



**João Pedro Teixeira  
de Figueiredo**

**Calibração de um anel de câmaras a bordo do  
ATLASCAR2**





**João Pedro Teixeira  
de Figueiredo**

**Calibração de um anel de câmaras a bordo do  
ATLASCAR2**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação científica do Doutor Miguel Riem de Oliveira, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Este trabalho teve o apoio financeiro dos projetos UIDB/00481/2020 e UIDP/00481/2020 - FCT - Fundação para Ciência e Tecnologia; e CENTRO-01-0145-FEDER-022083 - Programa Operacional Regional do Centro (Centro2020), no âmbito do Acordo de Parceria Portugal 2020, através do Fundo Europeu de Desenvolvimento Regional



**o júri / the jury**

presidente / president

**Doutor Marco Paulo Soares dos Santos**

Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

vogais / examiners committee

**Doutor Eurico Farinha Pedrosa**

Investigador Doutorado (nível 1), Instituto de Engenharia Eletrónica e Telemática de Aveiro (arguente)

**Doutor Miguel Armando Riem de Oliveira**

Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro (orientador)



## agradecimentos / acknowledgements

Em primeiro lugar, quero agradecer ao meu quase irmão Afonso, uma pessoa que me acompanhou, não só nesta jornada, mas em todas as mais importantes da minha vida, tanto a nível académico como pessoal. Mais uma vez o seu apoio teve um papel fulcral, sendo completamente incansável na motivação e na partilha de conhecimentos que levaram à minha evolução como pessoa e profissional.

Quero deixar uma palavra de agradecimento ao meu orientador, Miguel Riem Oliveira, por apesar de todas as dificuldades encontradas neste percurso nunca ter desistido deste projeto e por sempre me tentar motivar. Um muito obrigado por todos os conhecimentos que me permitiu adquirir e por todo o tempo que esta dissertação lhe tirou.

Por serem pessoas de extrema importância em todo o meu percurso académico, quero deixar uma palavra de apreço aos meus amigos e colegas Diogo Fonseca, Danny Ferreira, Francisco Nolasco e a todos os restantes que aqui não menciono. Todo o apoio e espírito de entreatajuda tornou isto possível. Sempre com um ambiente de grupo fantástico e uma grande amizade entre todos, que certamente levarei comigo para a vida.

Um especial agradecimento também à minha família, que me acompanharam nesta cansativa jornada. A minha mãe que sempre foi incansável e que tanta paciência demonstrou. Às minhas irmãs por todo o apoio (e algumas críticas). À minha avó por mesmo não sabendo muito bem o que se tratava, sempre mostrar entusiasmo pelo meu trabalho. Por fim ao meu pai e à Elsa por mesmo estando longe conseguirem mostrar a sua presença através de um grande incentivo.

Um muito obrigado à minha namorada Ana Teresa por ter começado a fazer parte da minha vida na melhor altura possível, levando-me sempre a procurar ser mais a nível pessoal e profissional. Ajudando-me sempre a ultrapassar os piores momentos.

Por fim, mas não menos importantes, e porque a motivação e o animo com a vida não advêm só do trabalho que se desempenha, um muito obrigado aos meus amigos Bruno Matos, João Neves, Manuel Rocha e João Jubero, que me acompanharam em inúmeros momentos de lazer e que muito contribuíram na evolução que traz a realização de um curso académico.



**palavras-chave**

Anel de Câmaras; Calibração Extrínseca; Veículo Autónomo; Otimização; Calibração Intrínseca; ROS; Calibração Simultânea de Sensores

**resumo**

Cada vez mais um sistema robótico inteligente recorre à visão artificial de forma a recolher informações do cenário em que se encontra. A capacidade de decisão nas suas ações está diretamente relacionada com a qualidade da recolha de dados do ambiente que rodeia o robô. Um veículo autónomo insere-se então nesta categoria de sistemas, uma vez que lhe é requerido que tome as decisões inerentes à condução. Devido à sua complexidade e a precisão requerida, por ser um sistema crítico, onde uma falha pode pôr em risco vidas humanas, torna-se necessário aumentar o número de sensores de percepção visual. De forma a que seja possível cruzar a informação de todas as câmaras é necessário que estas estejam calibradas extrinsecamente. Este modo de calibração passa pelo conhecimento da posição e orientação de todos os sensores de um sistema em relação aos restantes ou a um referencial global. Deste modo, o presente projeto visa instalar e realizar a calibração extrínseca de um anel de câmaras a toda a volta do ATLASCAR2 (projeto de um veículo autónomo desenvolvido pela Universidade de Aveiro). É realizado um processo semi automático, desde a recolha de dados, utilizando ROS, até à calibração extrínseca de todas as câmaras, simultaneamente, em relação a um referencial global, efetuando ainda a calibração intrínseca de cada sensor. O processo utiliza como auxílio um padrão de xadrez e consegue realizar a calibração mesmo quando o padrão de auxílio não é detetado por todas as câmaras ao mesmo tempo. Esta metodologia obteve resultados bastante positivos (erro médio final inferior a 1 píxel) e mostrou uma boa robustez em todas as situações em que foi testado. Este trabalho permite então obter uma calibração bastante precisa de sistemas robóticos em que é impossível a partilha de informação por todos os sensores constituintes.



**keywords**

Camera Ring; Extrinsic Calibration; Autonomous Vehicle; Optimization; Intrinsic Calibration; ROS; Simultaneous Sensor Calibration

**abstract**

More and more an intelligent robotic system uses artificial vision to collect information from the scenario in which it is located. The ability to decide on its actions is directly related to the quality of data collection from the environment surrounding the robot. An autonomous vehicle then falls into this category of systems, since it is required to make the decisions inherent to driving. Due to its complexity and the required precision, being a critical system, where a failure can put human lives at risk, it becomes necessary to increase the number of visual perception sensors. To be able to cross information from all the cameras, they must be calibrated extrinsically. This calibration model involves the knowledge of the position and orientation of all the sensors of a system relative to the others or a global reference. Thus, this project aims to install and perform the extrinsic calibration of a ring of chambers all around the ATLASCAR2 (design of an autonomous vehicle developed by the University of Aveiro). A semi-automatic process is performed, from data collection, using ROS, to the extrinsic calibration of all the cameras, simultaneously, relative to a global referential, also performing the intrinsic calibration of each sensor. The process uses a chess pattern as an aid and can perform the calibration even when the aid pattern is not detected by all the chambers at the same time. This methodology obtained very positive results (final average error under 1 pixel) and showed good robustness in all situations where it was tested. This work leads to a very precise calibration of robotic systems in which it is impossible to share information by all the constituent sensors.



# Conteúdo

<b>Conteúdo</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tabelas</b>	<b>v</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Calibração . . . . .	3
1.2 Otimização . . . . .	3
1.3 ATLASCAR2 . . . . .	4
1.4 Objetivos . . . . .	5
1.5 Contextualização . . . . .	6
1.6 Estrutura do documento . . . . .	7
<b>2 Estado da Arte</b>	<b>9</b>
2.1 Calibração entre dois sensores . . . . .	10
2.2 Calibração de sistemas com mais de dois sensores . . . . .	12
<b>3 Solução proposta</b>	<b>17</b>
3.1 Recolha e estruturação de dados . . . . .	25
3.1.1 ROS - <i>Robot Operating System</i> . . . . .	25
Nós, tópicos e mensagens . . . . .	25
RQT . . . . .	26
Rviz . . . . .	26
Pacotes ROS . . . . .	28
3.1.2 Calibração Intrínseca . . . . .	28
3.1.3 Ficheiro Json . . . . .	29
Informação dos sensores . . . . .	29
Configuração da calibração . . . . .	30
Informações das coleções . . . . .	31
3.2 Calibração Extrínseca . . . . .	32
3.2.1 Parâmetros a otimizar . . . . .	33
3.2.2 Parâmetros Iniciais . . . . .	39
3.2.3 Função Objetivo . . . . .	49
Projeção dos pontos na imagem . . . . .	51
Cálculo dos residuais . . . . .	54
3.2.4 Otimizador . . . . .	55

<b>4</b>	<b>Resultados</b>	<b>57</b>
4.1	Calibração de sistema com duas câmaras . . . . .	57
4.1.1	Parâmetros a otimizar . . . . .	58
4.1.2	Comparação entre função de erro calculada através da distância euclidiana no mundo 3D e da distância entre píxeis . . . . .	60
	Cálculo do erro no mundo 3D . . . . .	61
	Comparação utilizando apenas um par de imagens . . . . .	63
	Comparação ao longo de 10 calibrações com pares de imagens diferentes . . . . .	65
	Comparação utilizando 8 coleções dentro da mesma calibração . . . . .	68
4.2	Calibração de sistema com três câmaras . . . . .	70
4.2.1	Parâmetros a otimizar . . . . .	70
4.2.2	Comparação entre função de erro calculada através da distância euclidiana no mundo 3D e da distância entre píxeis . . . . .	72
4.3	Calibração de sistema com 5 câmaras . . . . .	74
4.3.1	Variação do número de coleções utilizadas na calibração . . . . .	75
4.3.2	Variação na classificação de coleção válida . . . . .	79
4.3.3	Variação dos parâmetros iniciais da otimização . . . . .	80
<b>5</b>	<b>Conclusão e Trabalho Futuro</b>	<b>85</b>
	<b>Bibliografia</b>	<b>87</b>

# Lista de Figuras

1.1	ATLASCAR2 . . . . .	5
1.2	ROS Camera Calibrator example . . . . .	6
3.1	Exemplo de imagem que captura o plano de xadrez na sua totalidade . . . . .	18
3.2	Fluxo dos vários processos na recolha e estruturação de dados. . . . .	20
3.3	Fluxo representativo das duas etapas do trabalho, bem como a ligação entre as mesmas. . . . .	20
3.4	Fluxo de processo de Otimização. . . . .	22
3.5	Classificação de uma coleção, consoante o número de imagens em que o xadrez é detetado . . . . .	22
3.6	Esquema do estudo realizado para uma câmara numa determinada coleção. Os pontos verdes representam os cantos do padrão detetado diretamente na imagem (sendo estes considerados reais) e os azuis os pontos projetados através das transformações geométricas. Através dos residuais (distâncias entre pontos reais e os projetados - linhas vermelhas tracejadas), o otimizador altera os seus parâmetros (que levam às matrizes de transformações) de forma a reduzir a distância entre os pontos. . . . .	23
3.7	Imagens das duas Câmaras . . . . .	24
3.8	Esquema do RQT que demonstra os nós (elipses), tópicos (retângulos exteriores) e mensagens (retângulos interiores), que são utilizados no projeto. . . . .	26
3.9	Imagem demonstrativa do ambiente Rviz que é apresentado ao utilizador. É neste que é requerida a sua intervenção de forma a captar imagens das câma- ras, realizar a calibração intrínseca das mesmas e criar o ficheiro JSON com informações relevantes e necessárias para a calibração extrínseca. . . . .	27
3.10	Exemplo de informação dos sensores no ficheiro JSON . . . . .	30
3.11	Exemplo de informação da configuração da calibração no ficheiro JSON . . . . .	31
3.12	Exemplo de informação das coleções no ficheiro JSON . . . . .	32
3.13	Representação da função objetivo . . . . .	33
3.14	Sequência de calibrações par a par . . . . .	34
3.15	Arranjos para sistemas com múltiplas câmaras . . . . .	35
3.16	Arranjos de 5 sensores com recurso às combinações entre os mesmos . . . . .	36
3.17	Arranjos de 5 sensores com referencial global . . . . .	37
3.18	Comparação do número de transformações a otimizar . . . . .	38
3.19	Multiplicação de transformações geométricas . . . . .	43
3.20	Multiplicação de transformações geométricas com matriz inversa . . . . .	44
3.21	Representação do ambiente do <i>first guess</i> para uma câmara . . . . .	45

3.22	Representação do ambiente do <i>first guess</i> para um xadrez . . . . .	47
3.23	Imagens utilizadas para calcular <i>first guess</i> . . . . .	48
3.24	Fluxo representativo da seleção de imagens para o cálculo de residuais . . . . .	50
3.25	Fluxo representativo da passagem de informação entre as classes . . . . .	51
3.26	Transformação entre referenciais 2D . . . . .	52
3.27	Projeções do Xadrez . . . . .	54
4.1	Esquema do sistema stereo . . . . .	58
4.2	Imagens das duas Câmaras . . . . .	59
4.3	Exemplo da utilização da função de deteção do xadrez na imagem de uma câmara . . . . .	60
4.4	Comparação de Método 3D e píxeis com um par de imagens . . . . .	64
4.5	Posições finais da projeção do xadrez na calibração de 2 câmaras . . . . .	65
4.6	Exemplos de imagens utilizadas nas 10 calibrações . . . . .	65
4.7	Comparação do erro médio final ao longo de 10 calibrações . . . . .	66
4.8	Comparação do número de iterações ao longo de 10 calibrações . . . . .	67
4.9	Variação dos residuais (em píxeis) com a calibração utilizando 8 coleções . . . . .	69
4.10	Comparação do erro médio final variando o número de coleções . . . . .	73
4.11	Comparação do número total de iterações variando o número de coleções . . . . .	73
4.12	Erro médio final consoante o número de coleções da calibração . . . . .	76
4.13	Representação dos residuais da calibração com 30 coleções . . . . .	77
4.14	Representação do erro de cada imagem da calibração com 20 coleções . . . . .	78
4.15	Representação dos residuais na calibração onde é alterada a posição inicial dos xadrezes . . . . .	82
4.16	Comparação dos valores iniciais e finais de duas calibrações . . . . .	83

# Lista de Tabelas

3.1	Matriz de detecções do xadrez, onde "0" representa que o padrão não é detetado e "1" que existe detecção. Cada coluna corresponde a um sensor e cada linha a uma coleção . . . . .	40
3.2	Matriz de interesse criado de forma a estabelecer ligação de todos os sensores ao que inicialmente se encontra no referencial global. . . . .	41
4.1	Erro médio final usando 1 par de imagens . . . . .	64
4.2	Média erro médio final usando 10 pares de imagens . . . . .	67
4.3	Variação dos residuais com a calibração utilizando 8 coleções . . . . .	70
4.4	Erro médio final e número total de iterações para 8 coleções . . . . .	74
4.5	Valores dos erros médios iniciais e finais, bem como o número total de iterações, de calibrações com diferentes números de coleções. . . . .	75
4.6	Valores dos erros médios iniciais e finais, nos diferentes eixos, consoante o número de imagens utilizadas para calcular residuais na função objetivo. . . . .	78
4.7	Valores dos erros médios iniciais e finais, bem como o total de iterações, da calibração aquando da variação do mínimo de imagens (fazendo variar o número de residuais) a detetar o xadrez para que a coleção seja considerada válida. . . . .	79
4.8	Valores dos erros médios iniciais e finais, bem como o total de iterações, da calibração aquando da variação do número exato de câmaras a detetar o xadrez para que a coleção seja considerada válida. . . . .	80
4.9	Valores obtidos como erro médio inicial e final modificando os parâmetros iniciais, de forma a os afastar dos valores considerados reais, para o estudo da sua influência no resultado final. . . . .	81



# Capítulo 1

## Introdução

Um robô é um mecanismo capaz de realizar tarefas de forma automática. Nos primeiros exemplares, todas as ações teriam de ser programadas pelo homem. O seu propósito era a realização de tarefas repetitivas, onde pouca informação era necessária e os processos simples. Estes acabaram por ter um grande impacto no mundo da indústria, onde os operários acabaram substituídos, por estes mecanismos, em tarefas repetitivas ou exigentes/desgastantes fisicamente, conseguindo ainda aumentar a rapidez e eficiência dos processos. Os robôs industriais, com o passar do tempo e desenvolvimento da área, atingem grandes precisões nos seus processos, porém, caso algo no seu espaço operacional se altere -um operário entrar no seu raio de ação (criadas grades de segurança para evitar acidentes), falta de material para a realização ou este fora do local esperado, etc.- o robô não teria capacidade de se adaptar e continuaria a realizar as ações da mesma forma. Assim, para voltar a operar de forma correta, era necessária a intervenção humana, mais uma vez.

Caso seja pretendido que um mecanismo realize mais do que uma tarefa, considerando uma tarefa como uma sucessão de movimentos previamente programados, é necessário que ele tenha já uma percepção do que o rodeia. Através da recolha de informação, o robô é capaz de alterar o seu movimento/tarefa a desempenhar. Este tipo de máquina permite a execução de processos mais complexos e não apenas processos repetitivos, pelo facto de conseguir tomar decisões. Com o intuito de tornar os mecanismos mais "inteligentes", isto é, capazes de detetar características do cenário e ter a capacidade de se adaptarem, começou-se a recorrer à instalação de sensores nos robôs.

De forma a recolher informações de diferentes naturezas, existem sensores de diferentes modalidades. Por exemplo, com a intenção de detetar cores ou imagem de algum objeto do cenário são utilizadas câmaras RGB; caso o objectivo seja detetar objetos pela sua temperatura, recorre-se a câmaras térmicas ou se a finalidade for detetar a distância a um obstáculo ou objeto é através de um LIDAR. Cada mecanismo, consoante o seu propósito, pode conter um ou mais sensores, da mesma modalidade ou não. Isto permite-lhe combinar informação recolhida de diferentes naturezas para que a sua leitura do ambiente, bem como a sua decisão, seja a correta. Um robô com esta capacidade é considerado inteligente, por ser capaz de se adaptar ao ambiente em que se encontra, continuando a realizar as suas tarefas.

Um tipo de um robô inteligente é por exemplo um veículo autónomo. A este é requerido que tenha a percepção do ambiente em que se encontra de modo a tomar todas as decisões

inerentes à condução de forma correta e segura.

Os estudos relacionados com veículos autônomos, neste momento, fazem parte de uma das áreas de maior interesse e investigação da indústria automóvel, sendo que praticamente todas as marcas de carros e grandes empresas tecnológicas estão já a desenvolver várias abordagens para atingir esse objetivo [15]. Atualmente, vários veículos já têm incorporados sistemas de auxílio à condução e em alguns casos, sob certas condições, já conseguem assumir o controlo da condução.

Segundo a NHTSA (National Highway Traffic Safety Administration), um departamento federal dos Estados Unidos da América, existem diferentes níveis de automatização de um veículo<sup>1</sup>:

- **Nível 0:** Sem qualquer tipo de automação no veículo, o condutor deve realizar todas as tarefas relativas à condução e à deteção de informação do ambiente exterior.
- **Nível 1:** O veículo é na mesma controlado pelo condutor, mas já possui algumas assistências a nível da condução.
- **Nível 2:** O veículo já é capaz de controlar a velocidade e a direção consoante algumas condições, mas é necessário que o condutor esteja comprometido a controlar a condução o ambiente exterior.
- **Nível 3:** O veículo já possui um bom nível de automação, sendo que o condutor deve estar sempre pronto a assumir a condução do mesmo, mas já não tem a necessidade de controlar o ambiente exterior enquanto o veículo inteligente tem o controlo dele mesmo.
- **Nível 4:** Existe um elevado nível de automação, o carro é capaz de realizar todas as ações e tomar decisões por ele próprio consoante as circunstâncias em que se encontra. O ocupante do veículo pode optar por controlar o mesmo.
- **Nível 5:** O veículo é totalmente autónomo sob quaisquer circunstâncias. Tem capacidade de realizar todas as tarefas afetas à condução e detetar todas as informações de relevo do cenário. Tal como no nível anterior, o ocupante pode optar por controlar o carro.

Com o nível máximo de automação num veículo em vista, a recolha de toda a informação exterior torna-se imprescindível. Se a máquina é capaz de operar em quaisquer condições é então necessário que saiba reagir a todas as situações rodoviárias que lhe possam surgir, sendo estas esperadas ou não. Numa condução quotidiana, os espelhos retrovisores é algo que utilizamos com bastante frequência para termos a percepção do que nos rodeia. Seja na realização de um estacionamento paralelo, um estacionamento perpendicular, mudar de faixa de rodagem ou qualquer tipo de manobra, etc. baseamos sempre o movimento do veículo na informação que recolhemos dos referidos espelhos. Caso algum objeto surja das laterais do carro, ou algo invulgar aconteça, ao olhar pelas janelas conseguimos analisar a situação e reagir de acordo com isso. Então, de modo a que um veículo autónomo consiga aperceber-se destas situações ou realizar tarefas básicas relativas à condução, torna-se relevante que este consiga obter informação de todos os ângulos ao seu redor. Assim, ter uma visão e uma percepção a toda a sua volta é algo que facilita a tomada de decisão.

---

<sup>1</sup>[https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13069a-ads2.0\\_090617\\_v9a\\_tag.pdf](https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf)

Um veículo inteligente é um sistema que integra vários sensores. Isto permite que daí seja retirada bastante e diferente informação. O objetivo do mecanismo é então poder tratar e juntar o que recebe dos seus sensores. Caso o carro possua câmaras em toda a sua volta, cada uma delas observa a sua parte do cenário. A correta junção das imagens retiradas permite que seja feita a deteção de um objeto, de uma característica da via, etc. e a sua posição em relação ao veículo. Porém, para que a fusão seja possível, cada uma das câmaras necessita de ter conhecimento da posição e orientação das restantes. O processo de estimar as posições relativas é denominado de calibração. É um processo exigente pela obrigatoriedade de um resultado preciso. Uma calibração extrínseca entre dois sensores pouco precisa pode resultar num cruzamento errado de informação, levando a uma má decisão por parte do robô.

## 1.1 Calibração

Num sistema robótico existem vários sensores, podendo eles ser de variados tipos, que funcionam de forma independente uns dos outros. Porém o objectivo é a junção das várias informações disponibilizadas para assim obter a melhor percepção possível da realidade ao seu redor. Isto é algo fundamental num robô que seja completamente autónomo, para que a decisão/acção seja a melhor e mais ponderada.

Para obter a informação de forma precisa é necessário que os sensores estejam calibrados entre si. Calibrar é então estimar a transformação geométrica de um referencial para outro, Equação 1.1, ou seja, calcular a variação de posição e orientação entre os sensores. Estes referenciais podem ser os dos sensores ou de algum elemento de interesse no cenário, por exemplo, um xadrez de calibração.

$${}^{\text{referencial 1}}\mathbf{T}_{\text{referencial 2}} = \begin{bmatrix} r11 & r12 & r13 & tx \\ r21 & r22 & r23 & ty \\ r31 & r32 & r33 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.1)$$

No caso de um sistema com mais do que dois sensores, a transformação calculada pode ser uma sucessão de transformações, podendo existir um sensor de referência em que as posições e orientações dos restantes são calculadas em relação a ele.

## 1.2 Otimização

Uma ferramenta de otimização tem como principal objetivo encontrar os valores ideais para os parâmetros desejados, para minimizar ou maximizar uma função. Geralmente, quando o propósito de uma otimização é minimizar o valor de uma função objetivo, esta intitula-se de função de custo. A otimização consiste num processo iterativo que faz variar os parâmetros de entrada dessa mesma função, consoante os valores que ela retorna, até estes valores (por exemplo, o valor do erro em caso de tentativa de minimizar a função) atingirem um patamar pretendido, caso isso seja possível.

A calibração extrínseca, entre dois sensores de um sistema robótico complexo, é possível quando estes possuem informação em comum. Neste projeto, por exemplo, será utilizado

um plano com padrão em xadrez, sendo necessário que este seja detetado por duas ou mais câmaras (representando a informação partilhada que permite avaliar a pose de cada sensor). Através de um processo de optimização é estimada a matriz de transformação (parâmetros a otimizar), onde a função do erro é a diferença entre a informação projetada e a considerada real (*ground truth*).

No processo de optimização, os parâmetros a otimizar serão portanto os valores da matriz de transformação. Os referentes à rotação ( $r_{11}$ ,  $r_{12}$ , ...,  $r_{33}$  na Equação 1.1) e os referentes à translação ( $t_x$ ,  $t_y$ ,  $t_z$  na Equação 1.1). Para diminuir a possibilidade de erros, utiliza-se uma função do OpenCV<sup>2</sup>, o Rodrigues, que simplifica a matriz de rotação para um vetor de 3 elementos. Ao usar a rotação na notação Rodrigues, o otimizador é impedido de atribuir valores que desnormalizem a matriz de rotação. Isto permite ainda diminuir de 12 para 6 o número de parâmetros a utilizar na otimização.

### 1.3 ATLASCAR2

O ATLASCAR2 é um projeto desenvolvido no Departamento de Mecânica da Universidade de Aveiro [35]. Tem como objetivo final tornar um Mitsubishi Miev um veículo autónomo.

Por ser um robô que deverá ser capaz de tomar as próprias decisões, deve estar munido de vários e diferentes sensores, que lhe permitam ter a melhor percepção possível da realidade à sua volta. Atualmente, o ATLASCAR2, Figura 1.1, possui apenas dois laser 2D LIDAR e duas câmaras RGB na parte frontal do carro, não sendo estes suficientes para que o veículo inteligente realize as tarefas e funções, acima descritas, que são preponderantes para o correto desempenho deste tipo de sistemas.

É então necessário proceder à melhoria do sistema, incorporando no mesmo um anel de câmaras.

---

<sup>2</sup>[https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html)



Figura 1.1: Imagem real do ATLASCAR2

---

## 1.4 Objetivos

Como principal objectivo da dissertação tem-se a instalação de um anel de câmaras no ATLASCAR2. Um anel é então um conjunto de  $N$  câmaras ligadas no mesmo sistema. Neste caso, a finalidade é de conseguir cobrir, através da visão artificial, toda a área que rodeia o veículo, para obter a maior informação possível em todos os ângulos.

Na Figura 1.2, é possível visualizar um exemplo de um anel de câmaras instalado no ATLASCAR2 e da sua utilidade ao possibilitar a deteção de informação em todo o seu redor. Após um processo de calibração das mesmas, o objetivo será a construção de uma imagem panorâmica como resultado da combinação das várias imagens retiradas.

Associado ao principal objetivo do trabalho surgem possíveis problemas/dificuldades. Estes são criados pelo elevado número de sensores instalados e do seu propósito. Para que o sistema esteja totalmente calibrado é necessário estimar a posição de todas as câmaras em relação às outras ou a um referencial comum. De forma a diminuir o erro associado à calibração, diminuir o peso computacional e tornar o processo mais simples, deve ser feita apenas uma calibração envolvendo já todos os sensores do anel. Isto leva a que numa calibração hajam várias câmaras sem informação em comum entre si, por exemplo, no caso de uma instalada na frente do carro e uma na traseira.

Atendendo aos objetivos já mencionados no capítulo anterior, o presente projeto focar-se-á na calibração de múltiplas câmaras RGB: sensores constituintes do anel também já referenciado. Como tal, é necessário um conhecimento dos diversos métodos de calibração entre sensores desta específica modalidade.

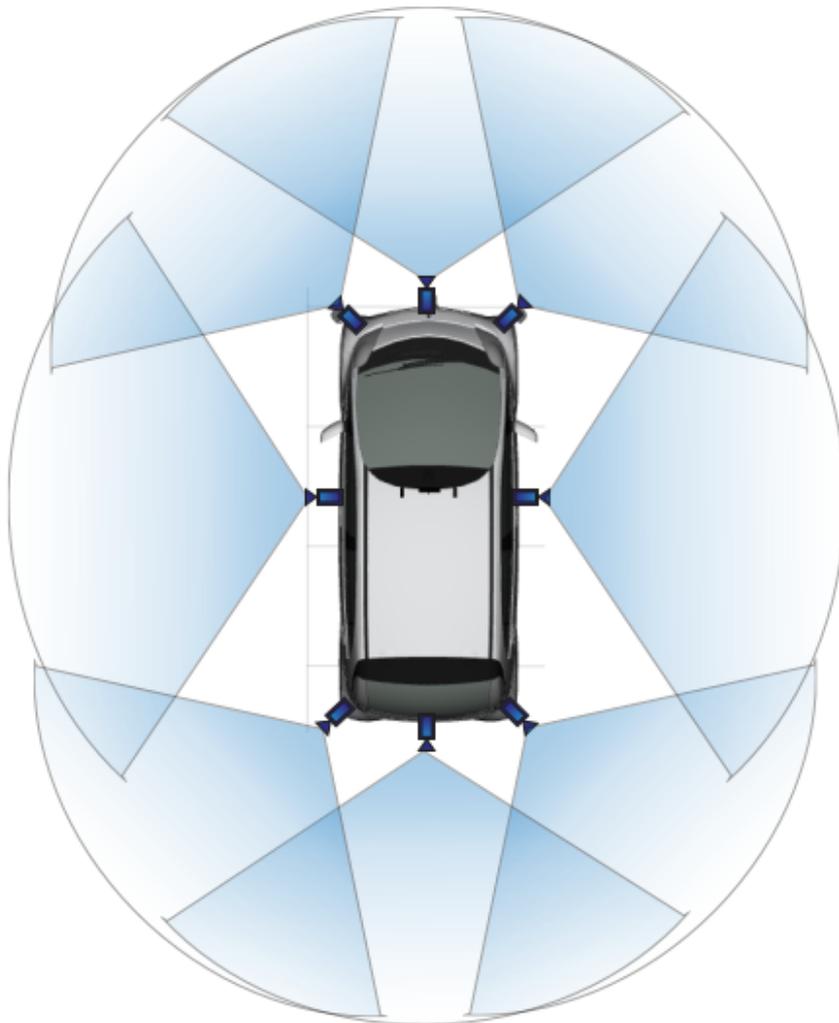


Figura 1.2: Representação do Anel de câmaras a bordo do ATLASCAR2

---

## 1.5 Contextualização

Esta secção tem como objetivo dar a conhecer ao leitor as circunstâncias em que este trabalho foi desenvolvido e este documento escrito.

O ano de 2020 ficou marcado pela grande pandemia, que todo o mundo afetou, com o vírus SARS-Cov2, o novo corona vírus. Por esta razão, todo o trabalho teve de ser adaptado à nova realidade de forma a tornar possível a sua realização. Deste modo, o objetivo que seria a instalação de 8 câmaras a bordo do veículo ATLASCAR2, a nível prático e experimental, foi alterado para 5 câmaras em ambiente domiciliário ou em laboratório. Todo o propósito se mantém, quer isto dizer que, todos os cuidados serão tidos em conta como se de um veículo autónomo se tratasse, uma vez que, o objetivo passa por ser possível aplicar este trabalho a

essa situação sem grandes alterações.

## 1.6 Estrutura do documento

Este documento encontra-se dividido em 5 capítulos, de forma a tornar todo o processo claro e perceptível para o leitor.

No primeiro capítulo, Introdução, que termina precisamente nesta secção, é apresentado o problema da calibração de um anel de câmaras que a dissertação visa resolver.

No segundo, Estado da Arte, são apresentados vários projetos já realizados na área da calibração, com diferentes métodos de aplicação e diferentes problemas, de forma a obter uma percepção do que já existe nesta área bem como as suas vantagens que podem ser aplicadas ao presente projeto e pontos a melhorar ou a ter atenção.

No terceiro capítulo, Solução Proposta, o mais extenso deste documento, é explicada a forma como o problema será resolvido, já tendo por base procedimentos retirados do capítulo anterior. Inicialmente será explicado de forma geral todo o procedimento necessário para a realização do processo. Posteriormente, em duas secções distintas, serão explicadas, pormenorizadamente, duas etapas distintas do projeto.

No seguinte capítulo, Resultados, serão expostos vários resultados de testes realizados ao longo da resolução do projeto, bem como os testes realizados à solução final, de forma a analisar a sua robustez e precisão.

Por fim, no quinto e último capítulo, Conclusão e Trabalho Futuro, são feitas algumas considerações e conclusões do trabalho realizado, bem como os próximos passos a realizar de forma a dar continuidade ao trabalho iniciado nesta dissertação.



## Capítulo 2

# Estado da Arte

A instalação de um sensor num sistema robótico permite-lhe retirar informação do ambiente em que se encontra. Conseguindo captar características do cenário que o rodeia, o robô, é capaz de alterar o seu comportamento, sempre dependendo do que recebe dos sistemas de percepção. Consoante as suas tarefas ou o seu propósito, a informação que necessita para adaptar o seu comportamento varia, ou seja, cada sistema robótico requer informação de diferentes naturezas. Com este objetivo, existem sensores de diferentes modalidades, isto é, capazes de extrair essas características distintas. Assim sendo, dependendo da complexidade do robô ou das suas funções, este pode conter vários sensores e de várias modalidades, tal como referido no capítulo anterior.

Para que a informação retirada de cada sistema de deteção se possa cruzar com a proveniente de outro(s), da mesma modalidade ou não, é requerido que estes estejam calibrados. Segundo [34], calibração é um processo onde a estimativa da posição e orientação vai sendo melhorada através de mudanças no software ao invés da realização de alterações no seu design exterior. Este procedimento requer um modelo onde os seus parâmetros representam o robô real.

Tal como descrito anteriormente, uma calibração requer que os sensores partilhem informação, de forma ao otimizador poder avaliar os parâmetros da pose dos constituintes do sistema. Desta forma, pode ou não ser usado um padrão ou objeto específico que se torne mais fácil de identificar por parte dos sensores.

Existem várias áreas que requerem a operação e implementação de robôs em que os resultados da calibração dos mesmos conduzem a uma melhoria significativa da eficiência e custos associados.

Ao longo dos anos foram desenvolvidos variados métodos de calibração entre dois ou mais sensores, quer da mesma modalidade, quer de múltiplas modalidades. Esta evolução tem contribuído para a construção de robôs mais complexos, capazes de captar e tratar grandes quantidades de informações, bem como a precisão dos seus processos.

De modo a que este projeto contribua para o desenvolvimento da área da calibração é necessário ter um conhecimento geral dos vários estudos já realizados. Assim, ao longo deste capítulo serão apresentados alguns desses trabalhos, considerados relevantes para o domínio em que o tema da dissertação se insere.

## 2.1 Calibração entre dois sensores

Na robótica, é de grande relevo a utilização de diferentes sensores em cada sistema. Existem múltiplos métodos de calibração extrínseca entre dois sensores.

Em [24], é feita a calibração extrínseca entre um LIDAR 3D e uma câmara, usando um plano sem qualquer tipo de desenho que sirva de apoio à calibração, ou seja, explorando as características do plano que sejam detetáveis por ambos os sensores. Colocando o alvo em diferentes posições e orientações, numa zona comum aos dois, o objectivo é que este e as suas arestas sejam detetados. A partir daí, é criada uma função objetivo que otimiza os parâmetros pretendidos. São comparadas duas diferentes funções de custo: a primeira, usualmente utilizada neste tipo de casos devido à sua fácil utilização, proposto em [28], utiliza um ponto do plano detetado pelo LIDAR e o plano visto pela câmara; a segunda utiliza a distância entre o ponto de uma aresta detetada pelo LIDAR à linha correspondente vista pela câmara. Esta última função de custo, obtém resultados com boa precisão, superando o método tipicamente utilizado. Em [21], é apresentado o tema da condução autónoma, que requer uma boa percepção do que rodeia o veículo. É afirmando que os modelos existentes pecam à medida que o alvo se afasta do LIDAR 3D e da câmara devido ao ruído ou ao equipamento utilizado. De forma a eliminar o problema é utilizado um plano de calibração, onde a qualidade do padrão desenhado é irrelevante, para que seja detetado um canto por ambos os sensores, sendo a função de custo a distância entre esses dois pontos. Para diminuir o erro de uma abordagem ponto para ponto, é feita uma sequência de *frames*, onde é determinada uma região de interesse com apenas o plano de calibração e a pessoa que o segura, sendo o canto do plano, que é utilizado como referência, o ponto mais alto dessa mesma região. Este método obtém bons resultados e diminui o problema encontrado inicialmente.

Em [11], realiza-se a calibração extrínseca entre um laser 2D e uma câmara, por ser considerado que grande parte dos robôs possuem estes dois sensores. Esta é feita através da observação de triédros nos cenários encontrados (cantos ou esquinas, por exemplo). O método não necessita então de nenhum padrão específico, tornando o processo de calibração mais rápido e simples. Por outro lado, em [40] e [38], já é utilizado um padrão de calibração (xadrez em ambos os casos) para a calibração extrínseca entre uma câmara e um laser 2D. Já em [18], recorre-se a uma forma em 'v', utilizando dois planos com ângulo entre si, para serem facilmente detetados pelo laser e posteriormente calibrados. Segundo [11], o método nele proposto obtém resultados superiores aos restantes referenciados anteriormente.

Com a maior procura de câmaras de profundidade (RGB-D) para sistemas robóticos, em [2] é proposta uma solução *user-friendly* para se proceder às calibrações intrínseca e extrínseca das mesmas, por ser considerado que os métodos padrão não correspondem à exigência pretendida. Para realizar a calibração é apenas necessário coletar dados num ambiente minimamente estruturado. Relativamente a outros métodos utilizados, consegue obter resultados com boa precisão. Por outro lado, em [37] para a calibração extrínseca de duas câmaras RGB-D, é preciso apenas mover uma esfera (por exemplo uma bola de basquete) na frente dos sensores, sendo que o modelo criado deteta o objeto e elimina o ruído da imagem. Para o mesmo método, obtiveram melhores resultados utilizando esferas do que um xadrez de calibração.

Por ser de relevo para o tema calibração, é conveniente ter uma percepção global dos vários sensores que são utilizados na área da robótica. Particularmente, no campo dos *Automated Driving Systems* (ADS) o uso de sensores de diversas modalidades torna-se imprescindível, e, por consequência, a sua calibração também.

Atendendo aos objetivos já mencionados no capítulo anterior, o presente projeto focar-se-á na calibração de múltiplas câmaras RGB: sensores constituintes do anel também já referenciado. Como tal, é necessário um conhecimento dos diversos métodos de calibração entre sensores desta específica modalidade.

De forma a estimar a transformação entre duas câmaras RGB, foram já desenvolvidas várias metodologias distintas. Apesar de diferentes, estas podem possuir características comuns. Assim, podemos encontrar exemplos que recorrem a um padrão como forma de auxílio à calibração, [42], [3], e outros que utilizam características do cenário, [27], [29], [41]. As calibrações podem realizadas de forma online, [20], [13], [36], onde por vezes existe a possibilidade se os sensores serem constantemente calibrados, ou de forma offline, [23] (Parâmetros são estimados offline, mas são monitorizados online). Por fim, os métodos podem ser ainda totalmente automáticos, [25], [7], [10], ou semi automáticos, [39], onde há tarefas que requerem a interação do utilizador. Existem ainda trabalhos, em [26], por exemplo, que procuram resolver a questão de uma estimativa inicial dos parâmetros extrínsecos muito afastada da realidade. Um *first guess* muito desviado, pode levar a que o processo de otimização termine num mínimo local e não no mínimo global da função de custo.

De forma a estimar os parâmetros extrínsecos, existem diferentes funções de custo que podem ser utilizadas. Em [12] é usada a distância entre pontos no mundo 3D como erro retornado pela função. Isto deve ao facto de ser utilizado um plano de calibração, que traz restrições vantajosas ao problema para este efeito.

Em [33] é realizada também uma calibração stereo, porém tem a particularidade de as câmaras possuírem propriedades com valores bastantes distintos (por exemplo, abertura da lente e distância focal). Para contornar essa adversidade, utilizaram um padrão com um xadrez, de menores dimensões, embutido noutra, de modo a que cada câmara possa observar o seu respectivo padrão. O xadrez possui várias cores com o propósito de com diferentes canais de cores ser possível observar um ou outro padrão.

Em [20] é realizada uma calibração extrínseca de um sistema stereo de forma automática e online, recorrendo aos contornos do cenário em que se encontra ao invés de utilizar um padrão de calibração. Visto que, este método apenas calcula os parâmetros extrínsecos, é necessário realizar inicialmente a calibração intrínseca das câmaras, aplicando um processo tradicional para o efeito. Em relação à calibração extrínseca, para estimar o valor dos respectivos parâmetros é utilizada uma otimização não-linear. Embora esteja este processo dependente do cenário em que se encontra, contorna a possibilidade de as câmaras possuírem diferentes frequências, o que poderia levar a que, para o mesmo momento, captassem ligeiras diferenças no cenário e luminosidade. Sendo um método online, é conveniente que os dados sejam o mais sucintos possível. Assim, ao filtrar a informação recolhida para apenas os contornos de relevo, diminui-se o peso computacional e duração do processo. A precisão desta metodologia é algo que varia consoante o ambiente em que a calibração é aplicada. É até mesmo referenciado pelos autores que a covariância dos resultados decrescem de modo acentuado assim que o sistema

stereo passou a visualizar um cenário com contornos salientes, propositadamente criado para o efeito. Os autores referem ainda que o método por eles aplicado possui resultados comparáveis a processos *offline* com padrões de auxílio à calibração.

Com o foco da dissertação assente na calibração de câmaras, é importante retirar deste último trabalho apresentado o interesse em resolver o problema que pode surgir pelas diferentes frequências dos sensores. Ao captar duas imagens em momentos muito próximos, mas não coincidentes, existe a possibilidade de aumentar o erro caso haja uma ligeira diferença nas mesmas. É algo que num ambiente estático não terá influência, mas caso haja algum tipo de movimento já acaba por alterar os parâmetros obtidos. É então de relevo ter em atenção qualquer pormenor que possa ter influência direta nos resultados, de forma a maximizar a robustez e precisão do trabalho desenvolvido. Para além disto, por uma calibração entre sensores exigir um complexo processo de otimização, faz sentido que o processo seja não linear. Quer isto dizer que recorre a várias iterações de forma a tentar atingir o mínimo da função de custo. Outro ponto positivo reside na preocupação pela redução do peso computacional, de forma a tornar o processo mais simples e rápido.

Por outro lado, existem alguns pontos negativos neste trabalho ou ao aplicá-lo no projeto desta dissertação. Primeiro, para tornar o processo de calibração geral mais simples e fácil de aplicar, seria de importância que o mesmo realiza-se a calibração intrínseca e extrínseca. Por fim, um processo de calibração de sensores num veículo inteligente, devido à importância que tem a sua elevada precisão, não deve ter os resultados dependentes do cenário em que este se encontra, mas sim ser garantido sempre o mesmo rigor.

Tal como referido anteriormente, um veículo inteligente é algo bastante complexo, que requer uma excelente percepção do ambiente em que se encontra. Por esse motivo, é um sistema que necessita de mais do que dois sensores. Então, após vários exemplos de como relacionar dois sensores, de diferentes modalidades ou não, é necessário perceber de que formas se podem relacionar mais do que dois sensores num sistema robótico. Este procedimento pode aumentar e muito a complexidade de um robô e das suas ações.

## 2.2 Calibração de sistemas com mais de dois sensores

À medida que os sistemas robóticos vão aumentando a complexidade, são necessários mais sensores para que as tarefas sejam realizadas. Porém, é exigido que estas sejam realizadas com a maior precisão possível, ou seja, através de uma boa calibração desses mesmos sensores. Aqui reside um dos maiores problemas da calibração. Os métodos anteriores apenas visam calibrar dois sensores simultaneamente, deste modo, é preciso adaptar a calibração ao novo problema em estudo, existindo diferentes métodos para resolver a questão.

É proposto, em [19], calibração de juntas de um sistema com pelo menos quatro sensores, um *Kinect* (sensor de movimento) e três câmaras externas. É realizada uma calibração simultânea de todos os sensores, contrariando o trabalho desenvolvido em [4], onde é aplicada uma calibração a cada par de sensores. Por fim, otimizando a função de custo final, conseguiu-se obter resultados satisfatórios num projeto em tempo real, melhorando a precisão em relação à calibração padrão.

Em [43] é apresentado um método para estimar os seis parâmetros (três da translação e três da rotação) relativos à calibração extrínseca de vários sensores heterogêneos. O processo é dividido em duas partes. Inicialmente é feita uma calibração planar, translações no plano  $xy$  e rotação sobre o eixo  $zz$ , entre dois sensores, calculada através de movimento do robô. Esta primeira parte serve como referência para a segunda, os restantes parâmetros e os restantes sensores do sistema. Para auxiliar a calibração recorre-se à observação do plano do chão por um curto período de tempo. É um processo que, aplicado tanto em ambientes *indoor* como *outdoor*, obteve sucesso.

Em [32], é estudada a calibração de diferentes sensores num robô PR2. É utilizado apenas um processo de otimização, pois estão modeladas as características do erro associado a cada tipo diferente de sensor. Um novo tipo pode ser acrescentado à otimização, desde que sejam fornecidas as informações do seu tipo de erro. Este método permitiu obter bons resultados e aumentar a precisão de movimentos do respectivo robô.

De forma a resolver a questão da calibração de um sistema com mais de dois sensores, em [16], é desenvolvido uma nova abordagem de forma a estimar os parâmetros extrínsecos de múltiplos sensores instalados numa plataforma em movimento, não utilizando qualquer padrão de calibração. De forma a contornar o ruído nas informações recolhidas pelos sistemas de percepção, recorre-se ao modelo de Gauss-Helmet que aplica correções não só aos parâmetros, mas também ao que é observado em cada iteração. No caso da pose relativa entre sensores, é adoptada uma estratégia composta por várias calibrações par a par, escolhendo um deles como referência e calibrando os restantes, um a um, em relação a este. Os parâmetros extrínsecos são retirados de um histograma com as rotações e translações das muitas observações de posição e orientação realizadas (mais de 1600). Este trabalho requer um *first guess* com elevado grau de precisão, mas permite obter resultados bastante precisos, quando comparados com o método dos quadrados mínimos, em casos de elevado ruído na informação proveniente dos vários sensores.

Visto que, são feitas calibrações independentes entre cada sensor com o de referência, este não é considerado um processo de calibração simultânea. Deste modo, para que isto seja possível, é necessário que todos possuam informação em comum com o sensor de referência. No caso de um veículo com navegação autónoma, o objetivo é cobrir a maior área possível em torno do mesmo, pelo que, haverão sensores que não possuam informação em comum entre eles. Assim, este método não poderá ser aplicado no tema desta dissertação. Tal como referido, a utilização de um modelo Gauss-Helmet, permite melhores resultados que o método dos quadrados mínimos. Porém, apenas é confirmado em casos de elevado ruído na informação. Devido à importância que uma recolha de informação com bastante qualidade possui num veículo inteligente, este deverá estar equipado com sensores pouco sensíveis a ruído e a calibração feita com o maior rigor possível. Deste modo, deve ser tida em consideração a abordagem que permita a melhor precisão possível, pelo que não é totalmente vantajosa a utilização deste modelo.

O trabalho desenvolvido em [30] estuda a calibração entre dois lasers LIDAR, um 2D e outro 3D, e duas câmaras RGB. É apresentado um método automático que utiliza uma esfera como alvo de calibração. No processo de calibração são realizadas três etapas: inicialmente, com a esfera estática, os vários sensores detetam o centro da mesma; seguidamente, é movimentada

a bola de forma a que o seu centro seja localizado em diferentes posições por todos os sensores; finalmente, é escolhido um sensor como referência e os restantes são calibrados em relação a ele. Este método envolve então várias calibrações par a par, mas de forma não sequencial, isto é, é feita uma de cada vez e qualquer uma é independente das outras. Este processo desenvolvido permite uma liberdade na modalidade e quantidade de sensores, desde que estes sejam capazes de detetar a esfera e, conseqüentemente, o seu centro. Nos testes foram obtidos erros com valores pequenos no que aos lasers diz respeito, porém, em relação às câmaras, limitações na correta deteção do centro da esfera levaram a um erro maior no final. Também em [31], é reafirmado todo este processo, bem como a realização de uma calibração stereo antes da calibração geral, de forma a diminuir o erro associado às câmaras. Assim, é de relevo ter em atenção que uma superfície esférica pode não ser o melhor padrão de calibração num sistema apenas composto por câmaras e que apenas se deve recorrer ao mesmo para facilitar a calibração com outra modalidade de sensores.

É estudada, em [8], a junção de várias câmaras de profundidade. De forma a obter os parâmetros extrínsecos, é necessário apenas observar planos de diferentes pontos de vista. Este método pode ser utilizado para um sistema com duas ou mais câmaras. O facto de não ser necessária informação partilhada, mas sim, tal como referido anteriormente, diferentes pontos de vista de um plano comum, é um dos grandes factores para o número de sensores poder ser variado. Usando a vantagem dos cenários por norma conterem grandes planos, tais como o chão, o teto ou até mesmo as paredes, este método não necessita de nenhum padrão de auxílio à calibração, considerando que o torna mais simples e fácil de usar. Sendo utilizado para duas ou para mais câmaras, é par a par que são encontrados os planos coincidentes em ambas as imagens para proceder à calibração. Para serem válidos para a calibração são consideradas três restrições: o ângulo entre os vetores normais aos planos serem inferiores a um determinado limite; a distância entre os planos e o centro das câmaras serem inferiores a outro determinado limite; o número de pixeis de cada região ser superior a 20% do total da imagem. Para que seja possível a calibração, é necessário que sejam observados três planos válidos perpendiculares, numa ou em várias imagens, permitindo esta última que seja visto o mesmo plano com diferentes orientações. Uma situação de duas câmaras com direcções opostas pode dificultar o processo caso estejam muito afastadas (um veículo, por exemplo, com uma virada para a frente do carro e outra na traseira), obrigando a uma rotação da câmara para a obtenção das correspondências necessárias.

Numa vertente mais ligada à utilização de câmaras em veículos, é proposto em [14] uma forma automática de calcular as calibrações intrínseca e extrínseca de um equipamento com várias câmaras, assumindo que não existem campos de visão sobrepostos. É realizada uma primeira calibração intrínseca para cada câmara, utilizando um plano com xadrez. Posteriormente, a partir de uma curta viagem e através de ambientes naturais são calculados os parâmetros extrínsecos. Por fim, através de um passo de *bundle adjustment*, é feita uma optimização dos parâmetros intrínsecos calculados inicialmente.

Em [1], é estudado um modelo de *bundle adjustment* em maior escala, utilizando cerca de dez mil imagens diferentes. É estudada a utilização de gradientes conjugados e a sua performance. Através de um *dataset* de imagens da rua, é criado o problema de *bundle adjustment* e é feita a comparação com outros métodos obtendo bons resultados.

Em [17], é realizada a calibração intrínseca e extrínseca de um sistema com número de câmaras variado. Para tal, tem como principal requerimento a existência de informação sobreposta entre os vários sensores de imagem. Assim, é utilizado um xadrez como padrão de calibração, tendo este de ser detetado por todas as câmaras. Este padrão permite tanto estimar os parâmetros intrínsecos, adaptando um método existente, e os parâmetros extrínsecos, através do método desenvolvido. Este último, utiliza o referencial do plano de xadrez como referência para a calibração, sendo os restantes calibrados em relação a ele. Esta metodologia apresenta vantagens pela sua flexibilidade e boa precisão quando comparado com métodos convencionais.

Tendo este trabalho em consideração, pela obrigação de existência de informação comum entre todos os sensores, não poderá ser aplicado, pelo menos de forma direta, à dissertação. Devido ao facto de, o veículo autónomo não conseguir garantir esse requerimento, por possuir sensores a toda a sua volta. Porém, através da utilização de um padrão de calibração, existe a possibilidade de estimar tanto os parâmetros intrínsecos como os extrínsecos, tornando o trabalho mais geral e completo.

É apresentado em [5] um método geral, totalmente automático, para a calibração de um equipamento com múltiplas câmaras instaladas. Esta metodologia não utiliza nenhum padrão como forma de auxílio, ou seja, não tem a necessidade de obter qualquer informação prévia relativamente ao cenário em que está inserido.

As câmaras são fixadas ao robô, que inicia movimentos pré-programados até obter informação a 360° por parte de todos os sensores. Para cada câmara é estimado o seu movimento e criado um mapa 3D com as características visuais detetadas, utilizando para isto o algoritmo MonoSLAM. Os diferentes mapas são então alinhados usando alinhamentos de semelhanças 3D e RANSAC [9]. Esta junção serve de ponto de partida para uma otimização de *bundle adjustment*, de forma a estimar a posição e orientação das várias câmaras.

Este método obteve bons resultados, mesmo variando o número de sensores bem como a sua disposição. Porém, apesar de ser algo que poderia levar a defeitos na constituição de cada mapa 3D, ter incorporada a calibração intrínseca no processo tornaria-o mais completo. Segundo os autores, esta metodologia funciona mesmo em casos onde não existe informação partilhada pelos sensores. Todavia, ao obter informação a toda a volta do robô, com os movimentos programados, existirá sempre informação comum entre todas as câmaras.

Relativamente ao estudo de sistemas com sensores variados e de diferentes modalidades, é desenvolvido, em [6], um método de calibração, aplicado num sistema composto por duas câmaras RGB e dois lasers LIDAR num veículo inteligente. Este trabalho defende que num veículo autónomo, devido à elevada precisão requerida, deve ser feita uma calibração simultânea de todos os sensores e não uma sucessão de calibrações par a par – sejam elas em relação a um sensor de referência ou em modo sequencial.

Numa fase inicial do processo, é avaliada a árvore de transformações associada ao robô em questão. Isto permite que, esta metodologia, seja aplicável a qualquer sistema robótico. A abordagem de calibração desenvolvida representa um processo semi-automático, ou seja, até ao início da otimização, é requerida a intervenção do utilizador em algumas tarefas. É requisitado que, num ambiente virtual, onde está representado o corpo real do veículo, sejam posicionados os sensores, em relação a este corpo, do modo mais próximo possível da realidade. Atribuindo assim, uma estimativa inicial já aproximada do que será o resultado final da calibração – o que reduz, acentuadamente, a possibilidade de o otimizador terminar num mínimo local da

função de custo, problema comum em processos de otimização não lineares.

De forma a garantir a calibração simultânea de todos os sensores, os parâmetros a serem otimizados englobam os parâmetros extrínsecos de cada um dos sensores. Para além destes, a função de custo recebe também os parâmetros intrínsecos das câmaras, inicializados por um método de calibração convencional, e a transformação geométrica do objeto de calibração. Este método estima, os valores mencionados, com base na diminuição do erro de reprojeção dos pontos reconhecidos, do objeto de calibração, por cada um dos sensores. Deste modo, é necessária a deteção do padrão de calibração por parte de todos os sistemas de percepção de forma simultânea, sendo, por isso, descartados todos os momentos em que o alvo não seja identificado.

Este trabalho desenvolvido apresenta resultados comparáveis com os existentes métodos, porém, não é feito para uma situação em particular, podendo o número e modalidade dos sensores ser variado, e é feita apenas uma calibração de forma a englobar todos os seus sistemas de percepção.

A metodologia que será desenvolvida nesta dissertação, depois de analisados todos os métodos apresentados anteriormente, terá como principal influência o trabalho desenvolvido em [6]. Surgem como fatores decisivos o facto de neste ser aplicada uma calibração simultânea de um sistema com diversos sensores (número e modalidade variáveis) e o facto de este ter sido aplicado ao mesmo objeto em estudo, ATLASCAR2, utilizando a mesma *framework* – Robot Operating System (ROS), introduzido e apresentado no seguinte capítulo deste documento. Não obstante, este trabalho não poderá ser aplicado, diretamente, ao problema apresentado no capítulo anterior: calibrar câmaras que, em simultâneo, não observem o mesmo cenário, pelas razões já mencionadas.

Como influências secundárias serão a maioria dos restantes trabalhos apresentados neste capítulo, pelos pontos positivos e negativos que deles foram possíveis retirar, com a finalidade de aumentar a robustez e precisão desta dissertação.

No próximo capítulo será introduzida a solução conceptual proposta para a resolução do principal problema da dissertação, calibração de um anel de câmaras, bem como os problemas que dele surgem. Depois de desenvolvido o estado atual da arte da área da calibração, será, portanto, um dos focos obrigatórios a calibração de sensores que não possuam informação partilhada, por não existir, atualmente, resposta para esse problema.

## Capítulo 3

# Solução proposta

Tal como já referido anteriormente, um veículo autónomo, ou seja, inteligente, é um sistema robótico bastante complexo. Um robô capaz de tomar decisões por si necessita de ter ao seu dispôr o máximo de informação sobre o cenário que o rodeia. Por esta linha de pensamento, o aumentar da complexidade das suas ações requer um maior número de sensores.

O objeto de estudo para este projeto é, tal como descrito no Capítulo 1, o ATLASCAR2 – um veículo autónomo. Este é um sistema crítico, ou seja, um sistema onde não são permitidos erros, visto que, qualquer que seja a natureza do mesmo, pode pôr em risco vidas humanas. Nesta lógica, a correta recolha da informação é um ponto fulcral para o seu bom funcionamento.

De modo a recolher o máximo de informação possível, é de relevo aumentar o número de sensores do veículo. A condução autónoma é uma ação que requer detetar várias características visuais do ambiente em que se encontra: linhas da estrada, sinais de trânsito, possíveis obstáculos em toda a sua volta ou até mesmo pontos de auxílio ao estacionamento. É um processo que visa substituir a condução humana, de forma a tornar a condução algo muito mais seguro e prático. Tal como acontece na condução humana, existem sempre situações que nos obrigam a ter a noção completa do veículo e do que nos rodeia. Então, torna-se imprescindível, o veículo possuir sensores de imagem em todo o seu redor, para que não seja necessária qualquer intervenção nem auxílio por parte do utilizador.

À medida que são adicionados sensores a um sistema, e de forma a que seja possível retirar informação válida de todos estes, é necessário que eles estejam, de certa forma, ligados entre si. Esta ligação resume-se à posição relativa entre cada um dos sensores. Quer isto dizer que, de forma a ser possível interpretar os dados provenientes destes, é necessário conhecer a posição e orientação de cada um. Isto permite que haja um cruzamento de dados, levando a um melhor conhecimento do ambiente. Caso isto não aconteça, cada um destes objetos de percepção deve ser interpretado como um sistema independente, porém, pelo menos ao utilizar vários da mesma modalidade, o objetivo será conjugar toda a informação recolhida de forma a melhorar a capacidade de decisão do sistema robótico. Ao processo de estimação da pose de cada sensor, relativamente aos restantes chama-se calibração extrínseca.

Nesta dissertação, o trabalho desenvolvido tem como objetivo a calibração extrínseca de um sistema com múltiplas câmaras de forma a que este possa ser instalado no ATLASCAR2.

Para que seja possível estimar a posição e orientação de um sensor em relação a outro é imprescindível que estes partilhem informação. Isto porque, é na zona da sobreposição que

se obtêm dados sobre a pose do outro sensor. No Capítulo 2 encontram-se enumerados vários exemplos relacionados com este tema. No entanto, o conjunto de sensores necessários para que um veículo autónomo capte a totalidade do cenário em que se encontra, obriga a que cada um não possua informação comum com todos os restantes em simultâneo. Por outras palavras, nunca haverá o mesmo objeto físico, no ambiente envolvente ao veículo, a ser detetado por todos os sensores no mesmo momento temporal. Aqui, reside um dos desafios do mundo da calibração.

Neste trabalho é apresentada uma proposta de solução para o problema da calibração descrito anteriormente.

Esta abordagem possui duas etapas distintas: recolha e estruturação de dados e calibração extrínseca. Na primeira parte, é realizada a captura de imagens das diferentes câmaras. Ao longo deste processo, é necessário que as várias câmaras vão detetando, na sua totalidade, um plano com um padrão de xadrez, por exemplo, Figura 3.1. Este padrão é de fácil deteção na imagem, isto porque, pelas suas características, não se confunde com o cenário em que os sensores se encontram. Para além disto, as imagens são já capturadas num ambiente onde a iluminação é controlada, de forma a aumentar a precisão de todo o processo. Nesta abordagem, a utilização do xadrez simplifica o facto de também se pretender realizar a calibração intrínseca dos sensores, visto que é algo que pode ser utilizado para ambos os processos de calibração.

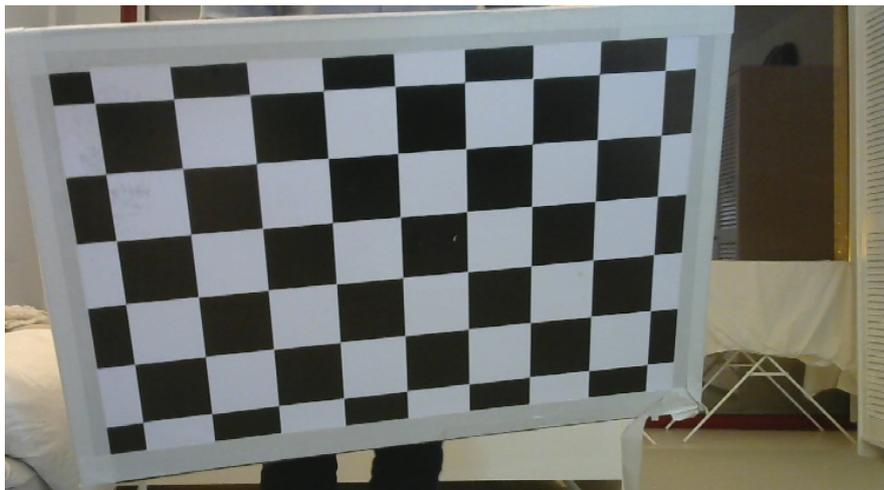


Figura 3.1: Exemplo de imagem que captura o plano de xadrez na sua totalidade

A primeira etapa será a única que exigirá a interação do utilizador. É-lhe requerido que capture várias imagens para a finalidade de calibrar extrinsecamente todos os sensores, e outras para a calibração intrínseca. Ao fazer o pedido de captura de imagem, o programa recolhe os dados de todas as câmaras em simultâneo. É muito importante que todas as imagens sejam recolhidas no mesmo instante, para que a informação em comum seja detetada na mesma posição e orientação real, sem erros de movimento associados. Caso duas câmaras tenham frequências diferentes do envio de mensagens, o xadrez de calibração pode estar com ligeira diferença nas imagens retiradas. Então, manter o plano na mesma posição por cerca de um segundo elimina o erro associado a um possível atraso na captura das imagens. Para o melhor entendimento do resto do procedimento descrito neste documento, é necessário a clarificação do conceito de coleção. Designa-se por 'coleção' cada grupo de imagens captadas por todos os

sensores relativas ao mesmo instante temporal. Como tal, sempre que o utilizador capturar um dado momento, será guardada uma nova coleção - conjunto de imagens obtidas por todas as câmaras.

Após ter coletado suficientes imagens (coleções), o utilizador deverá proceder à calibração intrínseca das câmaras. Este número de coleções necessárias é definido pelo próprio utilizador no início de todo o processo. No entanto, para garantir o maior sucesso nas tarefas futuras, aconselha-se a que a calibração intrínseca não seja realizada com base num número reduzido de coleções: habitualmente, as calibrações desta natureza requerem grupos de, no mínimo, uma dezena de imagens, como se pode verificar, a título de exemplo, na realizada com funções de OpenCV<sup>1</sup>. O processo de calibrar intrinsecamente tem como objetivo a obtenção dos parâmetros relativos à natureza e à construção de cada câmara e de cada lente associada - parâmetros estes necessários para a projeção de pontos reais captados em pontos na imagem (pixels). A metodologia criada e proposta para a calibração intrínseca e a sua utilidade são clarificadas neste mesmo capítulo.

Para a segunda etapa da abordagem deste trabalho - a calibração extrínseca - são requeridas várias informações. Todas estas deverão resultar da primeira etapa: recolha e estruturação de dados. As várias coleções de imagens capturadas pelo utilizador (com a finalidade de calibrar extrinsecamente) são, tal como já mencionado, parte integrante dos dados necessários. Não só estas imagens mas também os parâmetros intrínsecos, provenientes do processo de calibração intrínseca anteriormente referenciado, deverão constar na informação pretendida. De modo a facilitar o acesso e a consulta aos dados guardados, propõe-se um armazenamento organizado, intuitivo e devidamente estruturado. Atendendo ao referido, o sistema criará um ficheiro do tipo *JavaScript Object Notation* (JSON). Assim sendo, às imagens capturadas acresce este ficheiro, que contém múltiplos dados relevantes para etapas futuras, no conjunto de informação que será salvaguardada. A este conjunto - imagens e ficheiro JSON - designa-se *dataset*.

No esquema na Figura 3.2 é possível encontrar um fluxo representativo dos processos inerentes à recolha e estruturação de dados.

---

<sup>1</sup>[https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_calib3d/py\\_calibration/py\\_calibration.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html)

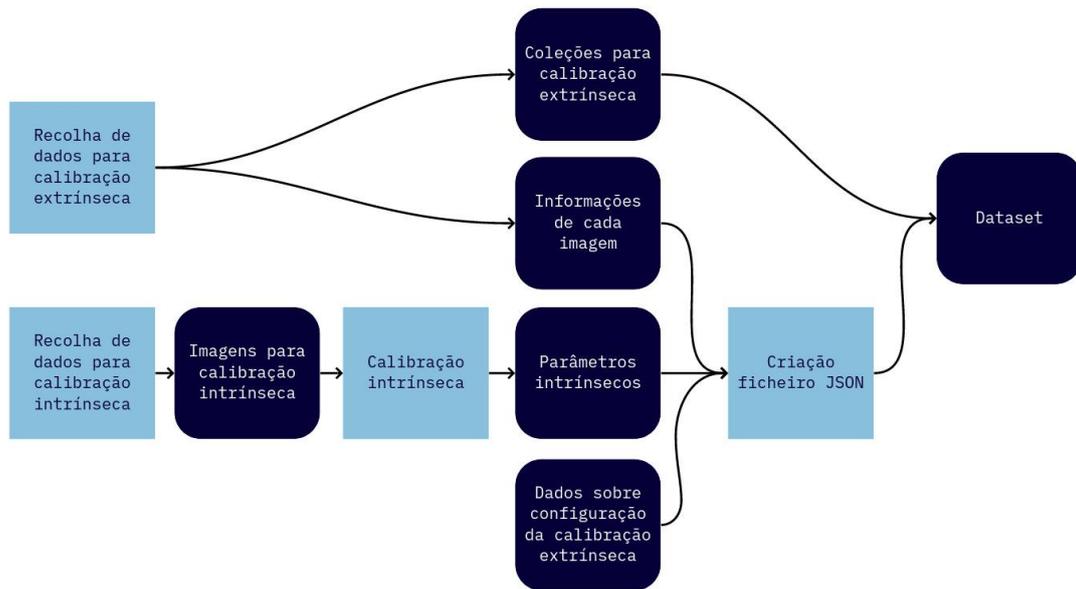


Figura 3.2: Fluxo dos vários processos na recolha e estruturação de dados.

Note-se que o *dataset* é a ponte de ligação entre as duas partes constituintes da presente abordagem, Figura 3.3. De facto, um *dataset* retirado numa dada ocasião não é mais do que informação estática que poderá ser utilizada para tratamento posterior a qualquer altura que o utilizador considere oportuna. Para o bom funcionamento da segunda etapa, basta apenas que, para além da existência de coleções guardadas, o ficheiro referido anteriormente contenha informações relativas aos sensores, coleções existentes e dados da calibração (xadrez, por exemplo). Quer isto dizer que, o programa referente à segunda metade do trabalho aceita diferentes *datasets*, procedendo à calibração de um sistema com sensores e coleções variados.



Figura 3.3: Fluxo representativo das duas etapas do trabalho, bem como a ligação entre as mesmas.

A segunda etapa do presente projeto consiste na calibração extrínseca de todas as câmaras em estudo, de modo simultâneo. Esta ocorrerá de modo automático, sem que seja requerida intervenção por parte do utilizador. O sistema, com o objetivo de calibrar todos os sensores presentes, isto é, de apurar a posição e orientação de cada uma das câmaras, realiza um processo de otimização. Uma vez que é possível caracterizar a pose de qualquer objeto através de três parâmetros de posição ( $x$ ,  $y$  e  $z$ ) e três parâmetros de orientação ( $r_1$ ,  $r_2$ ,  $r_3$ ), o objetivo desta segunda metade do trabalho (coincidente com o objetivo geral) é estimar estes mesmos seis parâmetros para cada um dos sensores. Como tal, e conhecendo já o funcionamento geral de um processo de otimização, propõe-se que os parâmetros a otimizar (valores de entrada da função objetivo) sejam os mencionados anteriormente. Para além destes, irão ser incluídos também, por necessidade de cálculo na função objetivo do otimizador, os valores de posição e orientação do padrão de calibração para cada uma das coleções registadas na primeira etapa. Acrescem ainda, aos argumentos de entrada da função objetivo do processo de otimização, os parâmetros intrínsecos de cada uma das câmaras, sendo também estes, portanto, otimizados.

Os seis valores que caracterizam a posição e orientação de cada sensor descrevem a transformação geométrica entre o referencial da câmara em questão e um outro referencial. Assim sendo, propõe-se que o referencial de um dos sensores disponíveis sirva de referencial global, ou seja, as poses a ser estimadas de todos os sensores serão sempre referentes a este referencial global. Note-se que a posição e orientação do sensor que inicialmente se encontra no referencial global também serão otimizadas. Por esta razão, o sistema de coordenadas de referência (global) será definido apenas no início do processo de otimização, sem nunca sofrer qualquer alteração durante o decorrer do mesmo.

Relembrando o que foi clarificado no Capítulo 1, uma otimização trata-se de um processo iterativo onde, partindo de um conjunto de valores iniciais, continuamente são testados novos valores dos argumentos de entrada da função objetivo (parâmetros a otimizar). Esta função possuirá sempre valores de saída que, dependendo do tipo de otimizador, serão pretendidos diminuir ou aumentar ou alguma combinação de ambos. É através da avaliação destes resultados e do objetivo traçado, que o otimizador irá procurar pelo melhor conjunto de valores que compõem os parâmetros a otimizar, de modo iterativo. Como tal, as variáveis de entrada da função objetivo serão sempre otimizadas de modo simultâneo para um mesmo processo de otimização. Na Figura 3.4 encontra-se o fluxo do processo que ocorre numa otimização, servindo como apoio ao que foi aqui descrito .

A proposta apresentada como segunda parte do trabalho - calibração extrínseca - é composta por apenas um único processo de otimização, uma vez que todos os valores de posição e orientação de todas as câmaras, em relação ao referencial global, são incluídos nos parâmetros a otimizar. Por esta mesma razão, a avaliação dos resultados e iteração seguinte por parte do otimizador serão sempre realizadas abrangendo as poses de todos os sensores em simultâneo. É de direta percepção que o resultado final - os parâmetros otimizados - irá contemplar as posições e orientações de todas as câmaras. Conclui-se, assim, de que se trata de uma calibração extrínseca de múltiplos sensores de modo simultâneo.

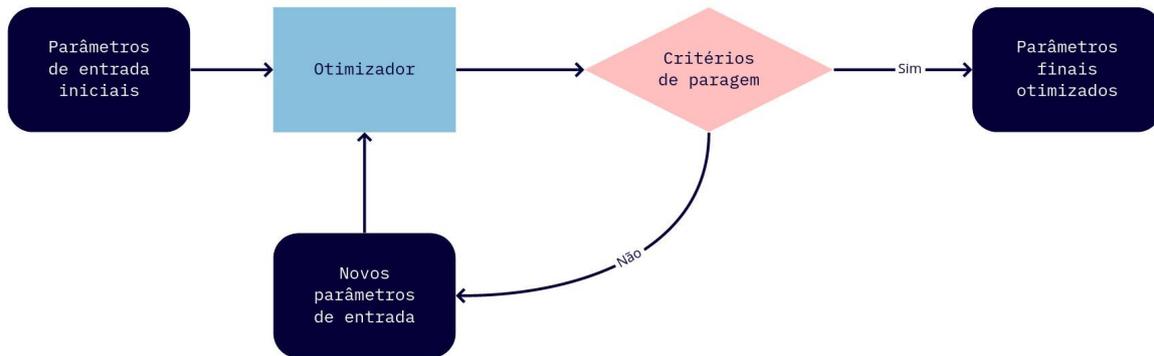


Figura 3.4: Fluxo de processo de Otimização.

A dificuldade associada à calibração de múltiplas câmaras instaladas em todo o redor de um veículo autónomo, que foi já múltiplas vezes referenciado, prende-se ao facto de em nenhuma coleção (conjunto de imagens referentes ao mesmo instante temporal) o xadrez ser detetado por todas as câmaras. Para futura referência, designar-se-à como coleção incompleta cada coleção de imagens onde não seja reconhecido o padrão de calibração em todas elas. Do modo inverso, uma coleção completa será aquela onde o xadrez é detetado em todas as imagens. Segundo o apresentado, no presente trabalho, o *dataset* resultante da fase inicial do processo apenas contém coleções incompletas. Para além disto, doravante serão também consideradas como inválidas todas as coleções que apenas possuem a deteção do padrão de calibração, numa única imagem, isto é, coleções onde apenas um sensor detetou o xadrez. A Figura 3.5 serve como suporte para uma melhor percepção da nomenclatura utilizada para classificar uma coleção, tendo por base o número de imagens onde o xadrez é detetado.

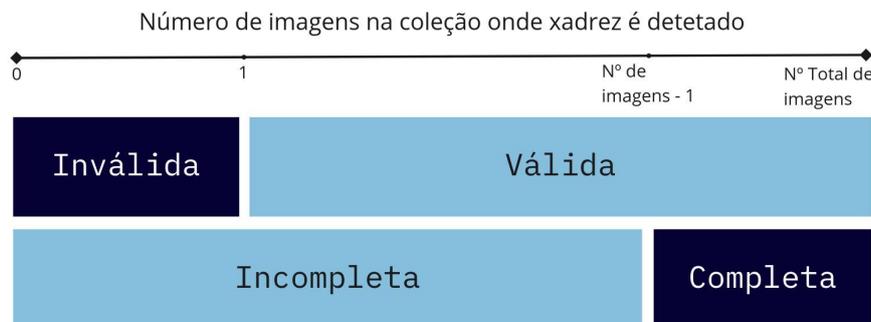


Figura 3.5: Classificação de uma coleção, consoante o número de imagens em que o xadrez é detetado

A metodologia para solucionar este tipo de problema reside na adaptação da função objetivo do processo de otimização. De facto, é esta função que, através dos parâmetros de posição e orientação de todos os sensores e de todas as distintas poses do xadrez, calculará valores que possam ser avaliados de modo a iterar os argumentos de entrada, aproximando-os da solução final. Propõe-se que, em primeiro lugar, sejam descartadas todas as coleções inválidas, ou seja, coleções que possuam apenas o xadrez detetado numa única imagem no máximo, restando assim apenas as coleções válidas e incompletas. A função objetivo, atra-

vés da informação contida no ficheiro JSON presente no *dataset*, conhecerá quais as imagens (sensores) onde foi detetado, na sua totalidade, o padrão de calibração. Estas serão as únicas imagens trabalhadas pela função de modo a produzir valores de saída que serão avaliados pelo otimizador. Cada imagem com deteção será estudada individualmente. Tendo os valores de posição e orientação da câmara e do xadrez da respectiva coleção, ambos em relação ao referencial global, nos seus argumentos de entrada, a função irá construir as matrizes das transformações geométricas e projetar os pontos referentes aos cantos do xadrez na imagem em estudo. Nesta mesma imagem, será também efetuada uma deteção do padrão de calibração. Desta deteção, resultarão novos pontos (em pixels) do xadrez detetado (considerados reais), que serão comparados aos pontos (também em pixels) projetados através das transformações geométricas supra referidas. É nesta comparação que serão medidas as distâncias de cada canto projetado do xadrez ao canto correspondente do mesmo xadrez no conjunto de pontos detetados. A função objetivo retorna todas estas distâncias, as quais serão, daqui em diante designadas como residuais, uma vez que o otimizador irá ter como objetivo diminuir o seu valor. Isto é, o otimizador procurará a melhor combinação dos valores de todas as posições (câmaras e padrões de calibração) para que a distância entre os pontos detetados e os pontos projetados seja a menor possível. A Figura 3.6 respresenta o processo realizado pela função objetivo (função custo), onde se pode verificar as transformações geométricas utilizadas, os pontos projetados e detetados para obter os valores residuais de uma das imagens em estudo.

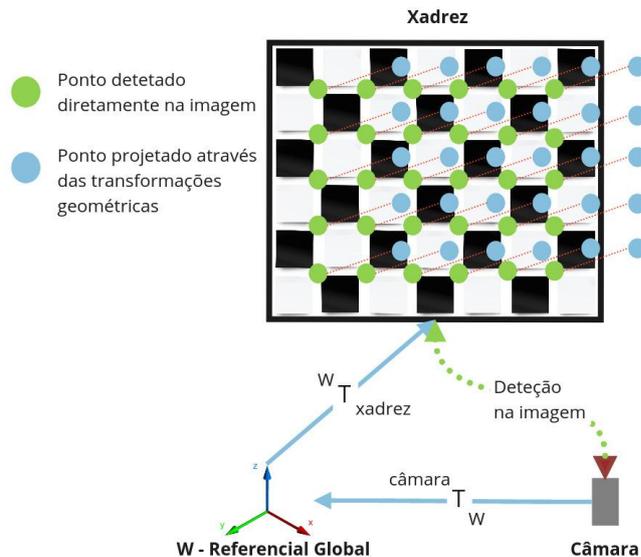


Figura 3.6: Esquema do estudo realizado para uma câmara numa determinada coleção. Os pontos verdes representam os cantos do padrão detetado diretamente na imagem (sendo estes considerados reais) e os azuis os pontos projetados através das transformações geométricas. Através dos residuais (distâncias entre pontos reais e os projetados - linhas vermelhas tracejadas), o otimizador altera os seus parâmetros (que levam às matrizes de transformações) de forma a reduzir a distância entre os pontos.

Com o que foi anteriormente explicitado, conclui-se, de modo direto, que cada residual está associado a um só canto de um xadrez de uma específica imagem. Cada imagem deteta apenas um padrão de calibração e é extraída de um só sensor. Como tal, cada residual

estará associado a apenas um padrão de calibração e a uma câmara. Tendo esta observação em consideração, e partindo da suposição que a imagem a ser avaliada seria a única com o xadrez detetado na coleção onde está inserida (coleção inválida), seria inconsequente para a melhoria dos resultados pretendidos (poses das câmaras), visto que o otimizador poderia, e provavelmente iria, alterar apenas os valores de posição e orientação do padrão de calibração dessa mesma coleção. Mais nenhuma outra imagem, e, portanto, mais nenhuns residuais, seriam afetados com esta alteração. Como tal, o otimizador teria total liberdade para tentar minimizar sem sequer necessitar de rever as posições dos sensores. Pode-se até afirmar que esta situação hipotética seria benéfica para uma avaliação global dos residuais, mas em nada contribuinte para a avaliação global do resultado pretendido.

Por esta razão, são descartadas, inicialmente, todas as coleções consideradas inválidas. Assim, sempre que o otimizador tentar alterar parâmetros de modo a diminuir valores residuais de uma só imagem, acabará por influenciar o valor de residuais de, pelo menos, outra imagem. De facto, se o otimizador apenas mudar a pose de um xadrez, alterará os residuais de todas as imagens dessa coleção onde existe deteção do padrão de calibração (no mínimo, duas imagens, Figura 3.7). Por outro lado, o facto do otimizador modificar os argumentos de entrada referentes à posição e/ou orientação de um sensor leva a uma alteração de todos os residuais das imagens, válidas, captadas por essa mesma câmara.

A filosofia de calibração simultânea, já mencionada, prende-se com todos os aspetos abordados: por um lado, apenas ocorre um único processo de otimização onde todos os resultados pretendidos se encontram já inseridos no conjunto de parâmetros a otimizar; por outro lado, nenhum destes parâmetros influenciará apenas um único valor residual, levando a que o otimizador tenha, constantemente, de procurar por soluções emparelhadas, encadeando a determinação das posições e orientações de todas as câmaras.

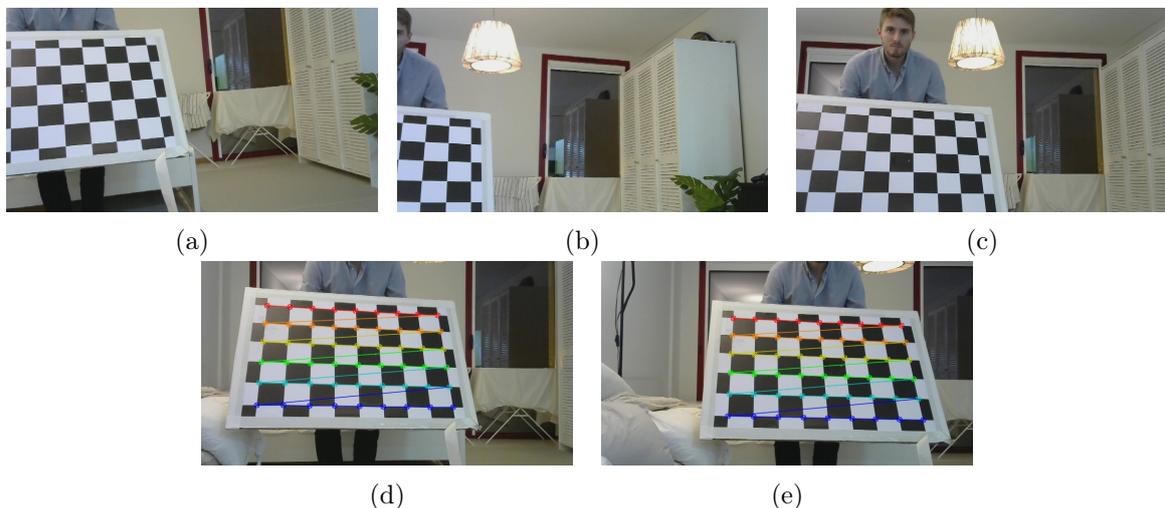


Figura 3.7: Imagens capturadas de uma única coleção num sistema com 5 câmaras: a) Câmara 1 (*left\_camera*), b) Câmara 2 (*center\_camera*), c) Câmara 3 (*center2\_camera*), d) Câmara 4 (*center3\_camera*), e) Câmara 5 (*right\_camera*). Como se pode verificar, esta coleção é considerada válida (e incompleta), sendo assim aceite e estudada na otimização. Neste caso, apenas duas das imagens contém deteção do xadrez: (d) e (e).

## 3.1 Recolha e estruturação de dados

Tal como descrito no início deste capítulo, o projeto está dividido em duas partes distintas. A segunda, que corresponde ao objetivo do trabalho e a primeira que é serve de apoio ao fornecer os dados necessários para a obtenção do resultado final.

Neste sub capítulo serão, então, analisados, em maior detalhe, os processos realizados na primeira metade do projeto, dando a conhecer as ferramentas utilizadas para cada um destes mesmos processos. De forma a obter a melhor percepção possível do que irá ser descrito, recomenda-se que se tenha em mente o esquema apresentado na Figura 3.2, onde se encontram os vários processos referentes à primeira etapa - recolha e estruturação de dados.

Com o primeiro objetivo da parte prática deste trabalho a ser a recolha da informação, é utilizada uma ferramenta do mundo da robótica - ROS, *Robot Operating System*. É considerada uma opção bastante válida e totalmente capaz de realizar as tarefas pretendidas, visto ser capaz de escutar informação de diferentes sensores do sistema.

De seguida, irá ser apresentada esta ferramenta, onde serão explicados conceitos chave para o bom entendimento da sua utilização, bem como a sua aplicação neste projeto.

### 3.1.1 ROS - *Robot Operating System*

Tal como descrito anteriormente, o ROS, é uma ferramenta bastante flexível que permite a criação do *software* de um sistema robótico. É um ambiente *open source*, isto é, de utilização gratuita, que possuem bibliotecas e funções que tornam a criação de um robô complexo um processo bem mais simples. É ainda possível a adaptação de excertos de código por parte de outros utilizadores, permitindo uma evolução e globalização da área da robótica.

Seguidamente serão dados a conhecer os elementos base do ROS que permitem que haja comunicação e recolha de informação.

#### Nós, tópicos e mensagens

Em ambiente ROS, um nó (*ROS node*) é o elemento base de tudo. São estes que executam todas as tarefas. Estes podem enviar ou recolher informação proveniente de outros nós. Esta informação é designada por mensagem, podendo ser de diferentes tipos (*int*, *float*, *string*, imagem, entre outros) consoante o que se deseje transmitir e/ou receber. Estas mensagens são transmitidas em tópicos. Estes são canais por onde circula a informação (mensagens), com uma certa frequência (*rate*). Cada nó pode publicar ou subscrever um ou mais tópicos. Por outras palavras, um tópico é sempre publicado por algum nó; e poderá ser subscrito por vários nós, ou até mesmo por nenhum. Isto é, um nó publica mensagens num tópico, sem qualquer informação sobre se este irá ser subscrito ou não, ou seja, não envia mensagens com um destinatário específico. Porém, quando um tópico é subscrito por diferentes nós, o ROS, internamente, gera canais paralelos para conetar os vários subscritores do tópico ao seu publicador, mesmo não o demonstrando - comunicação um para um.

No presente projeto, esta metodologia foi aplicada com vista a recolha de informação simultânea dos vários sensores ligados ao sistema.

Foi então criado um nó que publica os tópicos referentes às câmaras, sendo que, cada tópico circula informação de uma só câmara. Posteriormente, existe um outro nó responsável

por subscrever todos estes tópicos. Este último elemento, depois de receber as mensagens (em formato imagem) que estão a ser publicadas nos tópicos, é encarregue de juntar a informação, guardar as imagens, calibrar intrinsecamente e criar o ficheiro JSON presente no *dataset*.

## RQT

O RQT é uma ferramenta visual do ROS que permite observar o que está a acontecer em tempo real. É possível visualizar os diferentes nós ativos, bem como os tópicos que estes publicam e/ou subscrevem e as mensagens que estes transmitem. Na Figura 3.8 encontra-se um exemplo de uma utilização desta ferramenta no projeto em estudo, onde se pode perceber a existência de dois nós (elipses), e a existência de cinco tópicos (retângulos) que ligam ambos os nós. o nó *uvc\_camera\_stereo* publica as imagens de cada câmara nos tópicos respectivos e o nó *recolha\_informacao* subscreve cada um destes, tratando a informação recebida. Mesmo o RQT por si é um nó, não chegando, no entanto, a publicar nenhum tópico.

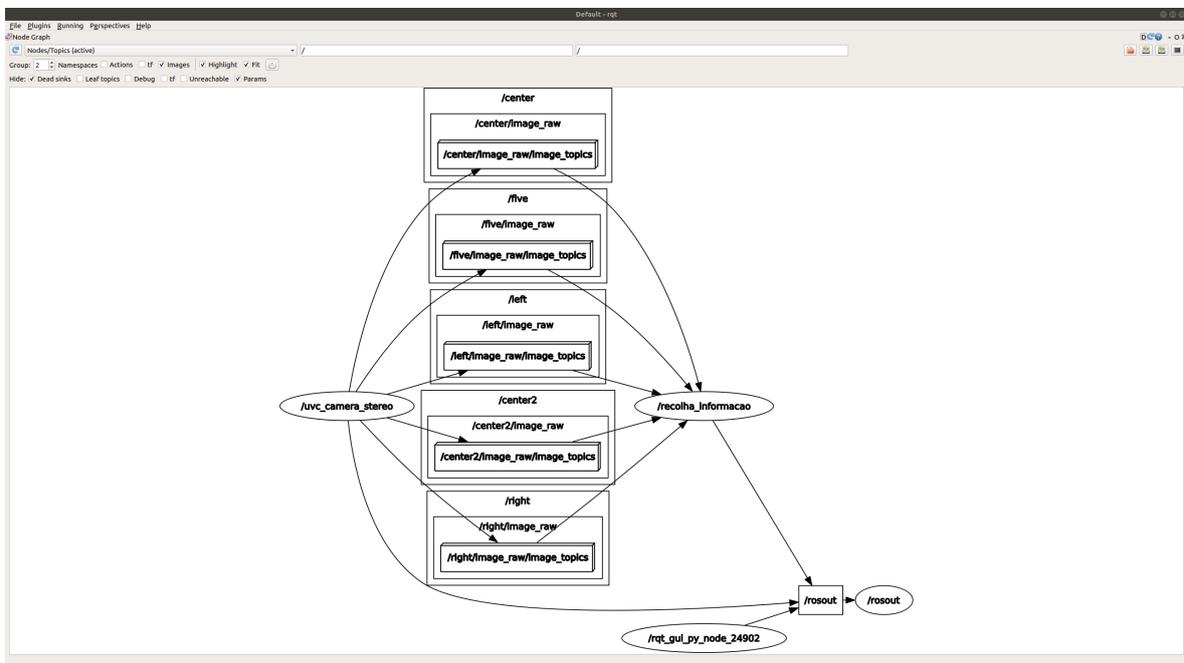


Figura 3.8: Esquema do RQT que demonstra os nós (elipses), tópicos (retângulos exteriores) e mensagens (retângulos interiores), que são utilizados no projeto.

## Rviz

O Rviz é um ambiente de visualização 3D, onde é possível, ou requerido, ao utilizador ter interação com este ambiente.

Na Figura 3.9 encontra-se o menu de interação do utilizador com o programa. É neste ambiente que o utilizador deverá realizar as tarefas necessárias para preparar a informação para a segunda parte do trabalho.

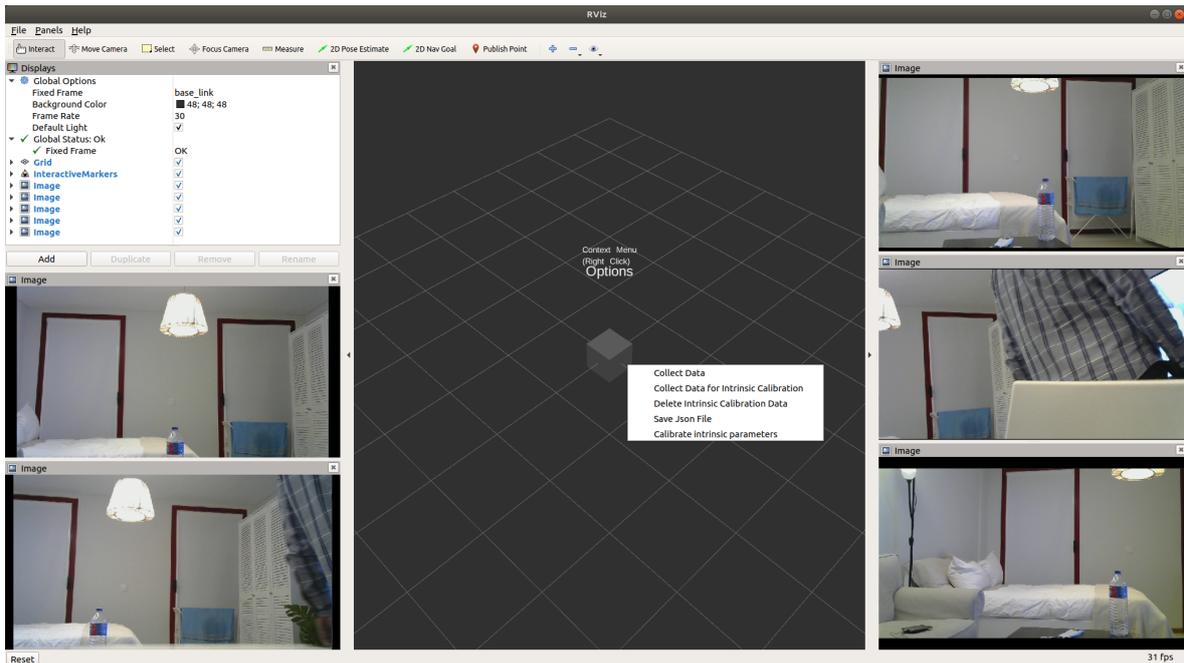


Figura 3.9: Imagem demonstrativa do ambiente Rviz que é apresentado ao utilizador. É neste que é requerida a sua intervenção de forma a captar imagens das câmaras, realizar a calibração intrínseca das mesmas e criar o ficheiro JSON com informações relevantes e necessárias para a calibração extrínseca.

A primeira destas tarefas será gravar imagens que são usadas para a calibração intrínseca. Para tal, o user deverá clicar em *Collect Data for Intrinsic Calibration*. Cada vez que este botão é premido, são guardadas as 5 imagens que as câmaras estão a detetar nesse momento numa pasta chamada *dataset\_intrinsc* no diretório atual. Como tal, é conveniente que o utilizador garanta que o padrão de calibração esteja a ser detetado por alguma câmara no instante de premir o botão e que acumule imagens onde, individualmente, todas as câmaras detetem o xadrez completo. De facto, o próprio utilizador, ao lançar o programa, deverá escolher qual o número mínimo que pretende de imagens sobre as quais se irá calibrar intrinsecamente.

Sempre que o programa é lançado de novo, a pasta *dataset\_intrinsic* contém já as imagens anteriormente guardadas. Como tal, existe a possibilidade de limpar este diretório, através do botão *Delete Intrinsic Calibration Data*.

Após a gravação do número mínimo de imagens (definido pelo próprio utilizador ao iniciar o programa), surgirá a opção *Calibrate intrinsic parameters*, a qual deve ser premida para calcular, então, os parâmetros intrínsecos. Caso o sistema reconheça que uma câmara específica (ou várias) não possua informação suficiente para calibrar os parâmetros intrínsecos desse mesmo sensor (nunca detetar o xadrez em nenhuma imagem), avisará o utilizador de que é necessário proceder à recolha de mais dados de forma a realizar esta acção com sucesso.

Caso a calibração intrínseca tenha sido concluída, é já possível guardar um ficheiro do tipo JSON, clicando em *Save json file*. No entanto, este JSON não conterá informação necessária suficiente, pretendida para a calibração extrínseca. Apenas iria conter os parâmetros intrínsecos calibrados. Deste modo, é necessário o utilizador recolher imagens, clicando em *Collect*

*Data*, para a calibração extrínseca que será realizada mais tarde, no processo de otimização. É nestas imagens guardadas, neste momento, que serão calculados todos os valores residuais, mencionados anteriormente e que serão detalhados posteriormente. Cada vez que o utilizador clica nesse botão, são guardadas as imagens que cada uma das câmaras está a detetar nesse momento; estas imagens são guardadas num diretório distinto daquele que foi utilizado para guardar as imagens referentes à calibração intrínseca: *dataset*.

No final de coletar todas as coleções que o utilizador considere necessárias, deverá clicar em *Save JSON file* onde o sistema irá adicionar ao JSON já existente com os parâmetros intrínsecos, ou criar um novo caso esse passo não tenha sido realizado (onde irá inserir esses mesmo parâmetros), toda a nova informação necessária: o nome de cada imagem detetada e gravada de cada sensor, a informação da deteção do xadrez para cada uma das coleções; e informação do padrão de calibração (Tamanho e número de pontos de interesse).

Este JSON é guardado no mesmo diretório que as imagens (*dataset*), uma vez que a segunda parte do projeto, a otimização, apenas terá acesso a este diretório. Sempre que o programa é lançado de novo, a pasta *dataset* é limpa, eliminando todos os ficheiros (imagens e JSON) que ela continha.

A recolha de dados para a calibração extrínseca e intrínseca são independentes e podem ser realizados com ordem aleatória, sendo que é exigido ao utilizador que este tenha realizado a calibração intrínseca para poder gravar os dados do processo. Sendo o objetivo da primeira metade do trabalho a recolha e estruturação de dados, e com os dados da calibração intrínseca a manterem-se guardados caso se volte a iniciar o programa, o utilizador pode desejar apenas adquirir novas imagens para o *dataset*. Assim, é requerido que este realize a calibração intrínseca cada vez que o programa é corrido, de forma a que estes valores sejam guardados no ficheiro JSON. Apenas este processo é obrigatório, visto que, o objetivo do utilizador ao correr o programa será sempre a recolha de imagens para a calibração extrínseca.

## Pacotes ROS

Em ambiente ROS, um pacote é algo que junta várias funcionalidades, ou seja, é um conjunto de vários elementos (nós, tópicos, etc.). Este projeto teve por base o pacote *uvc\_camera* que permitiu a criação do nó que publica os tópicos que contém as imagens das câmaras. Foi criado outro que contém o código relativo ao nó para a recolha e tratamento da informação.

### 3.1.2 Calibração Intrínseca

Com a finalidade de projetar um ponto do mundo 3D na imagem de uma câmara, tal como descrito anteriormente, algo bastante importante na segunda metade do trabalho - calibração extrínseca - é necessário ter informações relativas ao sensor em si, parâmetros inerentes a ele e independentes do espaço e cenário em que se encontra. Esta informação é encontrada na matriz intrínseca, Equação 3.1, de cada uma das câmaras. Cada câmara possui a sua matriz, podendo duas ter valores próximos caso sejam da mesma marca e modelo.

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

A matriz intrínseca contém então quatro valores importantes,  $f_x$  e  $f_y$  que correspondem às distâncias focais da câmara e  $c_x$  e  $c_y$  que são os píxeis do ponto principal da imagem,

normalmente o centro da mesma.

Para além desta matriz, as câmaras possuem coeficientes de distorção, também estes apenas associados ao respetivo sensor, de forma a dar conhecimento de um possível desvio da projeção retilínea na imagem que a câmara capta. Este fator acaba por influenciar diretamente a passagem para coordenadas 2D. Assim, existem coeficientes relativos à distorção radial,  $k_1$ ,  $k_2$ ,  $k_3$ ,  $k_4$ ,  $k_5$  e  $k_6$ , e outros referentes à distorção tangencial,  $p_1$  e  $p_2$ .

O número de coeficientes de distorção radial dependem da calibração intrínseca realizada, sendo pelo menos os dois primeiros necessários. Assim que o número mínimo de imagens, para realizar a calibração intrínseca, for garantido, é dada a instrução ao programa, por parte do utilizador, para efetuar a calibração dos parâmetros inerentes a cada câmara. Para tal, foi adaptada ao código a função `calibrateCamera`<sup>2</sup> do OpenCV. Esta mesma função retorna a matriz intrínseca na sua totalidade, Equação 3.1, e um vetor de distorções reduzido, Equação 3.2. Assim, neste projeto a calibração é feita tendo em conta os três primeiros radiais,  $k_1$ ,  $k_2$  e  $k_3$  e os dois tangenciais,  $p_1$  e  $p_2$ .

$$\mathbf{D} = [ k_1 , k_2 , p_1 , p_2 , k_3 ] \quad (3.2)$$

É recomendado que, para garantir bons resultados desta função, que sejam coletadas um mínimo de 10 imagens válidas, onde o xadrez é detetado, para cada câmara. Porém, é dada ao utilizador a possibilidade de ele mesmo definir este número mínimo. Isto deve-se ao facto de estes valores servirem apenas como ponto de partida, ou seja, na segunda metade do trabalho, ao realizar a calibração extrínseca, estes parâmetros serão também otimizados.

Estes parâmetros, matriz intrínseca e vetor de distorções, serão parte integrante do ficheiro JSON, contido dentro do *dataset* que é a ponte de ligação das duas partes do projeto.

### 3.1.3 Ficheiro Json

A estruturação e organização da informação é uma parte fundamental para o correto funcionamento do projeto. Com vista a realizar a calibração extrínseca, o programa necessita de consultar a informação recolhida na primeira metade do trabalho. Assim, ter todos estes dados organizados possibilita que sejam de fácil acesso.

Assim, ao premir o botão, em ambiente Rviz, para guardar as informações, é criado um ficheiro JSON. Este possui três diferentes níveis de informação, tal como é possível perceber pela Figura 3.2, onde existem três "caminhos" do fluxo que incorporam o ficheiro JSON: calibração intrínseca (*sensors*), dados relativos à calibração (*calibration\_config*) e informações das coleções guardadas (*collections*).

### Informação dos sensores

Neste nível de informação pode-se aceder aos dados dos sensores que serão imprescindíveis na segunda metade do trabalho. Neste menu, encontram-se sub níveis, sendo cada um deles relativo a cada sensor do sistema. Dentro de cada um referente a cada câmara encontram-se os parâmetros calculados na calibração intrínseca realizada (matriz intrínseca -  $K$  - e coeficientes de distorção -  $D$ ). A Figura 3.10 permite uma melhor percepção deste tema, ao representar a estrutura do ficheiro JSON dentro do menu referente aos dados dos sensores.

---

<sup>2</sup>[https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_calib3d/py\\_calibration/py\\_calibration.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html)

```

▼ object {3}
  ▼ sensors {5}
    ▼ left_camera {1}
      ▼ camera_info {2}
        ▼ K {9}
          0 : 869.080613
          1 : 0
          2 : 372.619394
          3 : 0
          4 : 856.727811
          5 : 193.495269
          6 : 0
          7 : 0
          8 : 1
        ▼ D {5}
          0 : 0.04878154868067422
          1 : 0.401181018460661
          2 : 0.002895646005398356
          3 : 0.003158290018303533
          4 : -1.5144287728811685
      ► center3_camera {1}
      ► right_camera {1}
      ► center2_camera {1}
      ► center_camera {1}
    ► calibration_config {2}
    ► collections {30}

```

Figura 3.10: Exemplo do nível de informação, num ficheiro JSON, relativa aos sensores, contendo os valores dos parâmetros intrínsecos (K corresponde aos valores da matriz intrínseca e D os coeficientes de distorção da câmara).

É ainda possível observar que cada indentação está relacionada com um nível inferior de informação. Assim, dentro do *object* (ficheiro), é possível perceber os três níveis anteriormente referidos: *sensors*, *calibration\_config*, *collections*.

### Configuração da calibração

Esta secção, permite ao processo realizado na segunda metade do trabalho ter acesso às configurações da calibração.

Por outras palavras, é aqui que se encontram as dimensões do xadrez, ou seja, o número de cantos no eixo  $x$  (dimensão "0") e no eixo  $y$  (dimensão "1"), e a largura de cada quadrado do padrão. Estes dados permitem facilitar a procura pelo padrão numa imagem, bem como perceber a sua posição no mundo real pela largura do desenho.

Para além da informação relativa ao xadrez de calibração, encontra-se também uma ordem que servirá de apoio à calibração extrínseca. Por norma, a ordem apresentada está relacionada com a disposição dos mesmos no sistema. Porém, não é obrigatório que seja desta forma uma

vez que irá apenas servir como um guia numa parte inicial do processo de calibração, onde será necessário que a ordem das câmaras seja conhecida.

Na Figura 3.11 representa o nível de informação, no ficheiro JSON, onde se encontram, então, estes dados relativos à configuração da calibração.

```
▼ object {3}
  ► sensors {5}
  ▼ calibration_config {2}
    ▼ sensor_order [5]
      0 : left_camera
      1 : center_camera
      2 : center2_camera
      3 : center3_camera
      4 : right_camera
    ▼ calibration_pattern {2}
      ▼ dimension [2]
        0 : 9
        1 : 6
        size : 0.019
      ► collections {30}
```

Figura 3.11: Exemplo do nível de informação, num ficheiro JSON, relativa à configuração da calibração, contendo as características do xadrez (número de esquinas em cada eixo e largura de cada quadrado) e a ordem dos sensores, que servirá apenas como um auxílio na segunda metade do trabalho - calibração extrínseca.

### Informações das coleções

Por fim, mas não menos importante, encontra-se o nível das informações relativas às coleções (conjunto de imagens retiradas de todos os sensores no mesmo instante temporal).

Cada coleção é representada por um valor que vai sendo incrementado à medida que são adicionadas novas coleções, com a primeira capturada a ter o valor "0".

Este nível de informação possui então a lista com todas as coleções guardadas na pasta *dataset*, sendo que a cada uma destas estão associadas informações relativas a cada sensor. Dentro do menu de cada coleção e de cada sensor encontra-se o nome da imagem retirada dessa mesma câmara e se o xadrez foi ou não detetado nessa imagem ("1" para caso positivo e "0" para caso negativo).

Na Figura 3.12 representa então este nível de informação, no ficheiro JSON, de forma a auxiliar a percepção do que anteriormente foi descrito.

```

▼ object {3}
  ► sensors {5}
  ► calibration_config {2}
  ▼ collections {30}
    ▼ 0 {1}
      ▼ data {5}
        ▼ left_camera {2}
          data_file : left_camera_0.jpg
          detected : 1
        ► center3_camera {2}
        ► right_camera {2}
        ► center2_camera {2}
        ▼ center_camera {2}
          data_file : center_camera_0.jpg
          detected : 0
      ► 1 {1}
      ► 2 {1}
      ► 3 {1}
      ► 4 {1}
      ► 5 {1}
      ► 6 {1}
      ► 7 {1}
      ► 8 {1}
      ► 9 {1}
      ► 10 {1}
      ► 11 {1}
      ► 12 {1}
      ► 13 {1}
      ► 14 {1}
      ► 15 {1}

```

Figura 3.12: Exemplo do nível de informação, num ficheiro JSON, relativa às coleções capturadas para a calibração extrínseca (segunda metade do trabalho), contendo o nome de cada imagem e se o xadrez foi detetado nessa mesma imagem. Cada coleção tem estas informações em relação a cada uma das câmaras.

---

## 3.2 Calibração Extrínseca

Após concluída a recolha e estruturação de dados, é então altura de realizar a segunda parte deste trabalho, coincidente com o objetivo do projeto: calibrar extrinsecamente. Uma calibração extrínseca passa pela determinação da posição relativa de todos os sensores de um sistema, neste caso, robótico.

Cada sensor possui, ou pode ser representado por, um referencial. Deste modo, de forma a todos os sensores estarem calibrados é necessário que se saiba a posição e orientação entre todos esses referenciais ou a pose destes para um comum. De forma a estimar estes parâmetros

(posição e orientação de cada referencial) é utilizado um processo de otimização. Devido à complexidade do problema, será uma otimização não linear, ou seja, as funções a otimizar não são do tipo linear.

Uma otimização é então um processo que tem como finalidade melhorar um conjunto de parâmetros, de forma a minimizar o erro (ou maximizar o lucro). O cálculo do erro (ou lucro) através dos parâmetros a otimizar faz-se através de uma função. Esta, designada por função objetivo, poderá ser do tipo função de custo, produzindo um erro que se pretenderá diminuir, ou função de lucro, produzindo um lucro que se pretenderá maximizar, dependendo da natureza do problema. A Figura 3.13 representa a relação entre os parâmetros a otimizar, a função objetivo e o erro calculado.

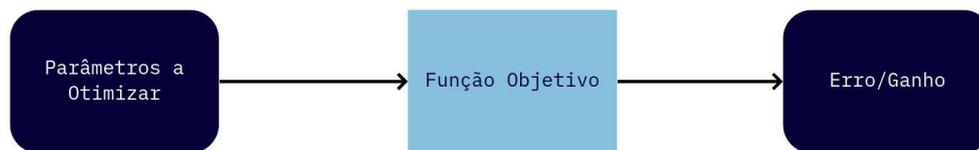


Figura 3.13: Esquema representativo do processo de uma função objetivo, que recebe uns parâmetros de entrada e computa um vetor com erros ou lucros.

Assim sendo, antes de se conhecer o processo realizado pela função objetivo, será necessário detalhar os valores de entrada que serão providenciados à função e, conseqüentemente, otimizados.

### 3.2.1 Parâmetros a otimizar

O objetivo do presente projeto passa por estimar as posições e orientações de todas as câmaras constituintes de um só sistema. Neste caso, a calibração extrínseca de todas as câmaras a bordo de um veículo autónomo. Isto permite que qualquer objeto detetado no campo de visão de uma destas possa ser localizado e, possivelmente, identificado.

Matematicamente, para determinar a posição de qualquer objeto num espaço tridimensional (como é o mundo real) basta apenas definir três variáveis:  $x$ ,  $y$  e  $z$ , Equação 3.3. Não só a posição, mas também a orientação de qualquer objeto segue a mesma lei. Assim sendo, a juntar-se aos três parâmetros relativos à posição, considerar-se-á mais três parâmetros:  $r_1$ ,  $r_2$  e  $r_3$ , Equação 3.4.

$$t = \{x, y, z\} \quad (3.3)$$

$$r = \{r_1, r_2, r_3\} \quad (3.4)$$

Como já foi mencionado no Capítulo 1, de modo a sintetizar o número de parâmetros, optou-se por se definir as orientações segundo a notação Rodrigues.

Note-se que estes valores de posição e orientação caracterizam uma pose de um referencial (por exemplo, de um objeto) em relação a outro (por exemplo, referencial abstrato ou de outro objeto). Assim sendo, os 6 parâmetros associados a cada câmara descreverão a relação entre o referencial dessa mesma câmara e outro qualquer referencial. Conseqüentemente, é necessário definir qual ou quais os referenciais base para a determinação das poses de todas as câmaras.

Como primeira hipótese, visto que o objetivo do trabalho é obter as poses de todas as câmaras, umas em relação às outras, surge a possibilidade de determinar a posição e orientação de cada câmara em relação à anterior. Na Figura 3.14 pode-se ver um exemplo destas relações de forma a tornar mais explícito o problema em questão, com 5 sensores constituintes do sistema.

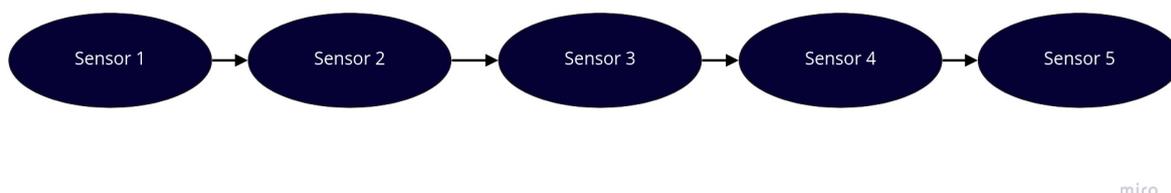


Figura 3.14: Sequência de calibrações par a par

Esta hipótese acarreta uma forte probabilidade de resultar em desvios de poses demasiado elevados na calibração geral. Tal acontece porque, apesar da calibração ser simultânea, os valores de posição e orientação estimados de um sensor (em relação ao anterior) terão um erro prévio associado à posição e orientação do sensor de referência - o anterior. A título de exemplo, a Figura 3.15 representa um cenário onde é possível verificar este tipo de desvios.

Como se pode visualizar, apesar da estimativa da pose do sensor 3 em relação ao sensor 2 estar correta, a posição e orientação final deste mesmo sensor 3, no sistema, está errada, devido a um erro prévio na determinação da pose do sensor 2 em relação ao primeiro.

De forma a culmar a lacuna explicada relativa à primeira hipótese, surge a sugestão de estimar a pose de cada câmara em relação a todas as restantes. Por outras palavras, pretende-se que a calibração considere todas as combinações possíveis de cada par de sensores. Na Figura 3.16 encontra-se a representação deste conceito que é agora sugerido.

Esta metodologia contorna o problema apresentado na primeira hipótese de arranjo sequencial dos sensores. No entanto, a consideração de todas as combinações possíveis entre duas câmaras do sistema, leva a um aumento do número de erros a serem calculados pela função objetivo, tornando o processo pesado computacionalmente e possivelmente mais demorado.

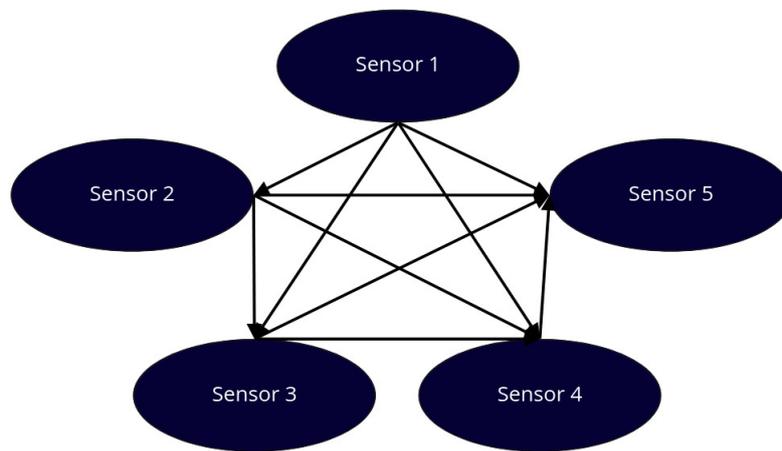


(a) Posição Real



(b) Exemplificação de uma possível calibração sequencial

Figura 3.15: Em (a) encontra-se a posição real de 3 câmaras, sendo que em (b) se pode encontrar um exemplo das posições estimadas por uma calibração sequencial. Apesar da câmara 3 em relação à câmara 2 estar numa correta posição, devido a um erro na posição da segunda câmara com a n<sup>o</sup>1, acaba por não estar na sua posição real.



miro

Figura 3.16: Sequência de calibrações par a par

Na tentativa de, sem cair novamente no acumular de erros, tornar o processo o mais leve possível, propõe-se a definição de um referencial que será global e absoluto a todo o sistema. As posições e orientações, a serem otimizadas, de todas as câmaras serão relativas a este mesmo único referencial. Na Figura 3.17 está esquematizado este método em que a pose de todos os sensores é estimada em relação a um referencial global.

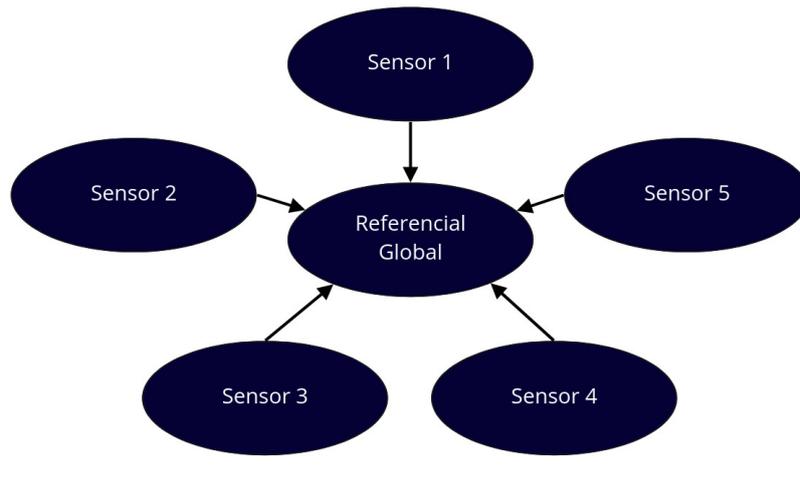


Figura 3.17: Modelo onde todos os sensores têm a sua posição e orientação relativa ao mesmo referencial global

Tendo em conta que o objetivo do trabalho é calibrar as câmaras que serão utilizadas para a condução autónoma de um veículo inteligente, tem-se como expectativa que o número de câmaras/sensores desse sistema seja relativamente elevado. Das três sugestões apresentadas, é de direta conclusão que a primeira será desde já descartada devido ao provável erro final que apresentará. De facto, o aumento do número de sensores constituintes do sistema leva a um aumento do erro associado, pelo que, não é o ideal para a situação em estudo.

As duas restantes metodologias, tendo em conta as combinações de câmaras duas a duas e calibrando tudo em relação a um referencial global, não apresentam, em princípio, problemas com erro acumulado. Utilizando o referencial global, por exemplo, a transformação do sensor 1 para o sensor 4, terá sempre e apenas os erros associados à calibração do primeiro para o referencial e do referencial para o quarto, independentemente do número total de sensores do sistema: nunca será mais do que uma combinação de duas poses estimadas para determinar a posição entre quaisquer dois sensores do sistema.

Sendo que, a nível de erro médio final, as duas últimas metodologias apresentadas não apresentam grande disparidade, como forma de decisão entre elas, olha-se para a exigência computacional de cada uma, uma vez que o objetivo desta calibração é a sua aplicação a um sistema robótico complexo, com muitos sensores, como é o caso de um veículo autónomo.

Na Figura 3.18 está explícita a forma como o número de poses a estimar aumenta em cada um dos casos, quando se adicionam sensores aos sistema. Na Equação 3.5 está descrita a função utilizada para calcular o número de combinações, onde  $N$  é o número de câmaras no

sistema. Inicialmente, a diferença entre ambos é praticamente nula, porém para um sistema mais complexo, tal como é exigido atualmente, com mais sensores a diferença é já acentuada. Para um sistema robótico com 10 sensores instalados, utilizando as combinações de câmaras são otimizadas 4,5 vezes mais transformações durante a calibração. Sendo uma função que aumenta exponencialmente, cada vez a diferença será mais acentuada.

$$\text{N}^\circ \text{ Transformações} = {}^N C_2 = \binom{N}{2} \quad (3.5)$$

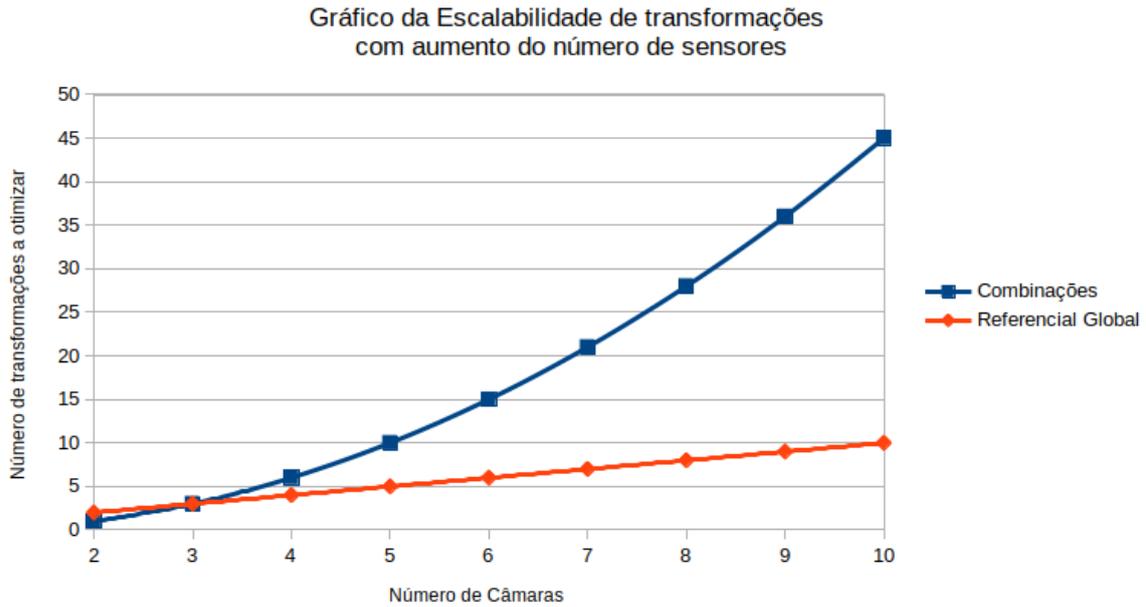


Figura 3.18: Comparação do número de transformações entre sensores a otimizar considerando combinações 2 a 2 e calibrando todos os sensores em relação ao referencial global

Tendo tudo isto em consideração, a utilização de um referencial global torna o processo mais eficiente. Para além de não acumular erros com o aumento de sensores, não requer demasiado esforço computacional, permitindo assim um elevado número de sensores associados.

Deste modo, os parâmetros, de posição e orientação, de uma câmara, descritos anteriormente, serão relativos a um referencial global, Equação 3.6. A definição deste mesmo referencial será detalhada posteriormente.

$$\Omega = \left\{ \overbrace{r_1, r_2, r_3, t_x, t_y, t_z}^{\text{Câmara 1}}, \overbrace{r_1, r_2, r_3, t_x, t_y, t_z, \dots}^{\text{Câmara 2}}, \dots, \overbrace{r_1, r_2, r_3, t_x, t_y, t_z}^{\text{Câmara N}} \right\} \quad (3.6)$$

Para além dos valores de posição e orientação de cada câmara, também os seus parâmetros intrínsecos serão otimizados, pelo que deverão ser incluídos no conjunto de valores de entrada da função de custo. Graças a esta inclusão, a calibração intrínseca realizada na primeira parte

do trabalho (ver Subsecção 3.1.2) não necessita de ser tão precisa e demorada, tornando esse processo mais simples. É por esta razão que o utilizador pode definir o número de imagens mínimas para efetuar a calibração intrínseca.

Deste modo, um vetor onde estarão incluídos os valores intrínsecos (matriz e coeficientes de distorção), Equação 3.7, estará presente nos valores que serão fornecidos à função objetivo.

$$\Psi = \left\{ \overbrace{f_x, 0, c_x, 0, f_y, c_y, 0, 0, 1}^{\text{Matriz Intrínseca}}, \overbrace{k_1, k_2, p_1, p_2, k_3}^{\text{Coeficientes Distorção}} \right\} \quad (3.7)$$

Por fim, não sendo objetivo da calibração, também serão considerados como parâmetros a otimizar a posição e orientação de cada um dos padrões de calibração presentes nas várias coleções contidas no *dataset*. Estes valores de posição e orientação de cada xadrez, Equação 3.8, serão também referentes ao referencial global mencionado anteriormente. Ao se otimizar as poses dos vários xadrezes (das múltiplas coleções) permite que não existam restrições de poses a nenhum dos intervenientes do processo, não obrigando a posição e orientação dos sensores a adaptar-se a um outro objeto fixo que poderia não estar na sua real posição.

$$\Upsilon = \left\{ \overbrace{r_1, r_2, r_3, t_x, t_y, t_z}^{\text{Xadrez, Coleção 1}}, \dots, \overbrace{r_1, r_2, r_3, t_x, t_y, t_z}^{\text{Xadrez, Coleção N}} \right\} \quad (3.8)$$

Concluindo, todos os valores apresentados, Equação 3.6, Equação 3.7 e Equação 3.8, constituem o vetor de parâmetros a otimizar. Este vetor, Equação 3.9, será providenciado à função custo, sendo todos os seus valores constituintes aproximados, continuamente, do seu valor real.

$$\Phi = \left\{ \overbrace{\Omega}^{\text{Poses Câmaras}}, \overbrace{\Psi}^{\text{Parâmetros Intrínsecos}}, \overbrace{\Upsilon}^{\text{Padrões de Calibração}} \right\} \quad (3.9)$$

Para que o processo de optimização se inicie, é necessário fornecer à função de custo os valores iniciais, que vão servir de ponto de partida para o otimizador. Este conjunto de valores é designado por *first guess*.

### 3.2.2 Parâmetros Iniciais

Conhecendo já os parâmetros que servem como entrada para a função objetivo, é necessário, então, para cada um deles, definir os seus primeiros valores. É a partir destes que o otimizador vai, a cada iteração, tentando chegar aos valores ideais para cada parâmetro.

Em modelos de otimização não lineares, mais complexos, como é o caso em estudo, os valores iniciais que são atribuídos aquirem uma maior importância e relevância. A função objetivo, não sendo linear, pode conter vários mínimos: o global e outros locais. É então facilmente perceptível que o objetivo da função passa por encontrar o valor mínimo global. Porém, caso a função passe por um mínimo local, pode acontecer terminar nesse mesmo valor, visto que à sua volta todos os valores fazem aumentar o erro total que retorna desta mesma função. Deste modo, um *first guess* perto do valor mínimo global diminui bastante as probabilidades de o otimizador encontrar, e conseqüentemente terminar, num mínimo local. Tendo isto em

consideração, foi criada, no programa, uma classe denominada por *FirstGuessCollections*, com vários métodos associados, para calcular uns valores iniciais aproximados dos reais:

## Método para criar lista de objetos

Proveniente da primeira fase do projeto, contido no dataset, encontra-se já o *first guess* dos parâmetros intrínsecos da câmara. Deste modo, fica perceptível que na calibração intrínseca realizada não seja exigido o rigor que normalmente se aplicaria, visto que apenas se procura um primeiro valor para estes parâmetros e que os valores retornados, apesar de não serem os reais, serão já bastante próximos.

Um dos primeiros métodos da classe a aplicar é o *Sensorclass*, que retira a informação do ficheiro JSON (matriz intrínseca e vetor de distorções) e cria uma lista de objetos. Cada um destes objetos corresponde a uma câmara e contém as informações necessárias relativas a cada uma destas.

## Método para criação de matriz de deteção

Este método, ao ler o ficheiro JSON contido no *dataset* providenciado, obtém a informação de quais as câmaras, para cada coleção, que detetam o xadrez. Através desta informação, cria uma matriz de valores binários (1 e 0) onde cada linha representa uma coleção e cada coluna representa um sensor. Por exemplo, num sistema com cinco câmaras onde foram registadas 27 coleções, a matriz criada por este método iria ter dimensão 27x5. Os valores de 1 simbolizam as deteções existentes entre as várias câmaras e coleções, ao passo que os valores 0 representam imagens onde não foram detetados quaisquer padrões de calibração. A Tabela 3.1 representa a matriz criada nesse mesmo exemplo.

Tabela 3.1: Matriz de deteções do xadrez, onde "0" representa que o padrão não é detetado e "1" que existe deteção. Cada coluna corresponde a um sensor e cada linha a uma coleção

	<i>left_camera</i>	<i>center_camera</i>	<i>center2_camera</i>	<i>center3_camera</i>	<i>right_camera</i>
Coleção 0	1	1	0	0	0
Coleção 1	0	0	0	0	1
Coleção 2	0	0	0	1	1
Coleção 3	0	1	1	1	0
Coleção 4	1	1	0	0	0
Coleção 5	0	0	0	0	0
Coleção 6	0	0	0	0	0
Coleção 7	1	1	1	0	0
...	...	...	...	...	...
...	...	...	...	...	...
Coleção 25	1	0	0	0	0
Coleção 26	0	0	0	1	1
Coleção 27	0	1	1	0	0

## Método para filtrar matriz calculada

Neste método é filtrada a matriz criada anteriormente: todas as linhas que contenham no máximo, apenas, uma deteção (ou seja, um '1') serão eliminadas. Tal como explicado no início deste mesmo capítulo, estas linhas da matriz (coleções) que serão descartadas designam-se por coleções inválidas, que não contêm informação de interesse para o correto funcionamento da otimização.

Durante esta filtragem, é criada uma lista com todos os números identificadores das coleções consideradas como válidas.

## Método que procura ligação de todos os sensores com o referencial global

Como detalhado na Subsecção 3.2.1, para se designar valores de posição e orientação de cada câmara, é necessário que estes valores descrevam a pose a partir de um referencial. Seguindo a escolha relatada nessa mesma secção, é utilizado um único referencial global para todos os sensores. Este referencial global será o referencial da primeira câmara na lista da ordem de sensores (ver Figura 3.11), ou seja, a *left\_camera*. Como tal, os valores de *first\_guess* de cada câmara representarão a sua pose inicial relativamente à posição inicial do primeiro sensor (*left\_camera*).

Este método, através da matriz de coleções válidas já calculada, irá detetar as coleções que permitam comparar as posições e orientações de cada câmara em relação à primeira. Esta ligação entre cada uma das câmaras com a *left\_camera* poderá ser determinada direta ou indiretamente; esta última opção acontece caso não exista nenhuma coleção onde a câmara em questão e a *left\_camera* tenham ambas detetado o xadrez. Para cada câmara, caso exista uma coleção com deteção comum entre esta e a câmara do referencial global, esta coleção (associada a esse par de câmaras) será adicionada a uma lista de interesse criada, onde ficarão guardadas as colções e os pares de câmaras respetivos. Coleções que emparelhem o mesmo conjunto de dois sensores já existentes nessa lista não serão consideradas. De facto, para se determinar as poses de cada câmara relativamente à *left\_camera* só é necessário uma única cadeia de ligações. Para o caso de um determinado sensor não possuir nenhuma deteção em simultâneo com a câmara do referencial global, o presente método irá procurar por emparelhar este sensor com outro (que não o global) que já exista na lista de interesse. Deste modo, através de cálculo direto, é facilmente possível determinar a posição da câmara em questão com a *left\_camera*, através de uma composição de duas (ou mais, se necessário) ligações consecuentes. Um exemplo de lista de interesse criada por este método pode ser encontrada na Tabela 3.2.

Tabela 3.2: Matriz de interesse criado de forma a estabelecer ligação de todos os sensores ao que inicialmente se encontra no referencial global.

Coleção	Sensor 1	Sensor 2
0	<i>left_camera</i>	<i>center_camera</i>
7	<i>left_camera</i>	<i>center2_camera</i>
3	<i>center_camera</i>	<i>center3_camera</i>
2	<i>center3_camera</i>	<i>right_camera</i>

Neste exemplo, verifica-se que o sensor `'center_camera'` deteteou o xadrez numa mesma coleção que a camara global -a coleção 0. Por outro lado, o sensor `'center3_camera'` já não deteteou o xadrez numa coleção comum com a `'left_camera'`, daí ter sido emparelhado com a `'center_camera'`.

## Método para criação de matrizes de transformação dos sensores

Combinando os valores de posição e orientação, é possível construir uma matriz que representa uma transformação geométrica. Uma matriz de transformação geométrica não é mais do que a relação entre dois referenciais. O mesmo ponto, num dado espaço, poderá ser descrito com coordenadas referentes a um referencial ou com coordenadas distintas associadas a um referencial distinto. A matriz de transformação geométrica entre os dois referenciais serve, por exemplo, para mudar o sistema de coordenadas com que se descreverá um mesmo ponto.

Os valores de translação (referentes à posição) matêm-se com a mesma estrutura com que já foram apresentados na Equação 3.3. Já os valores relativos à rotação (orientação) necessitam de serem descritos segundo a notação cartesiana (e não Rodrigues). A notação cartesiana de valores de rotação caracteriza-se por uma matriz de dimensão 3x3, onde cada coluna representa a rotação em torno de cada eixo. A função do OpenCV, Rodrigues<sup>3</sup>, permite, facilmente, transformar um vetor, referente à orientação, numa matriz, com as dimensões referidas, em coordenadas cartesianas. Na Equação 3.10 encontra-se um exemplo do que é a transformação de um referencial para outro.

$$\text{Referencial 1} \mathbf{T}_{\text{Referencial 2}} = \begin{bmatrix} r11 & r12 & r13 & tx \\ r21 & r22 & r23 & ty \\ r31 & r32 & r33 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.10)$$

De forma a realizar multiplicação de duas matrizes de transformação, ou até para calcular a inversa de uma matriz deste mesmo tipo, é realizada a sua homogenização. Visto que as suas dimensões seriam de 3 linhas por 4 colunas, é adicionada uma última linha.

Ao multiplicar duas matrizes, obtém-se uma matriz de transformação composta. A título de exemplo, na Figura 3.19 visualiza-se que multiplicando a matriz de transformação geométrica entre o ponto A e o ponto B com a matriz de transformação entre o ponto B e o ponto C, obtém-se a matriz de transformação geométrica de ponto A para o ponto C.

As matrizes de transformação geométrica poderão ser invertidas, representando, nesse caso, a transformação inversa, ou seja, entre os mesmos pontos mas em sentidos opostos. Por exemplo, na Figura 3.20, tendo a matriz de transformação geométrica entre o ponto A e o ponto B, e a transformação entre o ponto B e o ponto C, facilmente se determina a transformação entre A e C, através da seguinte equação (Equação 3.11):

$${}^A\mathbf{T}_C = {}^A\mathbf{T}_B \cdot {}^C\mathbf{T}_B^{-1} \quad (3.11)$$

Nesta última equação, o primeiro termo representa a posição e orientação do ponto C relativamente ao ponto A. Do mesmo modo, o segundo termo corresponde ao ponto B relati-

<sup>3</sup>[https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html#rodrigues](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#rodrigues)

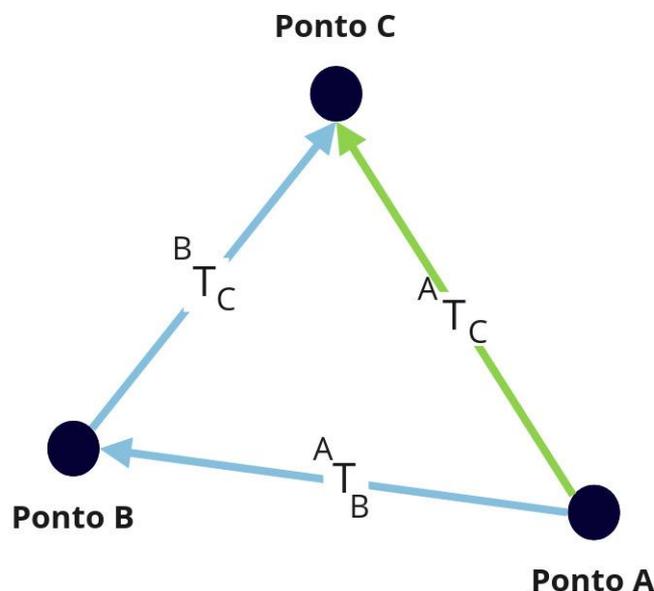


Figura 3.19: Imagem representativa de transformações compostas. A transformação do ponto A para o ponto B caracteriza o trajeto realizado para ir desde o ponto A para o B. Consequentemente, a transformação de B para C representa o mesmo. Assim, ao conjugar as duas obtém-se a transformação de A para C.

vamente ao primeiro. Já o último, corresponde à matriz inversa de C para B que é então a transformação geométrica de B para C.

Depois desta familiarização com o termo transformação geométrica e respectivas funcionalidades, é possível então perceber o processo realizado neste método. Este tem como objetivo a criação de uma matriz de transformação geométrica entre cada câmara e o referencial global (posição inicial do primeiro sensor), retornando, posteriormente, uma lista com todas estas matrizes. Visto que, o referencial global se encontra na posição e orientação inicial da primeira câmara, a matriz de transformação deste sensor para, este mesmo referencial, corresponde à matriz identidade. Esta matriz, Equação 3.12, não é mais que a transformação de um ponto para ele mesmo, ou de um referencial para ele mesmo. Isto acontece pelo facto de os referenciais serem exatamente iguais.

$$\text{Matriz Identidade} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

Para a determinação de cada uma das matrizes restantes, este método recorre, primeiramente, à lista de interesse, criada anteriormente, onde se encontram as ligações dos vários sensores ao referencial global, sejam estas diretas ou indiretas, bem como a indicação do nú-

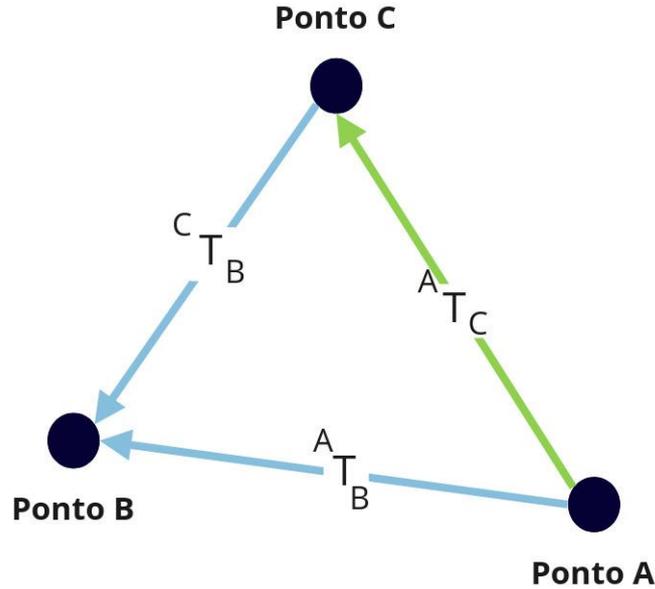


Figura 3.20: Imagem representativa de transformações compostas, considerando as matrizes inversas. A transformação do ponto A para o ponto B caracteriza o trajeto realizado para ir desde o ponto A para o B. Consequentemente, a transformação de C para B representa o mesmo. De forma a calcular a matriz de A para C, basta que se conjugue a transformação de A para B com a inversa de C para B

mero da coleção onde cada surge cada ligação. Sabendo então o número da coleção e o nome de ambos os sensores envolvidos, é então possível aceder à imagem de cada câmara nesta mesma coleção. Através de uma imagem, onde o xadrez é detetado, e das características intrínsecas da respetiva câmara (guardadas no método *sensorClass*) é utilizada uma função do OpenCV, *solvePNP*, que retorna dois vetores (posição e orientação) que permitem, com o auxílio da função Rodrigues, construir a matriz de transformação do referencial da câmara para o referencial do padrão de calibração. Deste modo, realizando o mesmo processo na imagem restante, determina-se a posição e orientação do xadrez relativamente a ambas as câmaras em estudo. Com esta informação, e tendo em consideração a referência teórica apresentada, é possível calcular a matriz de transformação geométrica de uma câmara para a outra. Isto acontece multiplicando a matriz de transformação de uma câmara para o xadrez pela matriz inversa da outra câmara para o mesmo xadrez, como se pode perceber pela Figura 3.21. Então, o cálculo realizado pode ser encontrado na Equação 3.13.

$$\text{Câmara } 2 \mathbf{T}_{\text{Câmara } 1} = \text{Câmara } 2 \mathbf{T}_{\text{Xadrez}} \cdot \text{Câmara } 1 \mathbf{T}_{\text{Xadrez}}^{-1} \quad (3.13)$$

Esta matriz inversa representa a transformação do referencial do padrão de calibração para o referencial do objeto de percepção. Caso um destes sensores seja o que contém o referencial

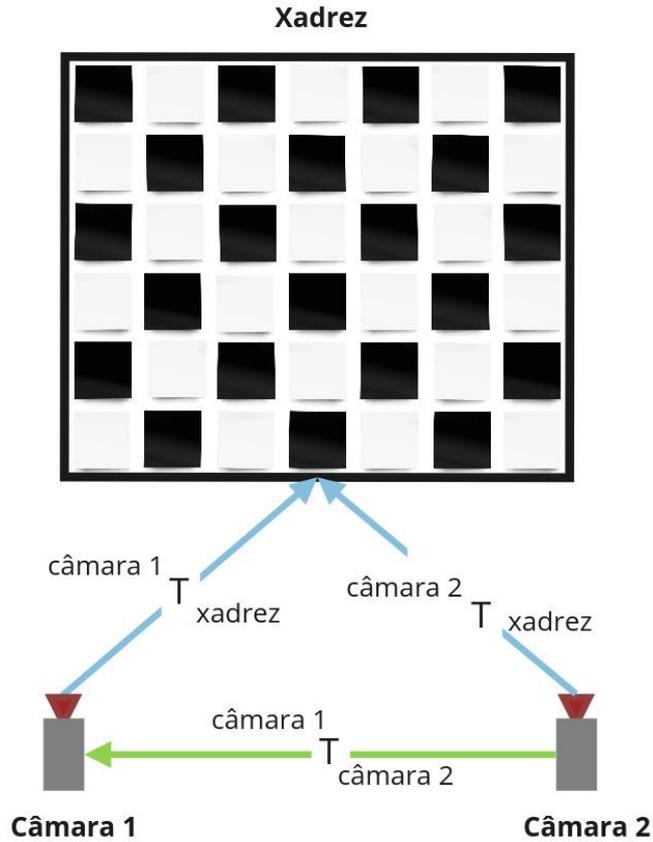


Figura 3.21: Nesta imagem está representado o esquema do que é necessário para calcular o *first guess* de cada câmara. A verde é a transformação desejada (da câmara 2 para a câmara 1) e a azul as transformações calculadas a partir da função solvePNP. Multiplicando a matriz de transformação da câmara 2 para o xadrez com a inversa da outra transformação calculada, obtém-se a matriz desejada.

global, a matriz calculada é já a matriz desejada. Caso contrário, um dos sensores terá já a sua matriz de transformação para este mesmo referencial, sendo isto perceptível pelo processo explicado no método anterior. Assim, a matriz calculada terá de ser multiplicada pela matriz de transformação desse sensor para o referencial global, Equação 3.14. Este procedimento é realizado para cada linha da lista de interesse, levando a que todos os sensores possuam, por fim, a sua posição e orientação iniciais estimadas.

$$\text{Câmara 2} \mathbf{T}_{\text{Referencial Global}} = \text{Câmara 2} \mathbf{T}_{\text{Câmara 1}} \cdot \text{Câmara 1} \mathbf{T}_{\text{Referencial Global}} \quad (3.14)$$

## Método para criação das matrizes de transformação geométrica dos padrões de calibração

Depois de determinados os valores iniciais para os parâmetros intrínsecos e para a posição e orientação de cada sensor, resta, por último, determinar a pose de cada xadrez, em relação ao referencial global, nas coleções consideradas válidas. Este método retorna, então, uma lista com as várias matrizes de transformação relativas ao padrão de calibração em cada coleção.

Para atingir este objetivo, o presente método percorre a matriz filtrada, contendo apenas as coleções válidas, onde para cada linha (cada coleção) determina a posição e orientação do xadrez. Para tal, partindo da primeira coluna (sensor), procura a primeira câmara que, nessa coleção, detete o padrão de calibração na respetiva imagem. Partindo da imagem, é utilizada a função do OpenCV, *solvePNP*, tal como no anterior método. Esta retorna a posição e orientação do xadrez relativamente ao sensor em estudo, sendo, mais uma vez, construída uma matriz de transformação, com o auxílio da função Rodrigues. Visto que o objetivo é obter a transformação geométrica deste padrão para o referencial global, e que cada sensor possui já uma matriz de transformação para esse mesmo referencial, multiplica-se a inversa da matriz calculada pela matriz de transformação deste mesmo sensor para o referencial principal (ver Figura 3.22). Ao calcular esta multiplicação, determina-se a matriz de transformação geométrica do padrão de calibração para o referencial global, Equação 3.15.

$$\text{Xadrez}\mathbf{T}_{\text{Referencial Global}} = \text{Câmara}\mathbf{T}_{\text{Xadrez}}^{-1} \cdot \text{Câmara}\mathbf{T}_{\text{ReferencialGlobal}} \quad (3.15)$$

Tal como já referido, este processo será repetido para todas as coleções.

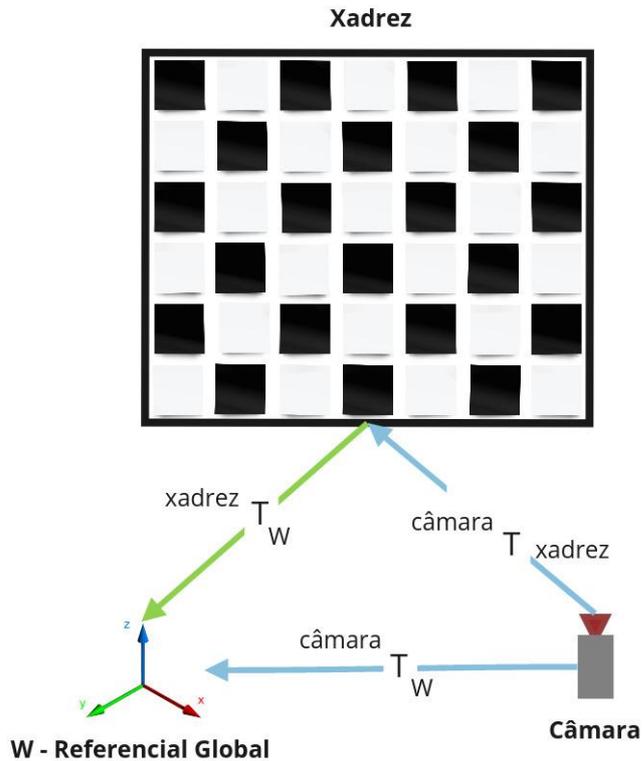


Figura 3.22: Nesta imagem está representado o esquema do que é necessário para calcular o *first guess* de cada xadrez (ou seja, o padrão de cada coleção). A verde é a transformação desejada (do xadrez para o referencial global) e a azul as transformações que foram calculadas. Multiplicando a inversa da matriz de transformação da câmara para o xadrez com a transformação da câmara para o referencial global, obtém-se a matriz desejada.

### Método que permite visualizar resultados da classe

Para efeitos de visualização dos resultados de todo este processo, tendo servido também como um auxílio durante a construção do mesmo, este método mostra a projeção dos cantos do xadrez nas imagens capturadas pelas câmaras, apenas usando as posições e orientações iniciais (*first guess*) calculadas. Para tal, é necessário ter o conhecimento de todos os pontos, referentes aos cantos do xadrez, no referencial do padrão de calibração. De forma a facilitar a sua determinação, é considerado o eixo  $z$  perpendicular ao plano que contém o padrão. Este processo é já realizado automaticamente pelas funções do OpenCV, que consideram também o eixo  $x$  como o lado do xadrez com maior dimensão e o eixo  $y$  o restante. Deste modo, sabendo a largura de cada quadrado, facilmente se cria uma matriz com os pontos dos cantos do xadrez no referencial do próprio objeto. Com a finalidade de projetar os pontos na imagem, é necessário que estes sejam vistos partindo do referencial da mesma. Assim, multiplicando cada ponto pela matriz de transformação da câmara para o xadrez, obtém-se a coordenada deste no referencial do sensor, Equação 3.16.

$$\mathbf{p}_{\text{Câmara}} = \text{Câmara} \mathbf{T}_{\text{Xadrez}} \cdot \text{Câmara} \mathbf{p}_{\text{Xadrez}} \quad (3.16)$$

Porém, todas as matrizes, a que o programa tem acesso neste ponto, representam a transformação para o referencial global, independentemente do objeto em que partir. Assim, é necessário multiplicar a matriz de transformação da câmara em questão pela inversa da matriz do xadrez da coleção, Equação 3.17.

$$\text{Câmara}\mathbf{T}_{\text{Xadrez}} = \text{Câmara}\mathbf{T}_{\text{Referencial Global}} \cdot \text{Xadrez}\mathbf{T}_{\text{ReferencialGlobal}}^{-1} \quad (3.17)$$

Este processo é realizado pela função *projectPoints*<sup>4</sup>, também do OpenCV, retornando, esta função, a matriz com a coordenada dos pontos, do xadrez, em píxeis. Esta função necessita das coordenadas dos pontos no referencial do xadrez, a informação da posição e orientação deste, em relação à câmara, e dos parâmetros intrínsecos do sensor. Com as coordenadas em píxeis, basta recorrer à função *drawChessboardCorners*<sup>5</sup> para projetar estes mesmos pontos na imagem. Na Figura 3.23 pode-se ver as projeções do *first guess* nas imagens utilizadas para o cálculo do mesmo.

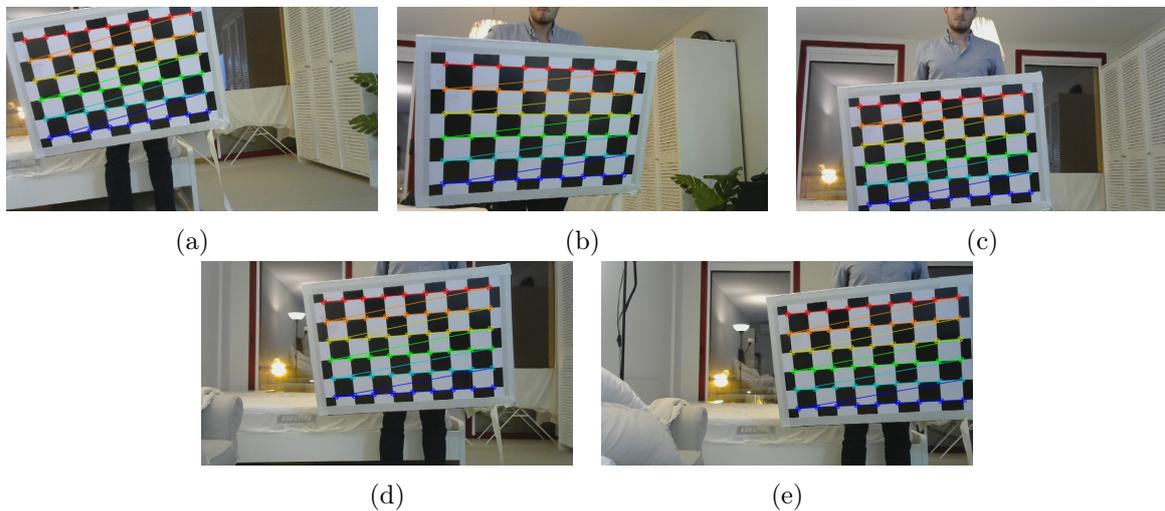


Figura 3.23: Imagens utilizadas para calcular o *first guess*: a) *left\_camera*, b) *center\_camera*, c) *center2\_camera*, d) *center3\_camera*, e) *center\_camera*. Como se pode verificar, para estas imagens o *first guess* está bastante preciso, porém, é algo expectável, visto terem sido as utilizadas para calcular o mesmo.

É possível ver que é bastante preciso o *first guess*, mas, sendo estas as imagens utilizadas para o cálculo de cada matriz relativa ao sensor, é de esperar que os resultados sejam melhores do que noutra qualquer imagem.

Todo este processo de cálculo dos parâmetros iniciais a fornecer à função objetivo não requer qualquer interação do utilizador. Aliás, a segunda metade do projeto é toda ela realizada de forma automática, após o programa ter acesso ao *dataset* construído na primeira parte.

<sup>4</sup>[https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html#projectpoints](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#projectpoints)

<sup>5</sup>[https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html#drawchessboardcorners](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#drawchessboardcorners)

Tendo calculados os valores de todos os parâmetros de entrada da função objetivo, parâmetros intrínsecos das câmaras (disponíveis no ficheiro JSON do *dataset*) e as posições e orientações de todos os elementos do sistema (câmaras e padrões de calibração) relativamente ao referencial global, será seguidamente explicitado o processo realizado por esta mesma função.

### 3.2.3 Função Objetivo

Durante um processo de otimização são realizadas várias iterações tendo como objetivo a aproximação dos parâmetros aos valores reais. Em cada uma destas iterações é chamada a função objetivo. Em cada vez que isto acontece, são modificados os parâmetros de entrada e esta função retorna novo vetor de residuais (vetor com os erros calculados). Este vetor de saída da função permite uma avaliação dos parâmetros de entrada, levando a que haja uma mudança positiva (diminuição do erro, neste caso). Assim, o processo otimiza os valores calculados anteriormente (*first guess*) de forma a obter os valores que retornam o menor valor de erro possível para cada residual.

Com este objetivo em mente, a função de custo percorre todas as coleções válidas e todas as imagens de cada coleção (conjunto composto por uma imagem de cada sensor, captada no mesmo instante temporal) presentes no *dataset* proveniente da primeira metade do trabalho. Para cada imagem é verificado se nesta é detetado o padrão de xadrez. Em caso positivo o programa percorre o restante código dentro da função objetivo, com vista a calcular os residuais associados a esta imagem; em caso negativo, o programa interrompe o processo e passa para o teste da imagem seguinte e assim sucessivamente. Terminando a coleção em estudo, automaticamente a função de custo inicia o estudo da coleção seguinte. Na Figura 3.24 pode-se encontrar um esquema deste processo realizado.

Como auxílio à função objetivo, é utilizada uma classe, *Calibration*, que armazena os vetores de posição e orientação, de cada câmara e de cada padrão de calibração, bem como os parâmetros intrínsecos de cada sensor e a lista das coleções consideradas válidas. Tal como descrito, a classe utilizada na Subsecção 3.2.2 retorna as matrizes de transformação relativas ao referencial global. Destas matrizes são retirados os vetores de posição e orientação desejados: sendo o primeiro extraído de forma direta desta matriz (primeiros valores da última coluna) e o segundo que passa pela transformação da matriz de rotações (com o auxílio da função Rodrigues). Esta última classe serve como apoio para a segunda, visto que, ao terminar o procedimento do *first guess* é transmitida informação para a nova classe que irá ser utilizada no processo de otimização. Na Figura 3.25 está representado o fluxo de informação entre as classes utilizadas neste projeto.

No caso de o xadrez ser detetado numa imagem, a função de custo começa, inicialmente, por recolher os parâmetros de entrada que se encontram armazenados na classe *Calibration*, relativos ao sensor em questão e ao padrão de calibração da coleção em estudo. Os vetores de posição e orientação, são logo de seguida, convertidos para matrizes de transformação (com dimensões 4x4). Isto permite, tal como visto anteriormente, que existam posições e orientações não apenas relativas ao referencial global, ao aplicar a multiplicação de matrizes.

De forma a tratar os parâmetros de entrada para retornar um vetor de residuais, a função objetivo tem como principal propósito a projecção dos pontos do xadrez em cada imagem onde o xadrez é detetado. Isto permite, tal como referido no início deste capítulo, medir a distância

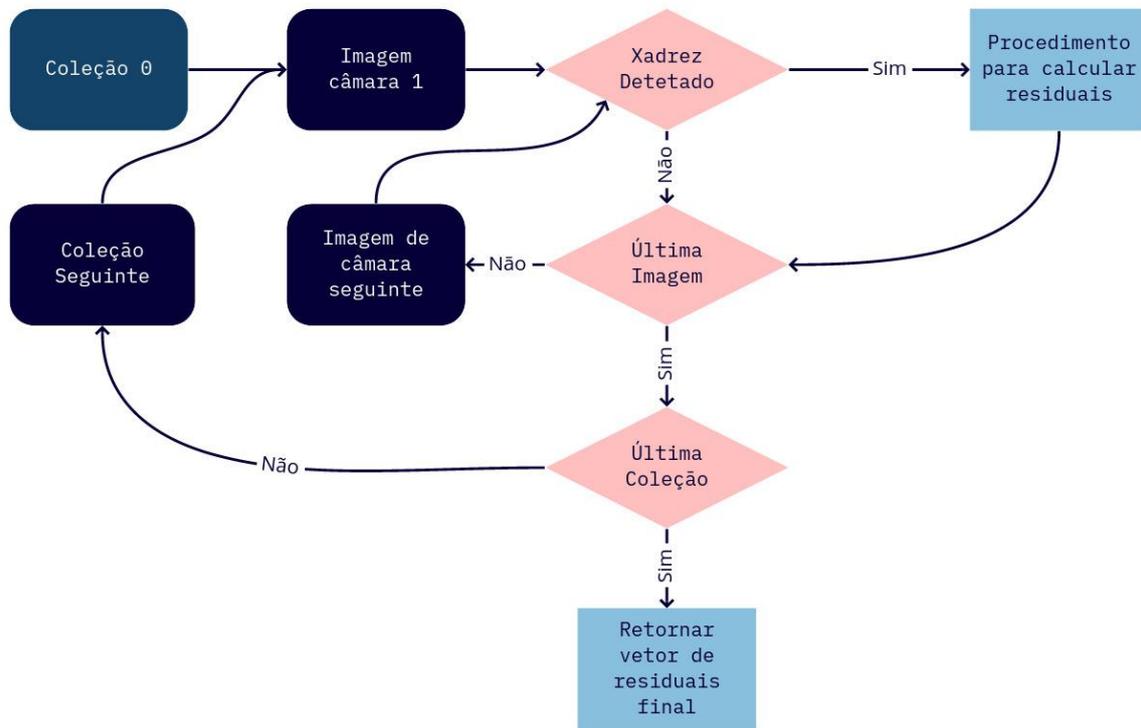


Figura 3.24: Nesta imagem está representado o que acontece na função objetivo de forma a seleccionar quais as imagens a serem utilizadas para calcular residuais. A função percorre todas as imagens de todas as coleções (começando na "0", retângulo inicial do fluxo), considerando apenas as que detetam o xadrez como válidas para o cálculo de residuais.

de cada um destes até ao respetivo ponto detetado na imagem, considerado real, sendo esta distância o erro associado a cada ponto.

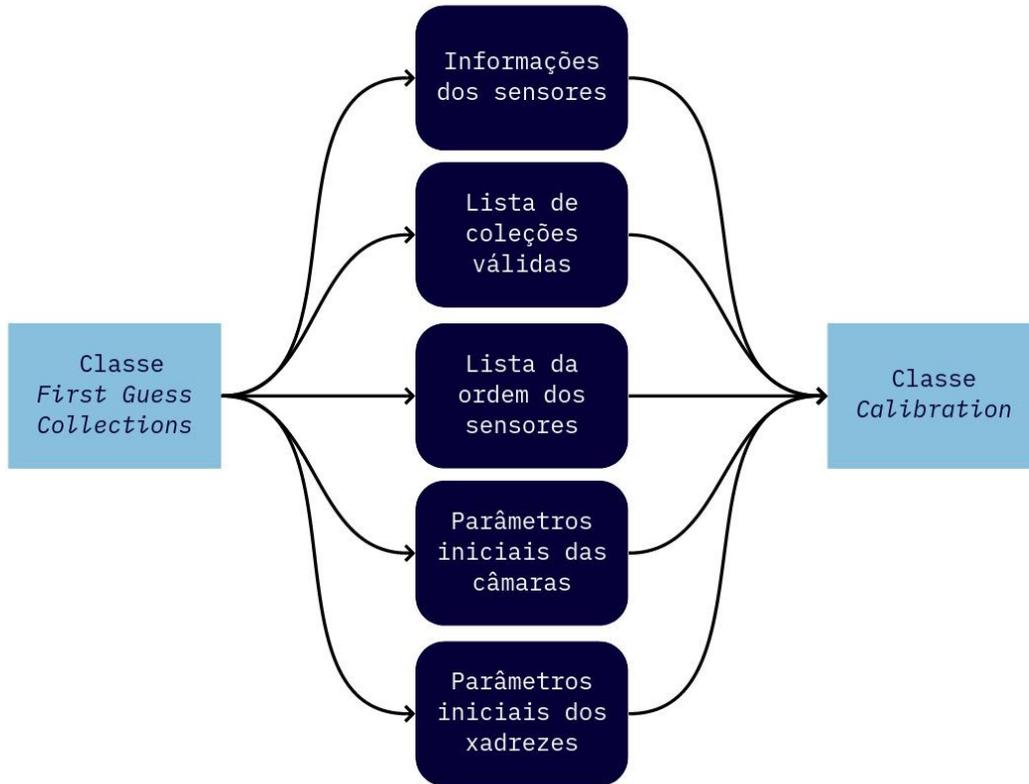


Figura 3.25: Nesta imagem está representado o fluxo de informação entre a classe do *first guess* e a classe de apoio à função objetivo, *Calibration*

### Projeção dos pontos na imagem

De forma a projetar os pontos do xadrez na imagem é necessário que estes possuam, em primeiro lugar, coordenadas no referencial da câmara respetiva. Para tal, é necessário estimar a transformação geométrica do referencial do sensor para o referencial do xadrez. Tal como se pode ver na Figura 3.22, apresentada anteriormente, esta transformação pode ser obtida através da transformação geométrica da câmara para o referencial global e da inversa da transformação do xadrez para este mesmo referencial (ou seja, da transformação do referencial global para o padrão de calibração), Equação 3.18.

$$\text{Câmara} \mathbf{T}_{\text{Xadrez}} = \overbrace{\text{Câmara} \mathbf{T}_{\text{Referencial Global}}^{\text{Parâmetros de entrada}}} \cdot \overbrace{\text{Xadrez} \mathbf{T}_{\text{Referencial Global}}^{-1}^{\text{Obtido através dos parâmetros de entrada}}} \quad (3.18)$$

Com base na Figura 3.26, onde se encontra o exemplo num mundo 2D, caso se possua uma transformação geométrica de um referencial W para um L (apenas translação de 3 em  $x$  e de 1 em  $y$ ), qualquer ponto com coordenadas neste último referencial pode ser visto pelo primeiro. Quer isto dizer que se pode descrever o ponto em relação ao primeiro referencial. Para obter então as coordenadas do ponto no referencial W basta apenas multiplicar a transformação deste para o referencial L pelas primeiras coordenadas do ponto. Assim, ao multiplicar a matriz de

transformação dos referenciais pelo ponto P, que no referencial L possui coordenadas (1, 0), obtém-se as coordenadas no referencial W (4, 1).

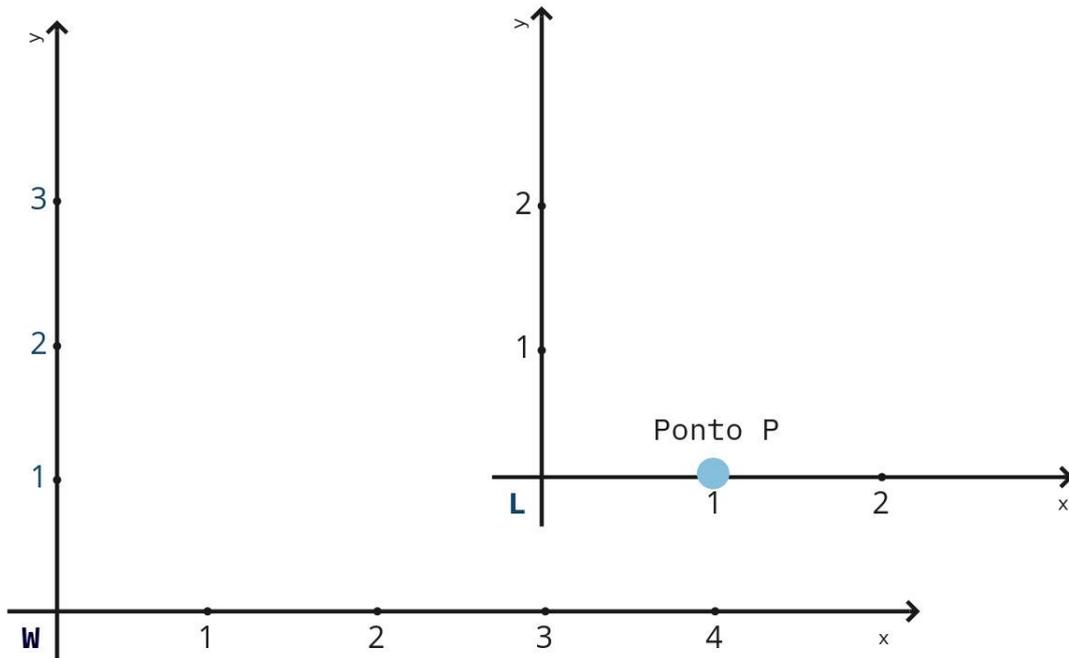


Figura 3.26: Nesta imagem estão representados dois referenciais, *World* (W) e *Local* (L), com posição relativa conhecida (translação de 3 em  $x$  e 1 em  $y$ ). Encontra-se um ponto P a azul, no referencial L, com coordenadas (1,0) que pode também ser visto do referencial W, ao aplicar a transformação geométrica de um referencial para o outro.

Deste modo, para determinar os pontos do xadrez no referencial da câmara, basta aplicar a Equação 3.19.

$$\mathbf{P}_{\text{Câmara}} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \text{Câmara} \mathbf{T}_{\text{Xadrez}} \cdot \mathbf{P}_{\text{Xadrez}} \quad (3.19)$$

Tal como descrito na Subsecção 3.1.2, para projetar pontos do mundo 3D para píxeis na imagem, é necessário ter conhecimento da matriz intrínseca e dos coeficientes de distorção. O processo é adaptado partindo do OpenCV<sup>6</sup> e terá como base os anteriores parâmetros.

Visto que o objetivo final passa por ter apenas duas coordenadas, relativas aos píxeis ( $u$  e  $v$ ), o primeiro passo é a homogeneização da coordenada  $z$ , dividindo todas as coordenadas por esta mesma, Equação 3.20, Equação 3.21 e Equação 3.22, passando então a ter novos valores

<sup>6</sup>[https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html#camera-calibration-and-3d-reconstruction](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#camera-calibration-and-3d-reconstruction)

para  $x$  e  $y$ , resultando num ponto com coordenadas 2D apenas.

$$x' = \frac{x}{z} \quad (3.20)$$

$$y' = \frac{y}{z} \quad (3.21)$$

$$1 = \frac{z}{z} \quad (3.22)$$

Tal como explicado anteriormente, as câmaras têm associadas a si mesmas uma distorção da imagem retilínea, podendo ser mais ou menos acentuada, consoante o modelo de câmara utilizado. Este fator acaba por influenciar diretamente a passagem para coordenadas 2D, sendo então aplicada uma retificação aos valores anteriormente calculados, Equação 3.24 e Equação 3.25. Nestas equações, entram os diferentes coeficientes de distorção,  $k_1$ ,  $k_2$ ,  $k_3$ ,  $k_4$ ,  $k_5$ ,  $k_6$  referentes à vertente radial e  $p_1$  e  $p_2$ , relativos à distorção tangencial.

De forma a simplificar ambas as equações de retificação, é utilizada uma variável  $r^2$ , sendo esta a soma dos quadrados de  $x'$  e  $y'$ , Equação 3.23.

$$r^2 = x'^2 + y'^2, \quad (3.23)$$

$$x'' = x' \left( \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} \right) + 2p_1 x' y' + p_2 (r^2 + 2x'^2), \quad (3.24)$$

$$y'' = y' \left( \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} \right) + p_1 (r^2 + 2y'^2) + 2p_2 x' y' \quad (3.25)$$

O número de coeficientes de distorção radial dependem da calibração intrínseca realizada, sendo pelo menos os dois primeiros necessários. Assim, neste projeto a calibração é feita tendo em conta os três primeiros radiais,  $k_1$ ,  $k_2$  e  $k_3$  e os dois tangenciais,  $p_1$  e  $p_2$ . A Equação 3.24 e a Equação 3.25 podem então ser simplificadas para a Equação 3.26 e Equação 3.27, respetivamente.

$$x'' = x' (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x' y' + p_2 (r^2 + 2x'^2) \quad (3.26)$$

$$y'' = y' (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1 (r^2 + 2y'^2) + 2p_2 x' y' \quad (3.27)$$

Com as coordenadas no plano 2D determinadas e retificadas, é então possível determinar a sua posição na imagem retirada da câmara. Para determinar  $u$  e  $v$ , é necessário multiplicar as coordenadas 2D pela respetiva distância focal. Estes valores correspondem à variação, em píxeis, relativamente ao centro da imagem. Visto que as coordenadas em píxeis têm a sua origem, píxel (0,0) numa extremidade, é necessário somar o valor da posição do centro à variação calculada para obter as coordenadas em píxeis do ponto desejado, Equação 3.28 e Equação 3.29.

$$u = f_x \cdot x'' + c_x, \quad (3.28)$$

$$v = f_y \cdot y'' + c_y \cdot \quad (3.29)$$

Computacionalmente, poderia ser exigente a realização deste método para cada um dos 54 pontos do xadrez por duas vezes e para cada iteração da otimização. Assim, é utilizada a função *projectPoints*<sup>7</sup>, do OpenCV. Esta transforma uma matriz de pontos 3D em pontos na imagem, tendo em consideração a distorção da câmara em caso.

### Cálculo dos residuais

Para o cálculo do erro associado à projeção realizada anteriormente, será necessário conhecer a posição real na imagem dos cantos do xadrez. Esta posição é detetada diretamente na imagem pela função *findChessboardCorners*, que retorna uma matriz com as coordenadas em píxeis de todos os pontos de interesse do padrão de calibração.

Os residuais serão então as distâncias absolutas em píxeis entre os pontos projetados e os detetados (considerados reais, *ground truth*) - a Figura 3.6 representa a natureza dos dois tipos de pontos de interesse na imagem. Matematicamente, esta distância é calculada pela norma do vetor que liga os dois pontos (projetado e *ground truth*), Equação 3.30, onde os valores *u* e *v* representam as suas coordenadas em píxeis.

$$\text{Distância em píxeis} = \sqrt{(\mathbf{u}_{groundtruth} - \mathbf{u}_{projetado})^2 + (\mathbf{v}_{groundtruth} - \mathbf{v}_{projetado})^2} \quad (3.30)$$

Na Figura 3.27, é possível encontrar um exemplo real, de uma das imagens em que o plano de calibração é detetado, das diferentes representações dos pontos do xadrez: na primeira estes são detetados na imagem (onde o desenho destes pontos é quase perfeito) e na segunda projetados através das matrizes de transformação que vão sendo estimadas (onde o desenho dos pontos se encontra claramente desfazado da realidade).



Figura 3.27: Projeção do xadrez na posição considerada verdadeira (a) e projeção do xadrez com posição variada ao longo da otimização (b)

<sup>7</sup>[https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html#projectpoints](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#projectpoints)

A distância, entre um ponto projetado e um detetado, representa o cálculo de cada um dos residuais (erros) associados a uma única imagem. Desta forma, para cada imagem, são calculadas os erros de todos os pontos do xadrez. Assim, o número de residuais associados a cada imagem é igual ao número total de cantos do xadrez. Na Equação 3.31 podemos ver o vetor de residuais de uma imagem, de uma câmara  $x$ , da coleção  $y$ .

$$\mathbf{r}_{\text{Câmara } x, \text{ Coleção } y} = \{\mathbf{e}_{\text{Ponto } 1}, \mathbf{e}_{\text{Ponto } 2}, \dots, \mathbf{e}_{\text{Ponto } N}\} \quad (3.31)$$

A função objetivo, assim que calcula os residuais de uma imagem, segue para a seguinte dentro da mesma coleção. Tal como visto no início deste subcapítulo, apenas são calculados os residuais das imagens onde o xadrez é detetado, sendo que as restantes são "descartadas" (não é executado o código dentro da função de custo). Assim, os residuais associados a cada coleção contém os vetores de residuais das câmaras, cujas imagens continham o xadrez na sua totalidade. Este vetor encontra-se na Equação 3.32.

$$\mathbf{r}_{\text{Coleção } y} = \{\mathbf{r}_{\text{Câmara } x}, \mathbf{r}_{\text{Câmara } y}, \mathbf{r}_{\text{Câmara } z}\} \quad (3.32)$$

Por fim, o vetor de residuais retornado pela função objetivo corresponde à junção dos vetores de residuais de todas as coleções presentes no *dataset*, Equação 3.33.

$$\mathbf{r} = \{\mathbf{r}_{\text{Coleção } 1}, \mathbf{r}_{\text{Coleção } 2}, \dots, \mathbf{r}_{\text{Coleção } N}\} \quad (3.33)$$

Quer isto dizer que a função objetivo retorna todos os residuais de todas as imagens onde o xadrez é detetado ao longo de todas as coleções. Como exemplo, numa situação em que existam 5 câmaras, 10 coleções, onde o xadrez seja identificado em 2 imagens por cada coleção, e o xadrez for de dimensões 9x6 (ou seja, 54 pontos), o vetor dos residuais irá conter 54 x 2 x 10 valores, ou seja, 1080 residuais.

### 3.2.4 Otimizador

Conhecendo todos os fatores relacionados com a calibração extrínseca, falta agora perceber um pouco melhor como corre o processo de otimização.

Este procedimento tem como base um otimizador. É este que recebe o vetor de residuais e consoante os erros faz mudar os parâmetros que serão entregues à função objetivo.

De forma a realizar o processo de otimização, optou-se pela utilização de um otimizador desenvolvido em [22]. Este consiste numa classe, *Optimizer*, com várias funções/métodos que facilitam todo o processo realizado. Este otimizador tem como base o otimizador *least\_squares*<sup>8</sup> da biblioteca *scipy*.

Com o objetivo de uma melhor percepção desta classe, seguidamente irão ser dadas a conhecer os seus métodos:

---

<sup>8</sup>[https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least\\_squares.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html)

### **Método *Optimizer.addDataModel***

Esta, como primeira função da classe, apenas tem como objetivo ter o conhecimento do modelo que contém todos os dados necessários (parâmetros a otimizar e alguma informação extra relevante). Neste projeto é utilizada a classe *Calibration*, já descrita anteriormente.

### **Método *Optimizer.pushParamScalar***

Este método tem como objetivo dar a conhecer ao sistema todos os parâmetros que serão otimizados. Estes parâmetros serão retirados da classe, numa primeira instância, e posteriormente atualizados. Deste modo, a função objetivo ao aceder à classe *Calibration* estará sempre a encontrar os últimos valores atribuídos pelo otimizador.

### **Método *Optimizer.setObjectiveFunction***

Este método apenas existe para tomar conhecimento de qual das funções do programa será a função objetivo da otimização. Esta será a função que, ao receber os parâmetros modificados pelo otimizador, irá retornar um vetor de erros.

### **Método *Optimizer.pushResidual***

Este método cria uma lista de residuais. Esta lista é criada antes do início da otimização, visto já ser possível ter conhecimento do número de imagens onde o xadrez é detetado (através do *dataset*). Deste modo, o otimizador já saberá a dimensão do vetor que espera receber, bem como o significado de cada um dos valores.

### **Método *Optimizer.setVisualization***

Este método permite ter uma visualização do que está a acontecer durante a otimização. Tem como base dois gráficos: um gráfico em que se observa a evolução do erro total e outro onde se observa o valor absoluto de cada residual a ser alterado, podendo comparar com o valor que possuía inicialmente.

### **Método *Optimizer.startOptimization***

Esta função dá início a todo o processo. Para tal é necessário que tenham sido corretamente executados todos os anteriores métodos.

## Capítulo 4

# Resultados

Neste capítulo, serão apresentados diversos resultados retirados ao longo de toda a realização desta dissertação.

Numa fase inicial, é importante explorar resultados de todo o processo que levou até método final de calibração. Isto permite uma melhor percepção do trajeto percorrido, bem como das decisões tomadas que levaram ao resultado final.

Posteriormente, será feita uma análise dos resultados obtidos em diferentes testes, utilizando já a solução final apresentada no Capítulo 3. A precisão deste processo de calibração será verificada em três distintas situações. Primeiro, serão realizadas diferentes calibrações variando o número de coleções que serão utilizadas para otimizar os parâmetros pretendidos; segundo, são efetuadas distintas calibrações onde as restrições que levam a classificação de uma coleção como válida são variadas; por fim, é feito um teste ao impacto que os valores atribuídos aos parâmetros iniciais têm no resultado final.

### 4.1 Calibração de sistema com duas câmaras

De forma a realizar uma calibração de um anel de câmaras, objetivo enunciado no capítulo introdutório, foi inicialmente aplicado o problema a um sistema stereo, ou seja, um sistema com apenas duas câmaras, Figura 4.1.

Esta calibração, tem a finalidade de estimar a posição e orientação de um sensor em relação a outro. Para tal, é utilizado um xadrez de apoio à calibração, de forma a que exista informação em comum em ambas as imagens retiradas pelas câmaras. Comparando os pontos do xadrez é possível, através de um processo de otimização, ir ajustando a transformação geométrica (posição e orientação) de uma câmara em relação à outra. Este processo visa, ao modificar os valores dos parâmetros a otimizar, diminuir o erro para os valores considerados reais. Ou seja, a distância entre a projeção do xadrez e a sua suposta posição real.

Neste exemplo utilizando apenas duas câmaras, será estudada a calibração utilizando apenas uma coleção (par de imagens, uma de cada câmara), variando a coleção utilizada até 10 calibrações distintas e por fim utilizando mais coleções para a mesma calibração. A utilização de apenas uma coleção é possível, apesar de não ser o ideal, devido ao facto de serem sempre coleções completas para o caso em estudo (coleções onde todas as imagens detetam o padrão de calibração).

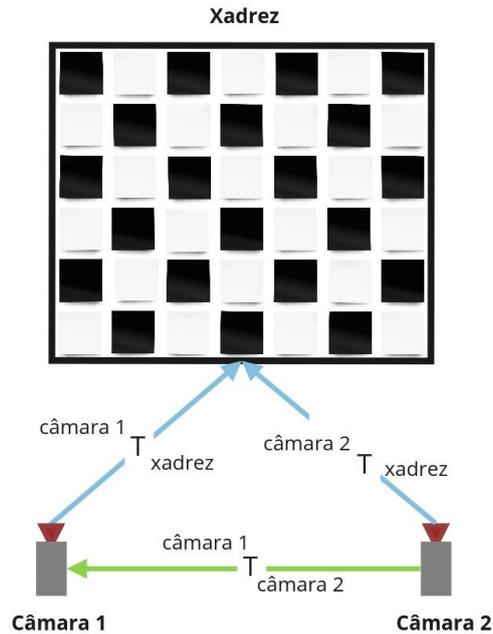


Figura 4.1: Esquema do sistema stereo

#### 4.1.1 Parâmetros a otimizar

Com o objetivo de simplificar a calibração, apenas se pretende otimizar a transformação da câmara 2 para a câmara 1, Equação 4.1. Quer isto dizer que, tanto a câmara 1 como o xadrez possuem posições e orientações fixas, variando apenas a câmara do lado direito do esquema anteriormente apresentado na Figura 4.1.

A transformação é uma matriz composta por uma matriz que contém a informação relativa à rotação e translação entre referenciais, neste caso, entre as câmaras (Equação 4.1).

$$\text{Câmara } 2 \mathbf{T}_{\text{Câmara } 1} = [R|t]. \quad (4.1)$$

A matriz  $R$ , das rotações ( $3 \times 3$ ), corresponde às várias rotações, em torno dos três eixos (coordenadas cartesianas), aplicadas ao referencial da câmara 2 para que este fique com a orientação do referencial da primeira câmara. Já o vetor  $t$ , corresponde às translações, em  $x$ ,  $y$  e  $z$ , para que os referenciais fiquem com a mesma posição.

O vetor de rotações (notação Rodrigues), com as dimensões já descritas (três valores), mais o vetor das translações (três valores), correspondem aos parâmetros a ser otimizados.

Tal como referido anteriormente, este processo de calibração tem como base um plano, com um padrão de xadrez, no campo de visão de ambos os sensores, Figura 4.2. Este padrão é facilmente detetável nas imagens retiradas pelas câmaras, existindo uma função do OpenCV própria para o efeito, *findChessboardCorners*<sup>1</sup>.

<sup>1</sup>[https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html#findchessboardcorners](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#findchessboardcorners)

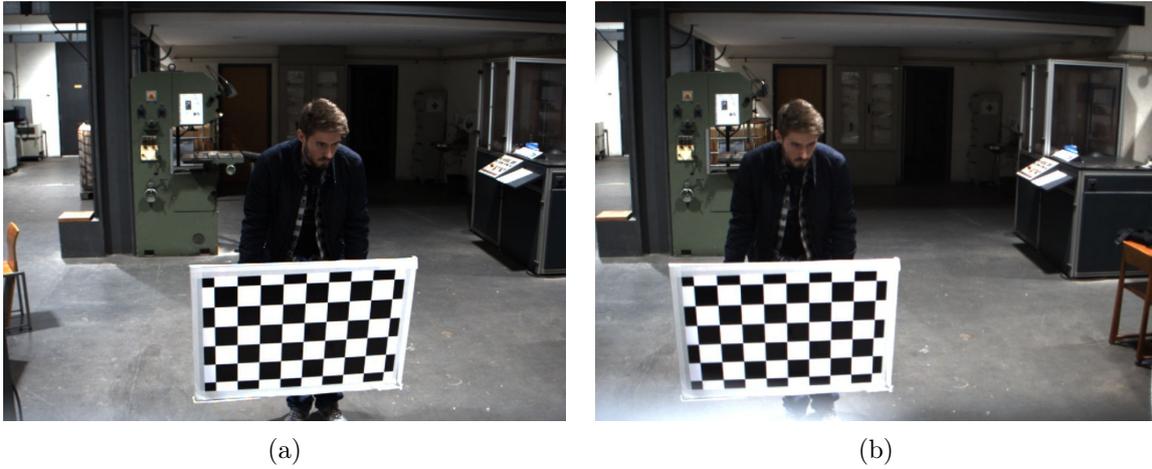


Figura 4.2: Imagens retiradas a partir da câmara do lado esquerdo (câmara 1) (a) e do lado direito (câmara 2) (b) do ATLASCAR2

Através da deteção dos pontos do xadrez, Figura 4.3, é possível determinar a transformação geométrica do referencial de cada um dos sensores até ao referencial do plano de auxílio à calibração. Com esta finalidade, existem já funções disponíveis, tendo sido utilizada novamente uma função do OpenCV, *Solve PNP*<sup>2</sup>, para o efeito.

Para o problema em questão, calibração de duas câmaras, utilizando um padrão como auxílio, é possível na Equação 4.2 perceber a definição do mesmo. Uma transformação é uma variação de posição e orientação de um referencial para outro. Deste modo, a forma como se calcula a transformação do referencial da câmara 2 para o referencial do xadrez possui duas hipóteses distintas. Na Figura 4.1 é possível perceber os dois trajetos possíveis, visto que se pode calcular de forma direta ou de forma indireta através da conjugação de transformações. Apesar de serem calculadas de maneira diferente, o resultado final deveria ser o mesmo, ou seja, a transformação do sensor 2 diretamente para o xadrez seria igual à transformação de um sensor para o outro e, posteriormente, do primeiro sensor para o padrão, caso todas as transformações tivessem o valor verdadeiro. Isto, porém, é algo que não se consegue devido ao erro que existe sempre associado a uma calibração.

$$\underbrace{\text{Câmara 2}}_{\text{ground truth}} \mathbf{T}_{\text{Xadrez}} = \underbrace{\text{Câmara 2}}_{\text{Matriz a Otimizar}} \mathbf{T}_{\text{Câmara 1}} \cdot \underbrace{\text{Câmara 1}}_{\text{Matriz fixa}} \mathbf{T}_{\text{Xadrez}}. \quad (4.2)$$

Tal como referido no início desta secção, assume-se que tanto a câmara 1 como o xadrez possuem posições e orientações fixas ao longo de todo o processo de calibração. Assim, ao aplicar a função *Solve PNP* na imagem deste sensor, o valor retornado será sempre constante. Deste modo, na Equação 4.2, a transformação da câmara 1 para o xadrez assume um valor fixo. Por outro lado, ao aplicar a mesma função na imagem da câmara 2, é calculada a

<sup>2</sup>[https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html#solvepnp](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#solvepnp)

transformação desta mesma câmara para o xadrez de forma direta e independente do sistema em que esta se encontra, isto é, independente do número de sensores a serem calibrados. Este valor é considerado como o valor mais próximo da realidade, sendo então uma referência no cálculo do erro por ser tomado como verdadeiro, *ground truth*. Também esta matriz de transformação não sofre alterações ao longo de toda a calibração.

Por estas razões, apenas a transformação entre as câmaras varia ao longo de todo o processo de calibração. Deste modo, apenas os valores desta matriz são variados pela otimização que decorre dentro deste processo, como referido anteriormente, são os parâmetros a otimizar.



Figura 4.3: Exemplo da utilização da função de deteção do xadrez na imagem de uma câmara

#### 4.1.2 Comparação entre função de erro calculada através da distância euclidiana no mundo 3D e da distância entre píxeis

A partir de imagens recolhidas através de duas câmaras colocadas no ATLASCAR2, o objectivo é, com a maior precisão possível, determinar a posição e orientação de uma em relação à outra. É necessário que ambos os sensores detetem um xadrez de apoio à calibração, visto ser a informação em comum que permite executar a calibração.

É considerada a câmara colocada sobre o lado esquerdo da frente do veículo como o sensor ancorado, isto é, a posição dela não varia, e o xadrez fixo. Assim, reduz-se o problema a apenas

calcular a transformação geométrica de uma câmara para a outra (parâmetros a otimizar).

O processo de calibração, possui, como parte integrante, uma otimização não linear. Partindo de uma posição inicial, *first guess*, esta otimização faz variar determinados parâmetros, ao longo das várias iterações que compõem o processo, até atingir um valor próximo do ideal. De modo a fazer variar os parâmetros de forma objetiva, é necessário dar um *feedback* ao otimizador dos valores por ele atribuídos. Assim, a função de custo recebe os parâmetros e retorna o vetor dos residuais, Equação 4.3, sendo este composto por 54 valores, cada um relativo a cada um dos 54 pontos do padrão de xadrez utilizado como auxílio à calibração. A forma como estes valores são calculados pode variar.

$$\mathbf{Residuais} = [ \mathbf{Erro}_{\text{Ponto 1, Xadrez}} , \mathbf{Erro}_{\text{Ponto 2, Xadrez}} , \dots , \mathbf{Erro}_{\text{Ponto 54, Xadrez}} ] \quad (4.3)$$

Para o caso em estudo, realizou-se uma calibração entre duas câmaras utilizando dois diferentes métodos. Ambos possuem um procedimento bastante similar, apenas diferindo no processo do cálculo do vetor dos erros residuais.

A primeira metodologia realiza este processo com base na distância entre os pontos do xadrez projetados através das transformações calculadas e os pontos do xadrez detetados diretamente na imagem. Todo o processo do cálculo dos residuais desta metodologia encontra-se detalhado na Subsecção 3.2.3.

O segundo método utilizado calcula a distância euclidiana entre pontos no referencial da câmara 2. Ou seja, ao invés de projetar os pontos do mundo 3D para a imagem, é feita a transformação dos pontos detetados na imagem para o referencial da câmara em questão. Com todos os pontos no referencial da câmara, cada residual corresponde à distância euclidiana entre o ponto detetado na imagem e o projetado pelas transformações calculadas. Este processo será, seguidamente, explicado mais detalhadamente.

### Cálculo do erro no mundo 3D

A existência de dois "caminhos" que levam à transformação da câmara 2 para o xadrez permite realizar uma comparação entre os mesmos. O primeiro, através da função do OpenCV, onde é calculada diretamente a transformação deste sensor para o plano, é considerada como o *ground truth*. Ao multiplicar a transformação de um sensor para o outro, pela transformação do primeiro sensor para o xadrez, obtém-se, também, a transformação da câmara 2 para o xadrez, sendo este o segundo "caminho". Este último, é feito variar ao longo da calibração pelo processo de otimização, devido ao facto de ser ajustada a variação de posição e orientação entre os sensores.

Ao multiplicar um ponto visto por um referencial pela transformação de outro referencial para esse mesmo, é possível obter as coordenadas desse ponto no segundo referencial. Deste modo, é possível calcular as coordenadas de cada um dos pontos do xadrez no referencial da câmara 2. O padrão de xadrez é composto por cinquenta e quatro pontos, correspondentes a cada interseção dos vários quadrados que o compõem. Estes pontos encontram-se distribuídos numa disposição 9 por 6, sendo o eixo  $x$ , do referencial do plano, ao longo do maior lado, o

eixo  $y$  ao longo do menor e por fim eixo  $z$  que é perpendicular ao xadrez. Este último eixo, ao adquirir estas propriedades, leva a que todos os pontos do plano tenham esta coordenada nula, pelo facto de a origem do referencial se encontrar num destes pontos que esteja numa extremidade. Com este último aspeto em consideração, a determinação das coordenadas de cada esquina do xadrez torna-se um processo mais simples, visto que os eixos  $x$  e  $y$  aumentam ao longo do padrão. Quer isto dizer que, nenhum dos pontos,  $p_{xadrez}$ , possui uma coordenada negativa. Assim, começando no ponto inicial  $(0,0,0)$ , apenas se vai adicionando nos dois primeiros eixos a largura dos quadrados do xadrez para se determinar as coordenadas dos restantes pontos.

Visto que, existem duas maneiras distintas de estimar a transformação, e conhecendo já as coordenadas dos pontos no referencial do xadrez, existem também, para cada um desses pontos, duas representações diferentes no referencial do sensor em estudo. Na Equação 4.4 encontra-se o ponto que será considerado como verdadeiro,  $p_{groundtruth}$ , por ser calculado diretamente. Já na Equação 4.5, encontra-se o cálculo das coordenadas do ponto com posição variada,  $p_{otimizado}$ , ou seja, o que está a ser otimizado.

$$\mathbf{p}_{ground\ truth} = \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \\ \mathbf{p}_z \\ 1 \end{bmatrix} = \text{Câmara } 2 \mathbf{T}_{Xadrez} \cdot \mathbf{p}_{Xadrez} \quad (4.4)$$

$$\mathbf{p}_{otimizado} = \begin{bmatrix} \mathbf{q}_x \\ \mathbf{q}_y \\ \mathbf{q}_z \\ 1 \end{bmatrix} = \text{Câmara } 2 \mathbf{T}_{\text{Câmara } 1} \cdot \text{Câmara } 1 \mathbf{T}_{Xadrez} \cdot \mathbf{p}_{Xadrez} \quad (4.5)$$

Com a existência de dois pontos diferentes, é possível determinar a distância entre eles. Visto que, um deles é considerado verdadeiro, esta distância é considerada como o erro entre o ponto que está a ser otimizado e a posição "real" (não será exatamente a posição real pelo facto de a função utilizada poder estar susceptível a erros). Para cada um dos pontos do xadrez é calculado o erro, Equação 4.6, sendo este processo realizado em cada uma das iterações da otimização.

$$\mathbf{Erro}_{3D} = | \mathbf{p}_{ground\ truth} - \mathbf{p}_{otimizado} | \quad (4.6)$$

Visto serem pontos no mundo 3D, esta diferença entre os pontos (erro) é calculada através da distância euclidiana, tendo esta em conta as três coordenadas constituintes de cada um dos respetivos pontos.

Tendo em consideração a Equação 4.4 e a Equação 4.5, a distância é calculada através da Equação 4.7.

$$\mathbf{Distância\ Euclidiana} = \sqrt{(\mathbf{p}_x - \mathbf{q}_x)^2 + (\mathbf{p}_y - \mathbf{q}_y)^2 + (\mathbf{p}_z - \mathbf{q}_z)^2} \quad (4.7)$$

De notar que ao longo deste capítulo o resultado da função objetivo que calcula a distância euclidiana aparecerá sempre com as unidades em píxeis. Isto deve-se ao facto de, no final da otimização realizada com esta função, se calcular o erro em píxeis com as transformações

geométricas estimadas. Este procedimento permite comparar diretamente os dois métodos utilizados, bem como ter uma melhor percepção da precisão desta função de custo (uma vez que se tem como erro médio final aceitável o valor de 2 píxeis e não se tem o correspondente em distância no mundo 3D).

### **Comparação utilizando apenas um par de imagens**

Existindo dois diferentes modos de calcular o vetor de erros da função de custo, é relevante perceber qual deles será mais benéfico implementar na solução proposta nesta dissertação.

Como primeira forma de comparação entre os métodos apresentados anteriormente, erro calculado pela distância euclidiana, entre o ponto otimizado e o considerado verdadeiro, no mundo 3D da câmara n<sup>o</sup>2 e o segundo método, que calcula o erro pela distância entre os píxeis na imagem da mesma câmara, é apenas utilizado um par de imagens. Este par de imagens corresponde a um imagem retirada de cada uma das câmaras.

Com isto, pretende-se avaliar o comportamento de cada um num contexto simples, onde é considerado para o problema que tanto a câmara n<sup>o</sup>1 e o xadrez possuem uma posição fixa. Deste modo, a calibração entre as duas câmaras torna-se o mais simples possível, visto que apenas se move uma câmara, com o resto do cenário estático.

Na Figura 4.4 é então feita a comparação destes dois métodos ao longo das respectivas iteração do processo de otimização. É analisado o erro médio dos 54 pontos constituintes do xadrez para cada uma das iterações. De forma a permitir que esta seja feita, é feita a conversão do erro relativo ao mundo 3D para píxeis.

Pelo gráfico apresentado, consegue-se perceber uma diferença entre as duas metodologias. As duas principais diferenças entre ambas residem no número de iterações. No número que demoram a atingir um erro médio baixo ou até mesmo próximo de zero e no número total de iterações do processo.

A distância euclidiana, para este exemplo, rapidamente converge para valores perto do real, com uma grande diminuição do erro médio. A distância medida em píxeis na imagem acaba por atingir esses valores mas já com um elevado número de iterações realizadas, fazendo decrescer o valor do erro de forma mais suave, contrariamente ao método oposto. O primeiro método, pelo que se retira do gráfico, necessita de cerca de metade das iterações para terminar em relação ao segundo método. Isto pode levar a um menor esforço computacional, caso sejam utilizadas mais coleções para a calibração.

Com este primeiro teste, é importante perceber o número total de iterações realizadas e o erro médio final de cada um dos métodos. Um número de iterações elevado leva a um maior esforço computacional e a um maior tempo total da calibração. Assim, é de relevo tentar diminuir ao máximo este valor, mas sem fazer aumentar o erro médio final da otimização. Este último permite avaliar a precisão de todo o processo, comparando o valor final com o valor considerado real. Por isso mesmo, é pretendido que seja o menor possível, pois significaria que a otimização se teria aproximado da realidade. É considerada uma boa calibração quando o valor médio final não ultrapassa dois píxeis.

Assim, pela Tabela 4.1, vemos que ambos os métodos possuem valores de erro final bastante positivos, nenhum ultrapassa 0.2 píxeis, sendo que seria aceitável até os 2 píxeis. Porém, o

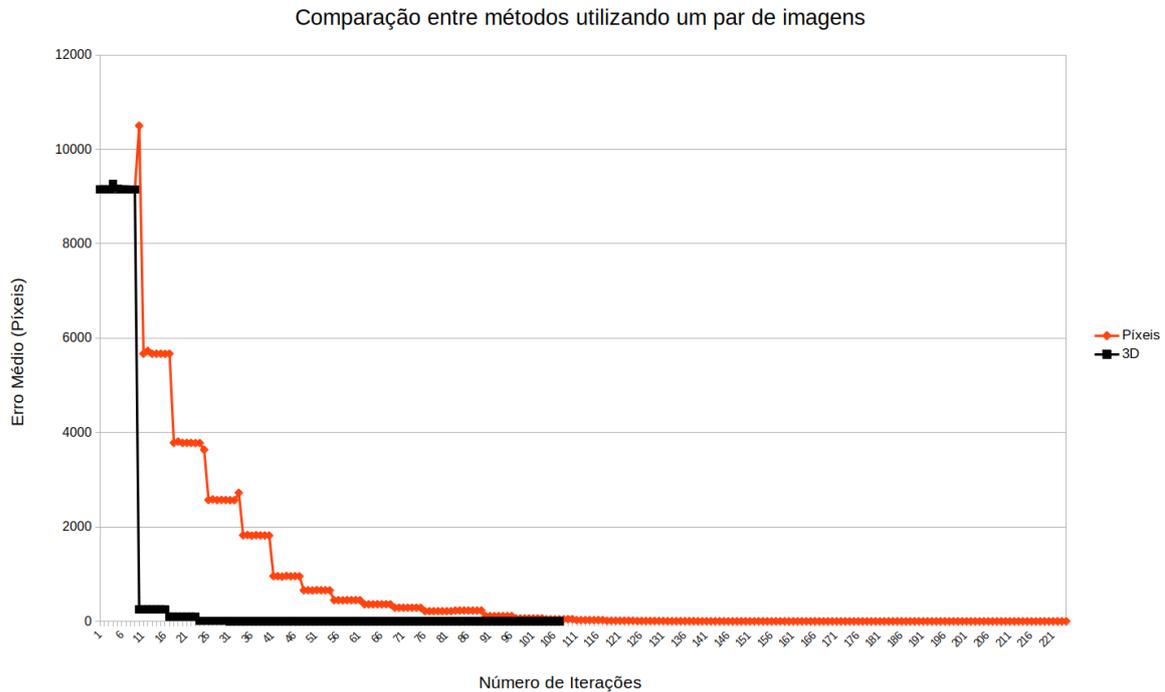


Figura 4.4: Comparação entre os dois métodos utilizando uma calibração com apenas um par de imagens

método do mundo 3D da câmara com menos de metade das iterações, atinge um valor de erro médio ainda mais baixo que a segunda metodologia (107 para 224 iterações).

Tabela 4.1: Erro médio final e número total de iterações de cada um dos métodos utilizados

Metodologia	Erro Médio Final (píxeis)	Número Total de Iterações
Distância Mundo 3D	0.076	107
Distância na Imagem	0.199	224

Neste caso em estudo, calibração de duas câmaras, considerando uma delas e o plano com padrão de xadrez numa posição fixa, ambos os métodos testados atingem valores bastante positivos. Ao calcular a distância entre os pontos no referencial 3D da câmara obtém-se um valor do erro menor, com menos iterações, podendo, por essa razão, existir uma diferença a nível de exigência computacional, mas nada de significativo para já, sendo preciso mais testes que o comprovem.

É notório o facto de ambas as metodologias, partindo de uma posição longe da realidade, erro médio inicial muito elevado, conseguem convergir para valores bastante próximos. A primeira tentativa, ou os valores iniciais dos parâmetros a otimizar, podem ter influência, positiva ou negativa, na precisão da otimização. Ao iniciar a otimização não linear com os valores muito distantes da realidade, pode acontecer que esta acabe num valor mínimo local e não no valor mínimo global, levando a um erro na calibração. Na Figura 4.5 é possível ver que, para ambos os casos, isso não aconteceu, estando ainda numa posição praticamente correta,

onde não é visível grande desvio em relação à realidade.



Figura 4.5: Projeção final do xadrez a ser otimizado pela distância no mundo 3D (a) e pela distância em píxeis na imagem (b)

#### Comparação ao longo de 10 calibrações com pares de imagens diferentes

De forma a verificar os resultados apresentados anteriormente, fez-se o mesmo processo dez vezes distintas. Quer isto dizer que, foram realizadas dez calibrações, cada uma delas com o seu par de imagens específico. Com o plano de xadrez em diferentes posições e orientações, é possível que os valores obtidos no erro médio final e no número de iterações realizadas, variem também. Na Figura 4.6 estão duas imagens da câmara 2, utilizadas em duas das dez calibrações, onde é visível a variação do plano de calibração.



Figura 4.6: Duas das imagens retiradas da câmara do lado direito, utilizadas em duas das dez calibrações

Na Figura 4.7 comprova-se que, realmente, a posição do xadrez acaba por ter alguma influência na precisão da calibração. Isto porque, existe uma variação do erro médio final ao

longo dos dez pares de imagens utilizados. Quando o erro é calculado pela distância em coordenadas 3D, esta diferença não é tão acentuada, existindo pouca divergência entre a calibração com menor e a com maior erro médio final. Porém, no caso da distância medida nas coordenadas em píxeis na imagem, já existe uma discrepância maior, atingindo mesmo uma diferença superior a 0,5 píxeis. Isto pode levar a crer que o primeiro método é mais flexível, não estando tão dependente da posição do padrão de xadrez, sendo, portanto, mais abrangente.

Adicionalmente, ainda é possível comprovar a tendência referida no anterior tópico, onde a metodologia no referencial 3D da câmara nº2 apresentava valores de erro médio inferiores. Para as dez calibrações, isto ocorre em grande parte, existindo apenas três exceções onde sucede o contrário.

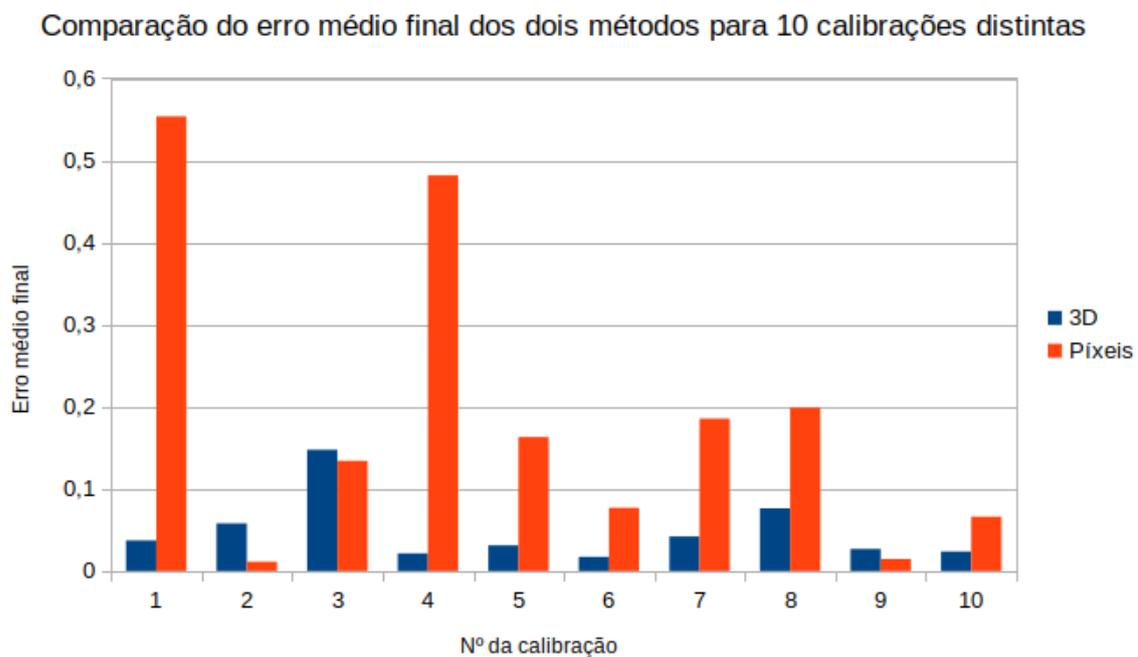


Figura 4.7: Comparação do erro médio final entre os dois métodos utilizando dez calibrações com imagens distintas

Para além do erro médio final, é importante também olhar para o número necessário de iterações para cada uma das calibrações. Assim, na Figura 4.8 encontra-se a comparação dos dois métodos ao longo dos dez processos realizados. Contrariamente ao que acontece no erro médio final, aqui em nenhum dos casos o método da distância em píxeis supera a primeira metodologia, ou seja, requer sempre um maior número de iterações para realizar a otimização. Em alguns casos, a diferença entre estes é bem visível, chegando a atingir quase 300 iterações de diferença em casos extremos. Nestas situações, existe já um esforço computacional adicional, bem como um aumento no tempo total do processo de calibração.

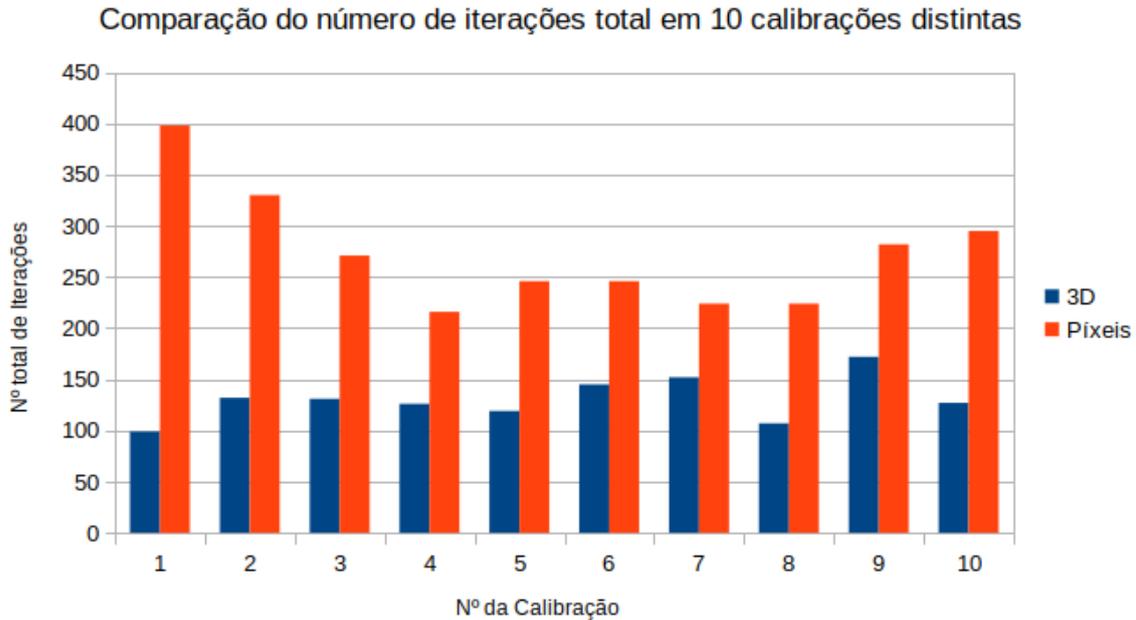


Figura 4.8: Comparação do número total de iterações entre os dois métodos utilizando dez calibrações com imagens distintas

Partindo das dez calibrações realizadas, foi feita a média do valor final do erro e do número total de iterações executadas e extraído o valor máximo de cada um destes dois tópicos, sendo possível observar tudo isto na Tabela 4.2.

Olhando então para a tabela, retira-se que a média do erro médio final das dez otimizações é extremamente baixo para ambas as metodologias. Isto porque, o valor seria aceitável sendo inferior a 2 píxeis e nenhum os valores máximos supera este patamar.

Ao nível do número total de iterações, a média de ambos também não apresenta motivos de preocupação sendo valores aceitáveis, porém a metodologia 3D possui um valor inferior à metade do outro método, podendo ser um ponto vantajoso. Já em relação ao valor máximo de iterações realizadas, medindo a distância em 2D na imagem, obtém-se um valor ligeiramente elevado.

Tabela 4.2: Média do erro médio final e número total de iterações e valor máximo de ambos de cada um dos métodos utilizados

Metodologia	Média do Erro Médio Final (píxeis)	Média do Número Total de Iterações	Valor Máximo de Erro Final (píxeis)	Valor máximo de iterações realizadas
Distância Mundo 3D	0.048	131.0	0.148	172
Distância na Imagem	0.189	273.2	0.554	398

Pelos testes expostos até ao momento, é possível verificar que ambos os métodos apresentam uma elevada precisão para o caso em estudo. A nível de esforço computacional, a segunda metodologia apresenta valores aceitáveis. Porém, neste processo apenas é exigida a calibração de duas câmaras, estando uma delas e o xadrez fixos, e apenas com uma coleção (par de imagens, uma de cada sensor). Ao tornar o processo mais complexo, por exemplo, a utilização de mais coleções para cada calibração, pode tornar-se um problema mais sério. Isto porque pode necessitar de um elevado número de iterações para convergir para um valor de erro dentro do limite aceitável.

### Comparação utilizando 8 coleções dentro da mesma calibração

Uma calibração de duas câmaras, requer que seja extraída informação de cada uma delas. Essa informação vem no formato de imagem. Assim, para que se realize uma calibração é necessária pelo menos uma imagem de cada sensor no mesmo instante. Estas formam um conjunto de imagens, sendo este chamado de coleção.

Ao utilizar apenas uma coleção na realização de uma calibração, leva a que o processo de otimização se torne mais simples, terminando com um erro bastante baixo. Porém, calibrando com um diferente par, o erro mantém-se reduzido, mas o resultado não será igual.

De forma melhorar o resultado da calibração, ou, pelo menos, torná-la não susceptível à qualidade de apenas uma coleção, são utilizadas mais coleções dentro do processo de calibração. Deste modo, o vetor dos residuais aumenta para  $54 \times n$ , sendo o primeiro valor correspondente aos pontos do plano de xadrez e  $n$  relativo ao número de coleções utilizadas. Aumentando então o vetor dos erros, aumenta a complexidade da otimização, podendo levar também a um aumento do erro final da otimização, bem como o número de iterações realizadas, comparando com apenas uma única coleção.

De forma a testar isto, ao exemplo dos anteriores tópicos, foram adicionadas sete coleções, fazendo um total de oito, às duas calibrações realizadas, onde o erro é calculado no mundo 3D e outra onde o erro é calculado na imagem.

Foi mantido o mesmo esquema do sistema, existindo duas câmaras e tendo como auxílio à calibração um plano com padrão de xadrez. Um dos sensores é considerado fixo, bem como o xadrez. Com isto, apenas uma transformação será otimizada, logo apenas 54 pontos para cada coleção, referentes ao padrão, serão utilizados para calcular o erro. O vetor dos residuais possuirá  $54 \times 8$  valores, ou seja, 432 valores.

Na Figura 4.9 encontra-se a variação do valor dos residuais, desde o início, *first guess*, até ao final da otimização com a metodologia da distância na imagem. Sendo também possível observar oito grupos de pontos, mais facilmente na secção a verde, a inicial, correspondentes às oito coleções utilizadas na calibração.

A diferença entre os dois métodos reside na função objetivo, ou a função de custo da otimização. Porém, o otimizador utilizado é o mesmo. Ou seja, apenas os valores retornados no vetor dos residuais serão diferentes. Para resolver a otimização não linear presente no problema em questão, é utilizado um otimizador da biblioteca do *scipy*, o *least\_squares*<sup>3</sup>.

---

<sup>3</sup>[https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least\\_squares.html#scipy-optimize-least-squares](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html#scipy-optimize-least-squares)

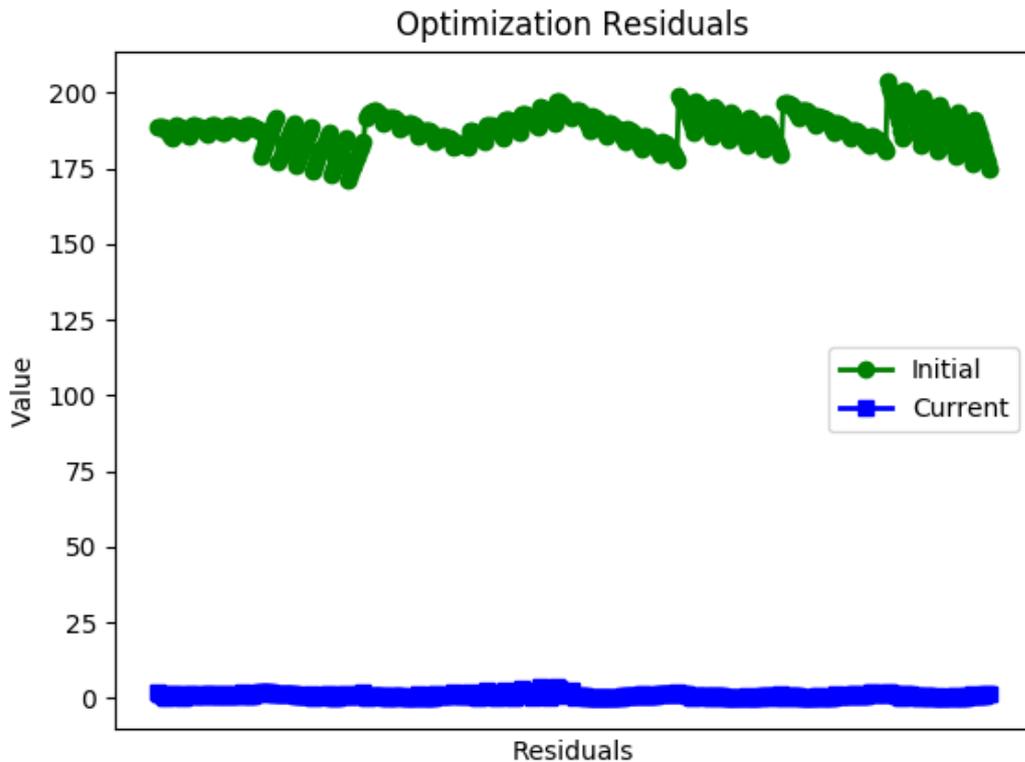


Figura 4.9: Variação dos residuais desde o início ao final da calibração, utilizando esta 8 coleções

Os valores dos parâmetros do otimizador acabam por influenciar não só o erro médio final, bem como as iterações permitidas. Assim, foram testados ambos os métodos com os mesmos parâmetros, neste caso, *ftol*, *xtol*, *gtol* e *diff\_step* com valor igual a  $10^{-4}$ .

Na Tabela 4.3 encontram-se os valores referentes às duas calibrações anteriormente referidas. Em relação aos tópicos anteriores é visível um aumento substancial do erro médio final para ambos os casos, tal como seria de esperar. Utilizando as oito coleções continua a existir um erro menor quando este é calculado no mundo 3D da câmara, mas a diferença é praticamente nula para o outro método. Porém, há um aumento substancial ao nível do número de iterações utilizadas. Isto porque, calculando o erro em píxeis na imagem, o otimizador apenas necessitou de 227 iterações, obtendo um erro inferior a 1 píxel. Por outro lado, a primeira metodologia necessitou cerca de 18 vezes mais iterações, o que é uma grande diferença a nível de esforço computacional. Deste modo, foram feitas alterações aos parâmetros de forma a reduzir o número de iterações para um valor perto dos 200, última coluna da Tabela 4.3. Todavia, ao reduzir o esforço computacional, o erro médio já ultrapassa o limite desejado (2 píxeis).

Tabela 4.3: Variação dos residuais desde o início ao final da calibração, utilizando esta 8 coleções

Metodologia	Erro Médio Final (píxeis)	Número Total de Iterações
Distância Mundo 3D	0.898	3960
Distância na Imagem	0.925	227
Distância Mundo 3D (Iterações Reduzidas)	2.294	251

Assim, pode-se concluir que, para mais de uma coleção, já não é rentável calcular o erro no mundo 3D, contrariamente ao que inicialmente se fazia prever. Porém, visto o foco deste trabalho envolver mais do que duas câmaras, é de relevo analisar ambos os métodos para esse caso.

## 4.2 Calibração de sistema com três câmaras

De acordo com o objetivo desta dissertação, calibração de um anel de câmaras, é então necessário estudar o comportamento quando são adicionados mais sensores ao sistema e é feita uma calibração mais complexa.

Ao exemplo anterior, onde foi realizada a calibração de duas câmaras, seguindo duas distintas metodologias, é adicionada agora uma terceira câmara entre as duas primeiras. O objetivo passa, então, pela calibração deste sistema novamente através do auxílio de um xadrez de calibração. Os três sensores partilham de informação em comum, ao detetarem simultaneamente o padrão de xadrez.

Sendo o objetivo a calibração do sistema, é necessário então descobrir a posição e orientação de todas as câmaras em relação às outras. O facto de existir um sensor adicional, torna todo o processo mais complexo, requerendo uma revisão aos parâmetros que irão ser otimizados.

### 4.2.1 Parâmetros a otimizar

A forma como se configura este problema, pode torná-lo mais simples ou mais complexo, fazendo também variar o erro associado. Deste modo, o ideal será encontrar a solução mais simples que permita obter um resultado final bastante positivo. Adjacente a este facto, tendo em vista a calibração de um anel de câmaras com o número de sensores podendo ser variado se necessário, implementar um método bastante flexível seria algo que se deveria ter em consideração.

Assim, neste caso em estudo é aplicado o sistema em que os sensores são calibrados em relação a um referencial global. Isto significa que a rotação e orientação de cada sensor será relativamente a esse mesmo referencial. Este método encontra-se detalhado na Subsecção 3.2.1.

Deste modo, utilizando esta metodologia, um elevado número de sensores não aumenta exponencialmente o esforço computacional, bem como o erro total da otimização.

Sendo a calibração do sistema de três câmaras realizada consoante o método do referencial global, o número de transformações a otimizar será igual ao número de câmaras utilizadas.

Assim, tal como sucedido no exemplo com dois sensores, os parâmetros a otimizar correspondem à posição e orientação de cada câmara, por exemplo, Equação 4.8. Da mesma maneira, a matriz de rotações será simplificada para um vetor de três elementos com recurso à função Rodrigues<sup>4</sup> do OpenCV.

$$\text{Câmara } {}^1\mathbf{T}_{\text{Referencial Global}} = [R|t]. \quad (4.8)$$

Neste caso em estudo, não é considerado nenhum elemento como fixo, pelo que todas as câmaras, bem como o padrão de xadrez, têm a sua transformação para o referencial global a ser otimizada. Isto significa então, que o vetor dos parâmetros que vão ser otimizados é composto pelos vetores de rotação e translação de cada câmara, bem como os do xadrez de cada coleção utilizada, Equação 4.9.

$$\text{Parâmetros} = \left\{ \overbrace{\mathbf{r}_{Cam,1}, \mathbf{t}_{Cam,1}, \dots, \mathbf{r}_{Cam,3}, \mathbf{t}_{Cam,3}}^{\text{Referentes às câmaras}}, \overbrace{\mathbf{r}_{Xadrez,1}, \mathbf{t}_{Xadrez,1}, \dots, \mathbf{r}_{Xadrez,n}, \mathbf{t}_{Xadrez,n}}^{\text{Referentes aos vários padrões}} \right\} \quad (4.9)$$

Tal como referido, caso sejam utilizadas mais do que uma coleção, existirão diferentes posições do xadrez, pelo que faz sentido que cada uma delas seja considerada um padrão independente. Deste modo, cada um possuirá uma matriz de transformação para o referencial global, matriz essa que tem de estar representada no vetor dos parâmetros. Ao otimizar a posição e orientação de cada xadrez, é possível atingir um valor mais perto do ideal para a calibração por dar liberdade a todos os elementos do sistema.

Tal como no exemplo de dois sensores, existe uma transformação de cada câmara para o xadrez da coleção em estudo, calculada diretamente através da função SolvePNP<sup>5</sup> do OpenCV. Esta é considerada como a posição verdadeira, *ground truth*. Esta é comparada com a multiplicação da transformação de cada câmara para o referencial global com a transformação do referencial global para o xadrez, Equação 4.10, ambas a serem otimizadas durante a calibração. Isto leva a que cada ponto do xadrez possua duas posições distintas para o sensor, quer seja no referencial 3D do mesmo, quer seja na imagem dele retirada.

$$\overbrace{\text{Câmara } \mathbf{T}_{\text{Xadrez}}}^{\text{ground truth}} = \overbrace{\text{Câmara } \mathbf{T}_{\text{Referencial Global}}}^{\text{Matriz a otimizar}} \cdot \overbrace{\text{Referencial Global } \mathbf{T}_{\text{Xadrez}}}^{\text{Matriz a otimizar (xadrez)}}. \quad (4.10)$$

Com a finalidade de otimizar os parâmetros referidos, é necessário avaliar a evolução dos mesmos. Assim, é necessário que a função objetivo da otimização receba os parâmetros e retorne um vetor de residuais. Isto permite que o otimizador possa modificar os valores de forma a diminuir o valor de cada um destes residuais.

Cada sensor, tendo duas transformações associadas, *ground truth* e a transformação a ser otimizada, possui dois pontos projetados para cada um do xadrez, seja no mundo 3D ou na

<sup>4</sup>[https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html#rodrigues](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#rodrigues)

<sup>5</sup>[https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html#solvepnp](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#solvepnp)

imagem. Assim, a diferença entre estes corresponde ao erro associado a cada ponto do xadrez para a câmara em questão. Organizando num vetor todos o erro de todos os pontos do padrão, ficamos com o erro associado a uma câmara para uma coleção, Equação 4.11.

$$\mathbf{Erro}_{\text{Cam } 1, \text{ Xadrez } 1} = \{\mathbf{Erro}_{\text{Ponto } 1, \text{ Xadrez } 1}, \mathbf{Erro}_{\text{Ponto } 2, \text{ Xadrez } 1}, \dots, \mathbf{Erro}_{\text{Ponto } 54, \text{ Xadrez } 1}\} \quad (4.11)$$

Utilizando transformações de cada câmara para um referencial global e considerando que nenhuma delas possui uma posição fixa, cada uma possui um vetor de erros com todos os pontos do xadrez, como se encontra na Equação 4.11.

Cada coleção, ou seja, cada par de imagens com o xadrez numa posição diferente, possui então um vetor de erros, sendo este composto pelos três vetores das câmaras do sistema, Equação 4.12.

$$\mathbf{Erro}_{\text{Coleção } 1} = \{\mathbf{Erro}_{\text{Cam } 1, \text{ Xadrez } 1}, \mathbf{Erro}_{\text{Cam } 2, \text{ Xadrez } 1}, \mathbf{Erro}_{\text{Cam } 3, \text{ Xadrez } 1}\} \quad (4.12)$$

Caso haja mais do que uma coleção na calibração, a função de custo percorre cada uma delas e por fim retorna um vetor de residuais integrando o vetor de cada uma das coleções utilizadas.

$$\mathbf{Residuais} = \{\mathbf{Erro}_{\text{Coleção } 1}, \mathbf{Erro}_{\text{Coleção } 2}, \dots, \mathbf{Erro}_{\text{Coleção } n}\} \quad (4.13)$$

## 4.2.2 Comparação entre função de erro calculada através da distância euclidiana no mundo 3D e da distância entre píxeis

Com o objetivo de calibrar um sistema com elevado número de sensores, e após vários estudos realizados no sistema com apenas duas câmaras, é realizado um estudo de forma a comparar os dois diferentes métodos do cálculo do erro por parte da função objetivo.

Deste modo, com a adição de uma câmara espera-se ter uma melhor percepção do comportamento de cada uma das metodologias.

Tendo isto em mente, é necessário focar em duas questões importantes: Erro médio final e o número total de iterações realizadas. O objetivo passará por perceber qual dos dois métodos - calcular a distância em píxeis na imagem retirada de uma câmara ou a distância entre os pontos no referencial 3D da mesma - permite uma melhor relação entre estes dois parâmetros.

Para tal, foi realizado um estudo onde, partindo de apenas uma coleção, se foi aumentando gradualmente o número de coleções utilizadas até um máximo de 8. De lembrar que neste exemplo todas as coleções são completas.

Assim, para cada calibração realizada, é registado o erro médio final e as iterações que o otimizador realizou. Assim, adaptando o método de cálculo dos parâmetros iniciais da otimização (Subsecção 3.2.2), foram obtidos os gráficos da Figura 4.10 e da Figura 4.11.

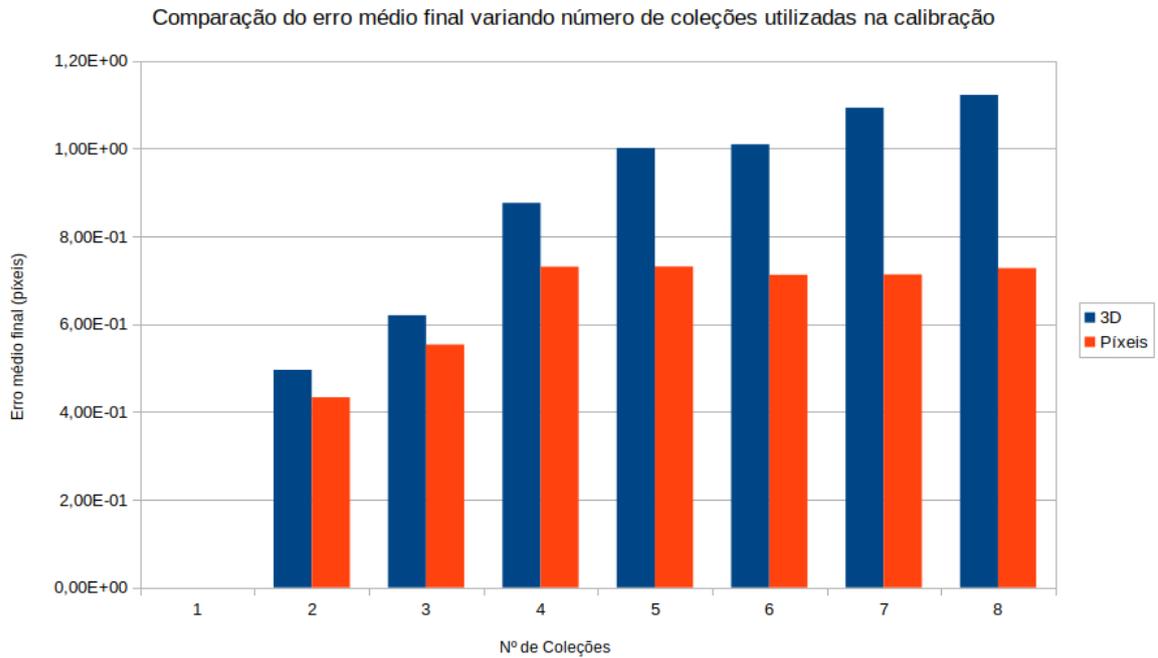


Figura 4.10: Comparação do erro médio final entre os residuais calculado no mundo 3D (azul) e os calculados na projeção do xadrez na imagem (laranja) variando o número de coleções.

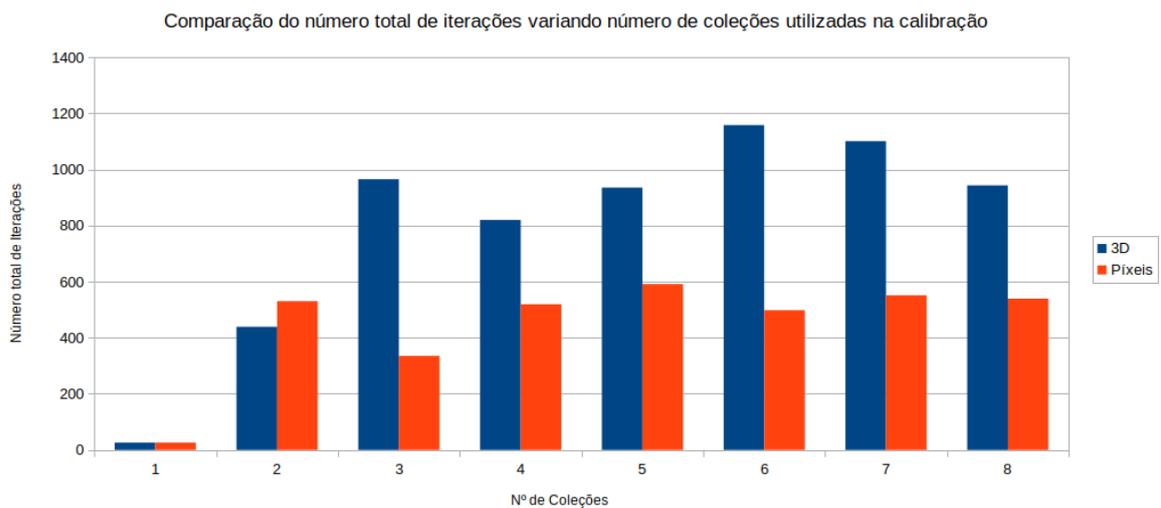


Figura 4.11: Comparação do número total de iterações entre os residuais calculado no mundo 3D (azul) e os calculados na imagem (laranja), variando o número de coleções.

Tendo por base estes últimos gráficos, facilmente se conclui que calcular os residuais através da distância entre os pontos da projeção de um xadrez e os mesmos pontos detetados diretamente na imagem traz um maior benefício nas duas vertentes em estudo.

Primeiramente, em relação ao erro médio final, para além de apresentar sempre valores mais baixos, estabiliza o seu valor a partir da utilização de 4 coleções, ao passo que o outro método até utilizando 8 coleções não mostra grandes certezas de que esteja num valor já estável.

Por fim, a nível de iterações realizadas, acontece o mesmo para o método da distância na imagem, uma vez que o número de iterações é praticamente sempre menor e o valor é estável (não há grandes discrepâncias ao adicionar coleções à calibração).

Na Tabela 4.4 estão os valores da calibração realizada com 8 coleções em específico. Sendo a calibração que utiliza mais coleções do presente estudo, é a que mais se assemelha ao objetivo final do projeto - calibrar um anel de câmaras - onde irão ser utilizadas várias coleções para calibrar um número elevado de câmaras.

Tabela 4.4: Comparação entre erro médio final e número total de iterações realizadas utilizando 8 coleções

Metodologia	Erro Médio Final (píxeis)	Número Total de Iterações
Distância Mundo 3D	1.122	943
Distância na Imagem	0.728	539

Neste exemplo específico, mais uma vez se observa a superioridade do método do erro na imagem, com um valor final médio inferior a 1 píxel e bastante menor que o método concorrente. Também a nível de iterações realizadas ganha por larga margem, sendo importante para diminuir a exigência computacional do processo.

Deste modo, conclui-se que na função objetivo irá ser calculado o erro pela distância entre a projeção do xadrez na imagem e os mesmos pontos mas detetados diretamente na mesma.

Assim, o próximo passo será realizar testar para um sistema com maior número de câmaras.

### 4.3 Calibração de sistema com 5 câmaras

Após o estudo de duas e três câmaras, onde foram testadas duas diferentes funções objetivo, chegou-se à conclusão de que o erro calculado a partir da distância, em píxeis, na imagem retirada de um sensor é a melhor solução, tanto a nível de erro total como a nível de esforço computacional.

Nesta secção irão ser apresentados os resultados retirados partindo da solução proposta para este projeto, Capítulo 3.

Diferenciando do exemplo do sistema com três câmaras, acresce o facto de nos parâmetros de entrada da função objetivo estarem incluídos os valores intrínsecos de cada câmara (matriz intrínseca e coeficientes de distorção).

Para este sistema, já não são consideradas apenas coleções completas, ou seja, coleções (conjunto de imagens retiradas simultaneamente de todas as câmaras) em que o xadrez seja detetado em todas as suas imagens. Sendo o objetivo do projeto aplicar a calibração a um anel de câmaras, de forma a detetar 360° em torno de um veículo autónomo, nunca será

possível obter uma coleção completa. Ao longo desta secção serão demonstrados diferentes testes realizados tendo isto em mente. Será, primeiramente, explorada a influência do número de coleções no resultado final da calibração. Segundamente, irá ser testada, a influência que tem a classificação de uma coleção como válida, ou seja, o número de imagens a detetar o xadrez para que uma coleção seja considerada válida. Por fim, irá ser analisado o impacto que a classe criada para calcular o *first guess*, Subsecção 3.2.2, tem na precisão da calibração.

### 4.3.1 Variação do número de coleções utilizadas na calibração

Tal como realizado para três câmaras, é aqui realizado um estudo sobre a influência do número de coleções utilizadas na calibração. Esta análise aparece como apoio às restantes realizadas, pois permite não só perceber se um menor ou maior número de coleções pode beneficiar ou prejudicar qualquer que seja o estudo que se esteja a realizar, bem como uma adição de valores que podem também ser comparáveis aos valores da análise em questão.

Assim, foi analisado um dataset com 40 coleções, tendo sido realizada a calibração utilizando todas estas coleções primeiramente e, posteriormente, foi-se diminuindo o número utilizado, de 5 em 5, até um mínimo de 10 coleções. Dentro das coleções selecionadas nem todas serão consideradas para a calirbação. Apenas as coleções em que o xadrez seja detetado em duas ou mais imagens são consideradas como válidas.

Para cada calibração foi calculado o erro médio com os parâmetros iniciais e o erro médio no final da otimização, de forma a perceber a evolução que foi alcançada. Foi também registado o número de iterações que o otimizador realizou, de forma a avaliar o esforço computacional entre calibrações.

Na Tabela 4.5 encontram-se estes valores, descritos anteriormente, bem como o número total de residuais que compõem o vetor retornado pela função objetivo. Este último valor está diretamente relacionado com o número total de imagens onde o xadrez foi detetado, no conjunto de todas as coleções.

Tabela 4.5: Valores dos erros médios iniciais e finais, bem como o número total de iterações, de calibrações com diferentes números de coleções.

Nº Coleções		Erro médio	Erro médio	Nº total	Número
Total	Válidas	inicial (píxeis)	Final (píxeis)		
40	24	4.30767	0.87266	2206	3780
35	19	4.29977	0.80121	1939	2538
30	15	2.24747	0.77155	1150	1944
25	11	1.41358	0.51546	1672	1350
20	9	1.30691	0.49306	1243	1134
15	8	1.36099	0.52133	1044	1026
10	5	1.14010	0.53382	1049	702

Relembrando que o objetivo da otimização seria obter um erro médio inferior a 2 píxeis, analisando a tabela anterior, percebe-se que o *first guess* calculado é já bastante próximo desse valor para elevadas coleções e até mesmo inferior quando o número total é reduzido. Esta ligeira diferença, entre as calibrações com mais e menos coleções, que chega a um máximo de 3 píxeis, pode ser explicada pelas imagens com que os parâmetros iniciais são calculados (explicado no Subsecção 3.2.2); e pelo facto de os padrões de calibração, no cálculo do *first guess*,

serem os elementos que maior erro acarretam, logo, um aumento de coleções leva a um direto aumento do erro médio inicial. Tal como será explicado seguidamente, na Subsecção 4.3.3, uma diferença tão ligeira não traz impacto nos resultados. Porém, mesmo o valor máximo de 4,31 píxeis, obtidos na calibração com 40 coleções, é já um valor bastante positivo para a fase inicial de uma otimização. Esta variação dos parâmetros iniciais em conjunto com o número de residuais a otimizar explicam a diferença obtida no total de iterações. À medida que o erro inicial baixa para perto dos dois píxeis o número de iterações tende a estabilizar (salvo uma excepção).

Olhando para o erro médio final, todos os valores se encontram abaixo do valor pretendido, porém existe uma variação deste valor ao calibrar com diferentes número de coleções. Na Figura 4.12 é possível ter uma melhor percepção da evolução do erro consoante o número de coleções válidas da calibração.

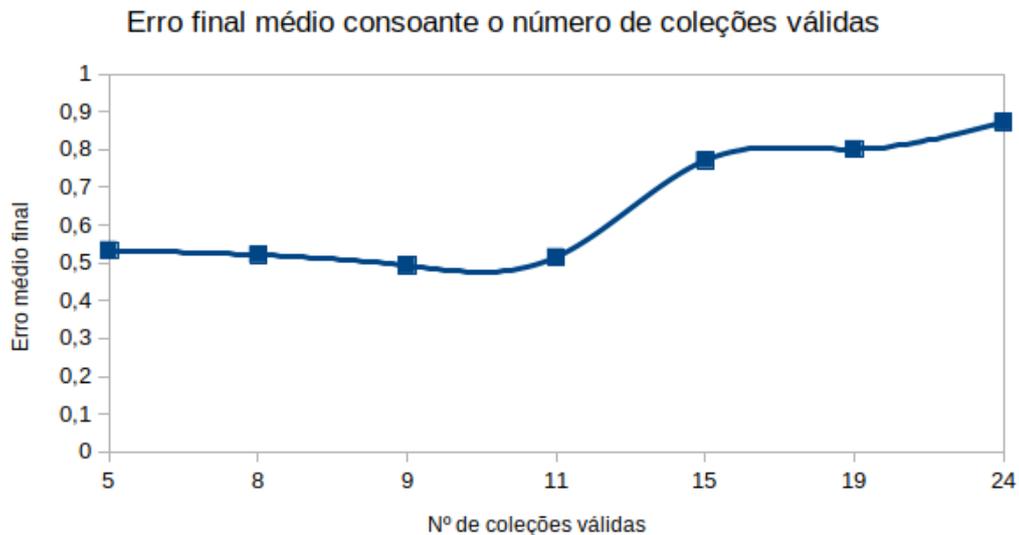


Figura 4.12: Variação do erro médio final com o número de coleções utilizadas na calibração.

Visualizando o gráfico, percebe-se que existe uma ligeira diminuição do erro médio até à casa da dezena, sendo que a partir desse valor o erro acaba por aumentar ligeiramente. Aumentando o número de coleções, leva a que também a probabilidade de encontrar uma coleção com menos qualidade aumente. Isto é perceptível no aumento que acontece das 11 coleções para as 15, que posteriormente não volta a acontecer daí para a frente. Ou seja, uma coleção entre a 11 e a 15 acaba por levar a um pior resultado final.

Na Figura 4.13 é possível perceber que nos últimos grupos de residuais se encontram dois grupos que chegam a atingir valores absolutos de 10 píxeis de erro, enquanto que os restantes excepcionalmente passam dos 6 píxeis. No entanto o facto de o erro médio final, apesar destes pequenos precalços, se manter bastante abaixo dos 2 píxeis, não suscitou a necessidade de eliminar estas coleções.

Para além do erro médio final, foi calculado o erro médio em cada um dos eixos da imagem,  $x$  e  $y$ . Na Tabela 4.6 encontram-se os valores iniciais e finais de cada uma das calibrações

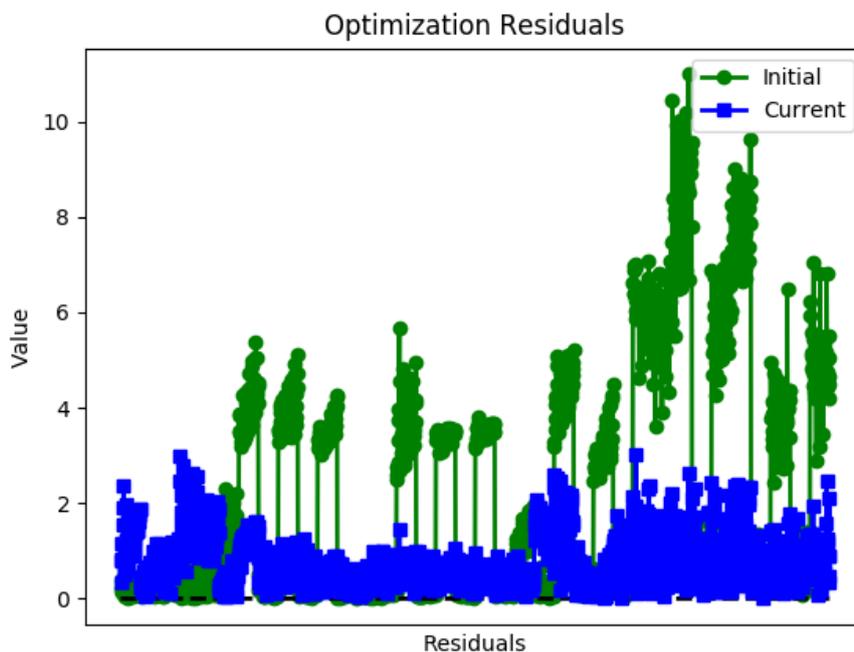


Figura 4.13: Representação do valor de cada residual na calibração utilizando 30 coleções. A verde encontram-se os valores iniciais e a azul os finais. (Imagem retirada da função de visualização do otimizador utilizado no projeto. Este otimizador possui o inglês como linguagem adotada)

anteriores, bem como o número total de imagens que foram utilizadas para o cálculo de residuais.

Desta tabela é perceptível que o maior desvio inicial se encontra em relação ao eixo  $x$ , porém o resultado final já aproxima bastante a diferença entre os dois eixos. Na Figura 4.14 é possível ver a diferença do erro médio de cada imagem, na fase inicial e na fase final, utilizada na calibração com 20 coleções.

Cada imagem possui 54 residuais (cada residual corresponde a um canto do xadrez), sendo calculado o erro médio desses 54 valores em cada eixo. Deste modo, o xadrez dessa imagem fica reduzido a um ponto com duas coordenadas, cada uma delas o erro médio nesse eixo. No gráfico da Figura 4.14 são então dispostos 21 pontos (calibração com 21 imagens válidas) azuis e outros tantos laranjas. Os azuis correspondem aos erros das imagens antes de ser realizada a otimização. Pode-se, então, ver uma grande dispersão dos mesmos, com erros a atingir 4 píxeis apenas num eixo. A laranja encontram-se representados os pontos finais, depois da otimização. É notória a melhoria em relação aos primeiros, estando todos muito mais compactos, sendo que as raras exceções se encontram dentro dos valores de erro aceitáveis.

Tabela 4.6: Valores dos erros médios iniciais e finais, nos diferentes eixos, consoante o número de imagens utilizadas para calcular residuais na função objetivo.

Coleções	Imagens utilizadas	Erro médio inicial em x (píxeis)	Erro médio final em x (píxeis)	Erro médio inicial em y (píxeis)	Erro médio final em y (píxeis)
40	70	3.8158576	0.52271485	1.689079	0.5836878
35	47	3.8707113	0.5312932	1.5852149	0.48789242
30	36	2.0106127	0.5036709	0.6833015	0.46071255
25	25	1.2249863	0.3580213	0.3931288	0.29150125
20	21	1.0953552	0.33719823	0.41891292	0.28722525
15	19	1.1743429	0.35484377	0.39336243	0.2981218
10	13	0.89746493	0.3866167	0.45935842	0.27959767

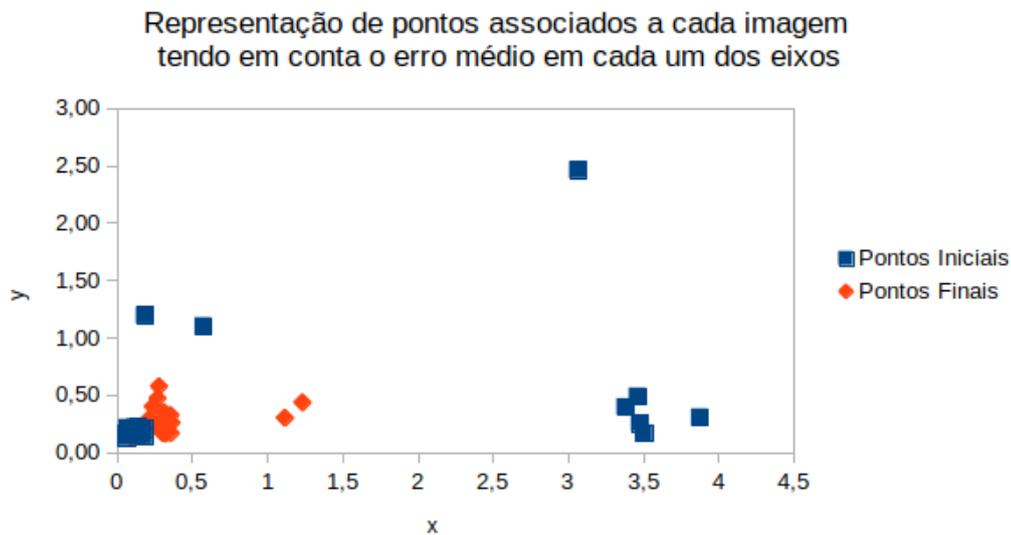


Figura 4.14: Representação do erro médio de cada imagem, tendo em conta a distância média (erro) em cada eixo, utilizada para calcular residuais na calibração com 20 coleções. A azul encontram-se os erros iniciais e a laranja os finais.

Tendo isto tudo em consideração, a variação de coleções não prejudica os resultados da calibração, deste modo, o utilizador tem uma maior margem de manobra, pois não lhe é requerido um elevado rigor na sua interação com o programa. Ou seja, o projeto desenvolvido mostra uma elevada precisão para qualquer conjunto de coleções, seja este com um elevado ou reduzido número de elementos, não obrigando ao utilizador a obter um número exato, ou ideal, de coleções de forma a atingir resultados positivos.

Conclui-se então que estes resultados demonstram uma boa robustez e flexibilidade do trabalho realizado.

### 4.3.2 Variação na classificação de coleção válida

Para que seja completa, uma coleção tem de possuir, neste exemplo específico, 5 câmaras a detetar o xadrez simultaneamente, ou seja, todas as câmaras do sistema. De forma a solucionar o problema proposto, calibrar um anel de câmaras, é necessário que a realização da calibração seja possível apenas com coleções incompletas. Dentro destes moldes, para o presente caso, uma coleção tem vários níveis dentro do grupo das coleções incompletas. Quanto mais câmaras detetarem o xadrez, mais perto esta fica de se tornar completa.

No Capítulo 3 é explicada o critério que torna uma coleção válida é o facto de, em pelo menos, duas imagens, desta mesma coleção, o xadrez de calibração seja detetado. Porém, nesta secção é alterada essa forma de definir a validade ou não de uma coleção, alterando o número de imagens mínimo. Disto resulta que as coleções aceites requiram um mínimo nível dentro das incompletas.

Deste modo, considerando um dataset de 40 coleções diferentes (contendo desde 0 imagens a ver o xadrez até 5, esta última uma coleção completa), é testada a calibração usando quatro diferentes tipos de coleções válidas. Começando pela definição da Figura 3.5, ou seja, qualquer coleção onde duas ou mais imagens detetam o xadrez é válida. Posteriormente, é realizada a calibração aumentando progressivamente o número mínimo de sensores a detetar o padrão de calibração até apenas serem aceites coleções completas, Tabela 4.7. De notar que ao aumentar esta restrição, o número de coleções válidas para a otimização tende a diminuir, porém, tal como visto anteriormente, considerando que é realizado um bom *first guess* (estudo realizado na Subsecção 4.3.3), os resultados considerando poucas coleções podem acabar por até sair ligeiramente beneficiados.

Tabela 4.7: Valores dos erros médios iniciais e finais, bem como o total de iterações, da calibração aquando da variação do mínimo de imagens (fazendo variar o número de residuais) a detetar o xadrez para que a coleção seja considerada válida.

Nº de imagens a detetar o xadrez	Coleções válidas	Erro Médio Inicial (píxeis)	Erro Médio Final (píxeis)	Nº total iterações	Total de residuais
2, 3, 4 ou 5	24	4.30767	0.87266	2206	3780
3, 4 ou 5	12	3.36383	0.79669	1432	2484
4 ou 5	4	3.43418	0.67979	1376	918
5	3	4.52573	0.82777	1431	810

Através da tabela dos erros deste estudo, percebe-se que à medida que são exigidas apenas coleções parcialmente mais completas o erro e o número total de iterações tendem a decrescer

ligeiramente, excepto para as coleções completas onde o erro volta a aumentar. Isto pode levar a, erradamente, concluir que para melhorar a precisão da calibração o número de câmaras a detetarem o xadrez simultaneamente deve ser o maior possível. Porém, olhando para o número de residuais de quando são utilizadas apenas coleções onde 4 ou 5 imagens visualizam o xadrez (melhor erro médio final), 918, e cruzando com os dados dos resultados anteriores, percebe-se que não é o ideal. Isto porque na Subsecção 4.3.1, onde são sempre aceites todas as coleções com a classificação atribuída no Capítulo 3, uma calibração a otimizar 1026 residuais obteve um erro final inferior ao obtido aumentando as restrições. Deste modo, percebe-se que o erro elevado neste mesmo exemplo com essa configuração se deve ao número de residuais a otimizar.

De forma a estudar melhor este facto, foi tido um consideração um novo exemplo. De forma a perceber a influência da classificação de coleção válida na calibração, será realizado o processo anterior, porém as coleções válidas apenas podem possuir um tipo de coleções. Ou seja, inicialmente serão aceites apenas as coleções onde o xadrez seja detetado em apenas e só duas imagens, depois apenas em três e desta forma até às 5. Na Tabela 4.8 encontra-se os valores dos erros médios calculados para cada uma das calibrações.

Tabela 4.8: Valores dos erros médios iniciais e finais, bem como o total de iterações, da calibração aquando da variação do número exato de câmaras a detetar o xadrez para que a coleção seja considerada válida.

Nº de imagens a detetar o xadrez	Coleções válidas	Erro Médio Inicial (píxeis)	Erro Médio Final (píxeis)	Nº total iterações	Total de residuais
2	12	5.02942	0.53413	1626	1296
3	5	3.57184	0.65860	1050	810
4	4	3.28655	0.69871	1251	864
5	3	4.52573	0.82777	1431	810

Com este exemplo já se observa um aumento do erro final médio com o aumento do número de sensores a detetar o xadrez numa coleção válida. Tal como será visto na Subsecção 4.3.3, as posições dos xadrezes são os parâmetros mais difíceis para o otimizador de obter com precisão. Deste modo, a existência de coleções com mais câmaras, a detetarem o padrão de calibração, trazem maior restrição ao ajuste da sua posição. Deste modo, com o aumentar do número de sensores aumenta o erro, porém, mais uma vez, nunca aumenta para valores preocupantes, mostrando a robustez do trabalho.

Com estes estudos realizados, demonstra-se que não existe a necessidade da calibração influenciar a disposição das câmaras num veículo autónomo, ou seja, não existe a necessidade de existirem iúmeras câmaras para captar cada ponto de vista. Apenas se requer que haja informação em comum entre dois sensores simultaneamente, isto porque, apenas com duas imagens a detetar o xadrez em caada coleção permitiu atingir um erro médio bastante positivo.

### 4.3.3 Variação dos parâmetros iniciais da otimização

Na Subsecção 3.2.2 é apresentada uma classe apenas dedicada ao cálculo dos parâmetros iniciais do processo de otimização. Obter valores próximos dos valores reais pode beneficiar em muito a otimização. Isto permite que todo o processo seja composto por pequenos ajustes

nos valores apresentados e não em grandes passos em busca de uma aproximação aos valores reais.

Ao auxiliar a otimização, poderá diminuir o esforço computacional, visto que um *first guess* longe do real poderia levar o otimizador a necessitar de mais tempo para atingir valores aceitáveis.

Esta secção, tendo como base a classe criada, tem como objetivo o estudo da influência do *first guess* na performance da calibração. É expectável que o número de iterações total da otimização aumente com o afastamento dos parâmetros de entrada da função objetivo em relação aos considerados reais, porém pode não ser o único fator a ser influenciado.

Deste modo, são comparados os resultados obtidos na Subsecção 4.3.1, onde foi realizada a calibração com 20 coleções, por ser a que melhores resultados obteve, com três diferentes situações. Partindo dos valores calculados pela classe, numa primeira situação, é considerado que todas as câmaras se encontram, inicialmente, sobre o referencial global, e não apenas a primeira como tem sido realizado. Num segundo cenário, é realizado um processo semelhante, mas para a posição dos padrões de calibração. Ou seja, após calculada a posição do primeiro, é considerado que todos os restantes se encontram nesta mesma posição, isto é, todos se encontram sobrepostos no início da otimização.

Na Tabela 4.9 encontram-se os valores retirados das calibrações referidas.

Tabela 4.9: Valores obtidos como erro médio inicial e final modificando os parâmetros iniciais, de forma a os afastar dos valores considerados reais, para o estudo da sua influência no resultado final.

Variação do first guess	Coleções válidas	Erro Médio Inicial (píxeis)	Erro Médio Final (píxeis)	Nº total iteraões	Total de residuais
Sem modificação	9	1.30691	0.49306	1243	1134
Alterando posição das câmaras	9	106.094	0.63667	2482	1134
Alterando posição dos padrões	9	97.9543	4.24510	3104	1134

Assim como era esperado, ao piorar propositadamente os parâmetros iniciais, o erro médio inicial é, de facto, muito superior ao anteriormente calculado nos outros exemplos. Este facto, leva a que o número de iterações necessárias seja também efectivamente superior, tal acima descrito. Com estes dados, conclui-se que o método realizado para calcular os valores do *first guess* permite um menor esforço computacional por se encontrarem mais perto da solução final.

Para além deste facto, ao calcular os valores dos parâmetros de forma precisa, permite que o valor final seja ainda próximo da realidade, ou seja, com um erro menor. Pela Tabela 4.9, facilmente se observa que o valor do erro final, onde não é feita nenhuma alteração aos parâmetros, é o mais baixo e substancialmente inferior a 2 píxeis (valor considerado aceitável).

Porém, quando a posição dos sensores é alterada, o erro também consegue atingir valores semelhantes. Este método, mostra que, apesar de ser necessário o dobro das iterações, é possível fazer grandes ajustes na posição e orientação dos sensores, até atingir valores muito perto do real.

Por outro lado, quando se altera a posição e orientação dos padrões de calibração, apesar do erro médio inicial ser inferior à calibração onde se modifica a pose inicial das câmaras, não se atingem valores dentro dos limites aceitáveis, mesmo ao fim de 3000 iterações. Este problema mostra que ao otimizador ajustar a posição e orientação de um xadrez é mais complicado do que de uma câmara. Isto deve-se ao facto de a posição dos padrões estar restringido pelas câmaras que o observam, sendo que quanto mais câmaras o detetam, mais difícil se torna para o otimizador. Ou seja, quando o otimizador ajusta a pose de uma câmara, isto reflete-se nos residuais de todas as imagens desta, enquanto que, ao ajustar a pose de um xadrez apenas altera os residuais das imagens onde é detetado. Quer isto dizer que o otimizador para tentar reduzir os residuais de uma imagem tentará provavelmente alterar o parâmetro que menos irá ter influência nos restantes valores de erro. Na Figura 4.15 pode-se observar os valores iniciais e finais de cada residual calculado pela função objetivo nesta calibração.

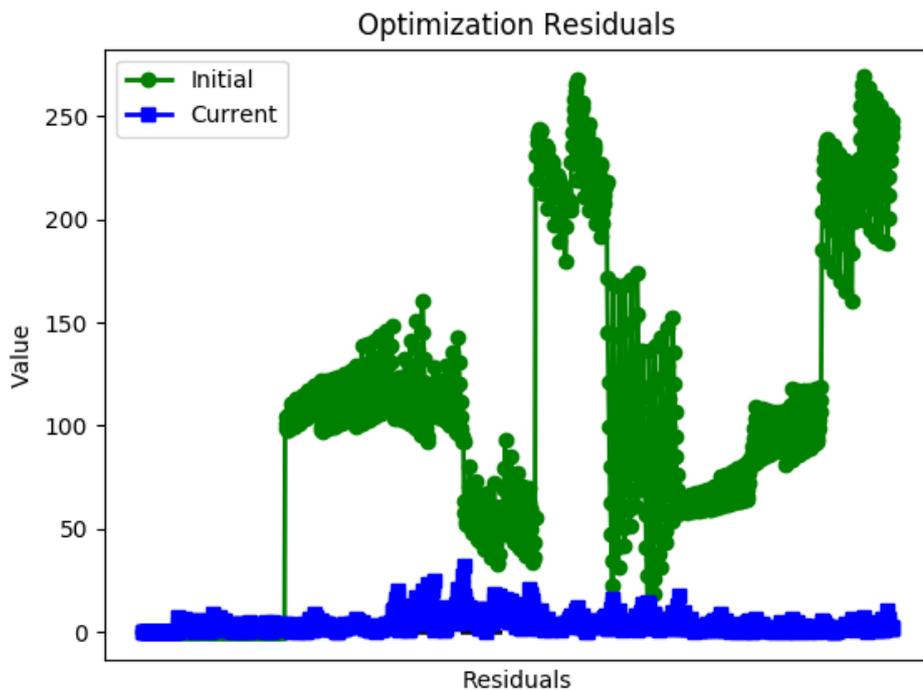
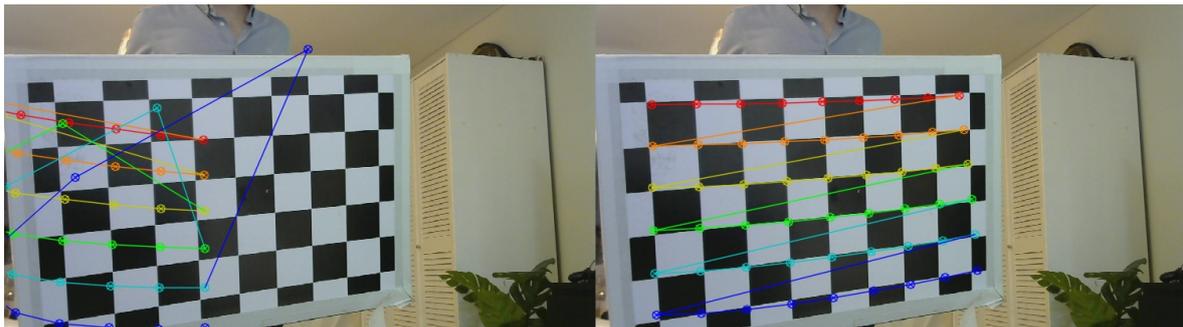


Figura 4.15: Representação dos residuais na calibração onde se altera a posição e orientação dos padrões de calibração. A verde encontram-se os valores iniciais e a azul os finais após a calibração.

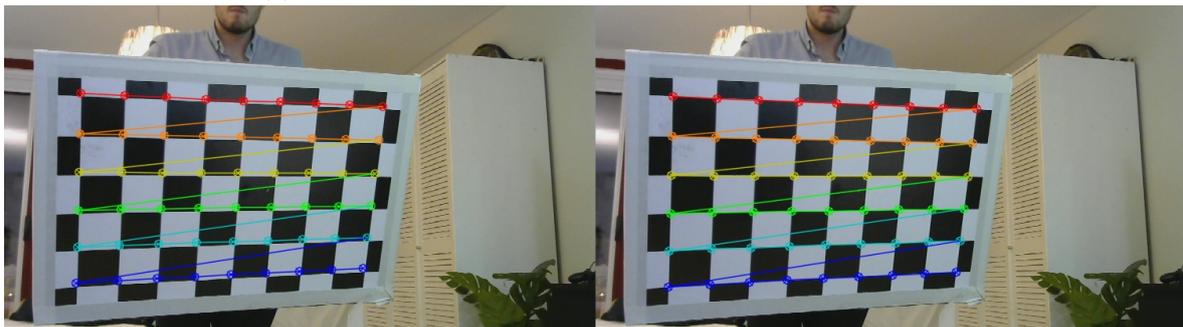
É possível então verificar que os finais ainda apresentam valores bastante elevado para o que seria de esperar.

Deste modo, conclui-se que a classe criada para calcular os valores dos parâmetros iniciais da otimização é fundamental para a obtenção de tão esperados resultados. Em todos os exemplos em que esta foi aplicada o erro médio final nunca excedeu o valor unitário de pixel, existindo uma margem substancial até ao limite estabelecido de duas unidades.

Na Figura 4.16 encontram-se as comparações das projeções dos cantos do xadrez com os parâmetros iniciais e posteriormente com os parâmetros otimizados para dois dos exemplos acima descritos: alterando a posição dos padrões de calibração e sem qualquer tipo de modificação aos parâmetros calculados pela classe do *first guess*.



(a) Calibração alterando a posição dos padrões de xadrez



(b) Calibração utilizando os valores retornados pela classe que calcula o *first guess*

Figura 4.16: Comparação das projeções dos pontos do xadrez com os parâmetros iniciais (imagens do lado esquerdo) e com os otimizados (imagens do lado direito) para dois exemplos: a) alterando propositalmente a pose dos padrões de calibração e b) utilizando apenas os valores calculados pela classe do *first guess*.

Esta última imagem, permite então retirar duas conclusões distintas: apesar dos maus valores iniciais do exemplo do topo, a projeção final não está muito longe do real, ou seja, a calibração conseguiu aproximar-se bastante bem dos valores pretendidos, mas não ao ponto de reduzir o erro para valores aceitáveis; e permite analisar a precisão que a classe criada traz ao trabalho, pois é possível visualizar que já o *first guess* se encontra melhor projetado que a projeção final do exemplo anterior, levando a um erro quase indetetável a olho nu na projeção dos parâmetros otimizados.



## Capítulo 5

# Conclusão e Trabalho Futuro

O principal objetivo deste trabalho passava pela instalação e calibração extrínseca de um anel de câmaras a bordo do ATLASCAR2, um veículo autónomo. Isto significa que se pretendia que o presente projeto fosse capaz de determinar a posição e orientação de todas as câmaras, constituintes deste mesmo anel, em relação às restantes.

Neste capítulo o objetivo foi alcançado parcialmente, uma vez que não foi possível a instalação do anel de câmaras a bordo do veículo autónomo, mas as simulações realizadas, em ambiente doméstico e laboratorial, permitiram realizar calibrações com elevada precisão.

Um dos problemas adjacentes ao objetivo proposto reside na impossibilidade da utilização de coleções completas (conjunto de imagens retiradas no mesmo instante onde o xadrez de calibração é detetado em todas elas). Esta adversidade advém do facto de se pretender obter uma visão em todo o redor do veículo inteligente, de forma a facilitar a recolha de informação em toda a sua volta. Porém, existindo uma captação de imagem a 360° em torno do veículo, é impossível que todas as câmaras detetem, no mesmo instante temporal, o mesmo padrão de calibração.

Perante os testes realizados ao programa criado, pode-se concluir que o trabalho desenvolvido possui uma elevada precisão e robustez. A realização de uma função que calcula um *first guess* muito próximo do considerado real permite obter, em todas as situações testadas, valores finais de erro muito inferiores ao limite máximo considerado aceitável (2 píxeis). Seja variando o número de coleções utilizadas na calibração, seja na forma como se classifica como válida uma coleção, os resultados apresentados são sempre muito positivos.

A dificuldade criada pela atual pandemia (Covid-19, explicada na Secção 1.5), que levou à não instalação do anel de câmaras, acabou por permitir o estudo realizado na Subsecção 4.3.2, onde foi possível retirar coleções contendo desde 2 a 5 imagens válidas - estudo bastante elucidativo no que à calibração utilizando coleções incompletas diz respeito.

Tendo a consciência de que o objetivo foi cumprido (obtenção de um valor final de erro aceitável), é de realçar algumas vantagens que este projeto pode trazer para a área da calibração. Em primeiro, como já referido, o facto de não serem exigidas coleções completas pode ser bastante benéfico para o mundo da robótica, principalmente para a condução autónoma. Isto permite que seja criado um sistema robótico com vários sensores sem informação comum entre todos eles. É apenas necessário que haja uma conexão, direta ou indireta, entre todos eles, ou seja, cada sensor tem de partilhar com, pelo menos, mais um. Deste modo, esse

outro sensor terá de partilhar com mais um sensor e assim sucessivamente, de forma a que exista uma cadeia que ligue todos os constituintes do sistema. Posteriormente, considerando a posição e orientação de todos os sensores em relação a um só referencial global possibilita um aumento do número total de sensores do sistema sem grandes consequências computacionais. Isto acontece pelo facto de cada sensor adicional apenas acrescentar à calibração mais uma transformação a otimizar, a posição e orientação deste relativamente ao referencial global.

Para além do referido, o projeto tenta sempre minimizar a hipótese de erro humano. Primeiramente, otimiza os valores dos parâmetros intrínsecos que recebe do *dataset*, compensando uma eventual falta de rigor na execução da calibração intrínseca. Segundo, obtém bons resultados independentemente do número de coleções utilizadas na calibração extrínseca.

Este número de coleções referido anteriormente, bem como o número de sensores do sistema, podem ser variados, visto que o programa que realiza a calibração extrínseca é configurado pelo ficheiro JSON contido no *dataset*. Deste ficheiro são retiradas as informações sobre o número de sensores e coleções, entre outras. Ou seja, caso neste conjunto venha informação relativa a apenas 3 sensores ou 5, com 10 ou com 20 coleções, a calibração é realizada de igual modo e de forma automática.

Após uma exploração dos benefícios que este projeto possa trazer à área em estudo, é de relevo terminar com o que pode ser a continuação do trabalho iniciado nesta dissertação.

Como primeiro passo terá de estar, logicamente, a instalação das câmaras no ATLAS-CAR2. Este sendo um dos objetivos iniciais da dissertação, não cumprido, terá de ser uma das seguintes etapas a realizar. Considerando que este último ponto apresentado se encontra realizado, pode-se encontrar formas de melhorar a solução apresentada. Sendo a calibração uma área sempre presente no mundo da robótica, este será um projeto que deverá estar em constante melhoria e desenvolvimento.

Uma das primeiras melhorias a aplicar reside na redução da necessidade de intervenção por parte do utilizador. Relembrando que o projeto se encontra dividido em duas partes distintas (recolha e estruturação de dados e calibração extrínseca), é na primeira que é requerida a intervenção do utilizador. Desta forma, tornar esta etapa inicial mais automática tornaria o processo mais simples de realizar. Nesta fase é necessário fornecer a lista de sensores ligados ao sistema, requer a cooperação de duas pessoas na recolha de coleções, caso não exista suporte para o padrão de calibração e ainda exige que haja uma diferenciação para as coleções utilizadas na calibração intrínseca. Esta última é útil por poder existir a necessidade de capturar mais coleções para a calibração extrínseca, mas não em número suficiente que permita a realização da calibração intrínseca. Porém, as duas primeiras poderão ser melhoradas de forma a tornar todo o processo mais fácil de realizar. O programa conseguir captar automaticamente as coleções ao detetar a validade das imagens retiradas das câmaras permitia que fosse necessário apenas uma pessoa para realizar todo o processo. Por fim, a deteção automática do número de sensores por parte do sistema tornaria o processo muito mais fácil e intuitivo.

O objetivo final é que todas as alterações futuras permitam que este projeto sirva como ponto de partida para a resolução dos problemas da calibração com coleções incompletas, algo muito importante para a área da condução autónoma, uma vez que os resultados obtidos foram bastante promissores.

# Bibliografia

- [1] Agarwal, S., Snavely, N., Seitz, S., Szeliski, R.: Bundle adjustment in the large. pp. 29–42 (11 2010)
- [2] Basso, F., Menegatti, E., Pretto, A.: Robust intrinsic and extrinsic calibration of rgb-d cameras. *IEEE Transactions on Robotics* 34(5), 1315–1332 (2018)
- [3] Broggi, A., Bertozzi, M., Fascioli, R.: Self-calibration of a stereo vision system for automotive applications. *Proceedings - IEEE International Conference on Robotics and Automation* 4 (03 2001)
- [4] C, D., Kannala, J., Heikkila, J.: Joint depth and color camera calibration with distortion correction. *IEEE transactions on pattern analysis and machine intelligence* 34 (05 2012)
- [5] Carrera, G., Angeli, A., Davison, A.J.: Slam-based automatic extrinsic calibration of a multi-camera rig. In: *2011 IEEE International Conference on Robotics and Automation*. pp. 2652–2659 (2011)
- [6] Castro, A.: Multi-modal sensor calibration on board the atlascar2. Master Thesis, University of Aveiro (2019)
- [7] Dang, T., Hoffmann, C., Stiller, C.: Continuous stereo self-calibration by camera parameter tracking. *IEEE Transactions on Image Processing* 18(7), 1536–1550 (2009)
- [8] Fernández-Moral, E., González-Jiménez, J., Rives, P., Arevalo, V.: Extrinsic calibration of a set of range cameras in 5 seconds without pattern (09 2014)
- [9] Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In: Fischler, M.A., Firschein, O. (eds.) *Readings in Computer Vision*, pp. 726 – 740. Morgan Kaufmann, San Francisco (CA) (1987), <http://www.sciencedirect.com/science/article/pii/B9780080515816500702>
- [10] Georg Mueller, R., Burger, P., Wuensche, H.: Continuous stereo self-calibration on planar roads. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. pp. 1755–1760 (2018)
- [11] Gomez, R., Briales, J., Fernández-Moral, E., González-Jiménez, J.: Extrinsic calibration of a 2d laser-rangefinder and a camera based on scene corners. vol. 2015, pp. 3611–3616 (06 2015)
- [12] Gu, F., Zhao, H., Ma, Y., Bu, P., Zhao, Z.: Calibration of stereo rigs based on the backward projection process. *Measurement Science and Technology* 27(8), 085007 (jul 2016), <https://doi.org/10.1088%2F0957-0233%2F27%2F8%2F085007>

- [13] Hansen, P., Alismail, H., Rander, P., Browning, B.: Online continuous stereo extrinsic parameter estimation. pp. 1059–1066 (06 2012)
- [14] Heng, L., Li, B., Pollefeys, M.: Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. pp. 1793–1800 (11 2013)
- [15] Hirz, M., Walzel, B.: Sensor and object recognition technologies for self-driving cars. *Computer-Aided Design and Applications* pp. 1–8 (01 2018)
- [16] Huang, K., Stachniss, C.: Extrinsic multi-sensor calibration for mobile robots using the gauss-helmert model (09 2017)
- [17] Kruger, L.E., Wohler, C., Wurz-Wessel, A., Stein, F.: In-factory calibration of multiocular camera systems. In: Osten, W., Takeda, M. (eds.) *Optical Metrology in Production Engineering*. vol. 5457, pp. 126 – 137. International Society for Optics and Photonics, SPIE (2004), <https://doi.org/10.1117/12.545396>
- [18] Kwak, K., Huber, D., Badino, H., Kanade, T.: Extrinsic calibration of a single line scanning lidar and a camera. pp. 3283–3289 (09 2011)
- [19] Liao, Y., Sun, Y., Li, G., Kong, J., Jiang, G., Jiang, D., Cai, H., Ju, Z., Yu, H., Liu, H.: Simultaneous calibration: A joint optimization approach for multiple kinect and external cameras. *Sensors* 17, 1491 (06 2017)
- [20] Ling, Y., Shen, S.: High-precision online markerless stereo extrinsic calibration (03 2019)
- [21] Lyu, Y., Bai, L., Elhousni, M., Huang, X.: An interactive lidar to camera calibration. pp. 1–6 (09 2019)
- [22] Madeira, T.: Enhancement of RGB-D image alignment using fiducial markers. Master Thesis, University of Aveiro (2019)
- [23] Marita, T., Oniga, F., Nedevschi, S., Graf, T., Schmidt, R.: Camera calibration method for far range stereovision sensors used in vehicles. pp. 356 – 363 (01 2006)
- [24] Mishra, S., Pandey, G., Saripalli, S.: Extrinsic calibration of a 3d-lidar and a camera (2020)
- [25] Mohamed, A., Culverhouse, P.F., Azambuja, R.D., Cangelosi, A., Yang, C.: Automating active stereo vision calibration process with cobots. *IFAC-PapersOnLine* 50(2), 163 – 168 (2017), <http://www.sciencedirect.com/science/article/pii/S2405896317335668>, control Conference Africa CCA 2017
- [26] Moutinho, N., Ferreira, R., Gaspar, J., Bernardino, A., Santos-Victor, J.: Markerless online stereo calibration for a humanoid robot (10 2014)
- [27] Mueller, G.R., Wuensche, H.: Continuous stereo camera calibration in urban scenarios. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). pp. 1–6 (2017)
- [28] Pandey, G., McBride, J., Savarese, S., Eustice, R.: Extrinsic calibration of a 3d laser scanner and an omnidirectional camera (01 2010)

- [29] Pelczynski, P., Ostrowski, B.: Automatic calibration of stereoscopic cameras in an electronic travel aid for the blind. *Metrology and Measurement Systems* 20 (06 2013)
- [30] Pereira, M.: Automated calibration of multiple lidars and cameras using a moving sphere. Master Thesis, University of Aveiro (2015)
- [31] Pereira, M., Silva, D., Santos, V., Dias, P.: Self calibration of multiple lidars and cameras on autonomous vehicles. *Robotics and Autonomous Systems* 83 (05 2016)
- [32] Pradeep, V., Konolige, K., Berger, E.: Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach. *Experimental Robotics* 79, 211–225 (01 2014)
- [33] Rathnayaka, P., Baek, S.H., Park, S.Y.: An efficient calibration method for a stereo camera system with heterogeneous lenses using an embedded checkerboard pattern. *Journal of Sensors* 2017, 1–12 (09 2017)
- [34] Roth, Z., Mooring, B., Ravani, B.: An overview of robot calibration. *Robotics and Automation, IEEE Journal of* 3, 377 – 385 (11 1987)
- [35] Santos, V., Almeida, J., Ávila, E., Gameiro, D., Oliveira, M., Pascoal, R., Sabino, R., Stein, P.: Atlascar - technologies for a computer assisted driving system on board a common automobile. In: 13th International IEEE Conference on Intelligent Transportation Systems. pp. 1421–1427 (Sep 2010)
- [36] Spangenberg, R., Langner, T., Rojas, R.: On-line stereo self-calibration through minimization of matching costs. In: Kämäräinen, J.K., Koskela, M. (eds.) *Image Analysis*. pp. 545–554. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
- [37] Staranowicz, A., Brown, G., Morbidi, F., Mariottini, G.: Easy-to-use and accurate calibration of rgb-d cameras from spheres. vol. 8333 (01 2014)
- [38] Vasconcelos, F., Barreto, J.P., Nunes, U.: A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder. *IEEE transactions on pattern analysis and machine intelligence* 34 (01 2012)
- [39] Wu, L., Zhu, B.: Binocular stereovision camera calibration. In: 2015 IEEE International Conference on Mechatronics and Automation (ICMA). pp. 2638–2642 (2015)
- [40] Zhang, Q., Pless, R.: Extrinsic calibration of a camera and laser range finder (improves camera calibration). vol. 3, pp. 2301 – 2306 vol.3 (01 2004)
- [41] Zhao, K., Iurgel, U., Meuter, M., Pauli, J.: An automatic online camera calibration system for vehicular applications. pp. 1490–1492 (10 2014)
- [42] Zhong, W., Dong, X.: Camera calibration method of binocular stereo vision based on OpenCV. In: Shen, C., Yang, W., Liu, H. (eds.) *AOPC 2015: Image Processing and Analysis*. vol. 9675, pp. 571 – 576. International Society for Optics and Photonics, SPIE (2015), <https://doi.org/10.1117/12.2202186>
- [43] Zuñiga-Noël, D., Ruiz-Sarmiento, J., Gomez, R., González-Jiménez, J.: Automatic multi-sensor extrinsic calibration for mobile robots (06 2019)