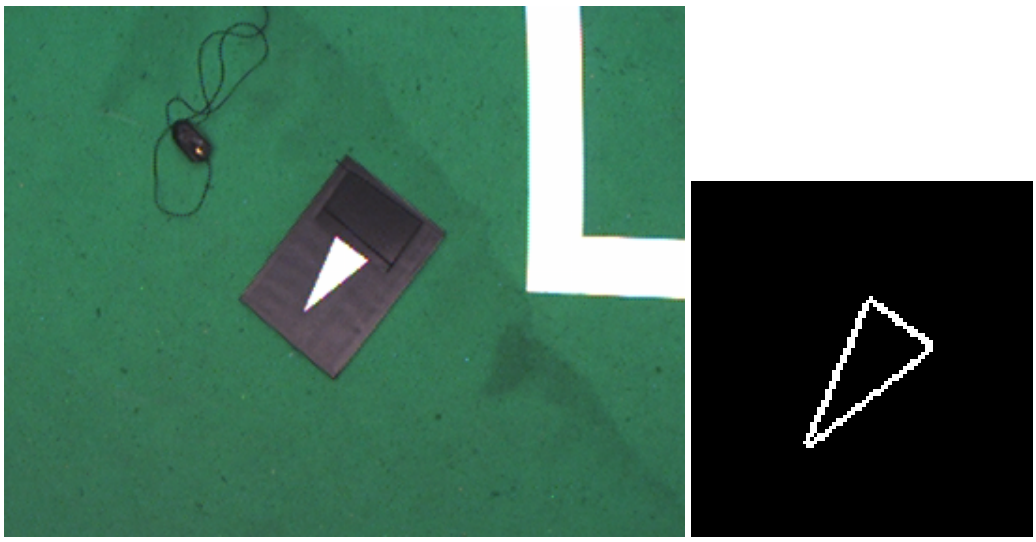

Figura 5.10: Oclusão de um lado completo

5.1.3 Tempos de Processamento

Foi agora testado o tempo de processamento do programa desenvolvido. Para tal, e de maneira a avaliar cada passo do processamento, foi verificado o tempo decorrido durante algumas etapas consideradas importantes. As etapas avaliadas são:

1. Conversão do formato de cor da imagem.
2. Sobel + Threshold + Detecção de cantos.
3. Detecção de segmentos.
4. Encontrar 3 segmentos ligados (triângulo).
5. Eliminação de falsos positivos + Seguimento.
6. Determinação da posição e orientação + Desenhos na imagem.

Foram realizados vários testes, em diversas situações, pois os tempos de processamento variam conforme o número de triângulos na imagem e com o número de cantos de um triângulo obstruídos, como pode ser verificado na tabela 5.2. É possível verificar que quanto maior é o número de triângulos detectados maior é a duração do processamento do programa, mais especificamente, das etapas nº 3, 4 e 5. Também pode ser observado que nos testes com oclusão de cantos os tempos referentes à etapa nº4 são superiores ao teste de 1 triângulo sem oclusão. Esta variação deve-se à utilização do método de seguimento na etapa nº5 quando algum canto não foi bem detectado.

Tabela 5.2: Tempos de processamento do programa (em milissegundos)

	Etapas							**Descrição do teste**
	1	2	3	4	5	6	Tempo total	
1	14	239	2	0	2	14	271	0 Triângulos
2	12	234	4	16	40	33	339	1 Triângulo
3	11	265	7	35	90	17	425	2 Triângulos
4	14	278	9	50	150	22	523	3 Triângulos
5	12	246	4	0	63	14	339	1 Triângulo com 1 Canto Obstruído
6	12	254	3	1	69	15	354	1 Triângulo com 2 Cantos Obstruídos

Outra observação retirada da tabela 5.2 é que a etapa mais demorada é a nº2, referente à detecção de orlas e cantos na imagem. Analisando-a mais pormenorizadamente, foi dividida em 3 sub-etapas: Sobel, Detecção de cantos e ordenação de cantos por ordem decrescente de melhor canto. Foram realizados 3 testes, com diferentes valores para os números de cantos encontrados, pois este é o único factor que faz variar o tempo de processamento desta etapa. Na tabela 5.3 podem ser observados os tempos de processamento obtidos. Pode ser verificado que o passo mais pesado é a detecção de orlas através da aplicação do Sobel.

Tabela 5.3: Tempos de processamento pormenorizados da etapa nº4 (em milissegundos)

nº de Cantos	**Sobel**	**Detecção de cantos**	**Ordenação de cantos**	**Tempo total**
21	145	45	13	203
32	151	53	36	240
37	151	53	45	249

5.2 Calibração da câmara

Depois de realizado o procedimento de calibração de câmara com o conjunto de 30 imagens que podem ser observados no anexo B, foram obtidos os parâmetros intrínsecos apresentados na tabela 5.4. Os parâmetros extrínsecos foram obtidos com base no referencial desejado e estão apresentados nas equações 5.1. O referencial obtido pode ser observado na figura 5.11.

$$
\begin{aligned}
R &= \begin{bmatrix} 0.117950 & -0.683240 & -0.720604 \\ -0.992251 & -0.109644 & -0.058455 \\ -0.039071 & 0.721915 & -0.690878 \end{bmatrix} \\
T &= \begin{bmatrix} -439.741875 \\ 416.271764 \\ 4414.167445 \end{bmatrix}
\end{aligned} \tag{5.1}
$$

Tabela 5.4: Parâmetros Intrínsecos

Parâmetro	**Valor**	**Incerteza**
f_x	656.005616	0.372503
f_y	655.161438	0.369508
c_x	385.154181	0.876290
c_y	296.578134	0.826043
skew	-0.000062	0.000267
k(1)	-0.365251	0.002258
k(2)	0.179178	0.006209
k(3)	-0.000426	0.000166
k(4)	-0.000600	0.000166

Figura 5.11: Referencial obtido

5.2.1 Remoção da distorção

Nas imagens 5.12 pode-se observar o gráfico de distorção, é possível observar o efeito em barril gerado pela lente da câmara.

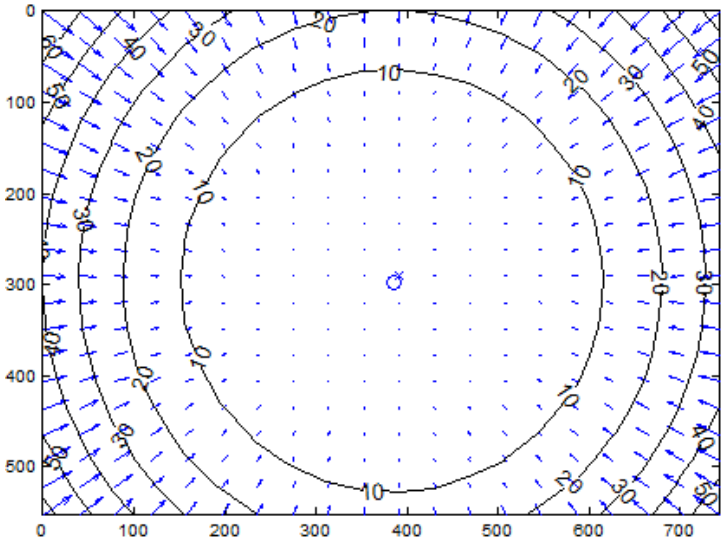

Figura 5.12: Gráfico representante da distorção gerada pela lente da câmara

Com os parâmetros intrínsecos calculados é possível remover a distorção da imagem. Nas imagens 5.13 é possível observar a distorção gerada pela câmara nas linhas de futebol robótico. Após a remoção da distorção, através dos parâmetros intrínsecos calculados, obtivemos as mesmas imagens mas sem distorção observáveis nas figuras 5.14. É possível verificar que as linhas do campo de futebol robótico apresentam agora formato rectilíneo, como deveriam. A imagem de detecção de orlas é inserida aqui também pois torna mais fácil a percepção da remoção da distorção.

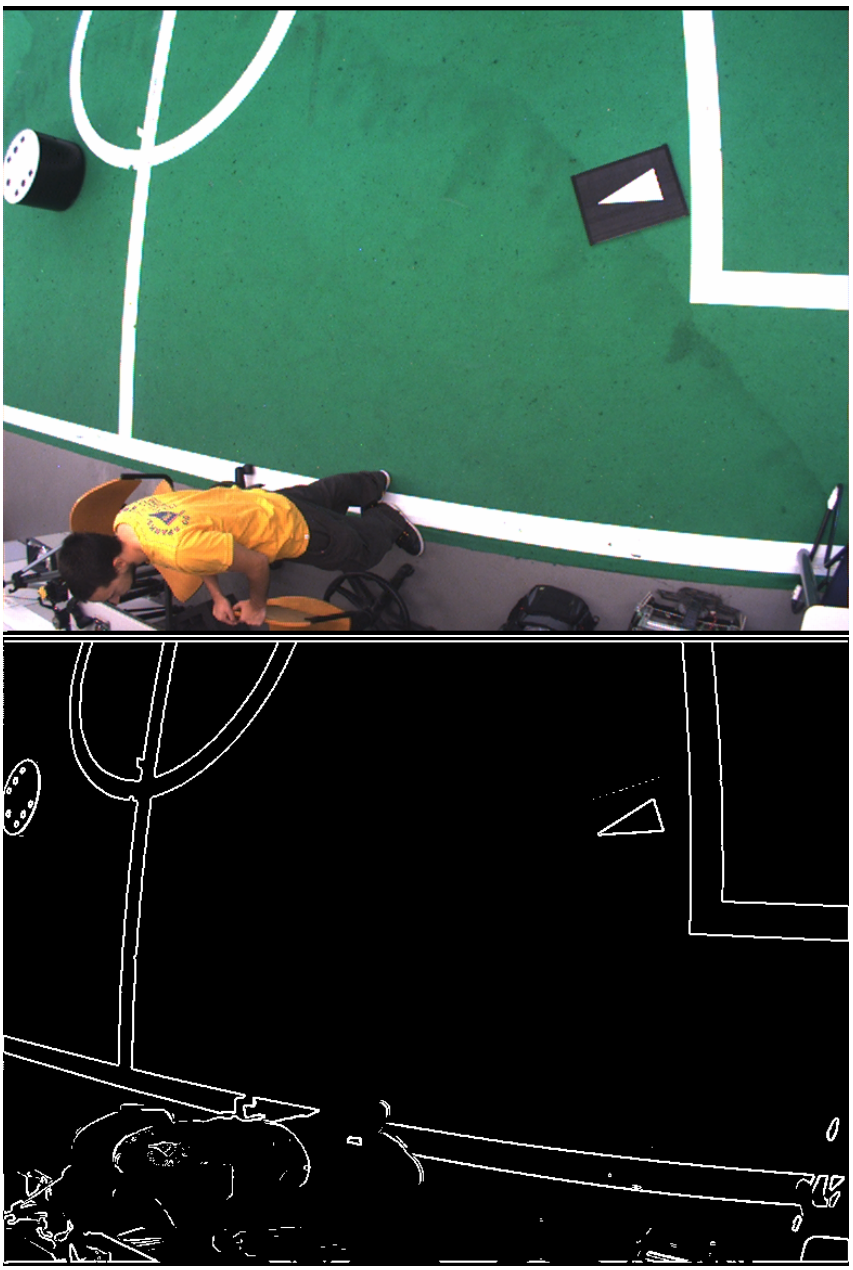

Figura 5.13: Imagem capturada pela câmara e as orlas detectadas na imagem para salientação da distorção nas linhas de futebol

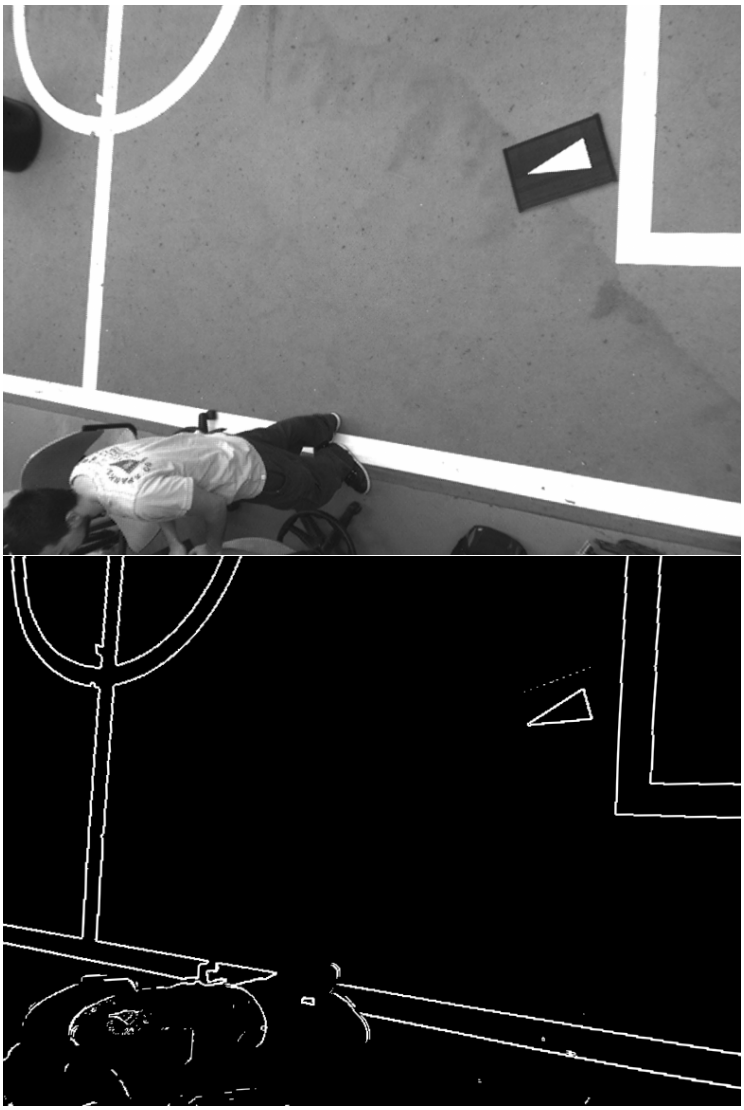

Figura 5.14: Imagens correspondentes às da figura 5.13, mas depois de remover a distorção

5.3 Precisão dos resultados

A precisão da localização obtida pelo programa desenvolvido depende de duas componentes de erro. Uma componente depende do ruído da detecção do centro do marcador, e a outra depende da calibração da câmara. A primeira é originada com a detecção dos cantos. Quando o triângulo se situa longe da câmara, a detecção de alguns dos cantos pode variar de um píxel para um vizinho, gerando um ruído no ponto central do marcador detectado e consequentemente um ruído na localização real do robô. Este ruído como já foi referido ocorre apenas quando o marcador se encontra bastante afastado da câmara, e apresenta um erro possível de no máximo de 1 píxel, que corresponde a uma situação de ruído dos três cantos do triângulo. Nos testes realizados tal situação não ocorreu, verificando-se apenas um ruído de no máximo 0.333 píxeis.

O ruído também tem pesos diferentes conforme a distância à câmara, visto que o valor de um

píxel aumenta com esta distância como pode ser observado na figura 5.15. Na figura 5.16 pode ser verificado a posição da câmara e distância desta ao referencial utilizado no campo robótico.

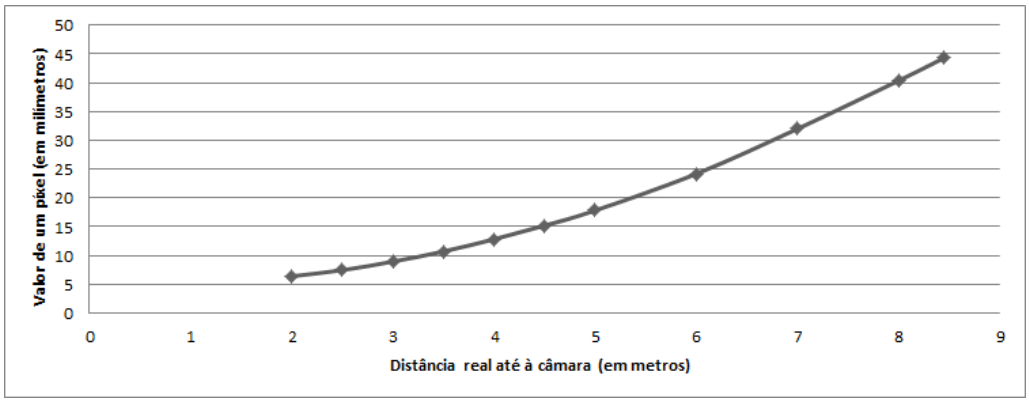

Figura 5.15: Função que relaciona o valor real de um píxel em mms com a distância real do ponto à câmara em metros.

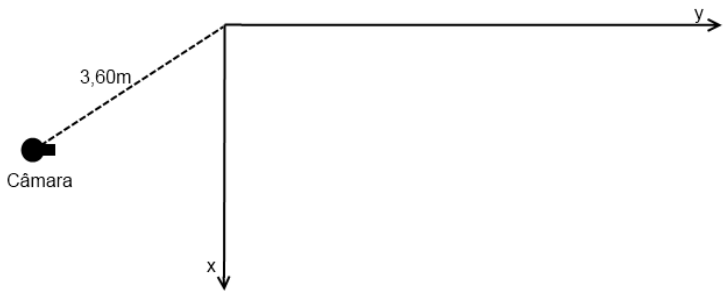

Figura 5.16: Posição e distância da câmara ao referencial utilizado.

A outra componente de erro depende da calibração da câmara. Na tabela 5.5 são apresentadas as comparações da localização real e a obtida no programa. É apresentado o erro absoluto de localização em cada ponto que corresponde à distância real entre o ponto medido e o estimado. É também apresentado o erro relativo que é descrito pela razão $\frac{Erro\ Absoluto}{Distancia\ da\ origem\ ao\ ponto}$. São testados 8 pontos. Podemos observar que a calibração foi bem realizada, apresentando erros na ordem dos milímetros. É possível também notar que os pontos onde se verificou maior erro absoluto são os mais afastados da câmara, embora o que apresenta maior erro relativo (amostra n^{o}6) seja um ponto próximo da origem do referencial. Estes erros devem-se às incertezas obtidas na calibração da câmara, que embora não sejam muito elevadas, produzem um erro absoluto que aumenta com o distanciamento do marcador.

Tabela 5.5: Comparação entre localização real e calculada

	Obtido		Real		Erro	
	X(cms)	Y(cms)	X(cms)	Y(cms)	Absoluto(cms)	Relativo(%)
1	163.70	0	163.47	-0.07	0.2404	0.15
2	89.20	307.50	88.73	308.00	0.6862	0.21
3	0	484	-0.26	483.37	0.6815	0.14
4	89.20	484	88.53	484.07	0.6736	0.14
5	134.10	414.30	134.06	412.96	1.3406	0.31
6	0	-12.70	0.27	-12.76	0.2766	2.18
7	101.60	319.65	101.55	318.42	1.2310	0.37
8	146.1	427	146.95	426.95	0.8515	0.19
	Média erro				**0.7477**	**0.4612**

Tabela 5.6: Comparação entre orientação real e calculada (em graus)

	Obtido	Real	Erro
1	0.23	0	0.23
2	90.67	90	0.67
3	-179.26	$\pm$180	0.74
4	-90.23	-90	0.23
5	3.23	0	3.23
6	90.99	90	0.99
7	176.35	$\pm$180	3.65
8	-89.06	-90	0.94
	Média erro		**1.335**

Na tabela 5.6 são apresentadas as comparações da orientação real e estimada do robô (marcador). O ângulo da orientação foi considerado aumentar no sentido anti-horário em que toma o valor 0º no eixo X, de acordo com o referencial utilizado (ver figura 5.11). Foram testadas apenas orientações facilmente perceptíveis à vista humana, num total de 8 amostras. As primeiras 4 amostras foram capturadas com o marcador próximo da origem do referencial, enquanto que, as restantes 4 correspondem a posições longe da câmara. Nas figuras 5.17 podem ser observados dois exemplos de testes da orientação. Está representada a captura das amostras nº 4 e 5, respectivamente.

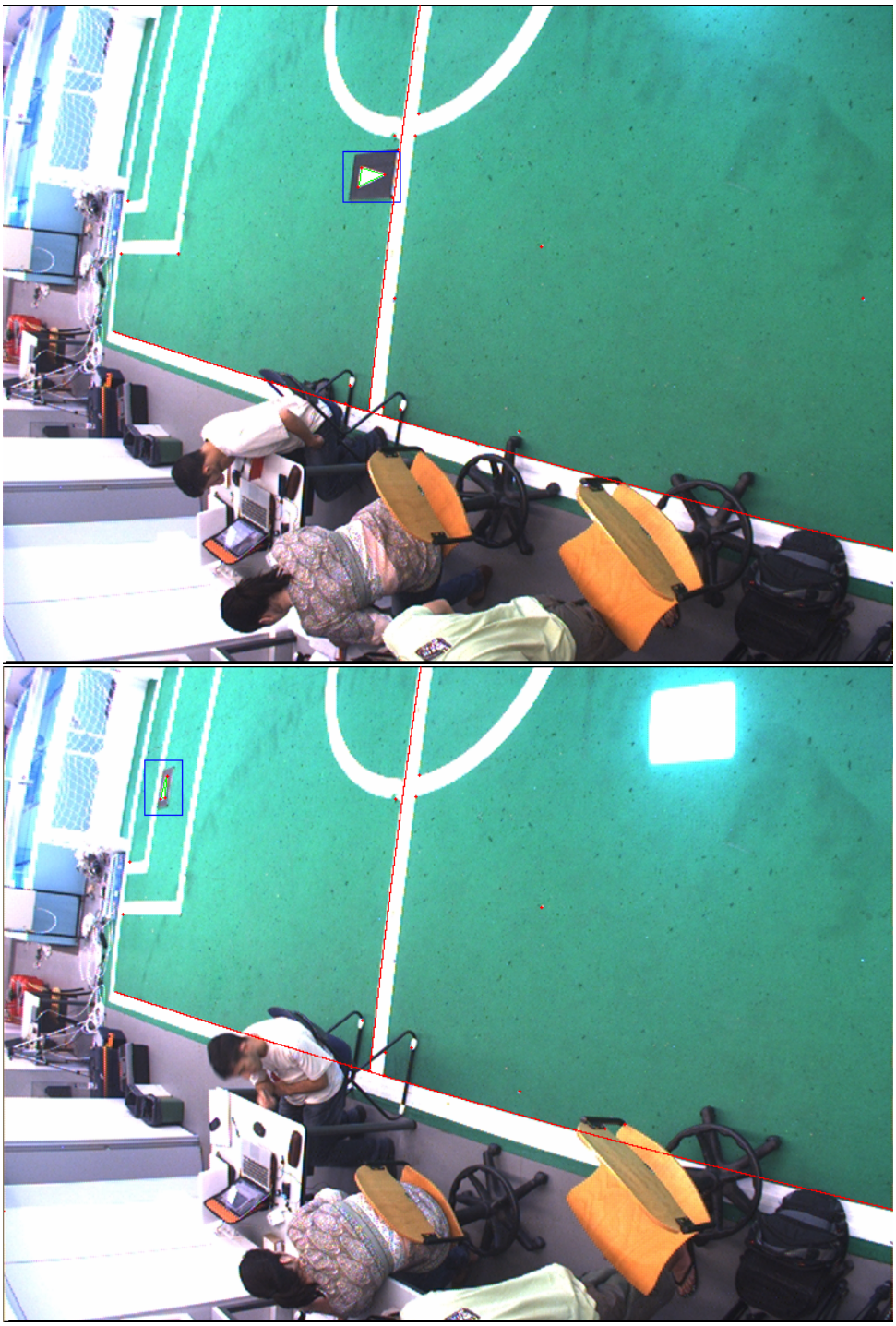

Figura 5.17: Amostragem nº4 e 5 para teste da orientação

É possível observar que na figura 5.18, o triângulo é colocado sobre o eixo X, apresentando uma localização de (135.72, 0.73, 1) que pode ser consultada na interface da aplicação produzida. Pode também ser consultada a orientação que se encontra o marcador, que neste caso apresenta o valor de 0.23º. Por baixo deste valor pode ser visto as dimensões correspondentes a cada lado do

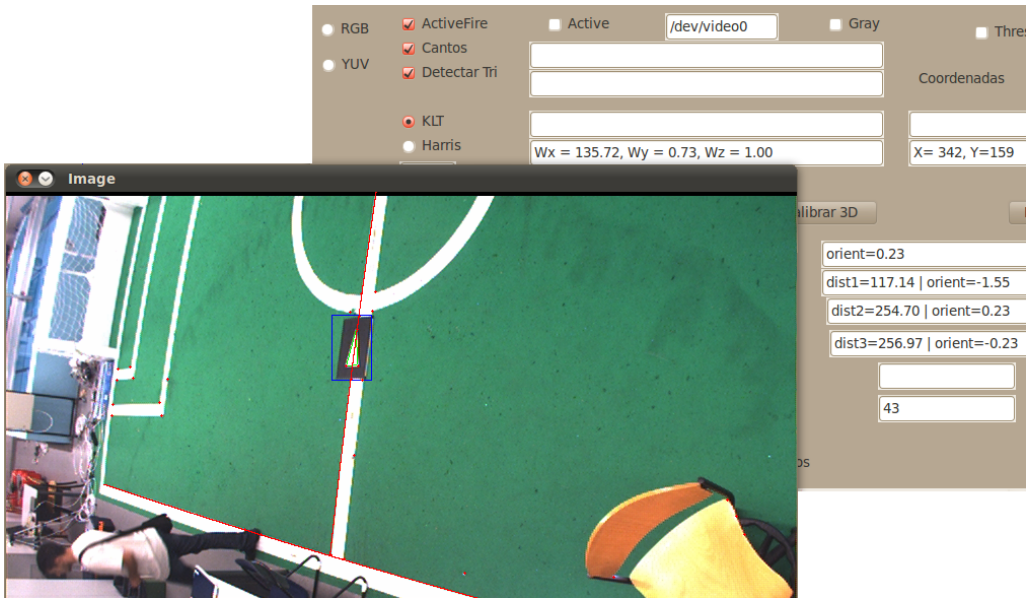

Figura 5.18: Amostragem nº1 para teste da orientação e apresentação da interface do programa desenvolvida

triângulo em milímetros e as suas orientações em radianos. Os comprimentos obtidos para cada lado do triângulo diferem um pouco dos reais (ver tabela 5.1) isto deve-se à localização do canto detectado pelo computador não corresponder exactamente à mesma localização perceptível pelo olho humano, e consequentemente os segmentos de recta tornaram-se ligeiramente mais curtos neste caso. Alguns erros de orientação podem também ser explicados pela imperfeita localização dos cantos, visto que a determinação da orientação do marcador depende directamente da direcção do eixo de simetria do triângulo, e se um canto não se encontrar no píxel exacto, altera logo os resultados desejados.

Capítulo 6

Conclusões e Trabalho Futuro

Neste capítulo será terminado o trabalho, apresentando as conclusões e comentários finais. Serão abordados os resultados obtidos tendo em conta os objectivos propostos no início da dissertação. Serão também apresentadas algumas sugestões para melhoramento do trabalho para futuros projectos.

6.1 Satisfação dos Objectivos

Nesta dissertação foi estudado e implementado um sistema de localização remota baseada em câmaras de vigilância.

Foi desenvolvido um método de detecção de marcadores com a forma de um triângulo isósceles, baseado na detecção de cantos e de orlas. O método inclui uma série de processamento de imagens que levam à detecção inequívoca do marcador desejado. Com base nos resultados obtidos, é concluído que a detecção apresenta bons resultados mesmo quando o marcador apresenta algumas oclusões. Foi realizado uma calibração de câmara necessária para a localização real do robô móvel. Esta calibração permitiu melhorar o sistema de detecção do marcador, devido à possibilidade de testar e confirmar a presença do marcador desejado, com base na medição real do comprimento dos seus lados.

Através dos resultados finais obtidos, pode ser concluído que os objectivos inicialmente propostos foram alcançados. Foi concluído que é possível determinar com boa precisão a posição e orientação de um robô com base em câmaras de vigilância. Esta proposta pode ser então uma boa alternativa para o problema da localização de robôs móveis, principalmente para grandes superfícies onde já exista um sistema de câmaras de vigilância.

6.2 Trabalho Futuro

Serão agora apresentadas algumas sugestões para melhoramento deste trabalho em futuros projectos.

Relativamente ao sistema de detecção do triângulo, poderia tentar ser testado outro método para detecção de orlas, já que o Sobel é a parte do processamento que mais tempo ocupa. Poderia também ser estudado e implementado um sistema de detecção de cantos com precisão sub-píxel, visto que a localização dos cantos está directamente ligada à precisão da posição e orientação do robô. Poderia também ser acrescentado um método de identificação de marcadores, cada robô possuiria um código único de maneira a tornar possível a distinção dos vários robôs detectados.

Poderá também ser melhorada a calibração de câmara com um melhor conjunto de imagens de forma a obter menos incertezas nos parâmetros calculados.

Outro melhoramento a ter em conta seria o de usar várias câmaras para detectar os robôs e criar um sistema externo a estas que utilizasse a informação proveniente de cada câmara para tornar a localização mais robusta. Poderia também fazer uso dessa informação para detectar alguns erros de calibração de alguma câmara.

Anexo A

Convolução de máscaras

A.1 O que é uma máscara?

Uma máscara é uma matriz cujos elementos são chamados de *pesos*. Todas as máscaras possuem uma origem, nas máscaras simétricas normalmente a origem é o píxel central, mas qualquer um pode ser escolhido dependendo do uso desejado.

A.2 Aplicação da máscara na imagem

A aplicação da máscara numa imagem produz um resultado do mesmo tamanho que a imagem original. O procedimento é o seguinte:

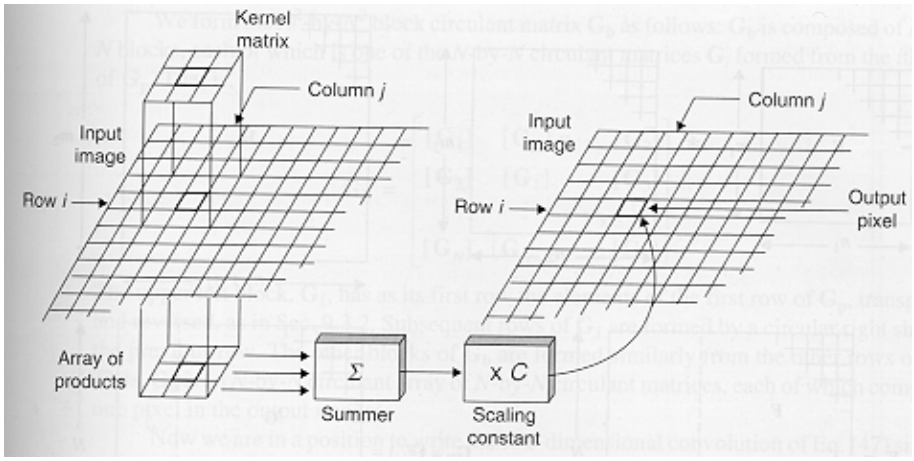

Figura A.1: Procedimento de aplicação de uma máscara (1). Retirado de [32]

- Para cada píxel na imagem, a máscara é colocada em cima, ficando com a origem da máscara sobre esse píxel.
- Os píxeis da imagem que foram sobrepostos pela máscara são então multiplicados pelo *peso* da máscara correspondente.

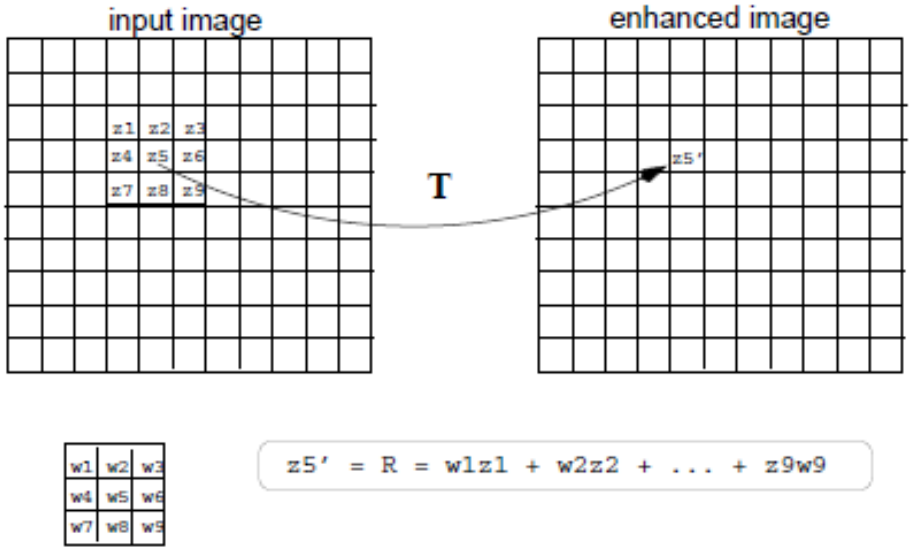

Figura A.2: Procedimento de aplicação de uma máscara (2). Retirado de [32]

- Os resultados são todos somados, e darão origem a um valor que será colocado na imagem de saída na mesma localização do píxel processado na imagem de entrada.

Para ajudar a entender, este procedimento pode ser acompanhado pelas figuras A.1 e A.2.

Anexo B

Conjunto de imagens utilizadas para calibração da câmara

As imagens utilizadas para a calibração da câmara foram as seguintes:

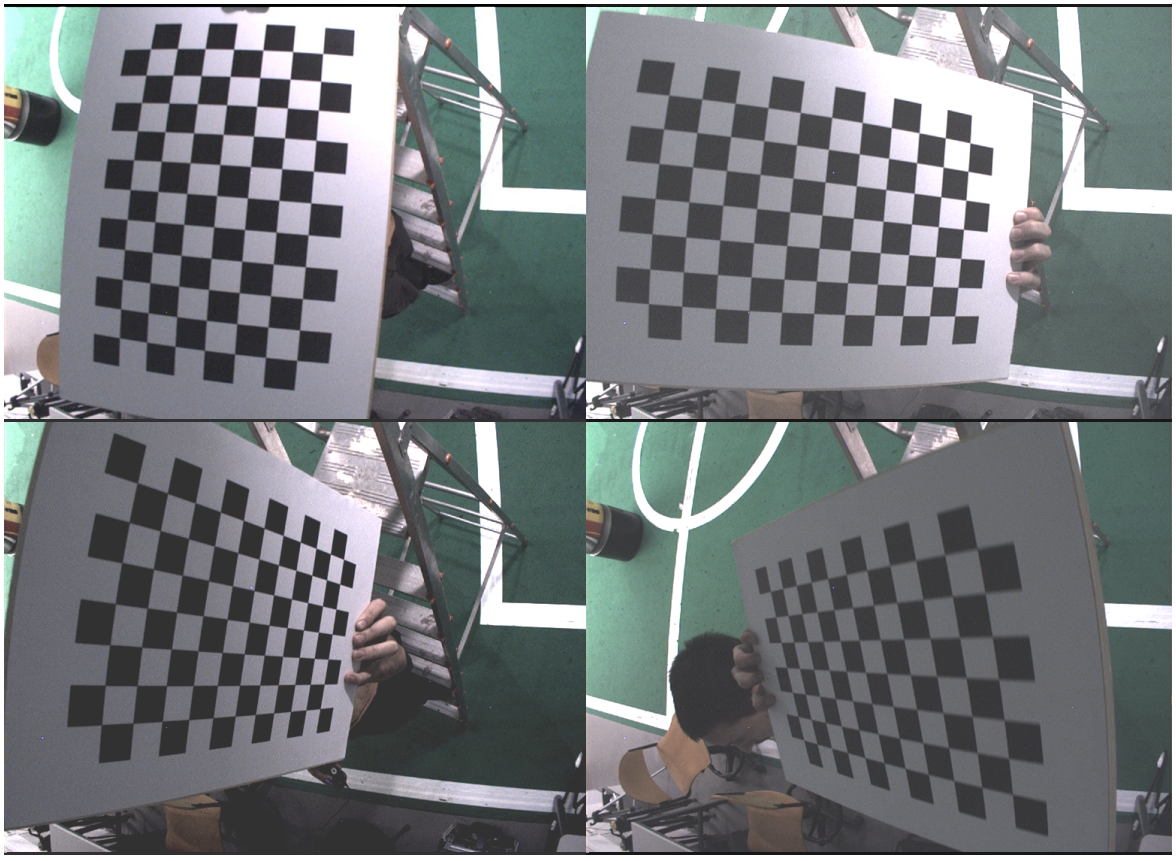

Figura B.1: Imagens utilizadas para calibração da câmara (1)

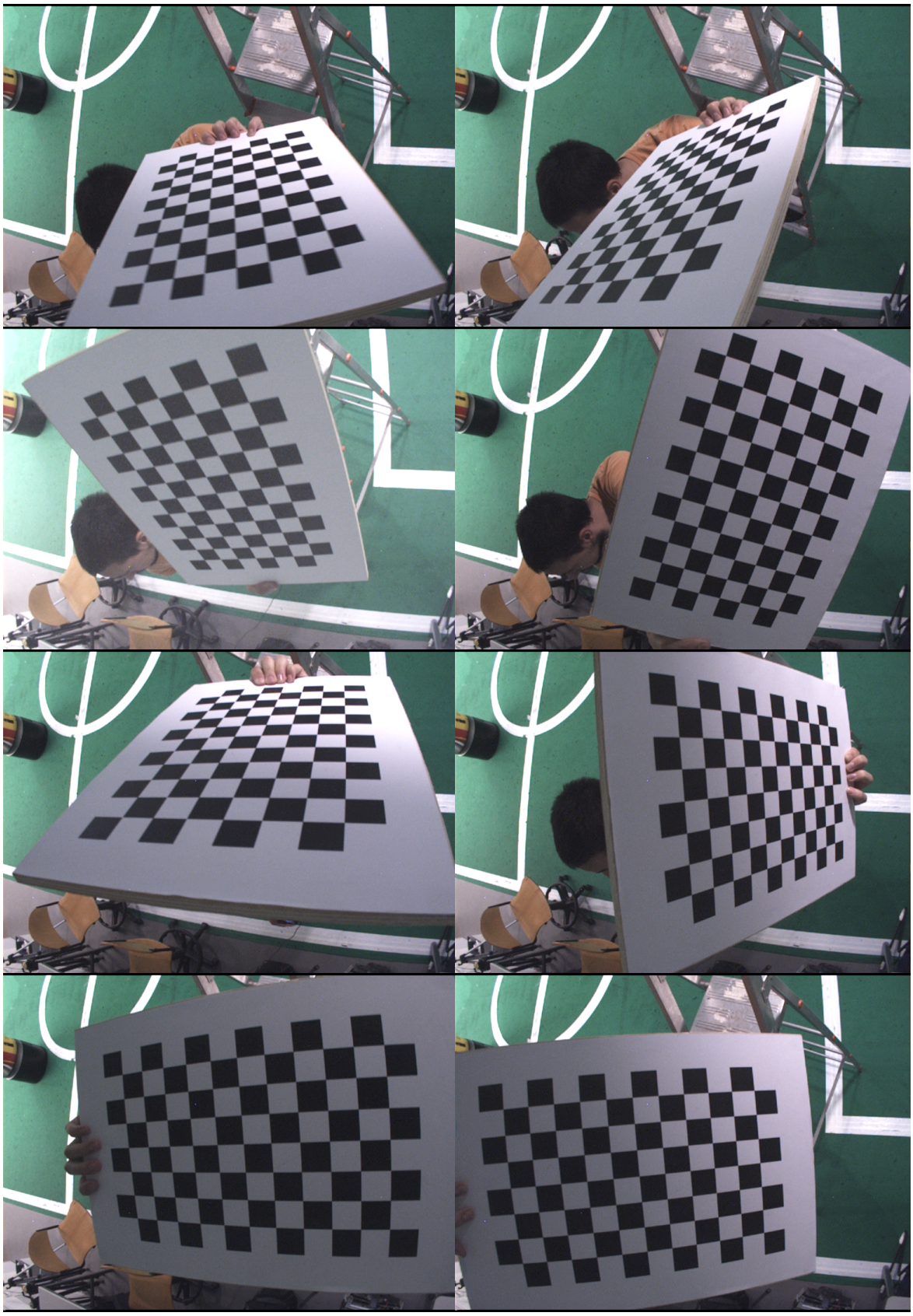

Figura B.2: Imagens utilizadas para calibração da câmara (2)

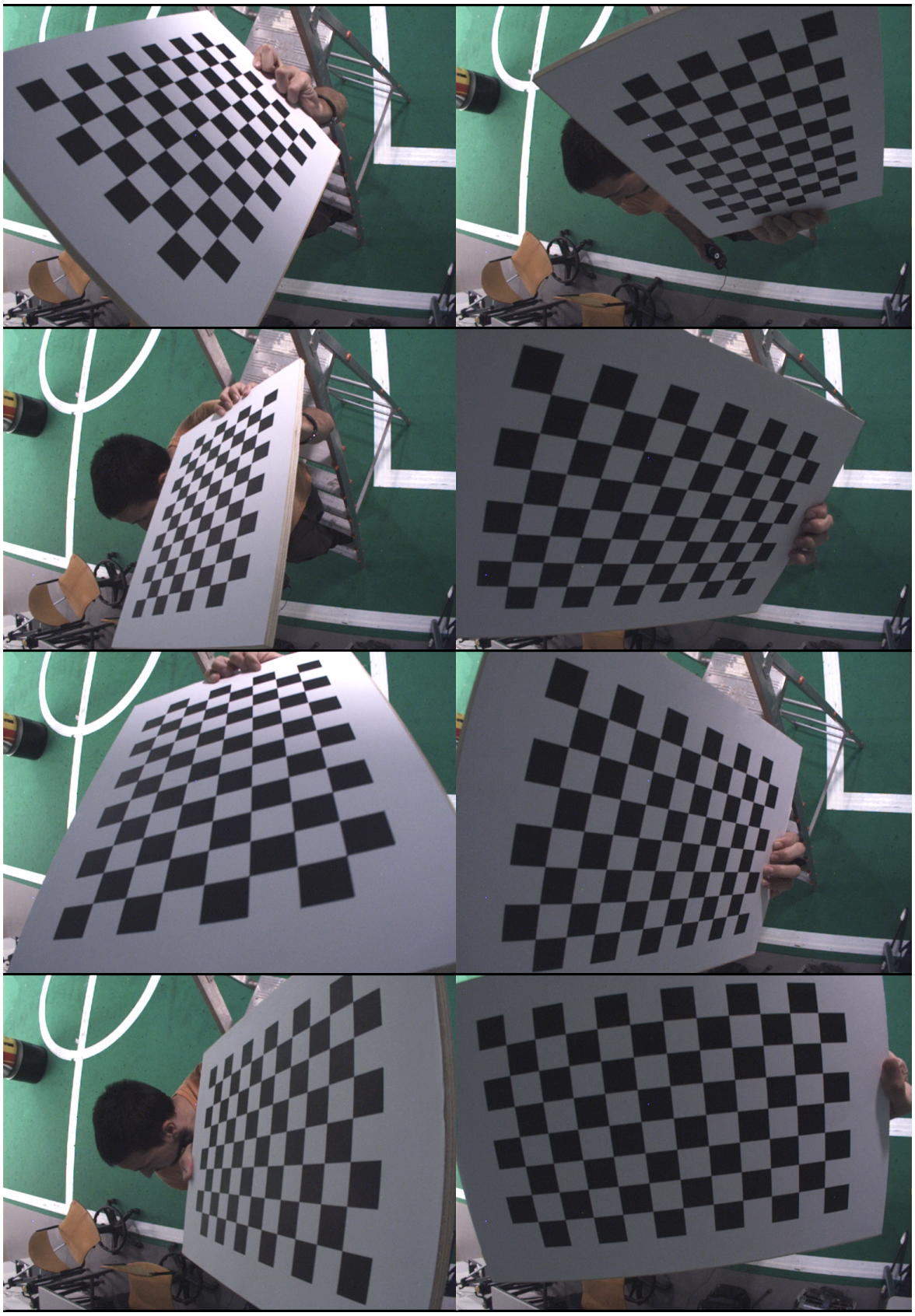

Figura B.3: Imagens utilizadas para calibração da câmara (3)

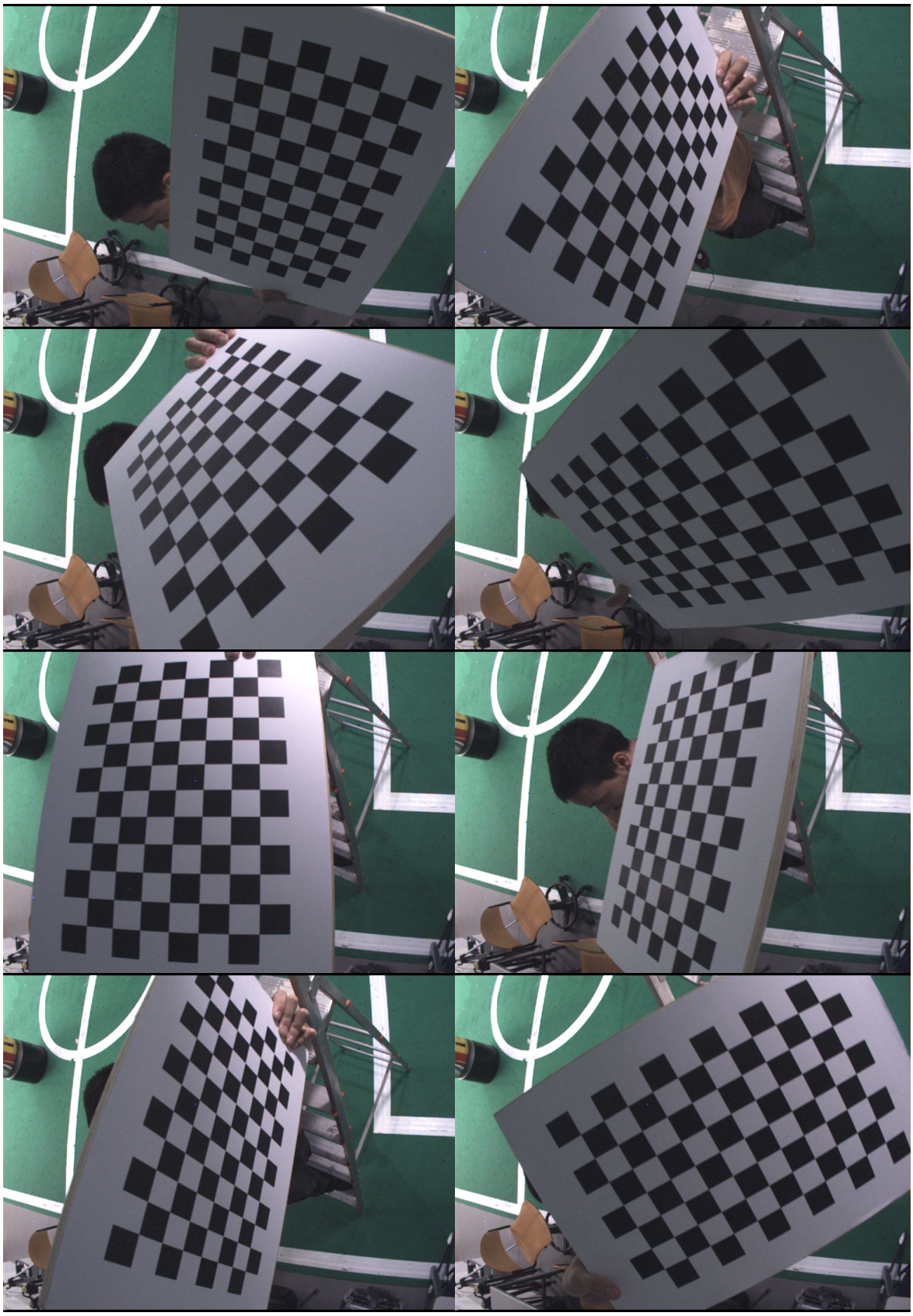

Figura B.4: Imagens utilizadas para calibração da câmara (4)

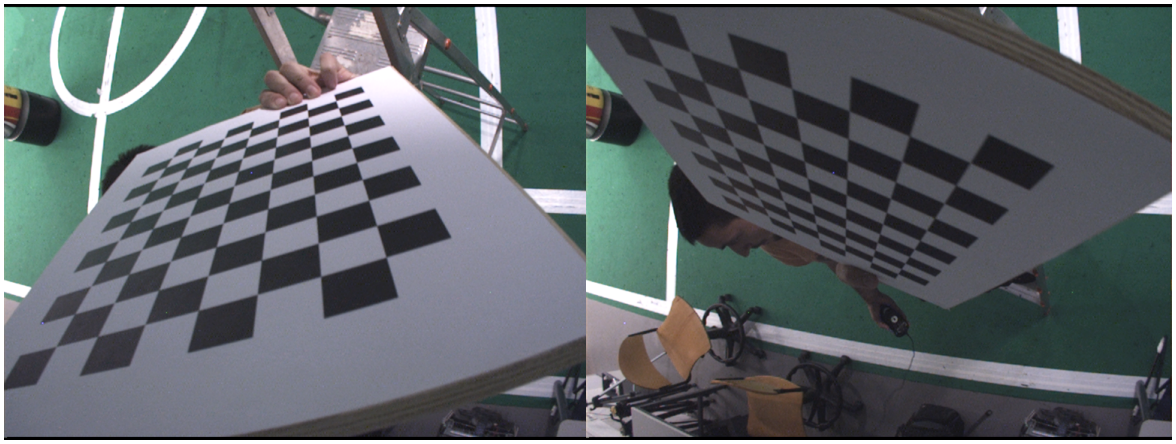

Figura B.5: Imagens utilizadas para calibração da câmara (5)

Referências

[1] Camilo Christo e Carlos Cardeira. Service oriented architecture for mobile robot localization. *Em IEEE Conference on Emerging Technologies and Factory Automation*, 6:888–891, 2007.

[2] Joo-Ho Lee e Hideki Hashimoto. Controlling mobile robots in distributed intelligent sensor network, October 2003.

[3] Drazen Brseie e Hideki Hashimoto. Mobile robots physical agent od intelligent space, 2009.

[4] Vlad Balea e Rikard Eriksson. Indoor navigation with pseudolites (fake gps sat.). Tese de mestrado, University of Linkoping, Janeiro 2005.

[5] Yoshi Hada e Kunikatsu. Multiple mobile robot navigation using the indoor global positioning system (igps). *International Conference on Intelligent Robots and Systems, IEEE*, 2:1005–1010, 2001.

[6] Yoshiro Hada, Edward Hemeldan, Kunikatsu Takase, e Harunori Gakuhari. Trajectory tracking control of a nonholonomic mobile robot using igps and odometry, 2003.

[7] Andry Pinto. Localização e mapeamento na robótica móvel em ambientes não-industriais. Tese de mestrado, Faculdade de Engenharia da Universidade Do Porto, 2010.

[8] Riccardo Cassinis, Fabio Tampalini, e Roberto Fedrigotti. Active markers for outdoor and indoor robot localization, 2005.

[9] DongJu Kim, JongSuk Choi, e Mignon Park. Detection of multi-active markers and pose for formation contro, October 2010.

[10] Charles B. Owen, Fan Xiao, e Paul Middlia. What is the best fiducial?, Setembro 2002.

[11] Alan Mutka, Damjan Miklic, Ivica Draganjac, e Stjepan Bogdan. A low cost vision based localization system using fiducial markers, Julho 2008.

[12] Xiang Zhang, Stephan Fronz, e Nassir Navab. Visual marker detection and decoding in ar systems: A comparative study, 2002.

[13] Mark Fiala. Designing higly reliable fiducial markers, Julho 2010.

[14] Mark Fiala. Artag, a fiducial marker system using digital techniques, Junho 2005.

[15] David Lima. Localização absoluta de robôs móveis em ambientes industriais. Tese de mestrado, Faculdade de Engenharia da Universidade Do Porto, 2010.

[16] Zijian Zhao, Yuncai Liu, e Zhengyou Zhang. Camera calibration with three noncollinear points under special motions, Dezembro 2008.

[17] Paulo Dias. Tsai camera calibration. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/DIAS1/.

[18] Berthold Horn. Tsai's camera calibration method revisited, 2000.

[19] Xiaoqiao Meng, Hua Li, e Zhanyi Hu. A new easy camera calibration technique based on circular points, 2003.

[20] Yihong Wu, Haijiang Zhu, Zhanyi Hu, e Fuchao Wu. Camera calibration from quasiaffine invariance of two parallel circles, 2004.

[21] Koen Douterloigne, Sidharta Gautana, e Wilfried Philips. Fully automatic and robust uav camera calibration using chessboard patterns, Julho 2009.

[22] Valsamis Douskos, Ilias Kalisperakis, e George Karras. Automatic calibration of digital cameras using planar chess-board patterns, 2007.

[23] A V. Douskos, A I. Kalisperakis, A G. Karras, e B E. Petsa. Fully automatic camera calibration using regular planar patterns, 2008.

[24] T. Guttormsen, A. Andersen, J. Holst, M. Keilow, e E. Kjæhr. Edge detection and its implementation in c++, 2008.

[25] Gerhard Roth. Corner (interest point) detection, Winter 2011. COMP 4900C.

[26] Robert Collins. Lecture 06: Harris corner detector, 2007.

[27] Tomas Werner. *Harris Corner Detector*. Center for Machine Perception.

[28] Jihong Shi Zhiyong Ye, Yijian Pei. An adaptive algorithm for harris corner detection. *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference*, páginas 1–4, December 2009.

[29] Dmitrij Csetverikov. Basic algorithms for digital image analysis: a course. Relatório té, Institute of Informatics - Eotvos Lorand University - Budapest, Hungary.

[30] Jean-Yves Bouguet. Camera calibration toolbox for matlab, July 2010. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[31] Armando Jorge Miranda de Sousa. *Arquitecturas de Sistemas Roboticos e Localizacao em Tempo Real Atraves de Visao - Aplicacoes no Dominio do Futebol Robotico*. Tese de doutoramento em engenharia electrotecnica e de computadores, Faculdade de Engenharia da Universidade do Porto, 2003.

[32] Trucco. Area/mask processing methods. Chapter 3.