Universidade Federal de Juiz de Fora

Faculdade de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

Vinicius Ferreira Vidal

Reconstrução 3D virtual de componentes reais com dados térmicos para inspeção e manutenção

Juiz de Fora

2019

Vinicius Ferreira Vidal

Reconstrução 3D virtual de componentes reais com dados térmicos para inspeção e manutenção

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Energia , como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Leonardo de Mello Honório

Coorientador: André Luís Marques Marcato

Juiz de Fora

2019

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Ferreira Vidal, Vinicius.
Reconstrução 3D virtual de componentes reais com dados térmicos para inspeção e manutenção / Vinicius Ferreira Vidal. – 2019.
113 f. : il.

Orientador: Leonardo de Mello Honório
Coorientador: André Luís Marques Marcato
Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica, 2019.

1. SLAM. 2. Modelo Tridimensional. 3. Manutenção. 4. Inspeção Térmica. I. Honório, Leonardo de Mello, orient. II. Reconstrução 3D virtual de componentes reais com dados térmicos para inspeção e manutenção.

Vinicius Ferreira Vidal

Reconstrução 3D virtual de componentes reais com dados térmicos para inspeção e manutenção

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Energia , como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Aprovada em: 13 de Maio de 2019

BANCA EXAMINADORA

Prof. Dr. Leonardo de Mello Honório - Orientador
Universidade Federal de Juiz de Fora

André Luis Marques Marcato

Professor Dr. André Luís Marques Marcato - Coorientador
Universidade Federal de Juiz de Fora

Professor Dr. António Paulo Gomes Mendes Moreira
Universidade do Porto

Professor Dr. Augusto Santiago Cerqueira
Universidade Federal de Juiz de Fora

Professor Dr. Mario Antonio Ribeiro Dantas
Universidade Federal de Juiz de Fora

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA (PROPP)
Rua José Lourenço Kelmer, s/n – Campus Universitário
Bairro São Pedro – CEP: 36036-900 – Juiz de Fora – MG
(32) 2102-3785/ bancas.propp@ufjf.edu.br

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE PÓS-GRADUAÇÃO *STRICTO SENSU*

Nº PROPP: 412.13052019.20-M **Nº PPG:** 279

Ata da sessão pública referente à defesa da dissertação intitulada Reconstrução 3D Virtual de Componentes Reais com Dados Térmicos para Inspeção e Manutenção, para fins de obtenção do título de mestre em Engenharia Elétrica, área de concentração Sistemas de Energia Elétrica, pelo(a) discente VINICIUS FERREIRA VIDAL (matrícula: 102100419 - início do curso em 13/3/17), sob orientação do(a) Prof.(ª) Dr.(ª) Leonardo de Mello Honório e coorientação do(a) Prof.(ª) Dr.(ª) André Luis Marques Marcato.

Aos 13 dias do mês de maio do ano de 2019, às 10:00 horas, no(a) Sala de Videoconferência do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora (UFJF), reuniu-se a Banca Examinadora da Dissertação em epígrafe, aprovada pelo Colegiado do Programa de Pós-Graduação conforme a seguinte composição:

Prof.(a) Dr.(a) Leonardo de Mello Honório - Orientador(a) e Presidente da Banca

Prof.(a) Dr.(a) André Luis Marques Marcato - Coorientador(a)

Prof.(a) Dr.(a) Augusto Santiago Cerqueira - Membro titular interno

Prof.(a) Dr.(a) Mario Antonio Ribeiro Dantas - Membro titular interno

Prof.(a) Dr.(a) António Paulo Gomes Mendes Moreira - Membro titular externo (com participação remota, conforme Resolução n. 04/2016-CSPP)

Prof.(a) Dr.(a) Luis Edival de Souza - Suplente externo

Prof.(a) Dr.(a) Edimar José de Oliveira - Suplente interno

-- -

-- -

-- -

Tendo o(a) senhor(a) Presidente declarado aberta a sessão, mediante o prévio exame do referido trabalho por parte de cada membro da Banca, o(a) discente procedeu a apresentação de seu Trabalho de Conclusão de Curso de Pós-graduação *stricto sensu* e foi submetido(a) à arguição pela Banca Examinadora que, em seguida, deliberou sobre o seguinte resultado:

- [X] **APROVADO (Conceito A).**
- [ ] **APROVADO CONDICIONALMENTE (Conceito B)**, mediante o atendimento das alterações sugeridas pela Banca Examinadora, constantes do campo Observações desta Ata e/ou do parecer em anexo.
- [ ] **REPROVADO (Conceito C)**, conforme parecer circunstanciado, registrado no campo Observações desta Ata e/ou em documento anexo, elaborado pela Banca Examinadora.

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA (PROPP)
Rua José Lourenço Kelmer, s/n – Campus Universitário
Bairro São Pedro – CEP: 36036-900 – Juiz de Fora – MG
(32) 2102-3785/ propg@ufjf.edu.br

Observações da Banca Examinadora (caso inexistam, anular o campo):

Nada mais havendo a tratar, o(a) senhor(a) Presidente declarou encerrada a sessão de Defesa, sendo a presente Ata lavrada e assinada pelos(as) senhores(as) membros da Banca Examinadora e pelo(a) discente, atestando ciência do que nela consta.

INFORMAÇÕES:

- Para fazer jus ao título de mestre(a)/doutor(a), a versão final da dissertação/tese, considerada Aprovada, devidamente conferida pela Secretaria do Programa de Pós-Graduação, deverá ser tramitada para a PROPP, em Processo de Homologação de Dissertação/Tese, dentro do prazo regulamentar de 90 dias a partir da data da defesa. Após a entrega dos dois exemplares definitivos, o processo deverá receber homologação e, então, ser encaminhado à CDARA.
- Esta Ata de Defesa é um documento padronizado pela Pró-Reitoria de Pós-Graduação e Pesquisa. Observações excepcionais feitas pela Banca Examinadora poderão ser registradas no campo disponível acima ou em documento anexo, desde que assinadas pelo(a) Presidente.
- Esta Ata de Defesa somente poderá ser utilizada como comprovante de titulação se apresentada junto à Certidão da Coordenadoria de Assuntos e Registros Acadêmicos da UFJF (CDARA) atestando que o processo de confecção e registro do diploma está em andamento.

Prof.(a) Dr.(a) Leonardo de Mello Honório

Prof.(a) Dr.(a) André Luis Marques Marcato

Prof.(a) Dr.(a) Augusto Santiago Cerqueira

Prof.(a) Dr.(a) Mario Antonio Ribeiro Dantas

Prof.(a) Dr.(a) Antonio Paulo Gomes Mendes Moreira

Prof.(a) Dr.(a) Luis Edival de Souza

Prof.(a) Dr.(a) Edimar José de Oliveira

Vinicius Ferreira Vidal
Discente

AGRADECIMENTOS

Ao meu pai Luiz Fernando e minha mãe Maria da Consolação, pela educação, dedicação e base para poder desenvolver meus estudos da melhor forma possível.

A minha irmã Larissa por ajudar em todos os meus trabalhos.

A todos os meus familiares pelo incentivo e dedicação durante o período da pós-graduação.

Ao meu orientador, professor Leonardo Honório, pela atenção e tempo dedicados no trabalho.

A todos os amigos do GRIn, pelos bons momentos e ajuda durante todo o processo.

Aos bons amigos, pelo apoio e incentivo no curso e pelos bons momentos durante esses anos.

“A luta é vencida ou perdida muito longe das testemunhas, atrás das linhas, no ginásio e na estrada, bem antes de eu dançar sob aquelas luzes.”

Muhammad Ali

RESUMO

Os avanços na última década na área da visão computacional, em termos tanto de *hardware* como de *software*, permitiram a expansão da aplicação da realidade virtual e localização e mapeamento simultâneos (SLAM) de câmeras no ambiente. Com isso diversas áreas foram beneficiadas, incluindo a manutenção de equipamentos. No caso de equipamentos no setor elétrico, diversos defeitos ocasionam aumento relevante de temperatura, que podem ser inspecionados por imagens térmicas, porém cuidados como influência do ambiente e ângulo de visão podem resultar em interpretações errôneas. Esse trabalho traz um sistema que visa atingir essas duas funções: realizar a reconstrução do ambiente real em 3 dimensões no mundo virtual a partir de imagens obtidas por câmeras, o que traz para o escritório a análise do ambiente externo e suas características; e projetar sobre esse modelo 3D dados de temperatura obtidos por uma câmera térmica, a fim de analisar de todos os ângulos o estado do objeto que se deseja avaliar quanto à presença de defeitos. Para isso, um par de câmeras estéreo e uma câmera térmica são alinhadas e sincronizadas para captação de imagens durante inspeção. Cada nuvem de pontos 3D é obtida com a técnica de visão estéreo empregada. Para a união, ou registro dessas nuvens, é empregada uma técnica de Odometria Visual para definir a pose das câmeras segundo as características das imagens do par estéreo, seguida de filtros segundo *overlap* entre as nuvens. Por fim, em cada nuvem obtida e sincronizada com a imagem térmica, técnicas de visão computacional são usadas para projetar os pontos em 3 dimensões na última e obter assim seu dado de temperatura observado pela câmera. Unindo todas essas técnicas, desenvolvidas com o auxílio do *framework* ROS em C++, é obtido um modelo virtual com informações visuais e térmicas do ambiente e objeto inspecionado pelo equipamento. Os resultados mostram a confiabilidade das técnicas empregadas, juntamente com modelos obtidos satisfatoriamente em diversos cenários.

Palavras-chave: SLAM. Modelo tridimensional. Manutenção. Inspeção térmica.

ABSTRACT

The last decade improvements in the area of Computer Vision, in terms of both hardware and software, promoted and expansion in Virtual Reality and cameras Simultaneous Localization and Mapping (SLAM) in the environment, which brings benefits in several areas, including equipment maintenance. In the Electrical Engineering sector case, many flaws result in the equipment temperature raise, which can be inspected by thermal images, although environment influence and image angle of capture may result in wrong information interpretation. This work proposes a system that aims to achieve these two functions: develop the 3 dimensions real environment reconstruction in the virtual world from captured images, which takes the analysis of the external environment characteristics to the office; and project on these 3D models the temperature obtained by a thermal camera, in order to analyze from many angles the equipment state regarding the presence of any faults. Fo that purpose, a pair of stereo cameras and a thermal camera were aligned and synchronized for image capture during the inspection. Every 3D point cloud obtained is obtained via the applied stereo vision algorithm. For cloud accumulation, or registration, a Visual Odometry technique is applied to gather camera pose from stereo image pair characteristics, followed by filters and overlap between clouds analysis. At the end, for each cloud synchronized with the thermal image, computer vision techniques are used to project the 3D points in the last one, and therefore obtain the temperature data observed. Putting all those pieces together, developed in the ROS framework in C++, a visual and thermal virtual model of the object to be inspected is obtained. The results show the reliability of the applied techniques, aside of models obtained in several scenarios.

Key-words: SLAM. 3D Model. Maintenance. Thermal inspection.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

SFM	*Structure From Motion*
3D	3 Dimensões
SLAM	*Simultaneous Location and Mapping*
IMU	*Inertial Measurement Unit*
RGB	*Red-Green-Blue*
RGB-D	*Red-Green-Blue-Depth*
LIDAR	*Light Detection And Ranging*
VO	*Visual Odometry*
ROS	*Robotic Operating System*
KITTI	*Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago*
SIFT	*Scale Invariant Feature Transform*
PnP	*Perspective n Points*
RANSAC	*Random Sample Consensus*

SUMÁRIO

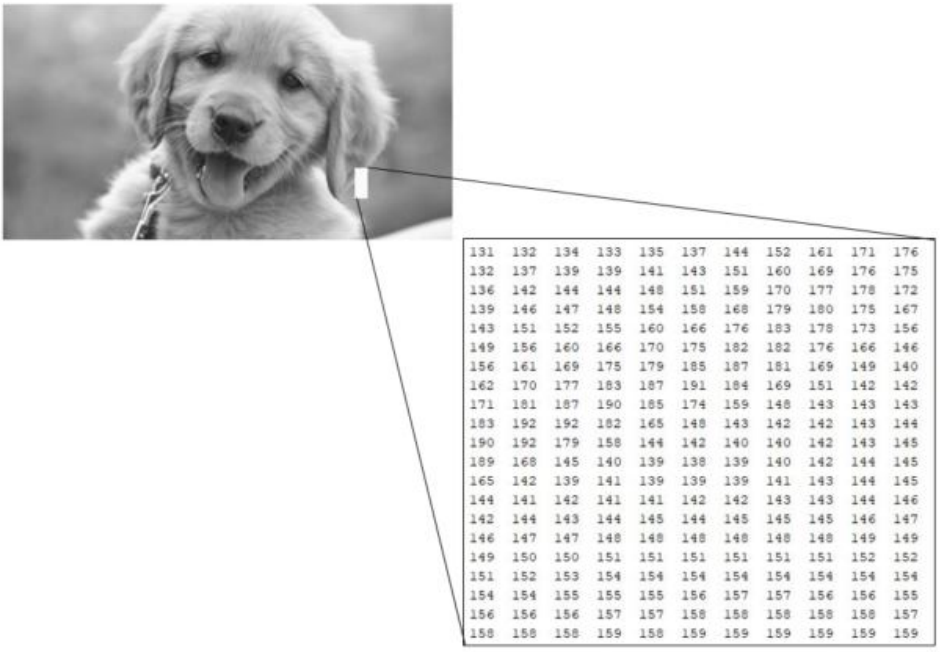
1 INTRODUÇÃO

O ser humano é capaz de perceber o mundo em aspectos tridimensionais a partir de sua visão de forma relativamente fácil [1]. Ao observar uma cena com os olhos é possível captar cor, profundidade e forma dos objetos e seres ali presentes. Em contrapartida, ao observar uma imagem como a da Figura 1, é possível ter ideia da forma e posição relativa entre as formas geométricas, porém seu tamanho real e detalhes do seu aspecto tridimensional podem ser perdidos. Logo, enxergar a cena pode ser uma tarefa simples, mas até mesmo para uma pessoa não é trivial a tarefa de obter informações tridimensionais acuradas a partir de uma imagem.

Figura 1 – Cena exemplo para análise tridimensional.

Essa facilidade para percepção através da visão não se repete para o caso computacional, no qual se torna complexa do ponto de vista de custo de processamento e técnicas algorítmicas. O computador percebe a imagem como uma matriz de números, exemplificada na Figura 2 (como será explicado no Capítulo 2), onde as noções de objetos e trajetórias não são tão diretas como para o cérebro humano, o que demanda algoritmos dedicados que se desenvolvem até hoje nas áreas industriais e acadêmicas [2].

Figura 2 – Imagem digital vista pelo computador.



A Visão Computacional, área que se surge da Computação Gráfica graças ao desenvolvimento de *hardware* e *software*, estuda esse tema. Técnicas desenvolvidas nas áreas da física e da computação gráfica conseguem modelar computacionalmente como a luz se reflete em superfícies e é absorvida pela atmosfera e lentes das câmeras, de forma a inserir em imagens objetos virtuais com aparências reais, como se ali estivessem em três dimensões. Já a visão computacional, área da ciência que surgiu no século passado (aliada ao desenvolvimento da inteligência artificial), tem por objetivo estudar essa percepção do mundo pelo cérebro humano através de imagens e modelá-la em forma matemática, realizando o caminho inverso das outras áreas citadas. Seria assim possível obter através de imagens interpretações sobre o que ocorre em três dimensões [1].

Este é um processo custoso e computadores ainda não conseguem realizá-lo com a mesma precisão de uma criança, pois demanda recuperar informações a partir de dados por vezes insuficientes, o que requer técnicas iterativas de otimização e aproximações sucessivas [3]. Porém, diversas aplicações são possíveis com o que se encontra disponível em termos de *hardware* e *software* nos dias atuais, dentre elas: identificar escritas a partir de imagens [4]; inspeção de máquinas e veículos [5]; identificação de padrões para monitoramento [6]; reconstrução de objetos em três dimensões a partir de um conjunto de imagens - *Structure From Motion*, SFM [7]; perseguir objetos em uma cena (*tracking*) [8]; monitoramento patrimonial (segurança contra invasões) [9]; entre outras [1].

Diversos trabalhos e técnicas na literatura propõem a utilização de câmeras para reconstrução do mundo tridimensional em escalas relativas ou reais [10], [11] e [12]. O resultado é um conjunto de pontos em 3D (três dimensões), denominado nuvem de pontos

[13] [14], a partir da qual se tem a noção da cena no mundo real. Propostas para a obtenção dessa nuvem e suas características são diversas, como: a reconstrução monocular [11], [15], envolvendo uma câmera; estéreo [16], envolvendo duas ou mais câmeras posicionadas de forma conhecida [17]; conjuntos de imagem obtidos através de uma câmera calibrada ou não (SFM) [18]; ou outras mais modernas vistas na última década com o uso de câmeras RGB-D (*Red-Green-Blue-Depth*, com informações visuais e de profundidade para cada *pixel*) [19] e aliadas a outros sensores de distância e orientação [20]. Dessa forma é possível mapear e se localizar no ambiente em questão.

Na área da robótica, a visão computacional está aliada diversas vezes à inspeção e localização em um ambiente, onde alguns trabalhos podem ser destacados na literatura [21], [22], [23] e [24]. Uma das aplicações possíveis e com desenvolvimento constante na atualidade é a utilização de câmeras como sensores para localização e mapeamento simultâneo do ambiente, de forma a correlacionar os resultados para obtê-los com maior precisão - do inglês, *Simultaneous Localization and Mapping*, SLAM (introduzido em mais detalhes na Seção 1.1.1). A Odometria Visual (do inglês *Visual Odometry*, VO), um nicho do SLAM, será aplicada neste trabalho, e os métodos envolvidos serão descritos ao longo dos capítulos da dissertação. Ela será aliada a outra aplicação também aqui utilizada, a qual envolve um tipo especial de imagem captado por câmeras térmicas.

O monitoramento térmico infravermelho, ou termografia infravermelha (IRT, "*Infrared Thermography*"), é o estudo de características de objetos segundo sua emissividade no que diz respeito à radiação térmica, na região do espectro infra vermelho, em sua maior parte localizado com comprimento de onda entre 2-14 micrômetros [25]. Essa técnica vem se desenvolvendo desde o final do século passado, e é de importância fundamental para detecção de falhas não vistas a olho nu em diversos setores da engenharia e afins, como na manutenção de equipamentos elétricos [26].

De posse de modelos tridimentionais virtualizados de objetos reais, a atribuição de temperatura aos pontos traria o monitoramento térmico infravermelho a um novo patamar de inspeção, com noção tridimensional do possível defeito existente. A temperatura é aferida de diversos pontos de vista, sendo assim em tese mais fiel à realidade do objeto. Tendo em vista essa proposta de pesquisa, o objetivo deste trabalho é apresentar uma metodologia que proponha técnicas de visão computacional para realizar a odometria visual a partir um par de câmeras estéreo (Seção 4.5) e atribuir a essa nuvem de pontos mapeada (Seção 4.3) a informação de temperatura vinda da câmera térmica, criando assim um modelo 3D térmico.

O monitoramento agiria sobre componentes do setor elétrico em ambientes interno e externo, detectando falhas por extresse térmico, como visto na literatura e discutido na Seção 1.1.

O processamento proposto é realizado somente com o uso de CPU, não requerendo

assim uma GPU dedicada.

A Seção 1.1 discorrerá sobre trabalhos relacionados aos pontos chaves apresentados aqui e no restante do texto, fornecendo embasamento para esta dissertação. Seu foco principal se dá nas aplicações do SLAM e VO para obtenção de mapas com mais qualidade e navegação confiável em abordagens diversas, bem como o monitoramento térmico infravermelho em diversos ambientes, objetos e situações. Ao fim, conclui-se com aplicações onde as duas técnicas são fundidas em um produto final.

1.1 REVISÃO BIBLIOGRÁFICA

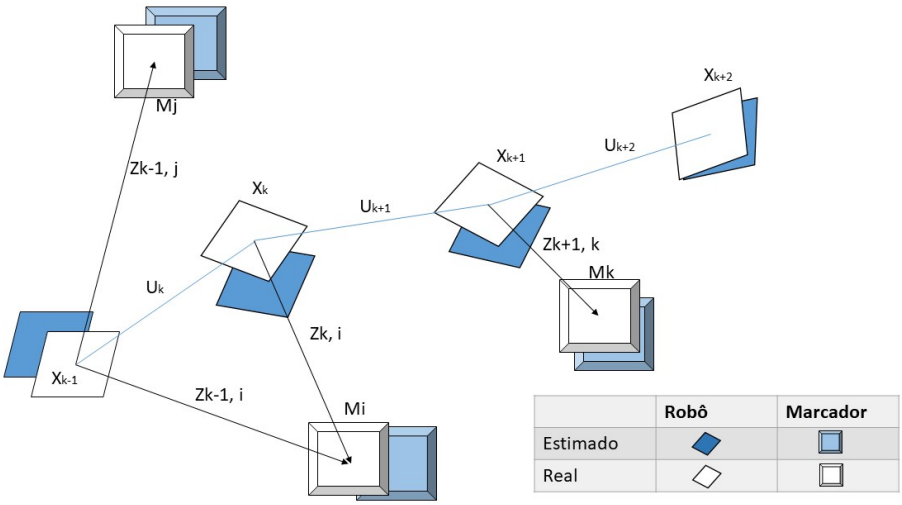
1.1.1 **SLAM e VO**

O início do problema de SLAM ocorreu na década de 80, com uma união de métodos probabilísticos à robótica e inteligência artificial [27]. Segundo Cadena *et al.* [28], ao invés de uma abordagem "cega" (como a odometria de rodas em um robô), o mapeamento de um ambiente é necessário para garantir visão intuitiva de um operador e o planejamento mais seguro do caminho, e de posse de um mapa conhecido a localização fica mais acurada ao comparar as medidas atuais com as conhecidas pelo robô.

O SLAM não é necessariamente realizado com uma câmera: no trabalho de Shen *et al.* [29] um sensor LIDAR (*Light Detection And Ranging*) é utilizado sobre um robô móvel terrestre para realizar o SLAM em ambientes fechados. O advento de melhores recursos de *hardware* (isso incluindo câmeras e processadores), juntamente com a era de pesquisas algorítmicas (2004-2015) [28], garantiram o uso de câmeras aliadas a outros sensores mais baratos como IMU (*Inertial Measurement Unit*, ou unidade de medição inercial), uma vez que mais características do ambiente eram captadas e um mapa mais fiel era obtido.

Para a formulação do problema em um primeiro momento, pontos de controle, ou marcadores, são considerados no mapa. Uma proposta de formulação do problema SLAM seria probabilística, com características Bayesianas [27]. Supondo as variáveis do problema como descritas e ilustradas na Figura 1.1.1:

Figura 3 – Posição do robô e dos marcadores reais e estimadas pelo algoritmo SLAM.



Onde:

- O vetor $\mathbf{x}_k$ contendo orientação e localização do veículo no instante k.
- $\mathbf{u}_k$ o vetor de entradas de controle aplicadas no instante $k-1$ para guiar o robô ao estado k.
- No vetor $\mathbf{m}_i$ está descrita a posição do marcador i, assumida invariante no tempo.
- $\mathbf{z}_{ik}$ seria a medição obtida pela câmera no instante k para o estado e marcadores.

O problema pode ser resumido em determinar os estados, dadas a probabilidade das medições atuais e passadas, além do conhecimento do mapa registrado ao longo da trajetória. Dessa forma fica clara a dependência entre os métodos do robô para a sua localização e o mapeamento simultâneos. Formas clássicas de solução deste problema envolvem a modelagem em espaço de estados dos vetores e a adição de ruídos gaussianos às medições, incitando o uso de um filtro de Kalman estendido; para o caso de considerar o modelo cinemático como não gaussiano, utiliza-se o filtro de partículas Rao-Blackwellized, ou o algoritmo FastSLAM. O leitor interessado em aprofundar sobre o tópico pode consultar as descrições em mais detalhes dos métodos em [27], e novas formas de modelagem e soluções envolvendo técnicas de linearização e otimização em [30].

Um ponto chave no objetivo do SLAM é o fechamento de laço, ou *loop closure*. Graças à noção do mapa e pelo uso de marcadores ou outros sensores como GPS, é possível reconhecer pontos já visitados, e a partir disso corrigir quaisquer desvios existentes por erros nas medições em relação à posição do robô, alcançando uma precisão de menos de 0,5% do comprimento do trajeto, como visto em [31]. Caso não haja esse fechamento, o

robô anda em um “corredor infinito”, sendo então denominada navegação por Odometria Visual (VO, do inglês *Visual Odometry*).

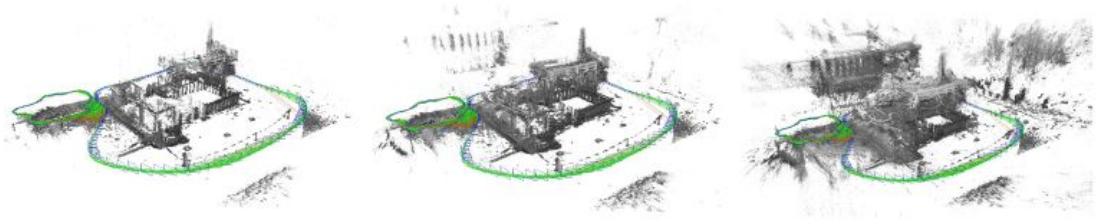
Há porém aplicações em que o mapa não pode ser povoado com marcadores, ou mesmo não vá ser revisitado em momento nenhum ao longo da trajetória, e então a técnica de mapeamento, mesmo que aliada à de VO, não interfere de forma direta na localização do robô. Frente a isso, técnicas modernas de cálculo de odometria a partir de imagens garantem bom mapeamento e localização para movimentos relativamente lentos, se comparados à taxa de aquisição de imagens e processamento das mesmas [28].

O desenvolvimento visto no SLAM nas últimas décadas esteve muito aliado a adventos de *hardware*, e seguindo duas tendências: o uso na robótica para navegação de robôs terrestres e aéreos, como drones [15] [32], e a introdução da realidade aumentada e virtual em produtos no mercado, extrapolando a esfera acadêmica [33].

Em 2006, o trabalho de [34] trouxe uma fusão de sensores de boa qualidade para a época, entre eles GPS, IMU e todo um sistema de navegação inercial, aliado a câmeras calibradas posicionadas sobre um veículo para mapeamento do espaço urbano. Os dados, quando confiáveis, são fundidos por um filtro de Kalman para estimação da posição e orientação da câmera (nomeada pose da câmera). Ao todo oito câmeras estão espalhadas sobre um veículo de forma a obter pouca sobreposição (*overlap*) entre as imagens, mas obtendo satisfatoriamente uma distribuição de 360 graus em torno do automóvel. Os mapas de profundidade calculados são fundidos de forma paralela com auxílio de GPU, obtendo nuvens de pontos e concluindo com um modelo 3D texturizado *online*.

O trabalho de [33] apresenta a técnica LSD-SLAM (*Large-Scale Direct Monocular SLAM*), a qual calcula as poses das câmeras por alinhamentos diretos das imagens, produzindo mapas densos e de larga escala, otimizados frente à estrutura global para melhor conformidade do mapeamento. As técnicas tradicionais de probabilidade são utilizadas para filtrar profundidades ruidosas das cenas. Por fim, o algoritmo é um dos poucos a ser executado de modo *online* com apenas uma CPU, pois a maioria demanda o uso de GPUs para estimativa de profundidades e registro das nuvens. Um resultado final de mapeamento e localização das câmeras se encontra na Figura 4.

Figura 4 – Reconstrução esparça e localização (trilho colorido) a partir de uma câmera monocular com o algoritmo LSD-SLAM.



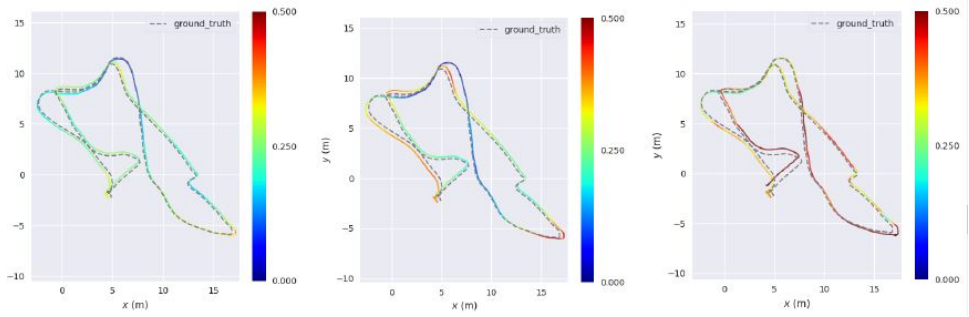
Fonte: [33].

Técnicas mais atuais, como *Kinetic Fusion* [35], Elastic Fusion [36] e ORB-SLAM2 [37] buscam otimizar da melhor forma e reduzir o processamento para melhor fusão (registro) das diversas nuvens de pontos e então gerar os mapas 3D e a pose mais correta da câmera para aquele trajeto mapeado.

Vários trabalhos são encontrados na literatura relacionados a VO, também citadas como VIN (*Visual Inertial Navigation*) ou VIO (*Visual Inertial Odometry*). Neles são apontados alguns casos de fusão de dados com IMU [38], enfatizando a importância da navegação auxiliada por visão e como a capacidade dos recursos de *hardware* atuais possibilitaram a aplicação em tempo real de diversos algoritmos.

Em [39], com somente uma câmera e uma IMU, através de técnicas tradicionais acrescidas de otimização o caminho é estimado. Resultados como o qualitativo ilustrado na Figura 5 demonstram comparações entre métodos diversos, onde o proposto se mostra melhor em relação ao erro médio do caminho considerado real.

Figura 5 – Comparação do algoritmo proposto pelo autor (a esquerda) de forma qualitativa frente a um trajeto proposto em relação a um *groundtruth* e outros algoritmos da literatura.



Fonte: [39].

A fusão das informações com IMU utilizando filtros de Kalman e suas variações para melhores resultados é vista em [40] e [41]. Em [42] uma abordagem com um par de câmeras estéreo é apresentado, assim como modificações no modelo probabilístico do filtro Bayesiano, obtendo melhores resultados em comparação a algoritmos de referência.

1.1.2 **Aplicações envolvendo monitoramento térmico**

No início da prática com equipamentos para IRT, Snell [43] descreve como vários fatores ambientais, dentre eles chuva, vento, umidade e ruídos intrínsecos aos equipamentos dificultavam a medição apropriada feita por pessoal, além de muitas vezes falta de capacitação dos mesmos. Pelo preço e dificuldade de manuseio dos equipamentos, seu uso efetivo foi colocado em questionamento, porém em Jadin [44] já pode ser visto melhorias na tecnologia das câmeras para de forma automática vir a sanar possíveis dificuldades ambientais e relatar dados ao usuário. Além disso, melhores regras para realização das

filmagens e avaliação foram determinadas de forma a combater erros nas medidas que gerariam resultados errôneos [45].

Dois métodos gerais são utilizados hoje em dia para avaliar a temperatura e existência de falhas:

- Método qualitativo: a avaliação é feita pela diferença entre as temperaturas na cena, encontrando zonas de calor ou *hotspots*. Várias aplicações não precisam da avaliação exata da temperatura, mas buscam por falhas em equipamentos que aqueçam uma zona em especial em relação ao resto do mesmo ou um ponto de referência determinado do objeto sujeito a condições semelhantes [46]. A partir de um padrão de avaliação, a técnica ΔT (Equação (1.1)) é executada: um ponto de referência é determinado, assim como sua temperatura aproximada T_1, e comparada a diferença de temperatura com um ponto da zona onde se tem interesse térmico (temperatura T_2). De posse dessa diferença, uma decisão deve ser tomada, como o exemplo da Tabela 1.1.2 para o padrão NETA.

$$\Delta T = T_2 - T_1 \tag{1.1}$$

Tabela 1 – Tabela NETA para tomadas de decisão frente a diferenças de temperatura em componentes.

Prioridade	ΔT entre componentes similares sobre mesma carga	ΔT sobre a temperatura ambiente	Ação recomendada
4	1-3	1-10	Possível defeito, cabe investigação
3	4-15	11-20	Indica possível defeito, reparar quando possível
2	-	21-40	Monitorar até que ações corretivas possam ser tomadas
1	>15	>40	Discrepância maior, reparar imediatamente

- Método quantitativo: nessa avaliação a temperatura é interpretada em seu valor exato absoluto inspecionado. Para uma avaliação mais precisa da radiometria, condições do ambiente, do material constituinte do objeto (em relação à sua emissividade, transmissividade e reflexibilidade) são levadas em consideração na medição. Uma temperatura de referência geral é adotada, normalmente a temperatura ambiente observada pela própria câmera, e a partir dela cada ponto na imagem é comparado, corrigido como dito anteriormente e retornado assim sua temperatura. A relação ΔT em (1.1) é sempre realizada com a temperatura ambiente como T_1.

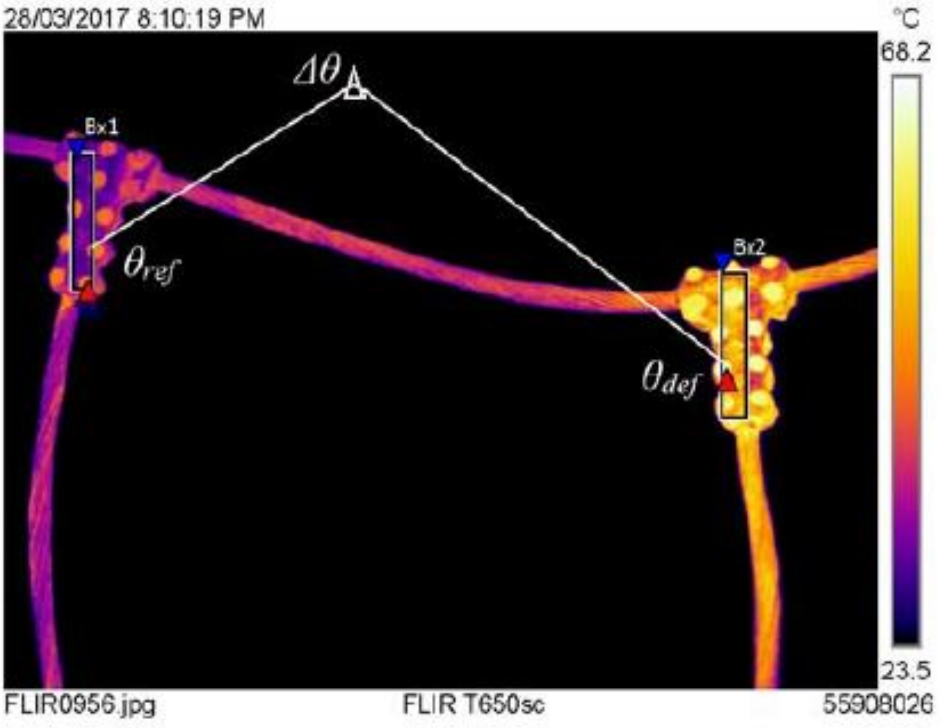
A radiação no espectro visível tem em sua maioria natureza reflexiva, em outras palavras refletindo a onda recebida por uma fonte luminosa, enquanto para o espectro infravermelho a maior parte apresenta natureza emissiva [47]. Isso garante imagens em ambientes com pouco advento de luminosidade, promovendo aplicações como segurança de patrimônio. Wong *et al.* [48] propõe um algoritmo para detecção automática de invasores em uma fábrica ou patrimônio a partir de câmeras térmicas em ambiente externo, algo que a noite não é aplicável para câmeras convencionais. Também é citado e proposto um estudo térmico dos equipamentos da fábrica e da vantagem em se usar a câmera térmica frente a técnicas tradicionais envolvendo testes e outras medições, reduzindo custos e tempo para o processo, que não precisaria ser necessariamente interrompido.

A grande aplicação de câmeras térmicas na engenharia diz respeito à identificação de componentes do sistema elétrico [49]. Fatores como más conexões, desgastes com o tempo e uso intenso ou desbalanceado vem a causar pontos de alta resistência e portanto de estresse térmico [50], o que resulta em ineficiência, falhas no funcionamento e perdas no sistema. A emissão de calor excessivo devido à sobrecarga, falha no sistema de refrigeração ou má condição do equipamento é presente em maior parte no espectro infravermelho, capturado pela câmera.

Cao *et al.* [51] cita a importância do monitoramento térmico em subestações de potência, com foco em causas de problemas em transformadores que geram calor em pontos específicos: correntes parasitas, fluxo magnético dissipado sobre a carcaça, e problemas gerais com o sistema de refrigeração a óleo. O método de avaliação utiliza as técnicas apresentadas como a Equação (1.1) relacionando a porcentagem entre a diferença da temperatura quente contra a ambiente e a parte fria do objeto em questão. Também é proposta uma tabela de tomada de decisão similar à apresentada em 1.1.2. Resultados com diferenças de mais de 15 graus indicaram possível falta de óleo na refrigeração pela carcaça de transformadores de 110 kV.

Dragomir *et al.* [52] foca o trabalho sobre o cabeamento de alta tensão em subestações. Como mencionado anteriormente, cuidados como o tipo de material (e suas propriedades emissivas e reflexivas), as condições ambientais e a carga presente no momento de inspeção são listadas e estudadas para uma melhor medição por parte dos autores, e a avaliação é feita de forma quantitativa, porém utilizando a temperatura de um ponto de referência do equipamento conhecida. Como resultado é apresentada, juntamente com correções e atribuição de carga, uma imagem de inspeção em junções em um cabo (Figura 6), uma boa tomada como referência e outra com elevada temperatura (a princípio defeituosa). A leitura crua indicou uma diferença de 20,7 graus Celsius entre os dois pontos, porém a extrapolação para a corrente nominal dos cabos elevaria o valor a 28,6 graus, o que foi concluído pela tabela apresentada como troca mais rápida possível do equipamento, não ultrapassando um mês.

Figura 6 – Juntas em cabos aferidas com imagens térmicas no monitoramento.



Fonte: [52].

Unindo os conceitos de SLAM e modelos 3D com o monitoramento térmico, o mapeamento 3D térmico é aplicado em trabalhos utilizando câmeras térmicas aliadas à câmeras RGB e RGB-D [53] [54], ou aliando a informação aos dados de sensores de distância e IMU [55]. A tese de Vidas [25] apresenta um *setup* de equipamento composto por uma câmera RGB-D e uma câmera térmica para ser usado manualmente por um operador. O sistema de termografia utiliza características das duas câmeras para realizar o SLAM, e assim mapear de forma *online* objetos de interesse, retornando modelos visuais e térmicos.

Borrman *et al.* [56] propõe o monitoramento 3D térmico automático por meio de um robô móvel, utilizando sensor de distância laser e duas câmeras RGB e térmica montados no robô. Os autores apresentam um método de navegação para a plataforma baseado em reconhecimento do ambiente, porém a aplicação é voltada para a reconstrução completa de um local fechado. Resultados mostram salas reconstruídas em escala de cor referente à temperatura medida.

1.2 DESAFIO E ORGANIZAÇÃO DO TRABALHO

Os trabalhos encontrados na literatura envolvendo modelos 3D térmicos focam em sua maioria aplicações em ambientes internos, com utilização de sensores de distância para melhor qualidade, mesmo em ambientes controlados. O desafio proposto para esta dissertação consiste na obtenção do modelo exclusivamente a partir de câmeras, a medida

que a navegação ocorre, com foco no objeto durante inspeção. Além disso, a aplicação da metodologia pode ser realizada em ambientes externos.

Para apresentar a base teórica, metodologia aplicada e resultados finais, a sequência deste trabalho se dá da seguinte forma:

- O Capítulo 2 introduz como é percebida a imagem e os sensores utilizados para captá-la (no caso as câmeras digital e térmica), modela a câmera matematicamente e introduz como é feita a identificação de características dentro da imagem.
- No Capítulo 3 a matemática utilizada para localização em um ambiente é desenvolvida para acompanhar a pose das câmeras e do modelo 3D, a movimentação e relação entre eles.
- O Capítulo 4 traz a teoria matemática envolvida no processo de transferência entre o mundo 2D para o 3D, criando modelos tridimensionais a partir de imagens. Também são apresentados os processos de calibração das câmeras visuais e térmica, a fim de obter melhor acurácia nesse procedimento. Por fim aborda a matemática básica para cálculo da odometria visual.
- O Capítulo 5 descreve a metodologia construida para o sistema: como foram aplicadas as técnicas teóricas para realização da odometria visual e obtenção da nuvem de pontos através do par de câmeras estéreo, bem como o registro das nuvens e a adição de informações térmicas às mesmas.
- Os resultados são apresentados no Capítulo 6, com exemplos de reconstruções e odometria visual frente a diversos cenários.
- Por fim, o Capítulo 7 mostra as conclusões do trabalho e trabalhos futuros.

2 CONCEITOS DE IMAGEM EM VISÃO COMPUTACIONAL

Neste capítulo serão apresentados o funcionamento das câmeras utilizadas, o modelo utilizado para relacioná-las ao mundo real no processo de formação da imagem, bem como formas de descrever pontos de interesse nas mesmas para processamento futuro.

2.1 IMAGENS

Imagens podem ser interpretadas como a transformação de uma cena em três dimensões para um plano de duas dimensões. Para tal transformação, a geometria de formação de imagens necessita de um modelo adotado, os quais foram desenvolvidos durante séculos para diversos objetivos [7].

Com início no século IV D.C. e retomada no período do Renascimento, artistas desenvolveram modelos geométricos para retratar a perspectiva de paisagens em três dimensões em suas telas, assim como formas de atribuir cor e iluminação a cada ponto das mesmas. Formas e traçados adotados por engenheiros e arquitetos da época influenciaram a consolidação de modelos a serem adotados [7].

Para as técnicas de visão computacional utilizadas neste trabalho, a imagem será interpretada como uma matriz de duas dimensões, onde cada entidade possui uma cor e brilho [7]. Define-se um mapa I, em uma região compacta de duas dimensões Ω pertencente ao domínio dos números reais, com valores positivos, onde I é a função definida pela Equação (2.1).

$$I : \Omega \subset R^2 \rightarrow R_+; (x, y) \rightarrow I(x, y) \tag{2.1}$$

2.2 CÂMERAS DIGITAIS

2.2.1 **Câmeras Visuais**

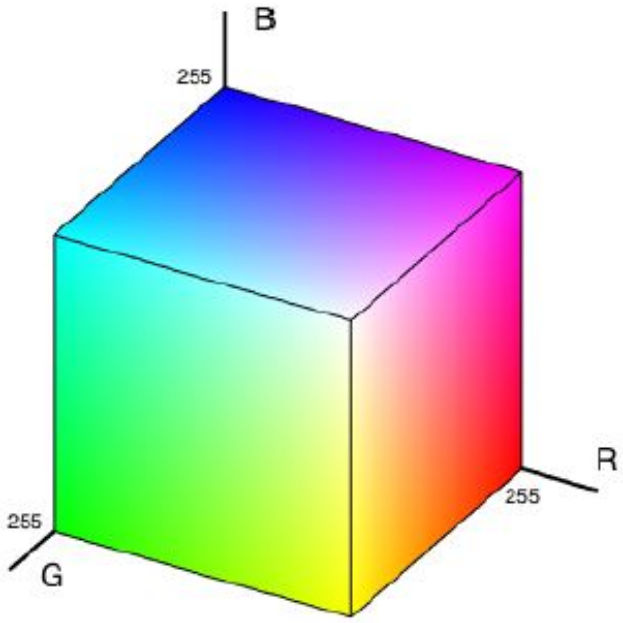
As câmeras digitais têm sido a tecnologia dominante no mercado nos dias atuais. A luz emitida pelas entidades da cena é capturada através da lente e armazenada em uma matriz de sensores, onde cada sensor converte os fótons recebidos em elétrons. Os dois tipos mais comuns são sensores CCD (*Charge Coupled Device*) e CMOS(*Complementary Metal Oxide Semiconductor*) [57].

Esse modelo de captura torna o espaço Ω definido na Equação (2.1) um plano matricial discreto, onde cada componente da matriz é chamado elemento pictório ou *pixel*. A quantidade de *pixels* existente na matriz resulta na resolução da imagem, a qual permite capturar a mesma cena em mais ou menos detalhes (Figura 7), fato crucial para a efetividade das técnicas de visão computacional.

Figura 7 – A mesma cena capturada com uma relosução de 600x800 à esquerda e 300x400 à direita.

Para o sistema computacional, cada *pixel* tem sua cor representada por *bytes*, em específico para este trabalho no espaço RGB (Figura 8). Também chamado de sistema cor-luz, ele basicamente combina o nível de vermelho, verde e azul encontrado em cada ponto no ambiente para resultar em outras cores diversas [58]. A matriz resultante possui então três dimensões, sendo sua largura e altura de acordo com a resolução da câmera para cada uma das cores, com cada elemento quantificado por um *byte*.

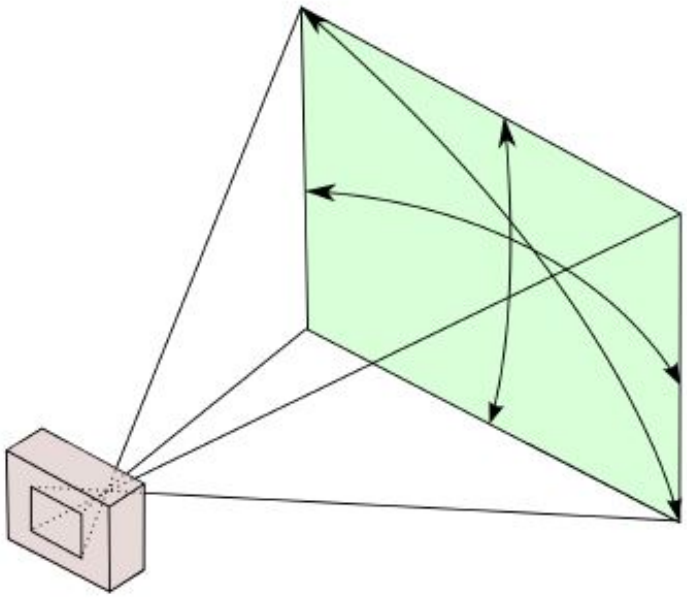
Figura 8 – Sistema de cores RGB com valores de 0 a 255 representados por cada *byte*.



Além da resolução mencionada previamente, câmeras digitais possuem abertura angular e *field of view* (campo de visão). Ao considerar o mundo visto pela câmera como uma esfera, a abertura angular diz respeito à sessão dessa esfera em que os raios são capturados, e a região capturada seria o campo de visão [7]. Na Figura 9, a região em

verde é o campo de visão, enquanto as linhas de raios que ligam as suas extremidades à câmera delimitam a abertura angular.

Figura 9 – Campo de visão e abertura angular de uma câmera digital.

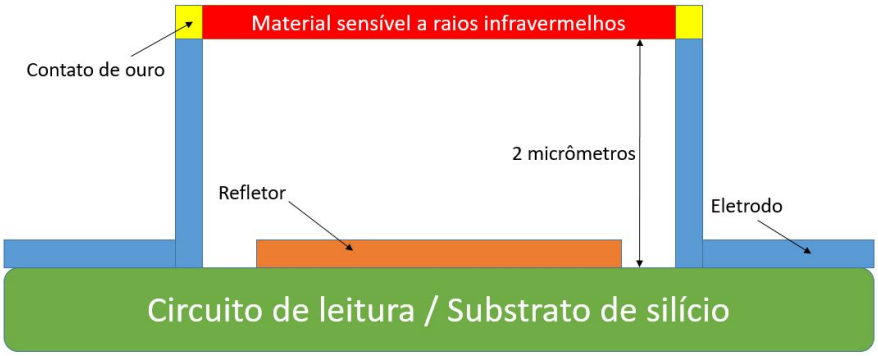


2.2.2 **Câmeras Térmicas**

As câmeras térmicas funcionam com princípios de captação de radiação semelhantes aos das câmeras digitais, porém utilizam sensores diferentes para obter informações em outra faixa do espectro luminoso. Câmeras comerciais em sua maioria, que não requerem refrigeração, são compostas por microbolômetros, elementos sensíveis à radiação de longos comprimentos de onda na região do infravermelho [59].

Os microbolômetros (ilustração na Figura 10) formam cada *pixel* da imagem, e são compostos por um material absorvente para radiação infravermelha e um refletor para maximizar essa absorção. Variações de resistividade desse material com a temperatura geram tensão de saída absorvida pelo sensor de saída. O nome dado à região onde esses microbolômetros se encontram arranjados de forma a maximizar a recepção através da lente da câmera é *Focal Plane Array* (FPA).

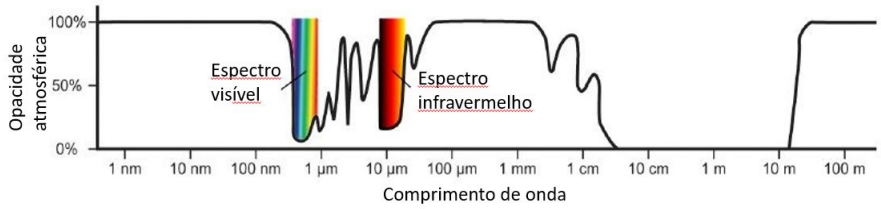
Figura 10 – Esquema de construção de um microbolômetro.



A maioria das câmeras térmicas contêm um termistor ou outro tipo de sensor de temperatura para medição interna, o que é compensado na leitura final do sensor. As lentes das câmeras devem ser feitas de material especial, como o germânio, pois o vidro é opaco à transmissão de radiação infravermelha [25].

A radiação a ser captada se encontra na faixa ilustrada na Figura 11. Nessa faixa de radiação encontra-se a maior parte do calor emanado devido à temperatura dos objetos em torno de 300 graus Kelvin [25].

Figura 11 – Espectro da luz, com a porcentagem de opacidade promovida pela atmosfera na vertical e o comprimento de onda na horizontal. Em destaque onde se localizam o espectro visível e o infravermelho mais relevante para aplicações.



Fonte: [59].

Uma vez recebida radiação por um objeto, a mesma se distribui em emissividade, reflectividade, absorvidade e transmissividade. Um objeto está em equilíbrio térmico quando sua emissividade se iguala à absorvidade. Para objetos opacos, com exceção de placas finas, a transmissividade pode ser desconsiderada, e ao aproximá-lo por um corpo negro, também a reflectividade.

Para calcular a emissividade de um objeto em $Watts/m^2$, aproximando-o por um corpo negro, e relacioná-la à temperatura do mesmo em Kelvin, a equação de Stefan-Boltzmann (como definida na Equação (2.2)), pode ser utilizada. Uma vez medida essa

emissividade e desconsiderando fatores do meio e do próprio material, o inverso das operações resultaria na temperatura do objeto.

$$j\star = \sigma T^4 \tag{2.2}$$

onde σ é a constante de Stefan-Boltzmann com valor de $5,670373 \cdot 10^{-8} W/m^2$, T a temperatura em Kelvin e $j\star$ a emissividade (radiância) do objeto.

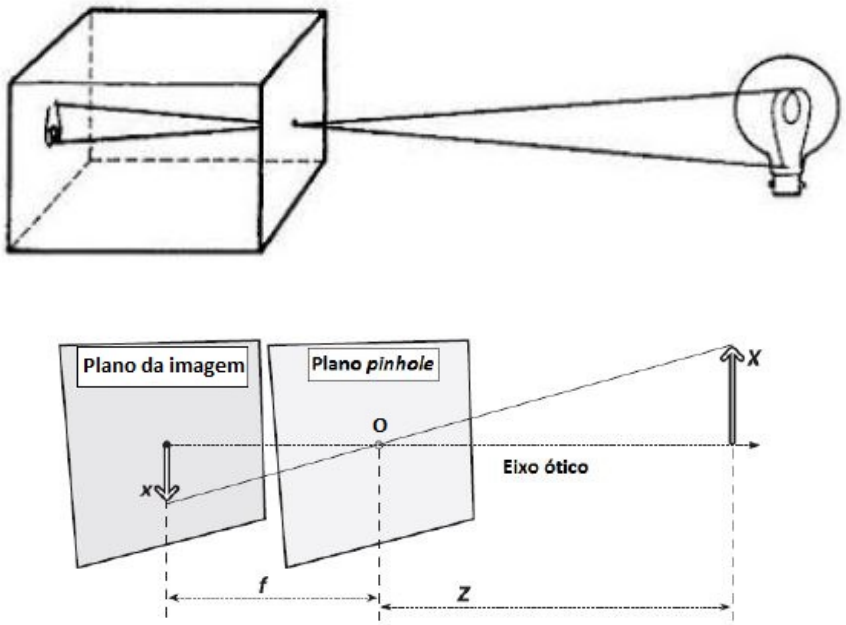
Outros fatores devem ser levados em consideração para obter melhor qualidade de leitura [60]. Como apresentado em [61], a calibração radiométrica de uma câmera térmica deve ser realizada para interpretar de forma acurada as leituras dos sensores como temperatura. Caso contrário, a câmera somente funciona de forma qualitativa para diferenciar partes da cena.

2.3 MODELO *PINHOLE* DE CÂMERAS

Para descrever como uma imagem é formada na matriz de sensores de uma câmera a partir de um objeto ou uma cena foi utilizado o modelo *pinhole* (orifício) [7]. Este é um modelo vastamente usado em visão computacional, sendo simples e acurado para diversas aplicações.

O nome deriva de um formato rudimentar de câmera, como uma câmera obscura, onde um recipiente escuro coleta a luz por um pequeno orifício em uma das suas faces, e forma a imagem na face oposta. No modelo, a luz passa por um único ponto O antes de se projetar no plano da imagem [62], que no caso da câmera digital é a matriz de sensores. A Figura 12 ilustra o modelo com a projeção de uma lâmpada no plano da imagem:

Figura 12 – Câmera que origina o modelo *pinhole*.



onde o *pinhole* O é chamado centro ótico; a linha deste ponto até o plano da imagem de forma perpendicular é o eixo principal, ou eixo ótico, e o ponto que esse eixo encontra o plano da imagem é o ponto principal [3].

Como visto na Figura 12, a geometria resultante da formação da imagem deriva das dimensões da câmera. A letra f representa a distância focal da câmera, propriedade fundamental para o relacionamento da imagem com o mundo em três dimensões [62]. A Equação (2.3) define a relação geométrica entre um ponto de coordenadas $\mathbf{x}$ a partir do eixo ótico e o ponto no mundo real.

$$\mathbf{x} = \frac{f \cdot \mathbf{X}}{Z} \tag{2.3}$$

Em 2.3, $\mathbf{X}$ são as dimensões do objeto em valores reais em relação ao eixo ótico. Dessa forma, de conhecimento do valor da distância focal f e de $\mathbf{x}$, todas as distâncias podem ser medidas por uma câmera, uma vez identificado um objeto de tamanho conhecido na imagem, ou em escala. Isso é um princípio fundamental obtido na calibração da câmera, descrita na Seção 4.4, e para a localização no mundo com odometria visual, explicada em mais detalhes na Seção 4.5.

2.4 DESCRIÇÃO E IDENTIFICAÇÃO DE UMA IMAGEM

Esta seção descreverá como, uma vez de posse da imagem, identificar pontos que a descrevam e calcular parâmetros para esses pontos, de forma a poder comparar imagens e

obter dados dessa comparação. Esse processo é fundamental para o reconhecimento da cena e de objetos [63].

2.4.1 Pontos de interesse - *Features*

O reconhecimento de pontos de interesse, ou pontos distintos em uma imagem, pode ser retomado a 1981 [64], quando Moraveck utilizou o reconhecimento de cantos (*corners*) na imagem para comparar o que era visto por um par de câmeras de visão estéreo. Estes pontos são normalmente *corners*, junções T ou *blobs*, e a característica mais importante que podem retornar é a repetibilidade, de tal forma que o mesmo seja encontrado em diferentes pontos de vista e condições de iluminação [65].

Um dos trabalhos mais consagrados na área foi o realizado por Harris [66], que ao longo dos anos também foi chamado de detector de *corners*, porém utiliza técnicas matemáticas para identificar locais com gradientes relevantes em todas as direções em uma determinada escala de análise. Primeiramente, a técnica foi empregada na comparação de imagens com pequenas disparidades, como em câmeras estéreo e movimentos de curto alcance [63]. Posteriormente, o trabalho de [67] mostrou resultados com bancos de imagem relativamente grandes e pontos de vista variados.

Aliados a técnicas de geometria de duas vistas [3], as imagens conseguem ser comparadas e as cenas reconstruídas. O método será explicado a seguir, na forma como é empregado computacionalmente, segundo Solem [62].

Para descobrir as taxas de variação de uma função utiliza-se a operação de derivação, ou diferenciação. Ao considerar a imagem $\mathbf{I}$ como uma função bidimensional discreta, é possível diferenciá-la em cada uma das suas dimensões, resultando a matriz $\mathbf{I}_k$ ao convoluir uma matriz $\mathbf{D}_k$ sobre essa função, onde k está para x ou y, como mostra a Equação (2.4):

$$\mathbf{I}_k = \mathbf{I} * \mathbf{D}_k \tag{2.4a}$$

$$\mathbf{D}_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \mathbf{D}_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.4b}$$

De posse das matrizes $\mathbf{I}_x$ e $\mathbf{I}_y$, define-se o vetor gradiente de imagem $\nabla\mathbf{I} = \begin{bmatrix} \mathbf{I}_x \\ \mathbf{I}_y \end{bmatrix}$ e a matriz positiva, semi-definida e simétrica $\mathbf{M}_I$, como na Equação (2.5). A matriz $\mathbf{M}_I$ é chamada matriz Hessiana.

$$\mathbf{M}_I = \nabla\mathbf{I}\nabla\mathbf{I}^T = \begin{bmatrix} \mathbf{I}_x^2 & \mathbf{I}_x \cdot \mathbf{I}_y \\ \mathbf{I}_y \cdot \mathbf{I}_x & \mathbf{I}_y^2 \end{bmatrix} \tag{2.5}$$

Dessa forma, cada ponto possui uma matriz $\mathbf{M}_I$, que por sua forma construtiva possui ranque 1 e autovalores $\lambda_1 = |\nabla \mathbf{I}^2|$ e $\lambda_2 = 0$. Para buscar por variações nesse ponto da imagem, portanto, utiliza-se um filtro tipicamente gaussiano $\mathbf{G}_\sigma$ (Equação (2.6)) de forma a obter novos autovalores que indiquem a taxa de variação entre o ponto e sua vizinhança.

$$\overline{\mathbf{M}_I} = \mathbf{G}_\sigma * \mathbf{M}_I \tag{2.6}$$

A depender dos autovalores da matriz Hessiana, três situações podem ocorrer com o ponto em questão:

- Se λ_1 e λ_2 são relativamente altos, existe um ponto de interesse.
- Se λ_1 for alto e λ_2 for baixo ou o contrário, este ponto pertence a uma região de aresta ou traço na imagem.
- Se ambos autovalores são baixos, esse ponto não apresenta nada, provavelmente um plano.

Para distinguir a primeira situação sem necessariamente calcular os autovalores, Harris [66] introduziu um método como apresentado na Equação (2.7).

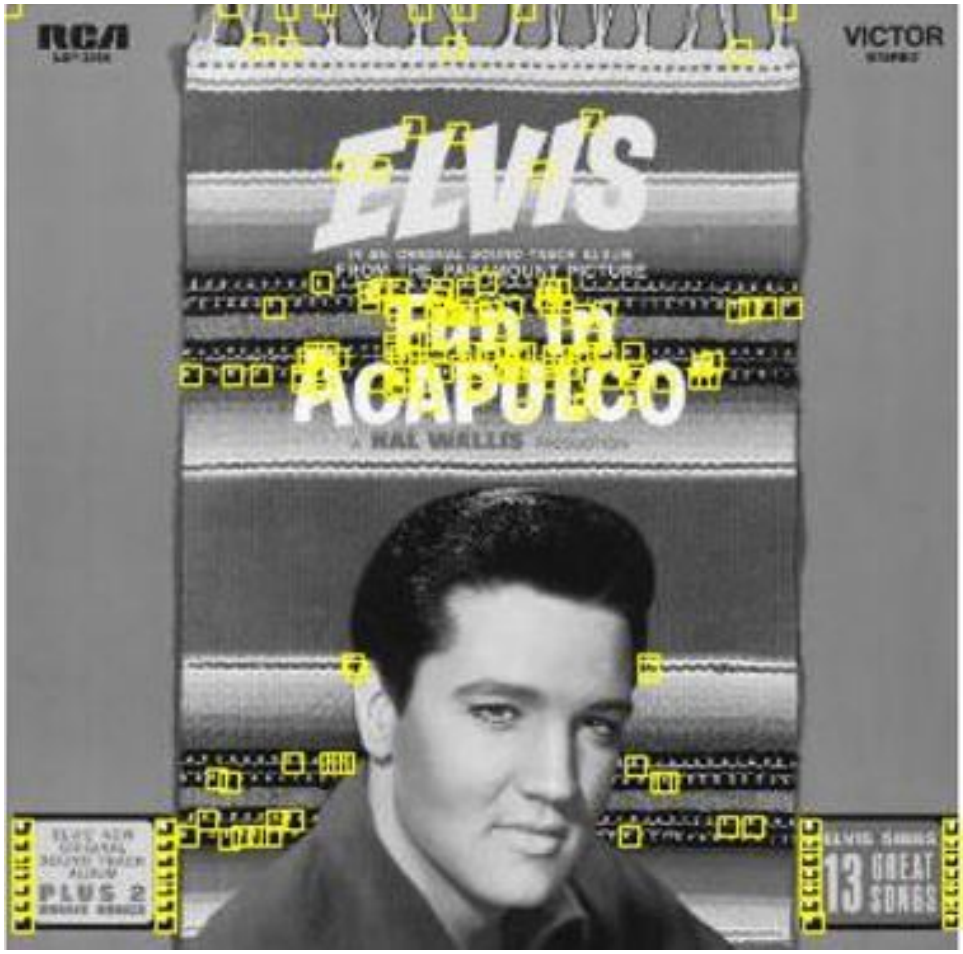
$$det\left(\overline{\mathbf{M}_I}\right) - k \cdot traço\left(\overline{\mathbf{M}_I}\right)^2 \tag{2.7}$$

Ao aplicar sobre os *pixels* da imagem e suas respectivas vizinhanças a operação 2.7, se o resultado ultrapassar um limite estabelecido, naquele *pixel* se encontra um ponto de interesse ou *corner*. Para se livrar do parâmetro k, um método alternativo é apresentado na Equação (2.8).

$$\frac{det\left(\overline{\mathbf{M}_I}\right)}{traço\left(\overline{\mathbf{M}_I}\right)^2} \tag{2.8}$$

Uma ilustração de pontos encontrados em uma imagem está na Figura 13. De posse de pontos de interesse distintos e confiáveis, uma forma de descrevê-los e compará-los se faz necessária, e será discutida em mais detalhes na subseção seguinte.

Figura 13 – *Corners* encontrados a partir da aplicação da metodologia apresentada e resultado da Equação 2.8.



2.4.2 **Descritores e *match* de pontos**

A comparação em busca de semelhanças (*match*) de imagens é um aspecto fundamental na visão computacional, seja para solucionar estruturas em 3D a partir de um banco de imagens, comparação de pares de imagens a partir de câmeras estéreo, ou como no caso deste trabalho: a reconstrução de um objeto ou uma cena em três dimensões e odometria visual [63].

Uma vez identificados pontos distintos em uma imagem, diversos trabalhos foram dedicados a descrever esses pontos com base em sua vizinhança [68] [69] [70], criando assim descritores, para a partir de tal informação comparar e identificar o mesmo ponto em duas ou mais imagens (*match* de pontos).

Descritores são então definidos como vetores designados a um ponto de interesse que descrevem a aparência da imagem ao redor do último. Quanto melhor o descritor, mais fiéis serão as correspondências em imagens diferentes. Como previamente introduzido, as correspondências ocorrem entre pontos em duas imagens diferentes que representam o mesmo local no mundo real.

Uma das primeiras abordagens para criação de descritores envolve os *corners* identificados conforme Subseção 2.4.1. Como descrito por Solem [62], a partir de uma matriz centralizada no *pixel* referente ao *corner*, pode-se realizar a correlação cruzada com uma matriz de mesma dimensão em torno de todos os *corners* de uma outra imagem; o maior valor obtido dessas operações tende a ser um *match* correto entre as imagens.

Supondo duas imagens $\mathbf{I}_1$ e $\mathbf{I}_2$, e uma matriz que englobe pontos $\mathbf{x}$ vizinhos ao ponto de interesse em cada imagem. A operação de correlação pode ser definida como a Equação (2.9):

$$c(\mathbf{I}_1, \mathbf{I}_2) = \sum_{\mathbf{x}} f\left(\mathbf{I}_1(\mathbf{x}), \mathbf{I}_2(\mathbf{x})\right) \tag{2.9}$$

onde f é uma função que varia segundo o método de correlação. Para a correlação cruzada, tem-se f como o produto escalar entre os termos, portanto entre os componentes $\mathbf{x}$ das duas janelas: $c(I_1, I_2) = I_1 \cdot I_2$.

Mais pesquisas demonstraram que, ao utilizar a operação de correlação cruzada normalizada, a comparação se torna menos sensível a variações gerais de iluminação. Essa operação está descrita na Equação (2.10):

$$ncc(\mathbf{I}_1, \mathbf{I}_2) = \frac{1}{n-1} \sum_{\mathbf{x}} \frac{\mathbf{I}_1(\mathbf{x}) - \mu_1}{\sigma_1} \cdot \frac{\mathbf{I}_2(\mathbf{x}) - \mu_2}{\sigma_2} \tag{2.10}$$

onde μ são as médias e σ os desvios padrão dentro da vizinhança selecionada em cada imagem.

Mesmo sendo uma abordagem simples e portanto computacionalmente eficiente, essa forma de descrição e comparação de pontos tem desvantagens que a torna inviável para diversas aplicações, as quais apresentam rotações e diferenças em escala entre as imagens comparadas. O método não consegue obter *matches* corretos em ambas as situações satisfatoriamente [63].

Um dos trabalhos mais marcantes e que definiu um patamar para essa área da visão computacional foi publicado por Lowe [63], no qual foi apresentado o descritor SIFT (do inglês, *Scale Invariant Feature Transform*) invariante à iluminação, escala e rotação, ao mesmo tempo sendo relativamente eficiente na comparação de pontos.

O primeiro passo para analisar a imagem e encontrar pontos de interesse parte do trabalho anterior do próprio autor [71], baseado no fato de que à época a melhor operação sobre imagens para encontrar tais características seria obter o Laplaciano de uma operação Gaussiana. Para tornar o processo mais eficiente computacionalmente, porém, a operação foi resumida à diferença de Gaussianas tomadas em diversas escalas.

Suponha uma matriz com um filtro gaussiano semelhante à presente em (2.6),

porém com a escala variável, como definida agora em (2.11):

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2.11}$$

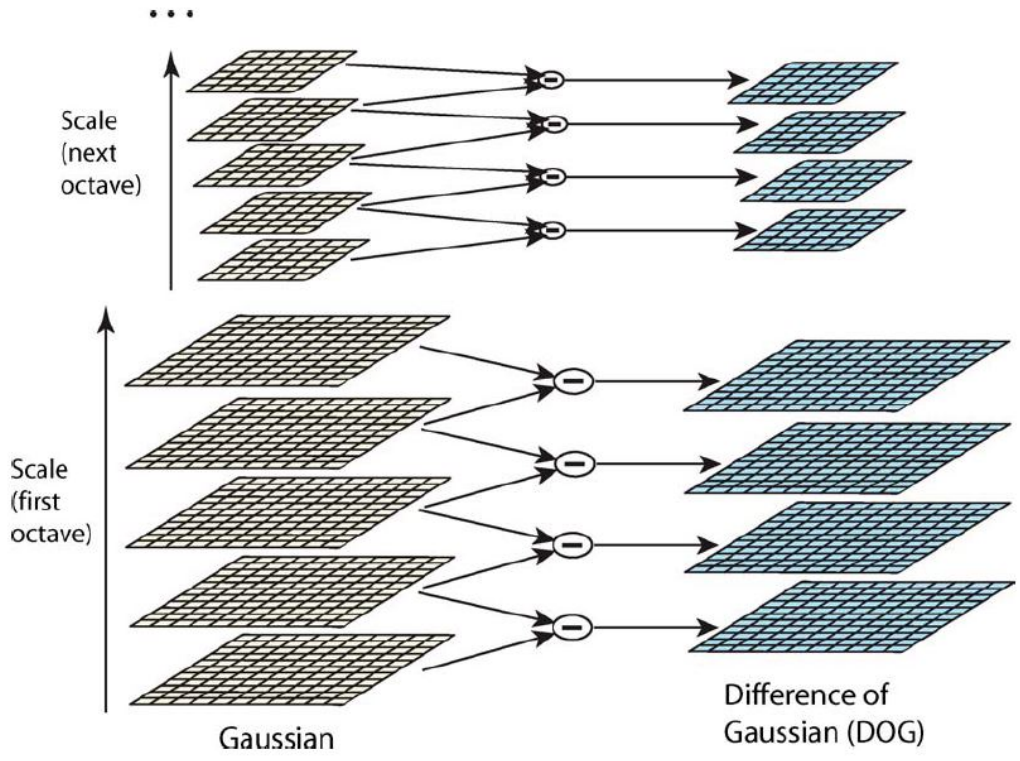
Para obter uma imagem filtrada é necessário convoluir a mesma com a matriz $\mathbf{G}$. Seguidas convoluções, variando o valor de σ, resultam no espaço de escalas da imagem. Na Equação (2.12) define-se a convolução, e em (2.13) a diferença de Gaussianas, onde máximos e mínimos indicam pontos de interesse no *pixel* em questão.

$$\mathbf{L}(x, y, \sigma) = \mathbf{G}(x, y, \sigma) * \mathbf{I}(x, y) \tag{2.12}$$

$$\mathbf{D}(x, y, \sigma) = (\mathbf{G}(x, y, k\sigma) - \mathbf{G}(x, y, \sigma)) * \mathbf{I}(x, y) = \mathbf{L}(x, y, k\sigma) - \mathbf{L}(x, y, \sigma) \tag{2.13}$$

A cada vez em que o valor de $k = 2$, a imagem original é redimensionada e novamente a sequência de convoluções é iniciada. A cada novo início tem-se uma oitava do espaço de escalas, como ilustrado na Figura 14.

Figura 14 – Sequência de redimensionamentos para encontrar *features* em diversas escalas, chamadas oitavas.

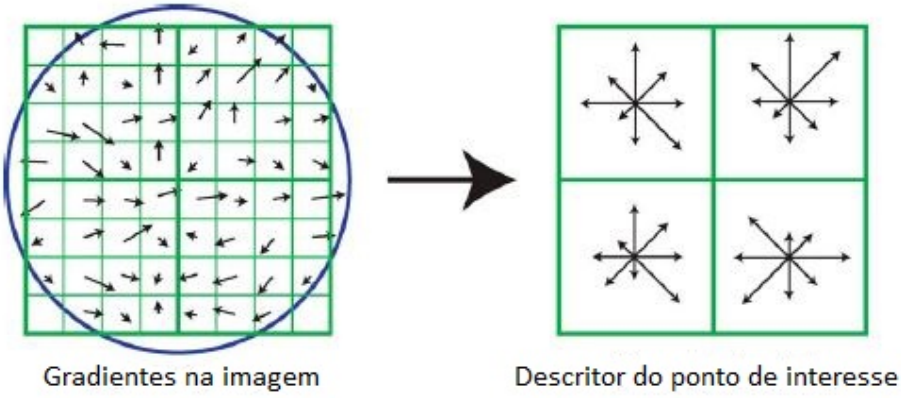


Fonte: [63].

Para solucionar o problema de variação perante rotações entre duas imagens, toma-se a imagem filtrada em determinada escala, e a partir daí são calculados os gradientes em torno do ponto em si e de seus vizinhos na região de entorno. O resultado é um histograma de 36 classes para os 360 graus ao redor do *pixel* do ponto de interesse. A classe dominante define ao final a orientação presente no descritor daquele ponto.

Para formar o descritor com as características da imagem no entorno do ponto de interesse, novamente uma alternativa à correlação cruzada normalizada é apresentada relacionada a histogramas: gradientes são tomados em uma região de 16x16 *pixels* no entorno do *pixel* central, filtrados por uma gaussiana para dar mais valor aos *pixels* centrais, e organizados em histogramas a cada 4x4 (como na Figura 15, que exemplifica para uma região de 8x8). Há 8 sessões dentro de cada histograma, portanto o trabalho conclui que a melhor opção utiliza 4x4x8=128 elementos para construir cada descritor.

Figura 15 – Gradientes obtidos em uma janela à esquerda e o descritor com histogramas em 4 vizinhanças à direita para identificar a janela.



Fonte: [63].

Na etapa de *match* de pontos, a métrica utilizada é a distância Euclidiana entre os vetores dos descritores, levando em consideração a escala e orientação em que foram encontrados. A decisão é tomada pelo método do vizinho mais próximo, acrescida de comparações com o segundo mais próximo para reforçar o fato de corresponder àquele ponto.

A busca pelos descritores é feita pelo método BBF(*Best-Bin-First*), que aumenta a velocidade no espaço de busca resultante. Para aceitar um *match* como correto, em geral é considerada a distância ao vizinho mais próximo menor que 0,8 vezes a distância ao segundo vizinho mais próximo.

Baseado nos princípios gerais adotados nesse algoritmo, diversos trabalhos foram realizados com o intuito de melhorar o balanço entre eficiência na comparação e custo computacional. Algortimos como GLOH são baseados nesses princípios, apresentando variantes que o tornam mais eficiente para *match* de pontos, porém com processamento um

pouco mais pesado [72]. Trabalhos relevantes, como os algoritmos SURF [65] e BRIEF [73], obtiveram resultados com custo computacional consideravelmente reduzido por simplificar diversas etapas introduzidas no SIFT, e são preferidos para aplicações em tempo real.

Como é o caso desse trabalho processar a comparação de imagens de forma *online* e com movimentos relativamente curtos entre imagens (*frames*) sequenciais, descritores simples nesse sentido serão aplicados, como será apresentado na Seção 4.5.

Uma vez que a forma básica e o objetivo de identificar características em uma imagem foi discutida e apresentada, a sequência do trabalho tratará de como descrever a posição 3D tanto da câmera quanto do objeto, e a partir de cálculos e teoria geométrica relacionar pontos na imagem às suas posições em 3D no mundo real, e vice-versa.

3 FERRAMENTAL MATEMÁTICO PARA LOCALIZAÇÃO ESPACIAL

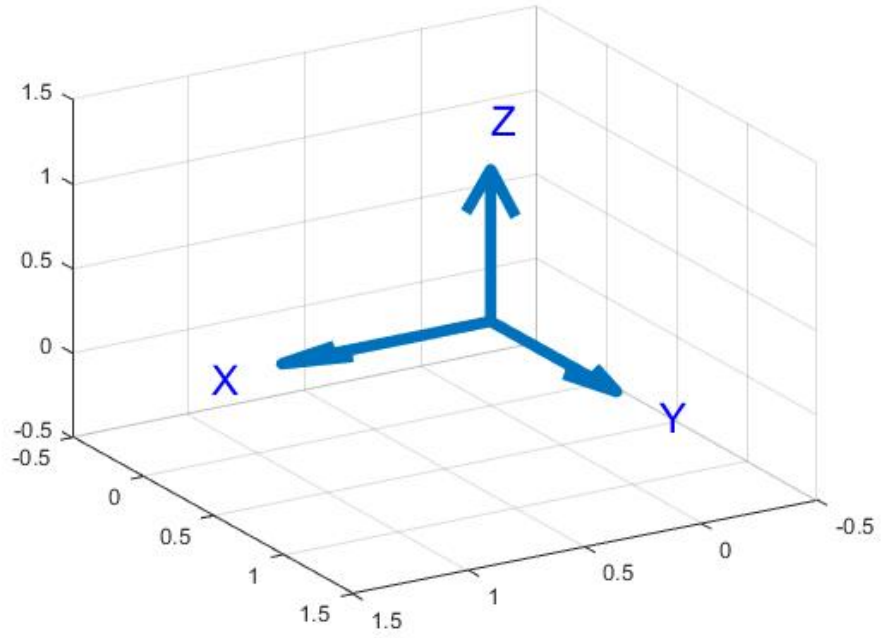
Este capítulo descreverá os fundamentos para a localização da câmera em relação à cena e ao objeto, assim como seu deslocamento ao longo do tempo, baseado nas definições encontradas em Corke [2]. Isso é primordial para o funcionamento da odometria visual, e por consequente para a reconstrução do objeto final em três dimensões. Como conclusão, todo deslocamento espacial deste trabalho está relacionado ao conceito de transformações homogêneas envolvendo rotações e translações aqui descrito.

3.1 *FRAMES* COORDENADOS E TRANSFORMAÇÕES

Uma necessidade fundamental na visão computacional aplicada à reconstrução 3D é a localização no ambiente existente. Um ponto pode ser facilmente representado por um vetor de três coordenadas relativo a uma origem dos espaços. Ao inserir uma câmera no ambiente, também é possível atribuir a ela uma posição e orientação, bem como relacioná-la ao ponto. Para isso são aplicados os *frames* coordenados em cada uma dessas referências.

O *frame* coordenado é um conjunto de eixos ortogonais entre si, com uma origem em comum, como ilustrado na Figura 16 para três dimensões com os eixos XYZ. À origem dos espaços na cena é atribuido um *frame*, ou um conjunto de eixos coordenados.

Figura 16 – Exemplo de *frame* com eixos coordenados XYZ ortogonais e origem definindo o ponto (0, 0, 0) da cena.

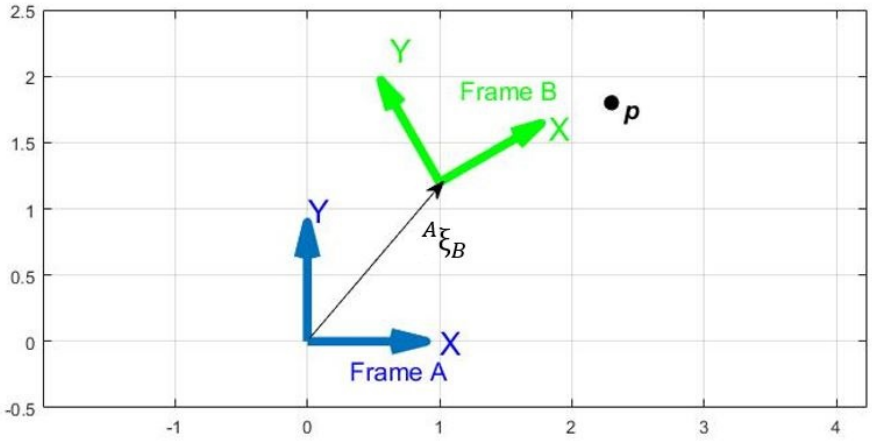


A posição designada e a orientação de um *frame* é chamada de pose, e está sempre

relacionada a um outro sistema de coordenadas conhecido. A pose relativa entre dois *frames* é definida pela letra ξ, também chamada de transformação homogênea.

Supondo dois *frames* A e B de duas dimensões XY, como ilustrados na Figura 17, um mesmo ponto $\mathbf{p}$ no espaço pode ser representado com coordenadas em respeito a ambos, e a relação entre elas se dá como na Equação (3.1).

Figura 17 – Ponto $\mathbf{p}$ existe para dois *frames* A e B, relacionados por uma transformação ${}^{A}\xi_{B}$.



$$ {}^{A}\mathbf{p} = {}^{A}\xi_{B} \cdot {}^{B}\mathbf{p} \tag{3.1} $$

onde um ponto em B é levado para o *frame* A.

As transformações podem ser invertidas e compostas para relacionar mais de um *frame* definido no ambiente, sendo o resultado disso exemplificado na Equação (3.2) para os *frames* A, B e C.

$$ {}^{A}\xi_{C} = {}^{A}\xi_{B} \bigoplus {}^{B}\xi_{C} \tag{3.2a} $$

$$ {}^{A}\mathbf{p} = \left({}^{A}\xi_{B} \bigoplus {}^{B}\xi_{C} \right) \cdot {}^{C}\mathbf{p} \tag{3.2b} $$

onde o operador $\bigoplus$ representa a composição de transformações homogêneas.

Assim, um ponto ${}^{C}\mathbf{p}$ descrito em coordenadas do *frame* C pode ser transformado em sequência até ser representado em coordenadas do *frame* A, e assim por diante.

Para compor essa transformação, duas operações podem ser realizadas sobre o *frame* existente, que são a translação entre os *frames* e a rotação em torno dos eixos coordenados. As seções seguintes discorrerão sobre a formulação envolvida nas transformações.

3.2 ROTAÇÃO E TRANSLAÇÃO PARA DUAS E TRÊS DIMENSÕES ESPACIAIS

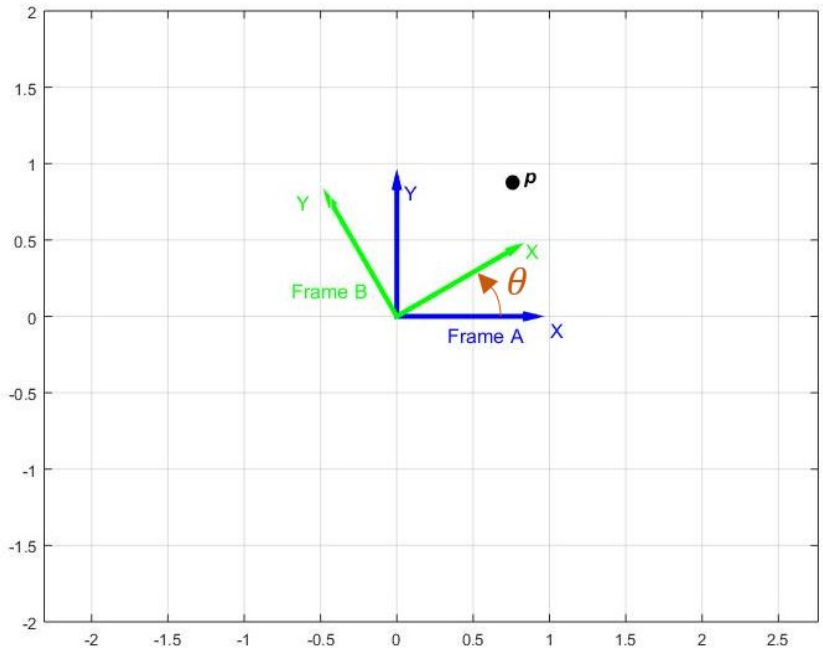
Supondo um espaço cartesiano de duas dimensões, com os eixos XY ortogonais sendo um *frame* de origem A. Para representar o ponto ${}^{A}\mathbf{p}$ são necessárias suas coordenadas x e y multiplicando os respectivos vetores unitários de cada eixo, como definido em (3.3):

$$\mathbf{p} = x \cdot \hat{\mathbf{x}} + y \cdot \hat{\mathbf{y}} = \begin{pmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.3}$$

Pode-se notar pela Figura 17 que o *frame* B está deslocado em relação à origem do *frame* A, assim como rotacionado. Dessa forma, a transformação envolvida entre os dois é composta pelo vetor ${}^{A}\xi_{B}\ (x, y, \theta)$, que escrita dessa forma envolve operações trigonométricas compostas. Portanto, para deduzir rotação, seguida de translação, o ponto $\mathbf{p}$ será escrito em termos de ambos os *frames* e será estudada a relação entre os mesmos.

Como na Figura 18, os *frames* A e B estão somente rotacionados e com as origens coincidentes. O mesmo ponto $\mathbf{p}$ pode ser representado em termos de ambos como em (3.3), e as relações geométricas observadas resultam a relação (3.4):

Figura 18 – Rotação em torno da origem de um ângulo arbitrário θ.



$$\begin{aligned} \hat{\mathbf{x}}_B &= \cos\theta\hat{\mathbf{x}}_A + \sin\theta\hat{\mathbf{y}}_A \\ \hat{\mathbf{y}}_B &= -\sin\theta\hat{\mathbf{x}}_A + \cos\theta\hat{\mathbf{y}}_A \end{aligned} \tag{3.4}$$

que na forma matricial é organizado conforme (3.5):

$$\begin{pmatrix} \hat{\mathbf{x}}_B & \hat{\mathbf{y}}_B \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}}_A & \hat{\mathbf{y}}_A \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{3.5}$$

Ao substituir o resultado na Equação (3.3), considerando o *frame* B, e igualar os termos a direita do resultado à mesma equação considerando o *frame* A, a relação final entre as coordenadas pode ser escrita como (3.6):

$$\begin{pmatrix} {}^{A}x \\ {}^{A}y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} {}^{B}x \\ {}^{B}y \end{pmatrix} \tag{3.6}$$

A matriz da Equação (3.6) é a matriz de rotação em duas dimensões ${}^{A}\mathbf{R}_B$, a qual tem como característica ser ortonormal, uma vez que suas colunas são ortogonais e possuem módulo igual a 1. Graças a isso, possue como propriedade a inversa ser a própria $\mathbf{R}^T$, e ter determinante igual a 1, ou seja, um vetor transformado por ela não altera o seu comprimento: ${}^{A}\mathbf{p} = {}^{B}\mathbf{p}$.

A próxima dedução envolve a translação entre dois *frames*. Supondo novamente os *frames* A e B, porém com origens diferentes assim como inicialmente proposto em 17. A rotação relativa entre os *frames* leva a um *frame* intermediário V, com origem coincidente à de A e eixos paralelos aos de B. Após a operação com a matriz de rotação, uma simples adição de coordenadas é suficiente para igualar os dois *frames* de A para B e completar a transformação:

$$\begin{aligned} \begin{pmatrix} {}^{B}x \\ {}^{B}y \end{pmatrix} &= \begin{pmatrix} {}^{V}x \\ {}^{V}y \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \\ &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} {}^{A}x \\ {}^{A}y \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \\ &= \begin{pmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \end{pmatrix} \begin{pmatrix} {}^{A}x \\ {}^{A}y \\ 1 \end{pmatrix} \end{aligned} \tag{3.7}$$

Portanto, como mostrado na Equação (3.7), os termos de translação foram incorporados à matriz de rotação, e as coordenadas do ponto acrescidas de 1 na terceira linha.

De posse das definições apresentadas, a Equação (3.8) traz a transformação completa ${}^{A}\mathbf{T}_B$ entre dois *frames* em coordenadas homogêneas com dimensões 3x3. Essa forma

apresenta vantagens computacionais por ser uma matriz quadrada.

$$\begin{aligned} {}^{A}\tilde{\mathbf{p}} &= \begin{bmatrix} {}^{T}\mathbf{R}_{B} & \mathbf{t} \\ \mathbf{0}_{1x2} & 1 \end{bmatrix} \cdot {}^{B}\tilde{\mathbf{p}} \\ &= {}^{A}\mathbf{T}_{B} \cdot {}^{B}\tilde{\mathbf{p}} \end{aligned} \tag{3.8}$$

Ao adicionar um terceiro eixo Z com a regra da mão direita em relação a XY, o espaço começa a ser analisado em três dimensões, logo a nova matriz de rotação deveria respeitar a relação (3.9):

$$\begin{bmatrix} {}^{A}x \\ {}^{A}y \\ {}^{A}z \end{bmatrix} = {}^{A}\mathbf{R}_{B} \begin{bmatrix} {}^{B}x \\ {}^{B}y \\ {}^{B}z \end{bmatrix} \tag{3.9}$$

A matriz deduzida na Equação (3.6) pode ser interpretada como uma rotação em torno do eixo Z de um ângulo θ. Para que a Equação (3.9) seja coerente, a nova matriz de rotação deve apresentar dimensões 3x3. Ao acrescer uma nova linha e coluna, com 1 em seu último elemento de diagonal e 0 nos demais, a matriz deduzida em (3.6) passa a operar nas três dimensões, porém sem nenhuma modificação à dimensão Z, como é definido pelas terceiras coluna e linha da Equação (3.10a).

As Equações (3.10b) e (3.10c) concluem, a partir de dedução semelhante à feita anteriormente, as rotações em torno dos eixos X e Y, respectivamente.

$$\mathbf{R}_{z}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.10a}$$

$$\mathbf{R}_{x}(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \tag{3.10b}$$

$$\mathbf{R}_{y}(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{3.10c}$$

A composibilidade é uma propriedade das matrizes de rotação ortonormais, de tal forma que a rotação desejada pode ser composta de rotações em torno de cada eixo desejado pela multiplicação de matrizes definidas em (3.10). Visto que o resultado de rotações sequenciais não é comutativo, há ordens pré estabelecidas na literatura para ordenar as formas de giro em torno dos diversos eixos [2].

Existem dois tipos de sequências de rotações para alcançar um ângulo final entre *frames* 3D: sequências Eurelianas ou Cardanianas [2]. No primeiro caso, há repetição não sucessiva de rotações em torno de um eixo, totalizando 3 rotações em sequência: XYX, XZX, YXY, YZY, ZXZ ou ZYZ. Para o segundo, há uma rotação em torno de cada eixo, respeitando uma das sequências: XYZ, XZY, YZX, YXZ, ZXY ou ZYX.

Uma das sequências muito adotadas por exemplo na aeronáutica em dinâmicas mecânicas é a sequência Euleriana ZYZ [2]. Para este trabalho foi adotada a convenção *roll-pitch-yaw* (Equação (3.11)), a qual utiliza a sequência XYZ, considerando que o eixo X aponta do centro para a frente, Y para esquerda e Z para cima da câmera.

$$\mathbf{R} = \mathbf{R}_x(\theta_r)\mathbf{R}_y(\theta_p)\mathbf{R}_z(\theta_y) \tag{3.11}$$

Para que possamos adicionar a translação após uma matriz de rotação (matriz de transferência em (3.7)) uma nova coluna deve ser inserida à direita da mesma com as coordenadas x, y e agora z. O novo vetor de translação $\mathbf{t} = \begin{pmatrix} x & y & z \end{pmatrix}^T$ forma a matriz de transferência em 3D. A Equação (3.12) define a transformação em coordenadas homogêneas para o espaço 3D.

$$\begin{pmatrix} {}^Ax \\ {}^Ay \\ {}^Az \\ 1 \end{pmatrix} = \begin{pmatrix} {}^A\mathbf{R}_B & \mathbf{t} \\ 0_{1x3} & 1 \end{pmatrix} \begin{pmatrix} {}^Ax \\ {}^Ay \\ {}^Az \\ 1 \end{pmatrix} \tag{3.12}$$

3.3 QUATERNIONS

> *Quaternions came from Hamilton after his really good work had been done; and, though beautifully ingenious, have been an unmixed evil to those who have touched them in any way, including Clark Maxwell.*
>
> Lorde Kelvin, 1892.

Quaternions foram definidos há mais de 150 anos atrás, e até os dias de hoje tem grande importância para roboticistas. Um quaternion é um número hipercomplexo, definido como um escalar s e um vetor $\mathbf{v}$ na Equação (3.13):

$$\begin{aligned} \breve{q} &= s + \mathbf{v} \\ &= s + v_1 i + v_2 j + v_3 k \end{aligned} \tag{3.13}$$

na qual $i^2 = j^2 = k^2 = -1$.

Uma propriedade intrínseca dos quaternions é que sua composição não é comutativa, assim como a composição de rotações. Porém, ao conseguirem representar a transformação em termos de um vetor e um escalar, tornam-se muito valiosos computacionalmente e vastamente utilizados em robótica e visão computacional pela eficiência computacional e prevenção de singularidade [2]. Sendo assim, serão aplicados nos códigos deste trabalho.

Toda rotação $\mathbf{R}$ entre dois *frames* 3D pode ser interpretada como um ângulo em torno de um vetor no espaço. Baseado nessa premissa, o quaternion é calculado representando essa rotação a partir da Equação (3.14):

$$s = \cos \theta/_2, \mathbf{v} = (\sin \theta/_2)\, \hat{\mathbf{n}} \tag{3.14}$$

onde θ é o ângulo de rotação em torno do vetor unitário $\mathbf{n}$. A composição de dois quaternions realizada pelo operador $\oplus$ é dada pela Equação (3.15).

$$\breve{q}_1 \oplus \breve{q}_2 = s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2 < s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2 > \tag{3.15}$$

Para levar em conta a translação e obter a transformação completa entre *frames* como em (3.12), (após a rotação realizada pelo quaternion) um vetor de translação $\mathbf{t}_{3x1}$ é adicionado para transladar o *frame*. Supondo dois *frames* A e B, isso é explicitado na Equação (3.16):

$${}^A\mathbf{p} = {}^A\xi_B \cdot {}^B\mathbf{p} = \breve{q} \cdot {}^B\mathbf{p} + \mathbf{t} \tag{3.16}$$

4 FUNDAMENTAÇÃO TEÓRICA PARA SFM E ODOMETRIA VISUAL

Este capítulo descreverá como interpretar o mundo real através de conjuntos de imagens, obtendo a estrutura 3D com o movimento e captação das câmeras - *Structure From Motion*, SFM [7]. Para isso, há a necessidade de relacionar os dados de *pixels* para unidades reais tridimensionais nas seções a seguir.

É de primordial importância o conhecimento de características da câmera para o resultado dos métodos: o quanto a lente distorce a imagem, e sua distância focal relacionam a imagem com a cena tridimensional. Isso é detalhado na Seção 4.1.

Para descobrir tais parâmetros das câmeras existem processos de calibração tanto para câmeras visuais quanto para térmicas, e os detalhes para realização do procedimento estão na Seção 4.4. De posse dos parâmetros, utilizando técnicas de identificação de imagens (Capítulo 2) e conceitos de geometria e álgebra linear, os pontos 3D são calculados a partir de duas ou mais imagens, sendo explicitados para duas imagens na Seção 4.2.

A odometria visual é apresentada na Seção 4.5, onde a partir de sequências de imagens e suas correspondências é estimado o movimento da câmera em relação ao mundo, o que é utilizado para unir vistas diferentes de uma mesma cena.

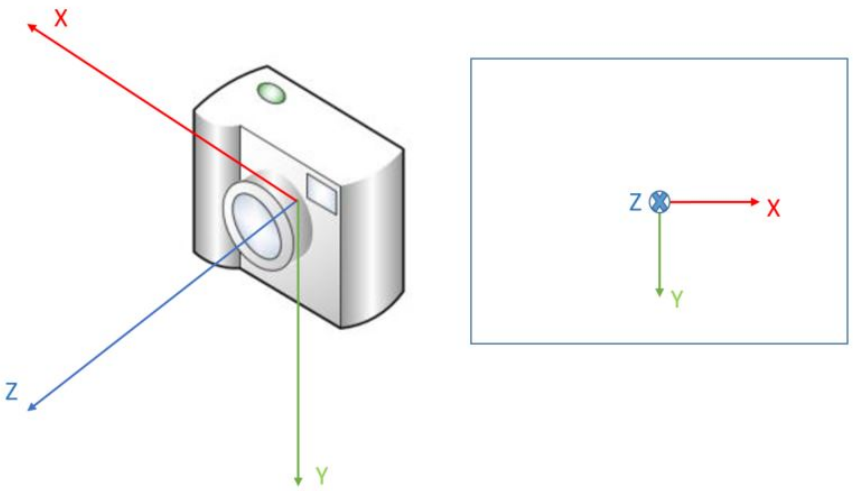
Por fim, a Seção 4.6 é responsável por detalhar como unir diversas nuvens de pontos em um modelo final 3D de toda cena filmada ao longo do processo, método nomeado registro de nuvens.

4.1 MATRIZ DA CÂMERA

4.1.1 **Parâmetros intrínsecos**

No cálculo da projeção de pontos do espaço na imagem é necessário saber a localização da câmera e a localização dos objetos da cena em relação à mesma. Utilizando os conceitos de *frames* coordenados apresentados no Capítulo 3 é posicionado um *frame* com origem no centro da câmera, ou ponto principal, e três eixos XYZ, como ilustrado na Figura 19. No caso o eixo Z está apontado para a frente da câmera para corroborar com o equacionamento proposto no decorrer deste capítulo.

Figura 19 – *Frame* da câmera com origem no centro da mesma, e vista do plano da imagem à direita.



A Equação (2.3) na Seção 2.3 mostra como obter um ponto $\mathbf{x}$ na imagem representando um ponto $\mathbf{X}$ no mundo real. Segundo equacionado em Hartley & Zisserman [3], o ponto $\mathbf{x}$ pode ser generalizado em coordenadas $\begin{pmatrix} x & y & z \end{pmatrix}^T$ na imagem (com z em escala), e o ponto no mundo real $\mathbf{X}$ é escrito em coordenadas homogêneas $\begin{pmatrix} X & Y & Z & 1 \end{pmatrix}^T$. Assim, a Equação (4.1) define a relação entre imagem, mundo real e distância focal para todas as coordenadas de forma matricial.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{4.1}$$

onde a matriz central pode ser escrita como $diag(f, f, 1)[\mathbf{I}|\mathbf{0}]$ e f é a distância focal da câmera no modelo *pinhole*.

A Equação (4.1) pode ser resumida em termos dos pontos e definindo a matriz central como matriz de projeção da câmera P neste modelo, ou simplesmente matriz da câmera [62]. Sendo assim, segundo Hartley & Zisserman [3], tem-se as Equações (4.2a) e (4.2b):

$$P = diag(f, f, 1)[I|0] \tag{4.2a}$$

$$\mathbf{x} = P \cdot \mathbf{X} \tag{4.2b}$$

Como considerado inicialmente no modelo *pinhole*, o ponto principal estaria localizado teoricamente no centro do plano da imagem. Como a matriz de *pixels* tem sua origem

de coordenadas no canto esquerdo inferior, deve-se somar um *offset* a fim de projetar as imagens a partir do ponto principal, o qual por sua vez na prática não se localiza exatamente no centro da matriz por imperfeições da câmera. A Equação (2.3) do modelo é adaptada para os eixos X e Y conforme as Equações (4.3a) e (4.3b), e a Equação (4.1) tem a matriz de projeção da câmera modificada como em (4.4).

$$x = \frac{f \cdot X}{Z} + p_x \tag{4.3a}$$

$$y = \frac{f \cdot Y}{Z} + p_y \tag{4.3b}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{4.4}$$

onde p_x e p_y são as coordenadas de *offset* do ponto principal no plano da imagem.

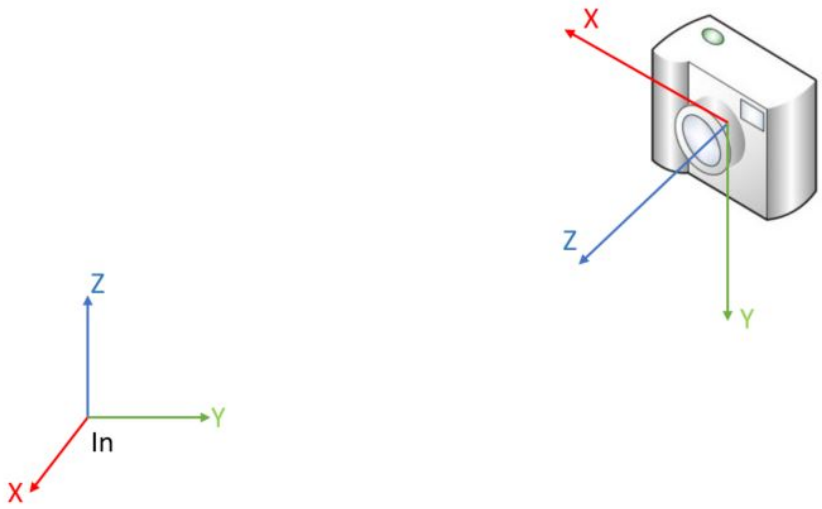
A matriz de projeção da câmera presente na Equação (4.4) possui somente parâmetros intrínsecos à câmera devido à sua construção e captação da imagem. Essa é a chamada matriz intrínseca **K**, definida em (4.5), e de posse dela podemos reprojetar pontos da imagem para o mundo real ao inverter a Equação (4.4) no *frame* da câmera.

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.5}$$

4.1.2 **Matriz da câmera P**

Para localizar os pontos 3D em relação ao mundo a partir do movimento da câmera, é necessário ter conhecimento da pose da última em relação a um *frame* inercial In definido como na Figura 20 [7].

Figura 20 – Câmera com seu *frame* em relação ao *frame* inercial *In* de referência.



Como visto no Capítulo 3, uma matriz de transformação homogênea contendo informações sobre rotação e translação deve ser agregada à Equação (4.4) a fim de contabilizar esse deslocamento em relação ao *frame* inercial. Ao adicionar à Equação (4.4) a matriz central definida na Equação (3.12), a nova Equação que relaciona pontos da imagem a pontos reais no *frame* inercial é definida em (4.6), sendo a Matriz da Câmera **P** [3] definida por inteiro em (4.7).

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0_{1x3} & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{4.6}$$

$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0_{1x3} & 1 \end{bmatrix} \tag{4.7}$$

Resumindo a Equação 4.6 ao substituir o resultado visto em 4.7, tem-se 4.8:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathbf{P} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{4.8}$$

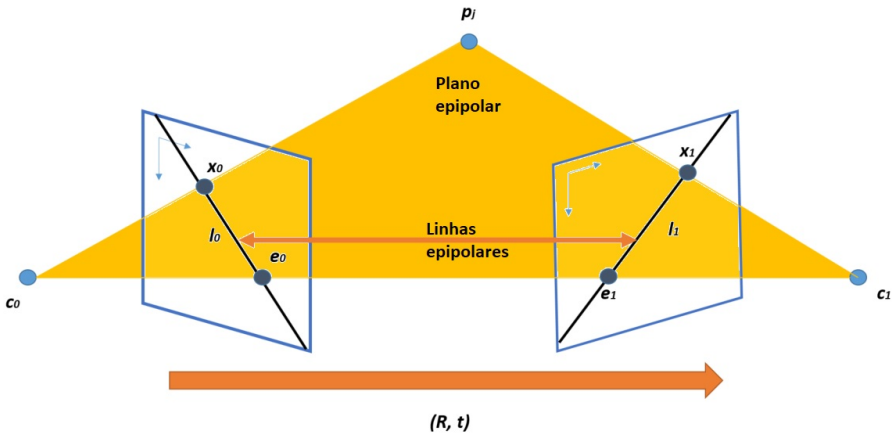
4.2 GEOMETRIA DE DUAS VISTAS

4.2.1 **Geometria Epipolar**

A partir de duas imagens de uma mesma cena é possível obter correspondências entre pontos de um mesmo objeto em cada uma das mesmas com diversas técnicas, como exemplificadas e definidas no Capítulo 2. O equacionamento apresentado na sequência desta seção, fundamentado em Szeliski [1], apresenta como calcular a posição do ponto localizado nas imagens no mundo real, bem como a posição da câmera no momento em que cada imagem foi retirada, sem a necessidade de conhecimento prévio sobre nenhum dos dois resultados.

Como base para o desenvolvimento matemático, suponha uma câmera com *frame* próprio se movendo em relação a um *frame* inercial, como ilustrada previamente na Figura 20. Inicialmente, a câmera se encontra no instante 0, com seu *frame* coincidindo com o inercial em origem e orientação, e ali é captada uma imagem de um ponto $\mathbf{p}_0$ da cena. Continuando o problema, a câmera se move para a pose do instante 1 com uma rotação $\mathbf{R}$ e translação $\mathbf{t}$, e ali capta uma imagem do mesmo ponto, porém agora explicitado como ponto $\mathbf{p}_1$. A Figura 21 ilustra essa situação genérica e apresenta a geometria epipolar, a qual surge no momento da triangulação dos pontos.

Figura 21 – Representação gráfica da geometria epipolar, onde $\mathbf{p}_j = \mathbf{p}_0 = \mathbf{p}_1$. Os vetores $\mathbf{c}_1 - \mathbf{c}_0$, $\mathbf{p}_j - \mathbf{c}_0$ e $\mathbf{p}_j - \mathbf{c}_1$ são coplanares formando o plano epipolar.



Na Figura 21 podem ser vistos os centros das câmeras nos respectivos instantes j (podendo j assumir valor 0 ou 1 como ilustrado), bem como o plano da imagem contendo o ponto $\mathbf{x}_j$, onde se localiza a projeção de $\mathbf{p}_j$ sobre a mesma. O raio que liga $\mathbf{c}_j$ a $\mathbf{x}_j$ é definido como raio direcional $\hat{\mathbf{x}}_j$, e d_j seria o escalar que multiplica esse vetor até o ponto

$\mathbf{p}_j$. É possível definir a Equação (4.9):

$$d_1\hat{\mathbf{x}}_1 = \mathbf{p}_1 = \mathbf{R}\mathbf{p}_0 + \mathbf{t} = \mathbf{R}(d_0\hat{\mathbf{x}}_0) + \mathbf{t} \tag{4.9}$$

onde $\mathbf{R}$ e $\mathbf{t}$ são a rotação e translação entre as câmeras da esquerda para a direita.

É provado que o vetor $\hat{\mathbf{x}}_j$ pode ser calculado da forma $\hat{\mathbf{x}}_j = \mathbf{K}_j^{-1}\mathbf{x}_j$, onde $\mathbf{K}$ é a matriz de parâmetros intrínsecos da câmera definida em (4.5). Realizando o produto cruzado do vetor $\mathbf{t}$, representado pela matriz correspondente $[\mathbf{t}]_x$, pela esquerda dos termos extremos da Equação (4.9), tem-se o resultado em (4.10):

$$d_1[\mathbf{t}]_x\hat{\mathbf{x}}_1 = d_0[\mathbf{t}]_x\mathbf{R}\hat{\mathbf{x}}_0 \tag{4.10}$$

Por fim, ao multiplicar pela esquerda os dois termos da Equação (4.10) pelo vetor do raio direcional $\hat{\mathbf{x}}_1^T$, o resultado vai a 0 (Equação (4.11)), uma vez que uma matriz $[\mathbf{t}]_x$ de produto cruzado multiplicada pelo mesmo vetor pela esquerda e direita resulta em 0 (em outro ponto de vista, um produto escalar de dois vetores perpendiculares retorna 0).

$$d_0\hat{\mathbf{x}}_1^T([\mathbf{t}]_x\mathbf{R})\hat{\mathbf{x}}_0 = d_1\hat{\mathbf{x}}_1^T[\mathbf{t}]_x\hat{\mathbf{x}}_1 = 0 \tag{4.11}$$

O plano que liga o ponto $\mathbf{p}$ ao centro das câmeras é chamado plano epipolar; os pontos $\mathbf{e}_j$ onde a linha de um centro ao outro cruza os planos das imagens são os pontos epipolares; e as linhas em cada plano da imagem que ligam $\mathbf{e}_j$ a $\mathbf{x}_j$ são as linhas epipolares.

A restrição definida na Equação (4.11) mostra a relação entre ambos os raios direcionais, e consequentemente os pontos $\mathbf{x}_j$ nas imagens. Essa restrição é resumida na Equação (4.12), onde a matriz $\mathbf{E}$ é chamada matriz Essencial.

$$\hat{\mathbf{x}}_1^T\mathbf{E}\hat{\mathbf{x}}_0 = 0 \tag{4.12}$$

O leitor pode perceber que a Equação 4.12 mapeia um ponto de uma imagem na linha epipolar l_j da oposta, por exemplo $l_1 = \mathbf{E}\hat{\mathbf{x}}_0$. Sendo assim, um ponto tem sua correspondência existente na linha epipolar da imagem oposta. A matriz $\mathbf{E}$ possui dimensões 3x3 e traz consigo informações sobre o deslocamento da câmera no mundo real, uma vez que restringe as correspondências entre as imagens.

Para ser calculada, são necessários no mínimo 8 pontos correspondentes nas duas imagens. Os elementos e_{ij} da matriz, com i e j variando de 0 a 2, são estimados ao multiplicar os valores dos pontos $\hat{\mathbf{x}}_j$ como na Equação (4.12). A multiplicação das

matrizes é aberta como na Equação (4.13), e oito equações formam um sistema linear a ser solucionado para os elementos de **E**:

$$x_{k0}x_{k1}e_{00}+y_{k0}x_{k1}e_{01}+x_{k1}e_{02}+x_{k0}y_{k1}e_{00}+y_{k0}y_{k1}e_{11}+y_{k1}e_{12}+x_{k0}e_{20}+y_{k0}e_{21}+e_{22}=0 \quad (4.13)$$

onde k vai de 1 a 8 pontos correspondentes.

A matriz **E** mapeia primeiramente os raios direcionais dos pontos nas respectivas imagens, e contem de forma exclusiva informações sobre a rotação e transação. Ao expandir a Equação (4.12) considerando os pontos $\mathbf{x}_j$, tem-se a definição apresentada em 4.14 e a matriz **F**, a qual incrementa **E** com os parâmetros intrínsecos da câmera. Ambas as matrizes possuem posto 2, como pode ser visto acima por mapear um ponto a um conjunto de pontos na linha epipolar. Um sistema semelhante ao apresentado com equações como em (4.13) pode ser utilizado para calcular **F**.

$$\hat{\mathbf{x}}_1^T\mathbf{E}\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}_1^T\mathbf{K}_1^{-T}\mathbf{E}\mathbf{K}_0^{-1}\hat{\mathbf{x}}_0 = \mathbf{x}_1^T\mathbf{F}\mathbf{x}_0 \quad (4.14)$$

Após descobertos os pontos correspondentes, definido o sistema e solucionada as matrizes **E** e **F**, é possível descobrir informações de rotação e translação entre as duas imagens (ou da câmera entre os dois instantes) e restringir correspondências entre pontos para realizar uma triangulação de forma correta, a fim de obter pontos **p**. Para obter **R** e **t**, a técnica de decomposição em valores singulares é utilizada [74].

Como apresentado em Hartley & Zisserman [3], no que diz respeito à decomposição, a matriz **E** deve ter dois de seus valores singulares iguais e um igual a 0. Como equacionado acima, é sabido que $\mathbf{E} = [\mathbf{t}]_x\mathbf{R}$. Pode-se decompor a matriz em seus valores singulares: $\mathbf{E} = \mathbf{U}diag(1,1,0)\mathbf{V}^T$, e a partir de desenvolvimento matemático e suposições sobre os ângulos de rotação e os sinais do vetor de translação, quatro soluções são propostas para a matriz de transformação $\mathbf{T} = [\mathbf{R}|\mathbf{t}]$ em (4.15):

$$\mathbf{T} = [\mathbf{U}\mathbf{W}\mathbf{V}^T|+\mathbf{u}_3] \quad ou \quad [\mathbf{U}\mathbf{W}\mathbf{V}^T|-\mathbf{u}_3] \quad ou \quad [\mathbf{U}\mathbf{W}^T\mathbf{V}^T|+\mathbf{u}_3] \quad ou \quad [\mathbf{U}\mathbf{W}^T\mathbf{V}^T|-\mathbf{u}_3] \quad (4.15)$$

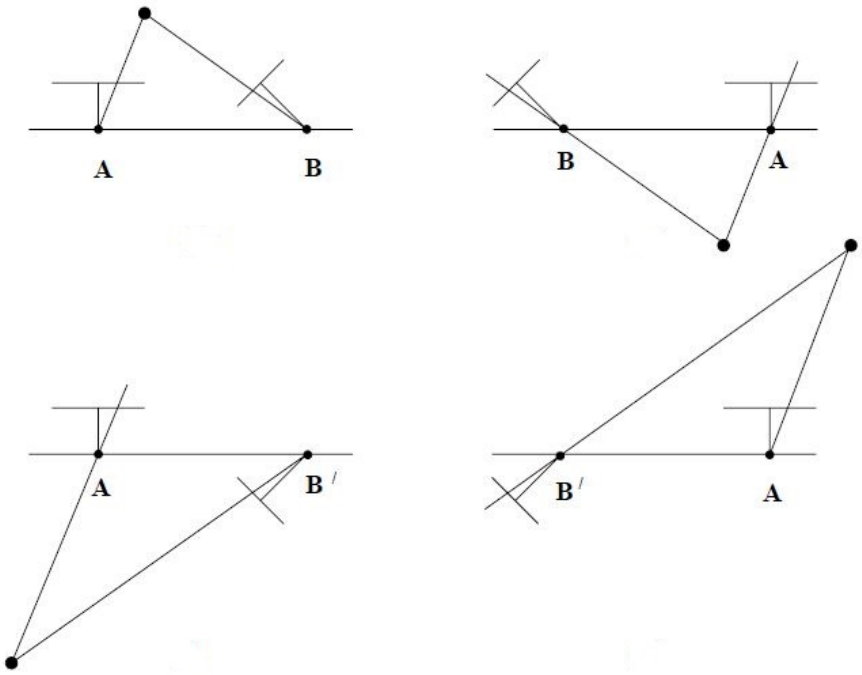
onde:

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{Z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.16)$$

Essa ambiguidade de soluções tem base matemática, porém para definir qual a solução correta real do problema é preciso triangular os pontos $\mathbf{x}_j$ correspondentes e observar em qual solução os pontos **p** obtidos se posicionam a frente das duas câmeras.

Uma interpretação clássica das quatro soluções possíveis está ilustrada na Figura 22, levando em conta os ângulos de rotação de câmeras nas posições 0 e 1, bem como sua posição relativa.

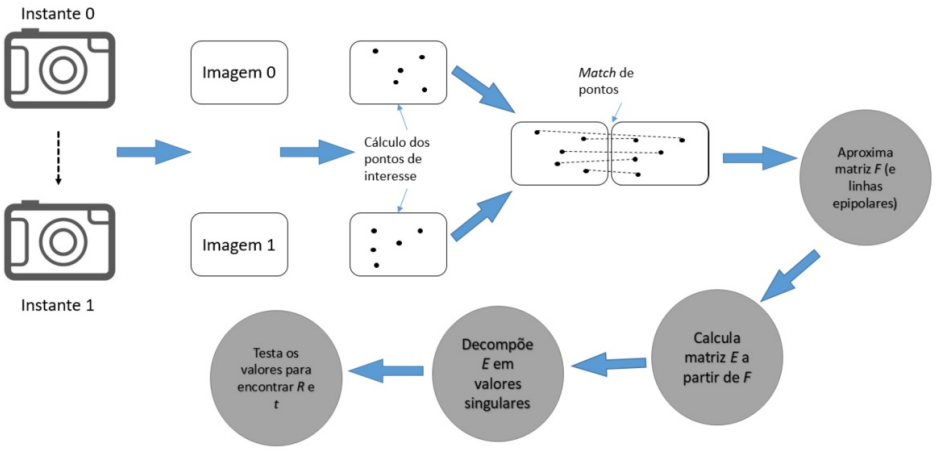
Figura 22 – Quatro soluções possíveis para o posicionamento das câmeras.



Fonte: [3].

Como resumo da sequência apresentada nessa seção, o fluxograma da Figura 23 identifica etapas para cálculos das matrizes referentes à geometria epipolar.

Figura 23 – Fluxograma sobre as etapas no cálculo da geometria epioplar.



4.2.2 **Visão Estéreo**

Foram apresentados diversos trabalhos referentes à reconstrução e obtenção de nuvens de pontos no Capítulo 1, inclusive por meio de câmeras estéreo. Visão estéreo é o nome dado à obtenção de informações tridimensionais a partir de um par ou maior conjunto de imagem vindas de dois ou mais sensores de forma sincronizada [32]. A técnica é capaz de retornar um mapa de disparidade (Figura 24), do qual se analisa a disparidade no eixo horizontal da matriz de *pixels* de pontos correspondentes entre as imagens. Essa disparidade, como será explicado mais adiante nessa seção, contribui para a triangulação dos pontos e obtenção final do mapa de profundidade e nuvem de pontos (Figura 25). É o princípio que os humanos utilizam, sendo capazes de reconhecer as três dimensões a partir do que os olhos observam simultaneamente.

Figura 24 – Mapa de disparidade a partir de duas vistas da mesma cena, com escala relativa de profundidade à direita.

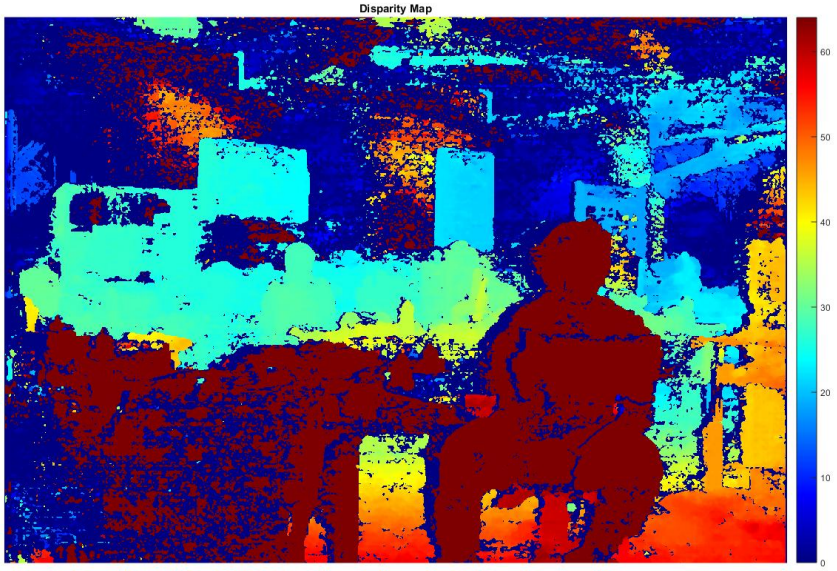


Figura 25 – Nuvem de pontos com a profundidade calculada para cada *pixel* do mapa de disparidade.

Uma das aplicações mais comuns da visão estéreo, inclusive a utilizada nesse trabalho, é o posicionamento de duas câmeras digitais de forma paralela a uma translação horizontal conhecida e medida com a maior acurácia possível (nomeada no inglês por *baseline*), de forma a seus planos de imagem ficarem alinhados [62].

Dessa forma, a cada par de fotos captada de forma sincronizada pode-se realizar o procedimento matemático para obter o mapa de disparidade e a nuvem de pontos.

Segundo Bradski [75], podem ser atribuídas quatro etapas para o processamento de pares de imagens estéreo, sendo eles:

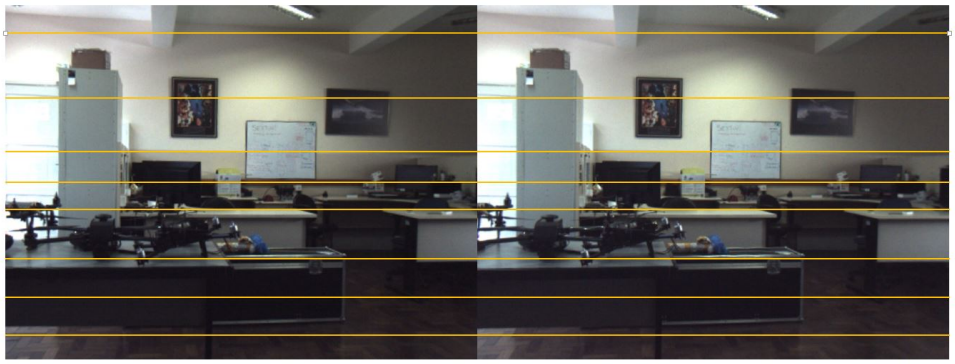
- Remover a distorção radial e tangencial das imagens segundo calibração das câmeras;
- Ajustar de forma fina a rotação e translação entre as câmeras, de forma a alinhar os planos das imagens, processo chamado retificação; Esses dois primeiros itens serão tratados em mais detalhes na Seção 4.4. Supondo as duas imagens ajustadas, segue o processo descrito aqui nesta seção;
- Encontrar a correspondência de pontos entre as imagens com técnicas vistas na Seção 2.4.2 ou métricas matemáticas baseadas em janelas [1], respeitando as restrições da geometria epipolar. Com isso é computado o mapa de disparidade;
- De posse do mapa de disparidade entre as correspondências é realizada a triangulação dos pontos para o mundo tridimensional, resultando na nuvem de pontos de saída.

Ao obter imagens retificadas, tendo em vista os conceitos apresentados na Seção 4.2.1 anterior, os centros das câmeras estão em tese perfeitamente alinhados, logo não há

necessidade de se calcular as matrizes **E** e **F**, que visam descobrir a rotação e translação entre as câmeras.

Pela geometria epipolar, com os planos das imagens paralelos e coincidentes, os epipolos estão localizados no infinito [3], e as linhas epipolares são paralelas nas duas imagens (Figura 26). Dessa forma o esforço computacional diminui consideravelmente, pois ao calcular um ponto de interesse em uma imagem, o mesmo deve se localizar na mesma linha da outra, o que diminui drasticamente o espaço de busca [1].

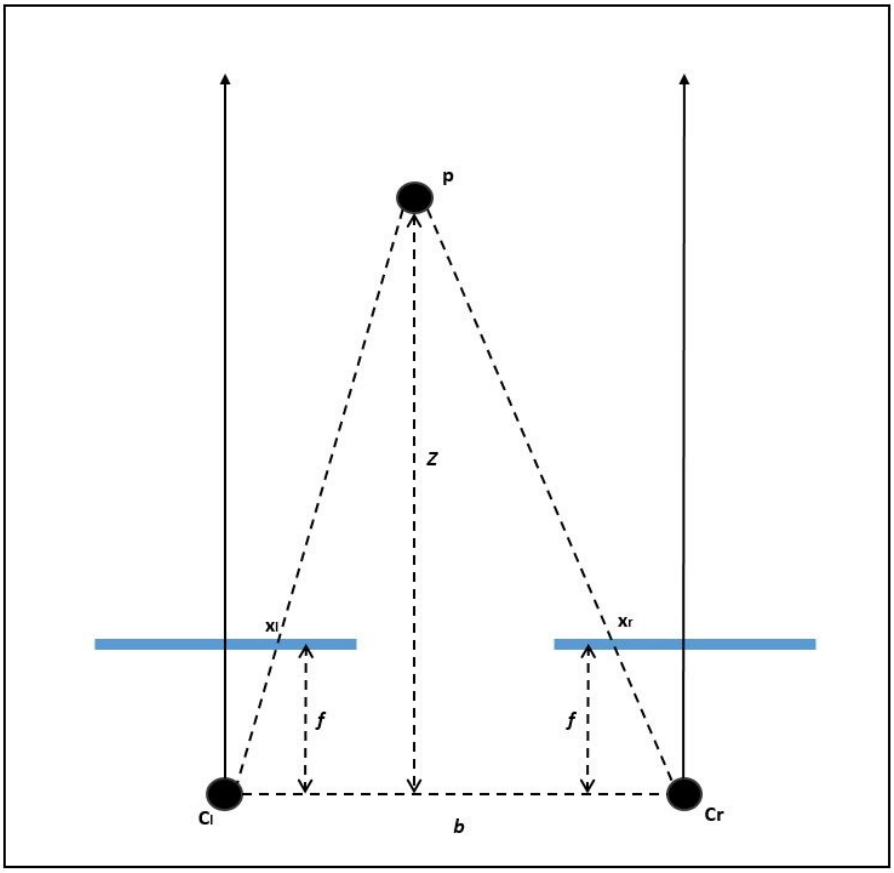
Figura 26 – Exemplo de linhas epipolares entre duas imagens retificadas.



A quarta etapa envolvendo a triangulação dos pontos é apresentada em Solem [62], na qual dada a disparidade do pixel referente à mesma *feature* nas duas imagens em termos de seus valores x_j, definido o valor da *baseline* b e conhecida a distância focal f das câmeras, é possível por relações geométricas obtidas na Figura 27 obter a profundidade do ponto Z no mundo real (Equação (4.17)).

É importante notar que o índice j, que na Seção 4.2.1 se referiam ao instante em que a imagem foi captada, agora se referem à câmera a qual a imagem pertence, sendo l para a esquerda e r para a direita.

Figura 27 – Uma vez os dois planos da imagem (em azul) perfeitamente alinhados e a imagem retificada, é possível calcular a profundidade Z a partir da relação geométrica formada.



$$Z = \frac{f \cdot b}{x_l - x_r} \tag{4.17}$$

Para cálculo do mapa de disparidade de forma esparsa, pontos de interesse e descritores são utilizados em ambas as imagens para obter as correspondências de forma mais exata possível entre as mesmas, e a partir daí triangular as correspondências em pontos **p** no mundo real. Esse método traz maior confiança no resultado e tem um custo computacional relativamente baixo graças ao espaço de busca reduzido, porém não é capaz de reconstruir ambientes com pouca textura, como paredes lisas e o céu uniforme.

Com recursos computacionais atuais, Szeliski [1] cita os métodos métricos para reconstrução densa utilizando janelas sobre os pares de imagens estéreo. A sequência para análise sobre a imagem envolve cálculos matriciais sobre uma região ao redor dos *pixels* da imagem. Tem-se o ponto correspondente ao da primeira imagem quando é obtido o maior valor dessa operação. A sequência para esse processo seria:

- Escolher a imagem base, como por exemplo a vinda da câmera esquerda;

- Definir uma janela de dimensões $m \times n$ para avaliar as vizinhanças dos *pixels* das imagens;
- Definir uma janela de dimensões $M \times n$, M maior que m anterior para a busca por correspondências na imagem da direita, uma vez que escolheu-se a esquerda como base;
- Varrer na imagem direita a região $M \times n$ centrada na mesma posição que o *pixel* $\mathbf{x}_l$ com uma janela $m \times n$, aplicando uma métrica para cada ponto correspondente da janela nas duas imagens. Métricas utilizadas na literatura são a soma de diferenças ao quadrado (SSD), soma da diferença absoluta (SAD), correlação cruzada normalizada (NCC), entre outras. Aqui é reforçado o conceito de que a janela só se desloca na horizontal devido à restrição da geometria epipolar;
- O *pixel* (ponto) $\mathbf{x}_r$ com melhor avaliação, como por exemplo maior valor para NCC, é escolhido como correspondente ao ponto $\mathbf{x}_l$, e a disparidade no eixo X é anotada para aquele ponto da imagem da esquerda;
- O processo se repete para todos os *pixels* da imagem da esquerda.

4.3 PROJEÇÃO DE UM PONTO TRIDIMENSIONAL NA IMAGEM

Toda a matemática apresentada no Capítulo 3 possibilita descrever a localização de uma câmera em um ambiente referenciado e os movimentos relativos entre a mesma e a cena observada. O processo descrito nas Seções 4.1 e 4.2 relaciona a imagem ao mundo real através de geometria (Subseção 4.2.2), propriedades da câmera e transformações homogêneas. Esse procedimento descrito é então usado para calcular o caminho inverso ao de interesse nessas seções, descobrindo o ponto $\mathbf{x}$ de uma imagem que corresponde ao ponto $\mathbf{p}$ no mundo real em relação ao *frame* da câmera.

Supondo o frame inercial O no ambiente, ao qual os pontos $\mathbf{p}_O$ da nuvem estão referenciados, e o *frame* A de uma câmera que capta ali imagens. A relação da imagem com os pontos observados está reescrita na Equação (4.18).

$$\mathbf{x} = \mathbf{P}_A \cdot \mathbf{p}_O = \mathbf{K}^A\mathbf{T}_O \cdot \mathbf{p}_O = \mathbf{K}[^A\mathbf{R}_O|\mathbf{t}] \cdot \mathbf{p}_O \tag{4.18}$$

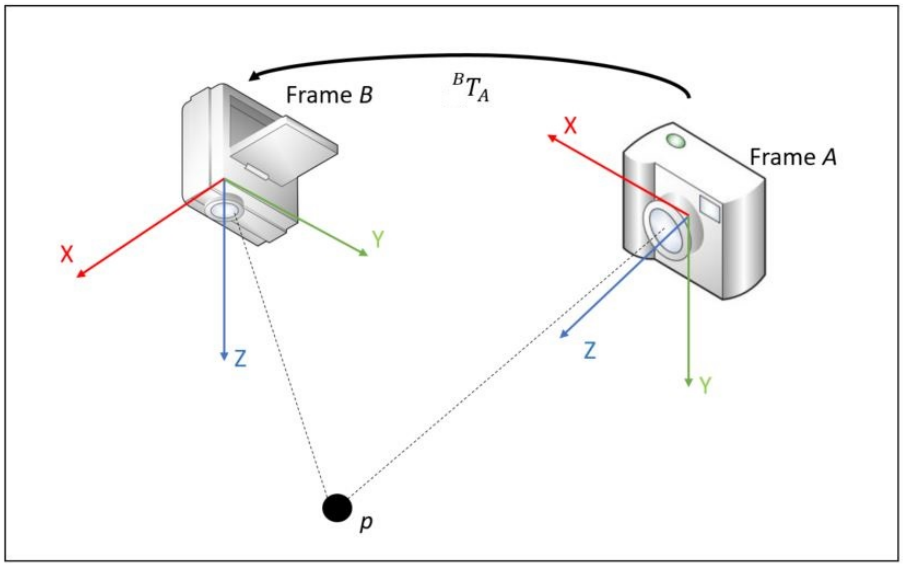
onde $\mathbf{P}_A$ possui o subscrito A para reforçar a matriz da câmera a que se refere.

Para o caso de mais de uma câmera realizando o processo de visão computacional, como é o caso desse trabalho, é necessário saber a relação da nuvem de pontos com todas as câmeras, e das câmeras entre si.

A Figura 28 ilustra uma posição relativa entre duas câmeras genéricas, onde uma matriz de transformação homogênea é capaz de relacionar as mesmas no espaço. Utilizando

a propriedade vista nas Equações (3.2) para relacionar diversos *frames*, os pontos $\mathbf{p}$ são vistos em uma segunda câmera com *frame* B incrementando a Equação (4.18) da forma vista em (4.19):

Figura 28 – Duas câmeras observando a mesma cena estão ligadas pela transformação ${}^{B}\mathbf{T}_{A}$ entre seus *frames* A e B.



$$\begin{aligned} \mathbf{x} &= \mathbf{P}_B \cdot \mathbf{p}_O = \mathbf{K}\,{}^{B}\mathbf{T}_O \cdot \mathbf{p}_O \\ &= \mathbf{K}[{}^{B}\mathbf{R}_A|\mathbf{t}][{}^{A}\mathbf{R}_O|\mathbf{t}] \cdot \mathbf{p}_O \end{aligned} \tag{4.19}$$

Concluindo o raciocínio apresentado, é possível projetar pontos do mundo real em qualquer câmera uma vez que se sabe a posição de ao menos uma delas em relação ao *frame* inercial, e a posição das outras em relação à primeira, com o uso da Equação (4.19).

4.4 CALIBRAÇÃO DAS CÂMERAS

Até o instante no decorrer deste capítulo foi descrita matematicamente a matriz da câmera com seus parâmetros intrínsecos, considerando-os já calculados ou sabidos. Esta seção traz os métodos utilizados para encontrar tais parâmetros para as câmeras do trabalho, tanto térmica quanto RGB.

4.4.1 **Procedimento geral**

Um método simples de calibração pode ser visto em Solem [62]. Considerando o modelo utilizado neste trabalho da matriz intrínseca $\mathbf{K}$ (Equação (4.5)), desprezando distorções e considerando o ponto principal no centro das dimensões da imagem, basta calcular a distância focal de uma forma semelhante à apresentada na Equação (2.3). O

procedimento utiliza um objeto retangular de tamanho conhecido dX e dY posicionado a uma distância dZ da câmera, paralelo ao plano da imagem. Medidas as dimensões do objeto na imagem em *pixels*, é possível estimar o foco da câmera, levando em conta uma possível diferença para as dimensões X e Y (quando a matriz de sensores possui *pixels* retangulares, o que acontece em câmeras mais baratas).

Sendo assim, a Equação (4.20a) descreve ambos os focos, com a matriz **K** concluída em (4.20b).

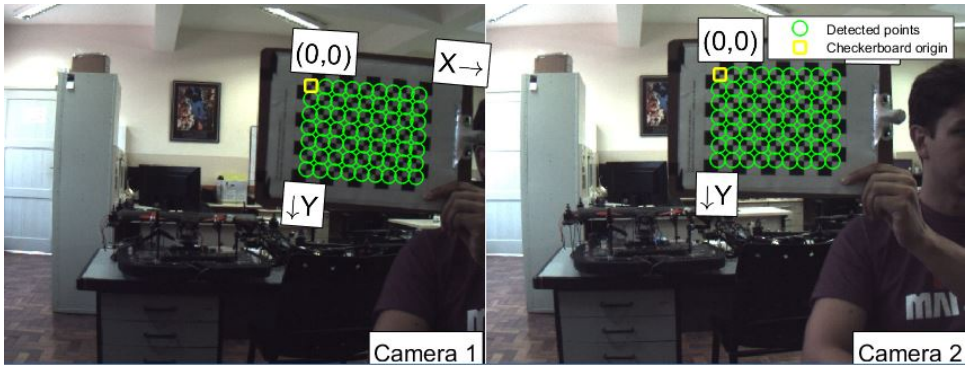
$$f_x = \frac{dx}{dX} dZ \quad f_y = \frac{dy}{dY} dZ \tag{4.20a}$$

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4.20b}$$

Neste trabalho foi considerado um processo que parte desse princípio básico, porém leva em consideração outros aspectos e imperfeições da câmera. O procedimento de calibração é inicialmente apresentado por Zhang [80] utilizando matemática vista em Hartley & Zisserman [3], porém utiliza um modelo incrementado para otimizar a calibração frente a distorções da câmera. Bradski [75] descreve o procedimento para o modelo *pinhole* e a matriz **K** adotada, bem como modelos de correção contra distorções mais comuns existentes nas lentes de câmeras digitais.

É necessário um padrão conhecido observado pela câmera, para trazer informações do mundo real contabilizadas em *pixels* [75] [80]. A forma mais comum adotada em procedimentos atuais é um tabuleiro de xadrez disposto em um plano, com sua medida das laterais dos quadrados conhecida, e o mesmo é observado de diversos pontos de vista para captação dos dados necessários. Na Figura 29 é possível ver a identificação de um tabuleiro durante o processo de calibração.

Figura 29 – Processo de calibração estéreo com vista identificada do tabuleiro por ambas as câmeras.



O início do desenvolvimento matemático envolve o conceito de homografia plana, a qual nada mais é do que a projeção de um plano sobre o outro. A partir disso pode-se propor uma homografia dos pontos no plano do tabuleiro de xadrez real para o plano da imagem. Suponha os pontos $\mathbf{p}$ sobre o tabuleiro e seus respectivos pontos $\mathbf{x}$ representados na imagem. É possível definir a matriz de homografia $\mathbf{H}$ como em (4.21):

$$\mathbf{x} = s\mathbf{H}\mathbf{p} \tag{4.21}$$

onde s é um fator de escala inerente à operação de homografia, primeiramente definida somente em escala, mas que será solucionada ao fim para medidas reais na calibração.

Ao supor que a matriz $\mathbf{H}$ é composta por uma parcela referente à rotação e translação da câmera em relação ao mundo, e supondo outra para os parâmetros intrínsecos da câmera, tem-se algo próximo ao apresentado para a matriz da câmera na Seção 4.1.2, e detalhado em (4.22):

$$\mathbf{x} = s\mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{p} \tag{4.22}$$

Para simplificar a questão, considera-se o plano do tabuleiro de xadrez com a coordenada Z em 0. Suponto coordenadas homogêneas para $\mathbf{p}$ na Equação (4.22), a simplificação anularia a terceira coluna da matriz $\mathbf{R}$, e a matriz $\mathbf{H} = \mathbf{K}\begin{bmatrix} r_1 & r_2 & \mathbf{t} \end{bmatrix}$ terá dimensões 3x3.

Ao comparar as colunas da matriz de homografia com essa repartição, pode-se definir:

$$\begin{aligned} h_1 = s\mathbf{K}r_1 \quad ou \quad r_1 = \lambda\mathbf{K}^{-1}h_1 \\ h_2 = s\mathbf{K}r_2 \quad ou \quad r_2 = \lambda\mathbf{K}^{-1}h_2 \\ h_3 = s\mathbf{K}t \quad ou \quad t = \lambda\mathbf{K}^{-1}h_3 \end{aligned} \tag{4.23}$$

na qual h_j são cada uma das três colunas de $\mathbf{H}$ e $\lambda = 1/_s$.

Sendo os vetores r_1 e r_2 da matriz de rotação ortonormais, os mesmos formam um ângulo reto no espaço e têm a mesma norma. Daí, duas condições surgem para que as matrizes existam a partir dos pontos observados, definidas em (4.24):

$$\begin{aligned} r_1^T \cdot r_2 = 0 \\ h_1^T\mathbf{K}^{-T}\mathbf{K}^{-1}h_2 = 0 \end{aligned} \tag{4.24a}$$

$$\begin{aligned} ||r_1|| = ||r_2|| \\ h_1^T\mathbf{K}^{-T}\mathbf{K}^{-1}h_1 = h_2^T\mathbf{K}^{-T}\mathbf{K}^{-1}h_2 \end{aligned} \tag{4.24b}$$

Simplificando o que é visto nas condições, define-se $\mathbf{B} = \mathbf{K}^{-T}\mathbf{K}^{-1}$, com seus elementos de índice ij variando de 1 a 3.

Com a multiplicação, a matriz tem a forma fechada:

$$\mathbf{B} = \begin{bmatrix} \frac{1}{f_x^2} & 0 & \frac{-c_x}{f_x^2} \\ 0 & \frac{1}{f_y^2} & \frac{-c_y}{f_y^2} \\ \frac{-c_x}{f_x^2} & \frac{-c_y}{f_y^2} & \frac{c_x^2}{f_x^2} + \frac{c_y^2}{f_y^2} + 1 \end{bmatrix} \tag{4.25}$$

Tendo em vista a simetria da matriz $\mathbf{B}$ e a forma das restrições em (4.24a) e (4.24b) é possível reescrever a operação matricial $h_i^T \mathbf{B} h_j = v_{ij}^T b$, onde:

$$v_{ij} = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i3}h_{j1} + h_{i1}h_{j3} \\ h_{i3}h_{j2} + h_{i2}h_{j3} \\ h_{i1}h_{j3} \end{bmatrix}^T \quad e \quad b = \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix}^T \tag{4.26}$$

Nesse ponto, as duas restrições podem ser resumidas à Equação matricial (4.27), e ao coletar n imagens do tabuleiro de pontos de vista diferentes é possível unir $2n$ equações em linha nesse sistema.

Com $n >= 2$ é possível ter solução para os parâmetros intrínsecos (focos, centros óticos e fator de escala) e extrínsecos (matriz de rotação e translação) nos grupos das Equações (4.28) e (4.29).

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v22)^T \end{bmatrix} b = 0$$
$$\mathbf{V}b = 0 \tag{4.27}$$

$$\begin{aligned} f_x &= \sqrt{\lambda / B_{11}} \\ f_y &= \sqrt{\lambda B_{11} / B_{11}B_{22} - B_{12}^2} \\ c_x &= -B_{13} f_x^2 / \lambda \\ c_y &= (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2) \\ \lambda &= B_{33} - (B_{13}^2 + c_y(B_{12}B_{13} - B_{11}B_{23})) / B_{11} \end{aligned} \tag{4.28}$$

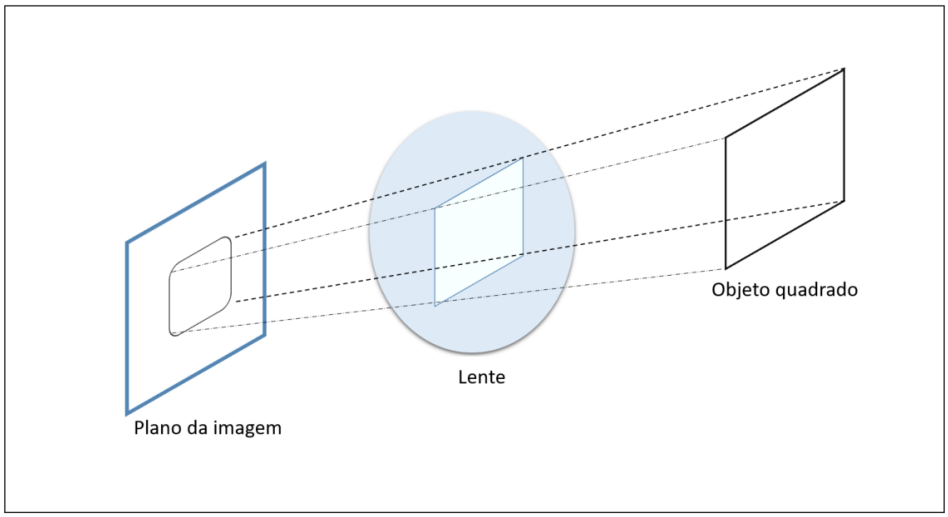
$$\begin{aligned} r_1 &= \lambda \mathbf{K}^{-1} h_1 \\ r_2 &= \lambda \mathbf{K}^{-1} h_2 \\ r_3 &= r_1 \times r_2 \\ t &= \lambda \mathbf{K}^{-1} h_3 \end{aligned} \tag{4.29}$$

Imperfeições das coordenadas dos pontos na imagem e distorções resultam nesse ponto um erro a ser corrigido por aproximações sucessivas e otimização. A partir das mesmas imagens retiradas para o processo descrito acima é possível obter parâmetros para corrigir duas formas de distorção mais comuns na maioria das lentes em câmeras digitais: a distorção radial, ou efeito barril, e a distorção tangencial [75].

Na prática, nenhuma lente é perfeita quanto à captação da luz de forma alinhada, por dificuldades mecânicas de fabricação e posicionamento frente à matriz de sensores. Na distorção radial, objetos mais distantes do centro da imagem tem seu aspecto distorcido, graças ao efeito da lente nas bordas, como ilustrado na Figura 30.

Para o modelo, é considerada distorção nula no centro, e crescente a medida que o raio r de distância em relação ao último aumenta. O efeito é quantificado por uma expansão em série de Taylor de grau dois na maioria dos casos, chegando a grau três para câmeras com maior distorção. O novo ponto corrigido tem as coordenadas como na Equação (4.30).

Figura 30 – Ilustração da distorção radial, que contorce retas a medida que se afastam do centro do plano de imagem.



$$x_{\text{corrigido}} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{4.30a}$$

$$y_{\text{corrigido}} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{4.30b}$$

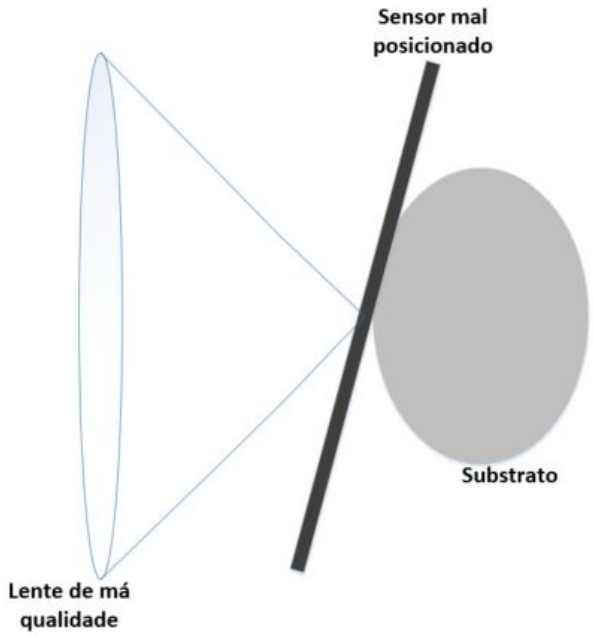
onde k_1, k_2 e k_3 são os coeficientes da série de Taylor mencionada.

A distorção tangencial vem a ocorrer graças a possíveis imperfeições no alinhamento dos planos da lente e matriz de sensores, e é corrigida para cada dimensão como apresentado na Equação (4.31). Uma ilustração do efeito está na Figura 31.

$$x_{\text{corrigido}} = x + [2p_1y + p_2(r^2 + 2x^2)] \tag{4.31a}$$

$$y_{\text{corrigido}} = y + [p_1(r^2 + 2y^2) + 2p_2x] \tag{4.31b}$$

Figura 31 – O mal posicionamento do sensor produz a distorção tangencial, onde imagens tem aparência mais próxima na região menos distânte à lente, e vice versa.



Em termos práticos, uma primeira calibração desconsidera quaisquer distorções existentes na lente, sendo assim uma primeira aproximação para parâmetros intrínsecos. Com conhecimento das características físicas do tabuleiro e onde os pontos $\mathbf{x}_d$ seriam esperados na imagem (na posição $\mathbf{x}_p$), várias equações, como descritas em (4.32), são colhidas e aproximadas para obter os coeficientes k_1, k_2, k_3, p_1 e p_2, e então os parâmetros intrínsecos são reaproximados, seguindo essa sequência até o critério de parada.

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = (1 + k_1r^2 + k_2r^4 + k_3r^6) \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} 2p_1y + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 2p_2x \end{bmatrix} \tag{4.32}$$

4.4.2 **Calibração da câmera térmica**

Para a calibração da câmera térmica em termos dos seus parâmetros intrínsecos, adotou-se os mesmos princípios da Seção 4.4.1. O contraponto e dificuldade encontrados para essa câmera são devidos ao fato de um simples tabuleiro impresso ou em madeira não ser visível à câmera pela sua temperatura [47].

Várias alternativas são encontradas na literatura para esse problema: Hilsenstein [81] construiu um tabuleiro em placa de circuito impresso, de forma a obtê-lo com dois materiais de propriedades térmicas diferentes; Harry *et al.* [82] montou uma matriz de fios aquecidos para ressaltar as linhas do tabuleiro.

Outra solução envolve aquecer um tabuleiro de quadrados pretos e brancos com uma lâmpada incandescente, a fim de os pretos se destacarem pela temperatura. Isso foi feito no trabalho de Dias [47], e está ilustrado em 32.

Figura 32 – Esquema para aquecer o tabuleiro em ambiente controlado com lâmpada incandescente de alta potência, com resultado à direita.

Fonte: [47].

4.4.3 **Câmeras estéreo**

Ao utilizar um par de câmeras estéreo como mencionado na Seção 4.2.2, Bradski [75] detalha o cuidado em encontrar não somente os parâmetros de cada câmera em separado, mas também a relação entre as duas no espaço, com a calibração e retificação estéreo.

A calibração estéreo estabelece uma câmera como referencial (no caso deste trabalho e adotada aqui como exemplo, a esquerda) e calcula a rotação ${}^{r}\mathbf{R}_{l} = \mathbf{R}_{rel}$ e translação ${}^{r}\mathbf{t}_{l} = \mathbf{t}_{rel}$ da outra em relação a essa (por consequente, a direita).

As duas câmeras captam imagens de forma sincronizada do tabuleiro de xadrez e realizam o procedimento descrito na Seção 4.4.1 para obter parâmetros intrínsecos e posição da câmera em relação ao mundo.

Supondo um ponto tridimensional $\mathbf{p}$ descrito no *frame* inercial de referência, o mesmo pode ser descritos nos *frames* das câmeras esquerda e direita com uma simples transformação da forma $\mathbf{p}_l = \mathbf{R}_l\mathbf{p} + \mathbf{t}_l$ e $\mathbf{p}_r = \mathbf{R}_r\mathbf{p} + \mathbf{t}_r$. Para relacionar um *frame* ao outro, o ponto $\mathbf{p}_l$ é transformado para $\mathbf{p}_r$ da forma $\mathbf{p}_r = \mathbf{R}_{rel}\mathbf{p}_l + \mathbf{t}_{rel}$. Unindo essas relações, pode-se determinar as Equações (4.33a) e (4.33b) com formulações fechadas para as matrizes:

$$\mathbf{R}_{rel} = \mathbf{R}_r\mathbf{R}_l^T \tag{4.33a}$$

$$\mathbf{t}_{rel} = \mathbf{t}_r - \mathbf{R}_{rel}\mathbf{t}_l \tag{4.33b}$$

Para cada par de imagens sincronizadas é obtida uma pose em relação ao mundo atual, as quais são substituídas nessas Equações para obter a rotação e translação relativas. Como existem imperfeições e arredondamentos de *pixels*, cada par resulta em uma matriz com pequenas diferenças, e a mediana das medidas é tomada como valor inicial para um algoritmo de otimização de Levenberg-Marquardt, que age sobre o erro da projeção dos pontos tridimensionais sobre as imagens esquerda e direita.

Uma vez conhecida de forma refinada a rotação e translação entre as câmeras pela calibração, entra em vista o processo de retificação das imagens [75]. Para que o algoritmo de reconstrução funcione da melhor forma possível é preciso que as linhas epipolares sejam horizontais, a fim de corresponderem às linhas da imagem. Isso é bastante improvável devido a imperfeições mecânicas no posicionamento das câmeras, logo um ajuste fino sobre o alinhamento das linhas deve ser realizado, o que consiste em corrigir distorções e manter paralelos os eixos principais das câmeras. Para este trabalho foi utilizado o algoritmo de Bouguet [83], o qual parte das matrizes $\mathbf{R}_{rel}$ e $\mathbf{t}_{rel}$ obtidas. Num primeiro momento, cada câmera é rotacionada com metade da rotação definida por $\mathbf{R}_{rel}$, sendo elas $\mathbf{r}_l$ e $\mathbf{r}_r$ para esquerda e direita, respectivamente.

A segunda questão envolve a posição esperada do epipolo na imagem da esquerda, $\mathbf{e}_1$, que deveria estar no infinito à esquerda. Uma vez os planos da imagem alinhados, este primeiro vetor para retificação fina é calculado considerando a direção formada pelo ponto principal no centro da imagem (c_x, c_y) e o vetor de translação $\mathbf{t}_{rel}$. Logo, $\mathbf{e}_1 = \frac{\mathbf{t}_{rel}}{||\mathbf{t}_{rel}||}$.

O próximo vetor para retificação deve ser ortogonal ao primeiro, de forma paralela ao plano de imagem e ortogonal ao eixo principal. Assim, simplificando $\mathbf{t}_{rel} = \mathbf{t}$, o produto

vetorial entre o eixo principal e $\mathbf{e}_1$ resulta em:

$$\mathbf{e}_2 = \frac{\begin{bmatrix} -\mathbf{t}_y & \mathbf{t}_x & 0 \end{bmatrix}^T}{\sqrt{\mathbf{t}_x^T + \mathbf{t}_y^2}} \tag{4.34}$$

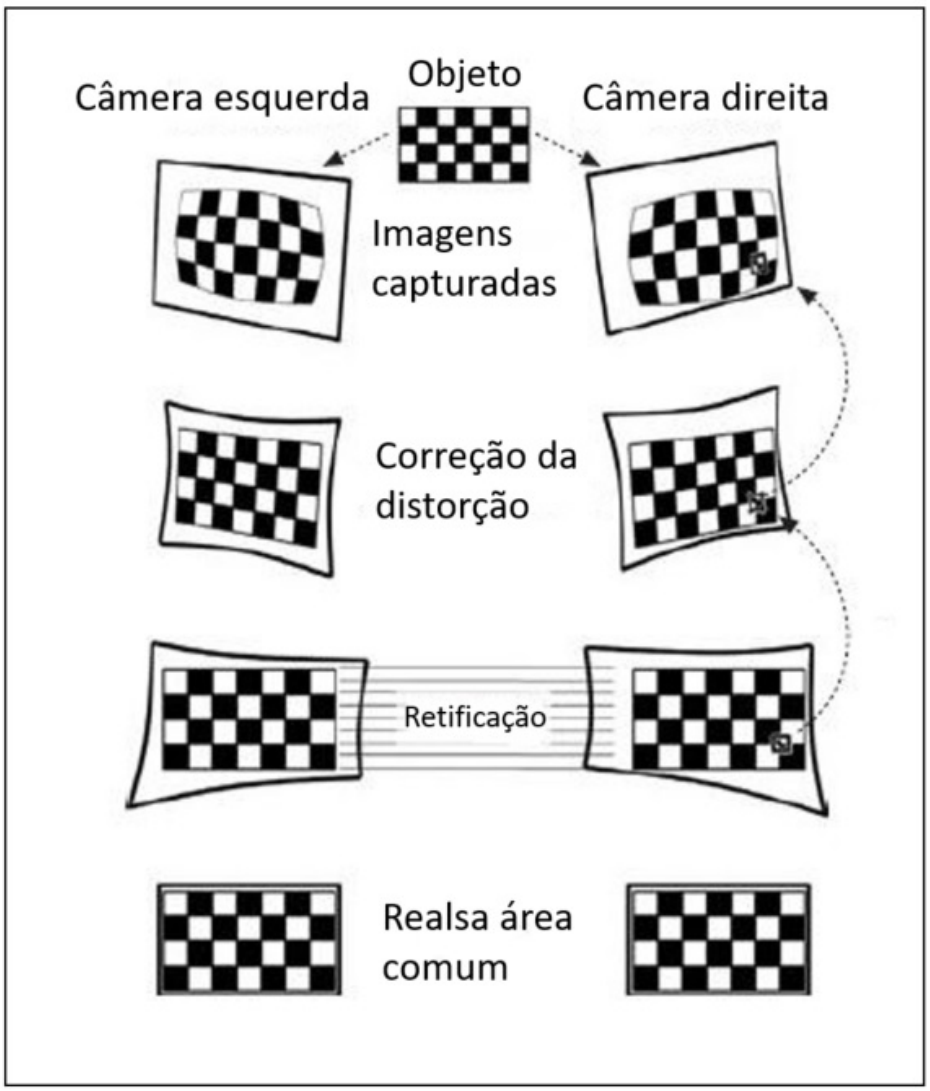
O produto cruzado entre $\mathbf{e}_1$ e $\mathbf{e}_2$ resulta no terceiro vetor, $\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2$, para assim ter-se a matriz de retificação definida na Equação (4.35), e as rotações finais sobre as câmeras como em (4.36):

$$R_{rect} = \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_3^T \end{bmatrix} \tag{4.35}$$

$$\begin{aligned} \mathbf{R}_l &= R_{rect}\mathbf{r}_l \\ \mathbf{R}_r &= R_{rect}\mathbf{r}_r \end{aligned} \tag{4.36}$$

A Figura 33 exemplifica o processo genérico que ocorre na correção da imagem a partir de dados obtidos com a calibração mostrados nas seções atual e 4.4.1:

Figura 33 – Fluxo de operações ilustradas para retificação final de imagens estéreo.



Fonte: [3].

4.5 ODOMETRIA VISUAL (VO)

Como já citado no Capítulo 1, sendo um nicho do SLAM, a odometria visual é um grande foco atual para a área de navegação e robótica, na última principalmente em ambientes em que não é possível o uso de GPS para localização (ambientes internos ou subaquáticos), ou é necessária maior precisão na mesma, sendo que isso só seria alcançado a princípio com sensores de mais alto custo, como *rangefinders* [15] [84].

Há dois métodos preferidos na literatura atual para realização da odometria visual [37]: o denominado método direto, baseados na aparência da imagem [85], ou os dados crus dos valores de *pixels*, analisando a disparidade de *frames* em sequência para obter a direção do movimento entre os mesmos; e baseados em *features* e descritores [86], que realizam a identificação e comparação de pontos chave na imagem, como descrito no Capítulo 2. A medição de odometria a partir dos *matchs* pode se dar utilizando a geometria epipolar vista

na Seção 4.2 ou métodos de otimização para aproximações sucessivas frente às posições relativas das features.

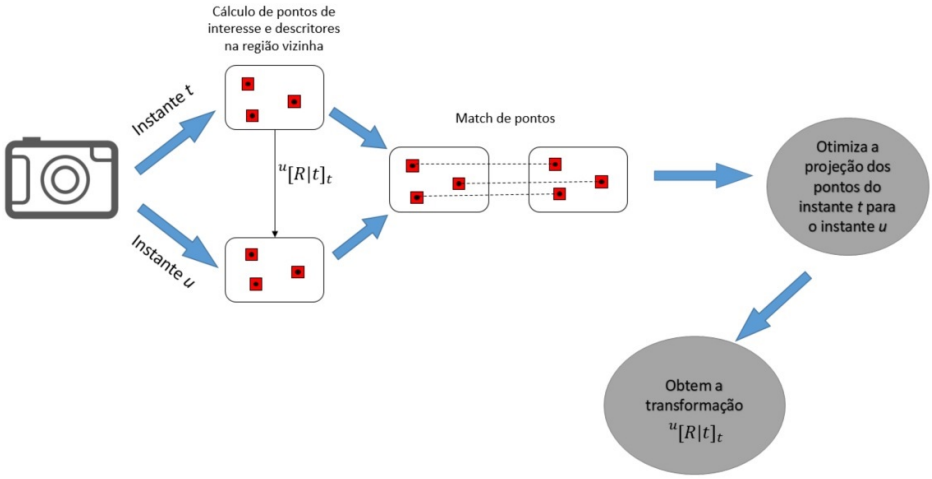
Segundo Kassir *et al.* [87], a maioria das aplicações utiliza o método de extração e comparação de *features*, por ser uma técnica vasta em termos de formas de abordagem em diversos ambientes e bastante difundida, obtendo resultados melhores que o método direto, o qual vem a requerer mais recursos de *hardware*.

Em contrapartida, Forster [15] traz riscos relacionados às *features*, as quais são dependentes de *thresholds*, sujeitas a falsas correspondências e má performance em ambientes com pouca textura ou câmeras ligeiramente desfocadas, ponto de vantagem para o método direto. Como conclusão, a técnica usada vem a depender fortemente do ambiente onde será inserida e da capacidade computacional disponível.

Várias variações são propostas para a otimização da VO [88] [38] [89] [90], as quais inserem técnicas como Filtro de Kalman, redes neurais e modelos dos robôs envolvidos para obter mais robustez contra imperfeições e *outliers*, assim como possíveis erros acumulados ao longo da aplicação. Nessa seção será descrita porém a sequência tradicional e base para a realização da odometria visual utilizando *features*, uma vez que o trabalho se dedica à inspeção em ambientes ricos em textura. Os conceitos apresentados estão baseados nas aplicações demonstradas em [91], [37] e [92].

O fluxograma da Figura 34 apresenta a linha seguida no processo. É importante reforçar que a odometria visual monocular não tem a capacidade de informar a escala real do mundo, também se perdendo em escalas diferentes ao longo de um processo: se a navegação é iniciada em um local com *features* próximas à câmera e após um tempo a câmera começa a confiar em *features* mais distantes, a escala do movimento será perdida por depender da escala do ambiente [91]. Como as câmeras estéreo tem conhecimento da translação no mundo real de sua *baseline*, essa é utilizada para relatar o deslocamento na unidade métrica adotada, além da reconstrução em três dimensões já apresentada na Seção 4.2.2.

Figura 34 – Fluxograma resumindo o cálculo da odometria para uma câmera entre os instantes t e u.



Um primeiro detalhe importante na confiança do resultado para odometria é a alta taxa de captura das câmeras envolvidas e uma movimentação suave pelo terreno, a fim de analisar com detalhes o movimento entre *frames* seguidos. Taxas de captura variando entre 20 e 60 Hz são vistas nas câmeras atuais aplicadas na literatura.

De posse de um par de *frames* obtidos nos instantes t e $u = t + 1$ são obtidos *features* em cada um. Em Zhang *et al.* [37] é discutido o uso de métodos SIFT e SURF para essa etapa e sequente descritores para os pontos, os quais demandam tempo de processamento alto para aplicações *online*, e os autores trazem a atenção para uso do detector FAST em parceria com o descritor ORB. Outros trabalhos contam com maior eficiência computacional usando descritores simples, como o *Harris*, que mesmo não sendo invariantes à rotação e escala retornam resultados satisfatórios uma vez que há pouco movimento entre *frames* em sequência [91] [93].

A partir de *features* obtidas é realizada a comparação das mesmas, ou *match*, citado na Seção 2.4. Para descritores simples a comparação é tida por meio de funções de custo assim como citadas na Seção 4.2.2: NCC, SAD, entre outras, em uma região de dimensões $n \times n$ de entorno do ponto.

A partir deste instante é possível utilizar os conceitos de geometria epipolar e encontrar as matrizes Fundamental e Essencial, e com elas a rotação e translação entre as duas imagens. Isso requer a solução de sistemas lineares, otimização e buscas por *outliers*, bem como operações matriciais relativamente custosas. Visto isso, na prática a otimização ocorre de uma outra forma para facilitar o processo *online*.

Suponha a pose da câmera no instante t como referência, e a transformação $^{u}\mathbf{T}_{t} = [^{u}\mathbf{R}_{t}|\mathbf{t}_{t}]$ que leva a câmera à posição do instante u. Para encontrar essa relação é utilizado em algumas aplicações o algoritmo de três pontos [94], variação do PnP

(*Perspective n Points*), que produz um sistema a ser solucionado e é a solução mínima para determinar a pose. Uma formulação lógica para esse algoritmo seria:

- Obter um conjunto de três pares $(\mathbf{x}_t, \mathbf{x}_u)$ correspondentes após o *match*, onde cada par corresponde a um ponto $\mathbf{p}$ no mundo real;
- Suponha matriz da câmera no instante t sendo $\mathbf{P}_t = \mathbf{K}[\mathbf{I}|\mathbf{0}]$, e para o instante u tem-se então com a transformação ${}^{u}\mathbf{T}_t$: $\mathbf{P}_u = \mathbf{K}[{}^{u}\mathbf{R}_t|\mathbf{t}_t]$;
- Reprojetando os pontos $\mathbf{x}_t$ da imagem no instante t para o mundo real com $\mathbf{P}_t^{-1}$ e projetando os mesmos para a imagem do instante u com $\mathbf{P}_u$, em teoria deveriam ser encontrados os próprios pontos $\mathbf{x}_u$. Esse é o objetivo do algoritmo para encontrar a transformação.

A Equação (4.37) possui o sistema a ser otimizado em função dos parâmetros da matriz de rotação e translação presentes em ${}^{u}\mathbf{T}_t$. O erro de projeção sobre a imagem do instante u deve ser minimizado, o que é feito com a soma dos erros ao quadrado de cada ponto $\mathbf{x}_{u\_i}$:

$$\min \sum_{i=1}^{3} \left[\mathbf{x}_{u\_i} - \mathbf{P}_u \left(\mathbf{P}_t^{-1} \mathbf{x}_{t\_i} \right) \right]^2 \tag{4.37}$$

Em termos computacionais, para contrapor o efeito de *outliers* entre os *matches*, o algoritmo RANSAC [95] é empregado, e explicado aqui de forma breve:

- Escolhe-se uma fração de k em um conjunto de n pontos e otimiza em busca da transformação a Equação (4.37) utilizando, por exemplo, o método de Levenberg-Marquardt [96].
- O resultado dessa primeira otimização é testado sobre os demais pontos do conjunto total. Se satisfatório, esta é a transformação; se não, a depender do resultado retira-se os pontos não satisfeitos e otimiza novamente com os restantes, ou simplesmente sorteia outros k pontos aleatórios de um total de n para otimizar o problema.

Variações são observadas para melhorar a eficiência ou obter resultados mais rapidamente com menor custo computacional, mas a conclusão necessária é a matriz de transformação em cada sequência de *frames* para obter a pose da câmera durante a trajetória.

4.6 REGISTRO DE NUVENS DE PONTOS

As aplicações introduzidas no Capítulo 1 e na Seção 4.5 são focadas no algoritmo para localização e no *pipeline*, que vai desde a escolha de *hardware*, interpretações de imagem, cálculo de odometria à obtenção final de nuvens de pontos para o modelo. Essa seção discorrerá sobre a obtenção do modelo final após todo esse processo.

A nuvem de pontos obtida pelo processo estéreo se localiza a princípio no *frame* da câmera. A câmera possui por sua vez a pose em relação ao *frame* inercial, implícita na matriz da câmera $\mathbf{P}$, como a transformação ${}^{c}\mathbf{T}_{I}$, onde I e c representam os *frames* inercial e da câmera, respectivamente. A odometria visual calcula esse valor a cada instante k (Seção 4.5). A reconstrução do ambiente deve ser realizada em relação ao *frame* inercial uma vez que este é estático e referência para o movimento. Portanto, a cada instante k a nuvem de pontos $\mathbf{PC}_k$ deve ser transformada em $\mathbf{PC}'_k$ como em (4.38) antes de ser acumulada em (4.39):

$$\mathbf{PC}'_k = {}^{c}\,\mathbf{T}_{I}^{-1}\mathbf{PC}_k \tag{4.38}$$

Uma vez referenciada no *frame* inercial, pode-se a princípio acumular as nuvens nos instantes k, como na Equação (4.39).

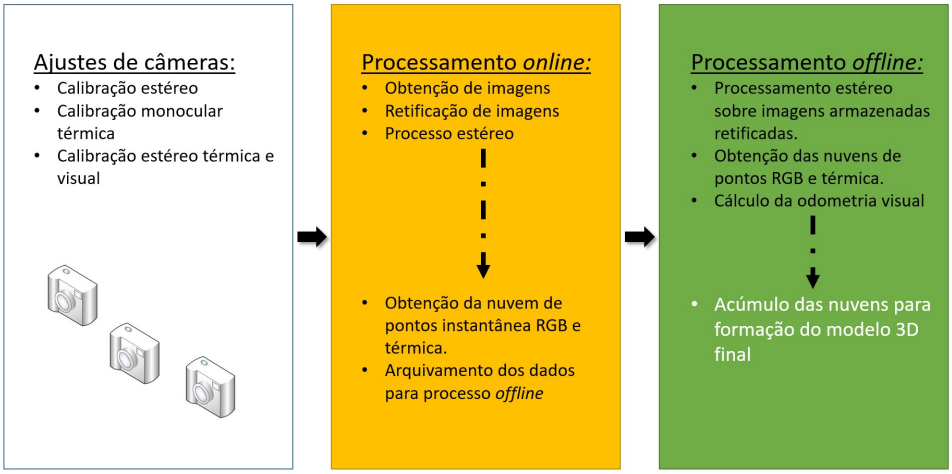
$$\mathbf{M} = \sum_{k} \mathbf{PC}'_k \tag{4.39}$$

5 METODOLOGIA PROPOSTA E ALGORITMOS UTILIZADOS

Nesse capítulo a sequência descrita no decorrer dos Capítulos 2, 3 e 4, de forma abrangente em teoria, será aplicada em uma linha de processamento, apontando os algoritmos usados em cada parte deste trabalho. O procedimento é inicialmente dividido nas três etapas de execução do trabalho: ajustes de câmeras, captura e processo *online*, e processo *offline* (Figura 35).

Ao fim do capítulo, na Seção 5.4, é introduzido o *framework* ROS, destacando como os algoritmos foram inseridos no sistema de forma prática.

Figura 35 – Etapa de ajuste de câmeras, processos *online* e *offline* em sequência.



5.1 AJUSTE DAS CÂMERAS

As câmeras são calibradas de forma estéreo para descobrir seus parâmetros intrínsecos e pose relativa, como apresentado na Seção 4.4. Sendo adotado o modelo *pinhole* para as mesmas e a forma de distorção das lentes radial e tangencial, todos os parâmetros k_1, k_2, k_3, p_1 e p_2 são calculados como apresentados na teoria.

As câmeras visuais produzem imagens com dimensões de 1600x1200 *pixels* para largura e altura, respectivamente. Para aumentar a taxa de captura de 10 para 20 Hz, a resolução é reduzida para 800x600 sobre o mesmo campo de visão. Para a câmera térmica a resolução em largura e altura é de 640x512, porém o campo de visão corresponde a cerca de um quarto do visto pelas câmeras visuais, localizado ao centro da cena observada. A sua taxa de aquisição gira em torno de 13 Hz.

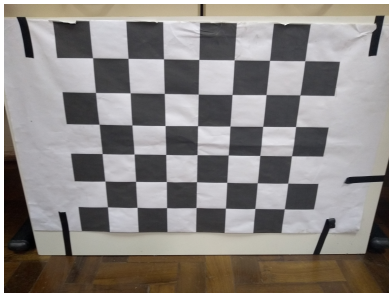
Em termos de *frames* para localização, é adotado um *frame* inercial para referência da odometria chamado “odom”, e a primeira câmera a se relacionar com este é a visual à esquerda, com o *frame* “left_optical”. Essa câmera por sua vez se torna a referência para o restante do sistema: a calibração estéreo das outras estão relacionadas ao *frame*

mencionado.

Para calibração estéreo das câmeras visuais é usado um tabuleiro impresso com 75 milímetros em cada lado dos quadrados, capturando distância focal, parâmetros de correção de distorção, parâmetros para retificação e portanto a pose relativa da câmera direita (com *frame* “right_optical”) para a esquerda.

Na calibração da câmera térmica foram usados dois procedimentos: primeiramente uma calibração monocular (Seção 4.4.1), seguida então de uma estéreo com a câmera visual esquerda. Na calibração monocular, realizada para obter parâmetros mais confiáveis em um ambiente controlado como o da Figura 32, um tabuleiro de 11 milímetros foi aquecido e visualizado pela câmera, obtendo assim sua distância focal e parâmetros de correção da distorção. De posse de um valor de referência o tabuleiro foi novamente aquecido e visualizado pelas câmeras térmica e esquerda. Os mesmos valores obtidos para as câmeras RGB são obtidos novamente para a térmica, no *frame* “thermal”, em relação ao “left_optical”. Os tabuleiros empregados estão retratados na Figura 36.

Figura 36 – Tabuleiros utilizados para calibração. À esquerda: calibração das câmeras RGB, 7x6 quadrados internos com 75 mm de lado; à direita: calibração monocular da câmera térmica, 9x6 quadrados internos com 11 mm de lado.

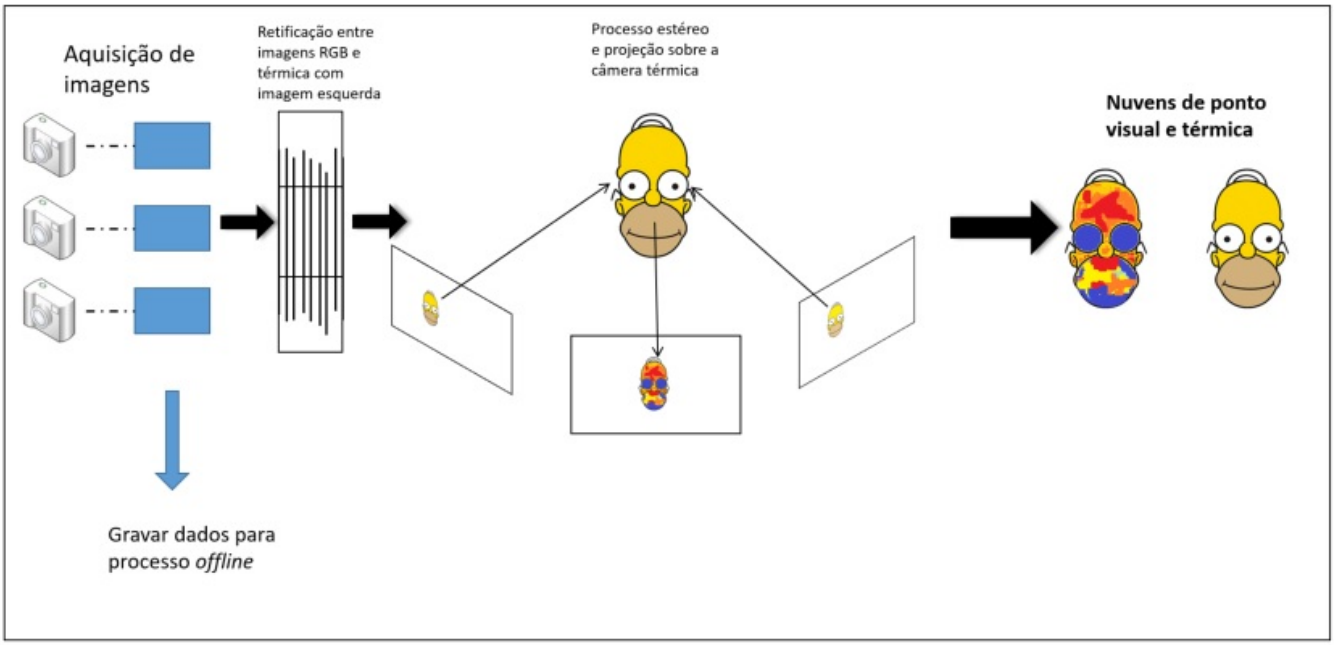


5.2 PROCESSO *ONLINE*

O *pipeline* apresentado em teoria no Capítulo 4 é parcialmente realizado de forma *online*, no que diz respeito ao estudo de imagens, processo estéreo, criação de nuvens de pontos e projeção da mesma sobre os dados térmicos.

Vale destacar que a principal função da etapa *online* consiste em coletar dados para o processo *offline* (no caso, as imagens das câmeras), pois assim é possível obter melhor resultado para a odometria visual e sincronização das imagens por meio de *software*. Essa etapa é seguida de processo de nuvens instantâneas RGB e térmica para simples aferição qualitativa do processo durante a captura dos dados. Uma sequência lógica das etapas está ilustrada no fluxograma da Figura 37, e as mesmas serão descritas em detalhes em sequência.

Figura 37 – Sequência lógica do processamento *online*, com o processo estéreo e projeção da nuvem de pontos estéreo sobre a imagem térmica.



Na primeira etapa são captadas imagens de forma teoricamente sincronizada da cena. As mesmas são retificadas como apresentado na Seção 4.4.3, o que leva a seguir para o processamento estéreo, a segunda etapa do fluxograma.

O algoritmo estéreo segue a linha apresentada na teoria na Seção 4.2.2, porém aqui será detalhada como aplicado por Bradski [75]:

- As imagens de entrada são normalizadas para reduzir efeitos prejudiciais de iluminação e reforçar a textura, passando sobre a mesma uma janela de tamanho variável (entre 5x5 a 30x30 *pixels*), e atribuindo ao *pixel* central o valor de cor calculado como em (5.1):

$$I'_c = \min\left[\max\left(I_c - \bar{I}\right), I_{lim}\right] \tag{5.1}$$

  onde I_c são as intensidades de cor do *pixel* central, $\bar{I}$ é a média de intensidade dos *pixels* da janela, e I_{lim} é um valor estabelecido para ajudar na uniformização final, bem como evitar valores negativos.

- Com imagens retificadas e normalizadas, as correspondências são buscadas sobre as mesmas linhas das imagens esquerda e direita, como manda a geometria epipolar. A fonte de comparação são os *pixels* da imagem esquerda. Supondo o ponto $\mathbf{x}_l = (x_0, y_0)$ como o ponto da imagem esquerda a ser buscado na direita, e sabendo da translação e rotação relativa das duas câmeras, é possível utilizar somente uma janela em torno do ponto com a operação de soma das diferenças absolutas SAD. Quanto menor o valor obtido, mais semelhantes são os pontos nas duas imagens. A busca vai

de $\mathbf{x}_l$ para a esquerda por k *pixels* (dita máxima disparidade definida) pelo ponto semelhante, e o melhor valor é definido como disparidade em $\mathbf{x}_l$.

- Para combater falsas correspondências e valorizar somente aquelas obtidas frente a texturas relevantes, uma taxa de singularidade $s > \frac{valor\_match - minimo\_match}{minimo\_match}$ é calculada com o valor da função candidato a *match* e o menor valor encontrado na busca pelo *match* do item anterior. Se a relação for superior a um valor pré-determinado, a correspondência é aprovada.

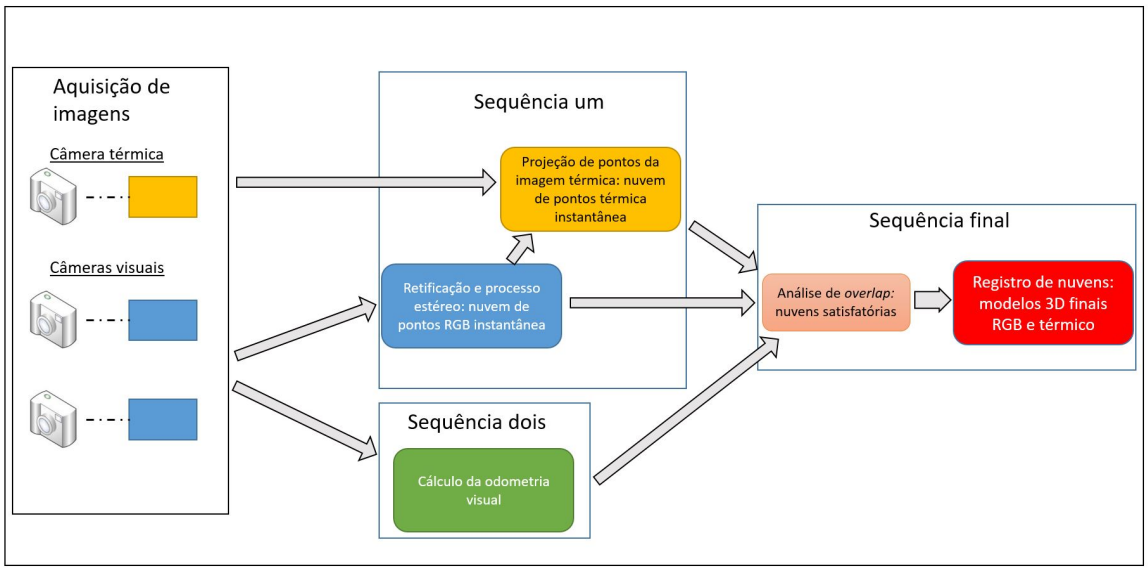
A terceira parte consiste em utilizar o mapa de disparidade filtrado obtido para calcular a profundidade (como visto na Equação (4.17)), e com a mesma obter o ponto em três dimensões. De posse dessa nuvem de pontos instantânea, os mesmos são reprojetados na imagem térmica utilizando a relação entre as câmeras ${}^{thermal}T_{left\_optical}$ e o processo descrito na Seção 4.3. Supondo que o ponto tridimensional $\mathbf{p}$ foi reprojetado para a região da imagem térmica no ponto $\mathbf{x}_{thermal}$, a coloração deste ponto na imagem é atribuída ao ponto tridimensional. Os pontos que recebem nova coloração formam a nuvem de pontos instantânea térmica. Ambas as nuvens podem ser visualizadas para dar ao operador do sistema noção da qualidade da reconstrução frente ao ambiente e iluminação.

5.3 PROCESSO *OFFLINE* - *PIPELINE* COMPLETO SOBRE OS DADOS COLETADOS

O processamento das técnicas propostas na metodologia sobre as imagens é feito de forma *offline*, para melhor sincronização de imagens e concluindo com melhor qualidade do modelo 3D final. Nessa etapa são revistas as imagens gravadas e calculadas as nuvens instantânea colorida e térmica, porém em paralelo tem-se também o cálculo da odometria visual, a análise de *overlap* entre as cenas e o registro de nuvens de pontos para resultado final.

O fluxograma da Figura 38 resume as etapas do processo e o trânsito de dados entre os algoritmos. Em um primeiro momento, tem-se a reprodução das imagens seguidas de retificação para o processo estéreo, assim como realizado durante o processo *online*.

Figura 38 – Etapas realizadas de forma *offline* pelo algoritmo para registro final de nuvens.



Em paralelo, a segunda etapa realiza a VO através das imagens RGB esquerda e direita. Aqui foi utilizado o algoritmo apresentado em [93] e com semelhanças em Ni [92], o qual expõe as etapas para encontrar a transformação entre *frames* sequentes nos instantes t e u das câmeras estéreo:

- Uma vez que são considerados movimentos pequenos e suaves entre as imagens, para detectar pontos de interesse são convoluídos *kernels* com filtros que realçam *blobs* e *corners* nas imagens com dimensão 5x5.

- Detectados pontos de interesse, com valores máximos e mínimos para ambas as classes filtradas, é passado outro filtro em regiões para não supressão de máximo ou mínimo, reduzindo o número de pontos na próxima etapa [97].

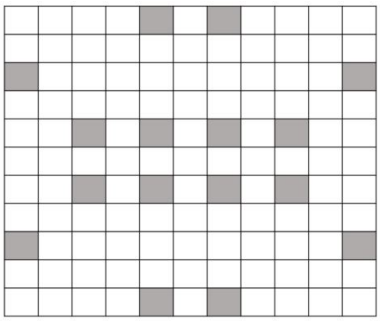
- O descritor empregado consiste na convolução de filtros de Sobel em uma janela 11x11 no entorno dos pontos, com a operação de SAD entre as janelas uma vez obtido o *match*.

  A Figura 39 ilustra os filtros para *corners* e *blobs* aplicados, bem como o descritor baseado na convolução de filtros de Sobel simplificado computacionalmente para maior eficiência [93].

Figura 39 – À esquerda: filtro para detectar *corners*; ao centro: filtro para detectar *blobs*; à direita: descritor em região 11x11 para comparar redondezas de pontos de interesse entre imagens.

-1	-1	0	1	1
-1	-1	0	1	1
0	0	0	0	0
1	1	0	-1	-1
1	1	0	-1	-1

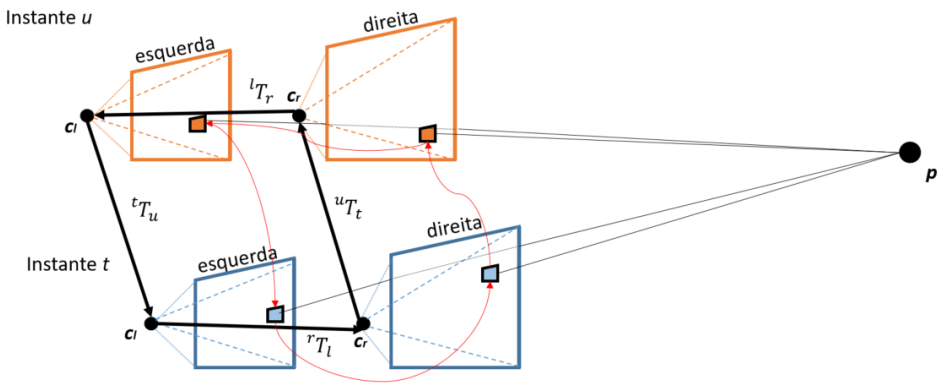
-1	-1	-1	-1	-1
-1	1	1	1	-1
-1	1	8	1	-1
-1	1	1	1	-1
-1	-1	-1	-1	-1



- Neste ponto um artifício é empregado, uma vez que está sendo usado um par de câmeras estéreo. Como explicado na Seção 4.5, o *match* é realizado entre imagens obtidas em sequência para descobrir a transformação entre as mesmas, mas isso é adotado para uma câmera somente. A ideia seria fazer um *match* em círculo, de tal forma que a foto da esquerda no instante u é comparada com a mesma do instante t; a última por sua vez é transferida e comparada com a da direita do instante t; a *feature* desta última imagem é levada à comparação com a imagem direita do instante u, e por fim volta à comparação entre a última citada e a primeira do ciclo.

  Após o *match* entre 4 regiões em torno de pontos de interesse, fazendo um círculo de transformações entre os dois instantes é necessário que o ponto encontrado ao final do círculo coincida com o de partida satisfatoriamente para confiar que, ao utilizarmos esse ponto para descobrir a movimentação entre as cenas, não estaremos lidando com pontos diferentes (*outliers*). A Figura 40 ilustra esse círculo, o qual garante mais confiança ao *match*.

Figura 40 – Esquema para *match* de features em círculo do conjunto estéreo.



- De posse dos N pares de pontos tidos como correspondentes entre as câmeras da

esquerda e da direita nos dois instantes, são projetados para o mundo real os pontos $\mathbf{p}$ do instante t com a matriz das câmeras estéreo conforme Seção 4.2.2.

- Os pontos $\mathbf{p}$ serão reprojetados nas imagens esquerda e direita no instante u. Considerando a matriz ${}^{u}\mathbf{T}_{t}$ como a transformação entre *frames* nos instantes t para u para ambas as câmeras, contendo rotação e translação, a diferença entre as duas transformações está no mundo real, tendo a esquerda como referência e a direita afastada da medida b referente à *baseline*. Descontando isso, considera-se a reprojeção do ponto como $\mathbf{x}_i^j = \pi^{(j)}(\mathbf{p}_i, r, t)$, sendo $j = l$ ou r, i o índice da correspondência entre o $\mathbf{p}$ e a *feature* que o originou e r e t os parâmetros de rotação e translaçãoa serem descobertos.

- Por fim, a Equação (5.2) representa o erro de reprojeção a ser minimizado para cada ponto nas duas imagens do instante u, feito por meio do método Gauss-Newton. A partir do *match* em círculo mostrado na Figura 40 ocorre o processo de projeção e reprojeção entre os instantes, que se assemelha ao visto nas equações da Seção 4.5 para uma câmera.

$$\sum_{i=1}^{N} ||\mathbf{x}_i^{(l)} - \pi^{(l)}(\mathbf{p}_i, r, t)||^2 + ||\mathbf{x}_i^{(r)} - \pi^{(r)}(\mathbf{p}_i, r, t)||^2 \tag{5.2}$$

Finalmente, a terceira etapa consiste em unir dados vindos das duas etapas anteriores para registro das nuvens de pontos. Como descrito matematicamente na Seção 4.6, de posse da odometria entre instantes subsequentes e nuvem de pontos $\mathbf{PC}$ para cada um dos mesmos, o simples acúmulo das mesmas seria suficiente para construir o modelo final.

Supondo K instantes de duração para o monitoramento realizado, o modelo final sem nenhum tratamento $\mathbf{M}$ pode ser definido como na Equação (5.3):

$$\mathbf{M} = \sum_{k=1}^{K} {}^{I}\mathbf{P}_k \cdot \mathbf{PC}_k \tag{5.3}$$

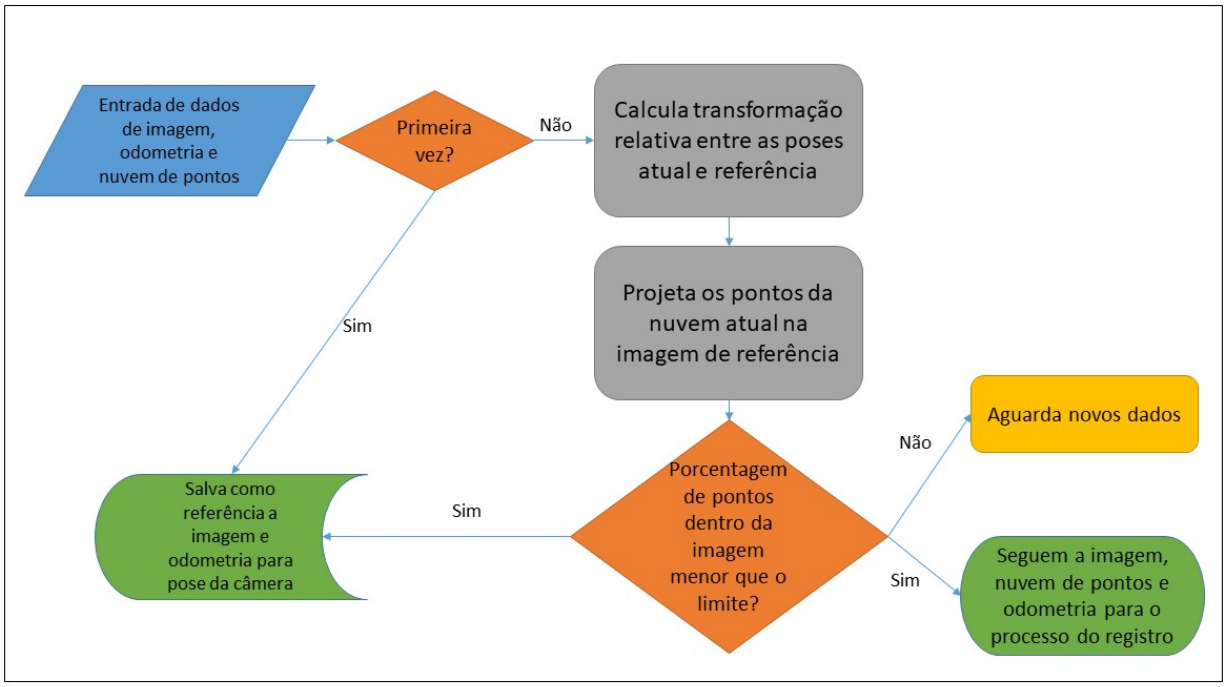
onde ${}^{I}\mathbf{P}_k$ é a matriz da câmera com a transformação entre o *frames* da câmera no instante k e o inercial.

Em contrapartida, a simples soma da Equação 5.3 proporciona várias repetições dos mesmos pontos durante a acumulação, causando deformações e excesso de memória desnecessários. Um algoritmo introduzido nesse ponto, a fim de solucionar esse problema, envolve o conceito de *overlap* entre capturas da cena, mencionado no primeiro item da Sequência final apresentada na Figura 38.

Supondo uma nuvem instantânea capturada no *frame* da câmera no instante k, e a pose relativa ao *frame* inercial ${}^{I}T_c^k$ é tomada como referência. Em teoria, todos os pontos dessa nuvem podem ser projetados sobre as imagens obtidas pela câmera no mesmo

instante, pois daí se originou a nuvem. À medida que a câmera se movimenta em relação ao *frame* inercial, as nuvens obtidas em instantes $k+m$ contêm pontos não vistos inicialmente na pose de referência, logo ao projetar esses novos pontos utilizando a transformação relativa entre poses ${}^{k}T_{k+m}$ os mesmos não se encontram dentro da imagem obtida em k, a referência. Quando P por cento dos pontos obtidos da nuvem $k+m$ não estiverem dentro da imagem de referência, ou seja, o *overlap* entre as cenas for aceitável, esta nuvem pode ser considerada como vinda de uma parte da cena diferente, e então a Equação 5.3 pode ser aplicada para acumular o modelo 3D final, tanto RGB quanto térmico, e concluir a metodologia. Essa solução reduz memória utilizada e custo computacional, por não aplicar o registro de nuvens desnecessárias. O fluxograma da Figura 41 ilustra o processo de análise de *overlap* entre nuvens.

Figura 41 – Fluxograma representando as etapas do algoritmo de *overlap*.



5.4 *FRAMEWORK* ROS

O sistema ROS (*Robotic Operating System*) é um *framework* desenvolvido de forma colaborativa ao redor do mundo por especialistas, com o intuito de padronizar e unir expertises de diversas áreas resultando em um sistema organizado, sincronizado e robusto para aplicações robóticas [98]. Neste trabalho foi utilizada a distribuição na versão *Kinetic Kame*, disponível para o sistema Linux Ubuntu na versão 16.04.

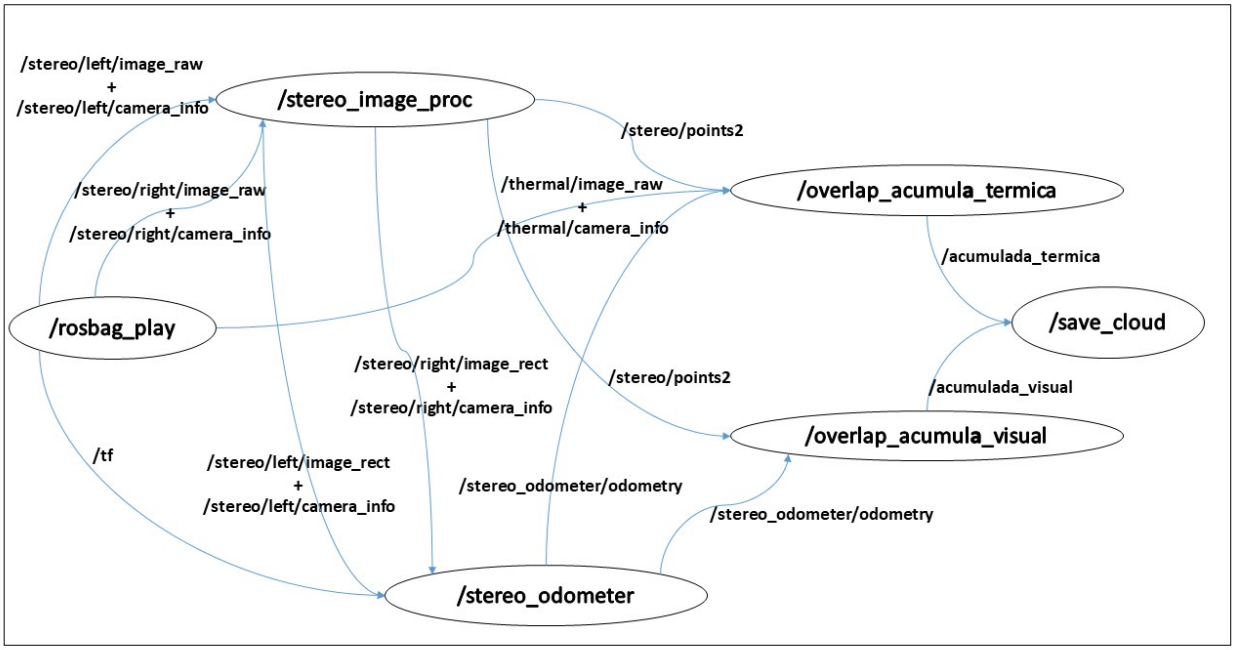
O ROS possui um sistema de comunicação robusto, o *middleware*, o qual sincroniza mensagens para facilidade do usuário. Várias mensagens e funcionalidades comumente

empregadas na robótica, como dados de IMU, lasers, câmeras e sensores de distância já possuem mensagens pré-definidas, enquanto organização de *frames*, processamento e união de dados diversos para aplicações como localização e mapeamento também já estão codificados [99].

A programação envolve as linguagens C++ e Python, abrangendo diversos módulos (denominados pacotes) com os mais variados propositos e algoritmos. O sistema se baseia em um ambiente onde executáveis (denominados nós) recebem e emitem dados (lançados no chamados tópicos) de forma temporizada e organizada por um mestre, processando os mesmos [99]. Os mais diversos tipos de dados são codificados nesses tópicos de forma simples e precisa para o usuário, desde números inteiros até nuvens de pontos extensas.

De forma particular para esse trabalho, o ROS foi utilizado com vital importância para sincronização das imagens vindas das câmeras e processamento sequencial das nuvens de pontos, tanto com atribuição de temperaturas quanto sua acumulação. Existem tópicos para imagens, informações sobre as câmeras, nuvem de pontos e odometria, e nós os recebendo e emitindo, processando as retificações, projeções e nuvens de pontos resultantes. A Figura 42 ilustra a organização dos nós e tópicos como funcionando no sistema criado, com destaque para os mais relevantes.

Figura 42 – Fluxograma simplificado com os principais nós (círculos) e tópicos (setas) do sistema desenvolvido em ROS.



Na Figura 42 pode-se ver o nó */rosbag_play* publicando as imagens cruas e transformações em tópicos para serem processadas pelos nós */stereo_image_proc* (retificação de imagens e processo estéreo de nuvens de pontos) e */stereo_odometer* (odome-

tria visual). As saídas são os tópicos */stereo/points2* para nuvem de pontos e */stereo_odometer/odometry* para odometria, recebidos pelos nós */overlap_acumula_termica* e */overlap_acumula_visual* seguintes para cálculo de *overlap*, projeção de dados térmicos e acumulação das nuvens. Por fim, as nuvens acumuladas são salvas para processamento futuro.

6 RESULTADOS

Nesse capítulo serão apresentados os resultados obtidos com a aplicação dos algoritmos em *softwares* programados para o *framework* ROS, como apresentado no Capítulo 5.

O primeiro resultado, preliminar aos demais, envolve a calibração das câmeras RGB de forma estéreo, e da câmera térmica, com valores e metodologias mostrados na Seção 6.1. De posse das calibrações é possível calcular todos os passos que envolvem a VO e avaliar sua qualidade, o que será feito na Seção 6.2, apresentando a metodologia para tal.

Como continuidade, as nuvens de pontos RGB obtidas de modo estéreo são registradas, com resultados qualitativos apresentados na Seção 6.3. Para avaliação da qualidade das nuvens de forma quantitativa, medições são apresentadas na Seção 6.4.

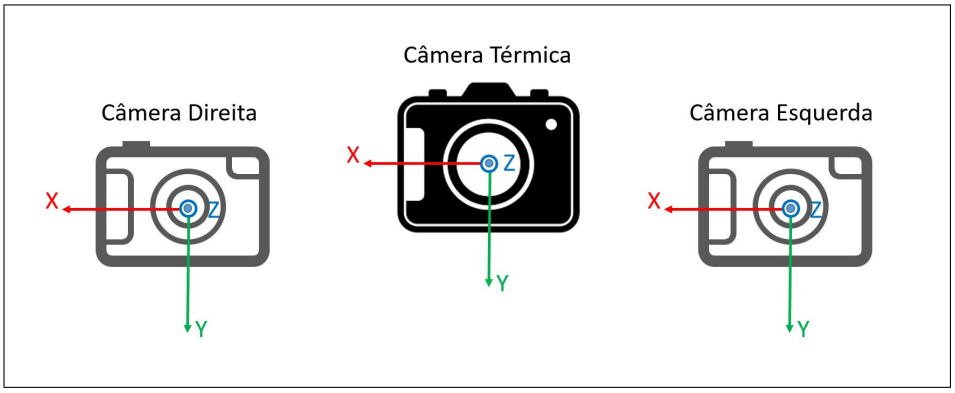
Por fim, a Seção 6.5 traz a nuvem de pontos já avaliada e com os dados de temperatura após projeção dos pontos na imagem térmica. Modelos 3D finais são apresentados e discutidos.

6.1 CALIBRAÇÃO DAS CÂMERAS RGB E TÉRMICA

O primeiro resultado buscado foi em relação à calibração satisfatória das câmeras RGB e térmica, seguindo os procedimentos descritos na Seção 4.4.

Como *frame* referencial foi adotada a câmera esquerda, com origem no centro da câmera, eixo Z apontando para frente, X para a direita e Y para baixo, como ilustrado na Figura 43. As outras câmeras têm eixos dispostos da mesma forma, com seus *frames* tendo origem nos respectivos centros.

Figura 43 – Esquema do conjunto de câmeras como montadas no trabalho, com seus respectivos *frames*.



O procedimento de calibração monocular de cada uma das câmeras RGB retorna seus parâmetros intrínsecos e para correção de distorção, enquanto a calibração estéreo

retorna a matriz de retificação das imagens. A calibração da câmera térmica também é feita de forma monocular, e sua pose é relacionada à referência por rotação e translação implícitas na matriz da câmera **P**.

Os valores das matrizes e parâmetros de distorção se encontram no Anexo A.

6.2 AFERIÇÃO DE ODOMETRIA

A primeira etapa de teste do algoritmo de VO envolveu o uso de um *benchmark* conhecido na literatura, desenvolvido pelos pesquisadores da KITTI (Karlsruhe Institute of Technology and Toyota Institute) em 2011, com o objetivo de uso para pesquisa e avaliação de algoritmo para visão estéreo, odometria visual, SLAM e reconhecimento de objetos [100]. Vários cenários são disponibilizados, dentre eles vias urbanas, rurais e estradas.

Um *benchmark* específico para avaliação da odometria é fornecido pelo trabalho [101], totalizando 10 sequências, das quais 5 foram escolhidas de forma geral em cenários variados para avaliação. Todos possuem uma referência de posicionamento verdadeira, ou *ground truth*, e arquivos com calibração para as câmeras que obtiveram as imagens.

Para avaliar a confiança sobre a odometria visual em aplicação em campo, a mesma foi testada também frente a um *ground truth*, que seria medido durante o trajeto pela câmera ZED, da fabricante StereoLabs. Essa câmera é um produto comercial de código fechado, retornando informações com precisão de milímetros em sua trajetória [102]. A última é posta sobre o conjunto de câmeras utilizado no trabalho.

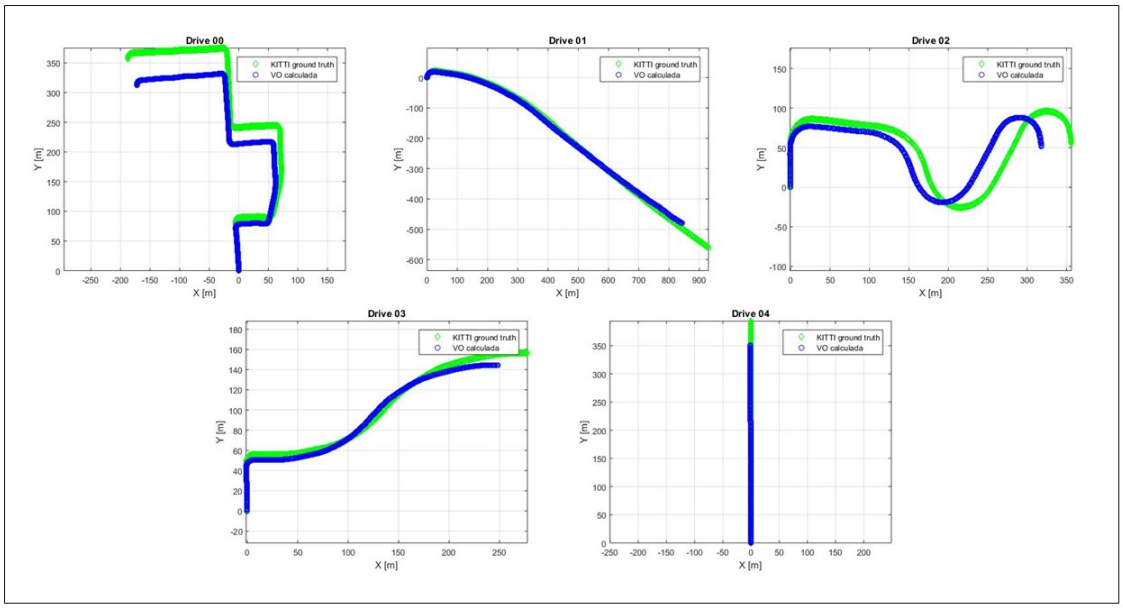
Os trajetos realizados em campo tiveram distância total inferior a 10 metros, distância proposta como suficiente para medição no entorno de um equipamento a ser inspecionado, e os dados de cada fonte foram armazenados. Os mesmos foram feitos com as câmeras posicionadas manualmente, o que em teoria dificultaria a acurácia do algoritmo devido a movimentos irregulares.

6.2.1 **Banco de dados KITTI para VO**

A base de dados KITTI contém diversos arquivos denominados "*Drive XX*", com XX variando entre 00 e 09, obtidos por um veículo equipado com câmeras estéreo, GPS RTK, um LIDAR, entre outros. Estes possuem o posicionamento verdadeiro do veículo ao longo da trajetória, a qual contém trajetos em centros da cidade, bairros residencias, estrada, campo e campus universitário. Os arquivos de 00 a 04 foram escolhidos para teste por apresentarem todos esses cenários.

Os resultados de medição para a odometria são representados na Figura 44. Nela os trajetos são plotados para avaliação qualitativa.

Figura 44 – Resultados do algoritmo de VO (em azul) frente aos caminhos propostos pelo banco de dados KITTI (*groundtruth*, em verde).



A Tabela 6.2.1 resume os resultados percentuais de erro de translação, um dos critérios de avaliação propostos pelo próprio site que fornece o banco de dados, para comparação entre algoritmos. Nela pode ser visto que o erro percentual em todos os trajetos, considerados relativamente extensos frente à aplicação, se mantiveram entre 10 e 11 por cento. Para um estudo focado na aplicação proposta, ao observar somente os 10 metros iniciais de cada cenário, o erro se encontra na maioria dos casos abaixo de 5 por cento, o que não deveria ultrapassar 50 centímetros durante o trajeto de inspeção e é considerado aceitável para o registro de nuvens de pontos, de acordo com trabalhos apresentados no Capítulo 1 e como visto na prática na Seção 6.3. O caso do alto erro para o arquivo *Drive 00* pode se dar ao fato de objetos se moverem na cena durante os 10 primeiros metros, enquanto o ótimo desempenho para o cenário *Drive 04* se deve ao fato de ser o mais retilíneo sobre cena estática, o que é empiricamente determinado por facilitar o cálculo do algoritmo.

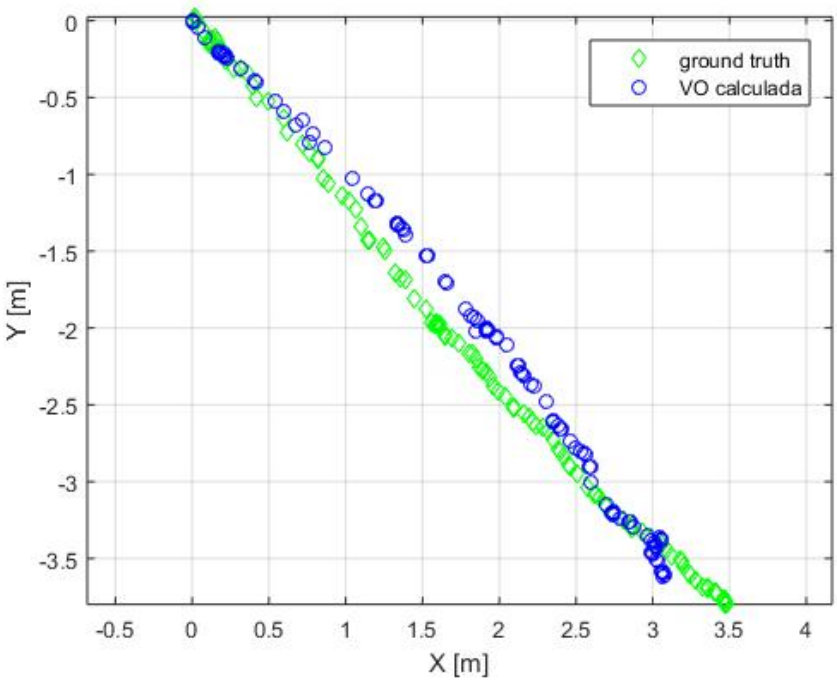
Tabela 2 – Resultados quantitativos percentuais para a translação sobre o banco de dados KITTI. O erro de translação inicial corresponde aos 10 primeiros metros de cada trajeto.

Arquivo	Trajeto total (metros)	Erro de translação total (%)	Erro de translação inicial (%)
Drive 00	685,14	11,09	14,02
Drive 01	1128,50	10,13	3,65
Drive 02	572,92	11,23	2,73
Drive 03	359,93	10,04	4,34
Drive 04	393,56	10,79	0,08

6.2.2 **Cenário de campo 1**

Neste cenário, um trajeto retilíneo foi feito ao longo da cena de inspeção. A Figura 45 ilustra os dados plotados em metros para odometria calculada frente ao *ground truth* no plano 2D, onde houveram maiores deslocamentos.

Figura 45 – Caminho retilíneo apontando para o objeto monitorado, comparando entre resultado do algoritmo e *ground truth*.

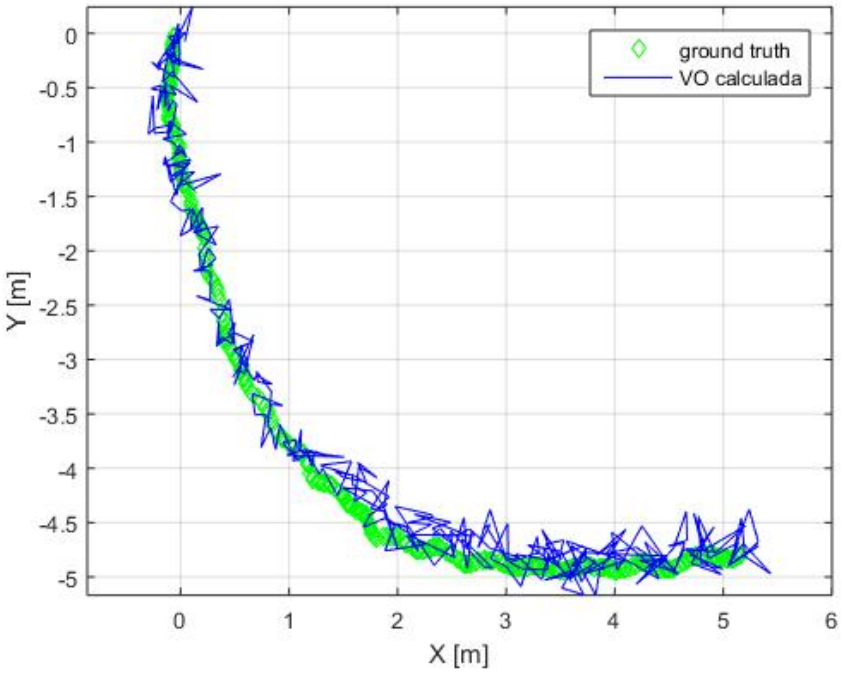


Neste caminho houve um erro médio de 23 e 11 centímetros para X e Y, respectivamente, com erro máximo de 37 centímetros entre as duas referências. O erro percentual de translação total entre as duas medidas esteve em 5,8% do *ground truth*. O resultado se manteve em torno de 6% para os outros 5 testes realizados sob o mesmo critério.

6.2.3 **Cenário de campo 2**

Para o cenário 2 foi realizado um trajeto curvilíneo no entorno de um objeto de interesse de inspeção. Novamente os dados são plotados na Figura 46, para a odometria calculada e *ground truth*.

Figura 46 – Trajeto em arco em torno do objeto inspecionado, comparando o algoritmo ao *ground truth*.



Para o trajeto curvilíneo o erro médio foi de 12 e 10 para X e Y, respectivamente, com erro máximo de 48 centímetros entre as duas referências.

Esse cenário está mais fiel à aplicação, visto que mantém o foco no objeto ao centro da trajetório circular. Outros 4 testes foram feitos de forma semelhante, sempre sobre cenas estáticas, com erros máximos de translação total em 3%.

6.3 NUVENS DE PONTOS RGB QUALITATIVAS

Para avaliação qualitativa da nuvem de pontos, ambientes diversos foram visualizados e testados. Esta é uma forma de análise prévia à quantitativa, para observar o algoritmo de visão estéreo. Como pode ser observado, a qualidade da odometria comprovada previamente garante um registro de nuvem satisfatório no quesito visual.

A noção de profundidade 3D aparenta satisfatória nessa avaliação, bem como formato dos objetos. Os resultados comprovam que ambientes externos ou com maior quantidade de características apresentam uma nuvem de pontos mais rica, como já esperado pela teoria. Em contrapartida, ambientes internos, com iluminação excessiva ou falha, e monocromáticos dificultam a busca por características na etapa *match* do par de imagens estéreo, o que resulta em "buracos" na nuvem de pontos. As imagens a seguir ilustram esses cenários.

Na Figura 47 está reconstruída uma esquina do corredor do prédio da Faculdade de

Engenharia com duas pessoas. É um ambiente de difícil iluminação, pode-se notar ainda alguns buracos na reconstrução, porém a profundidade e formatos gerais do corredor são respeitados.

Figura 47 – Modelo do corredor reconstruído a partir de registro de nuvens, analisado de forma qualitativa.

Nas Figuras 48 e 49 é possível observar reconstruções em 3D do ambiente interno do laboratório. É notável a qualidade visual das nuvens, porém existem buracos pela característica monocromática das paredes.

Figura 48 – Reconstrução do laboratório em 3D.

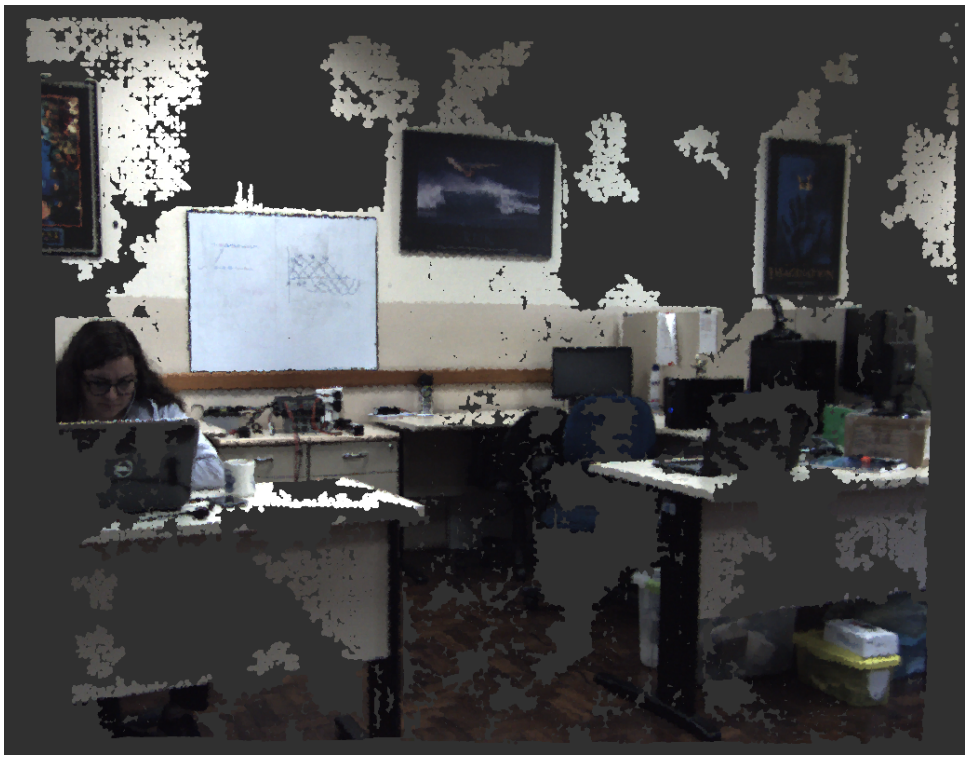
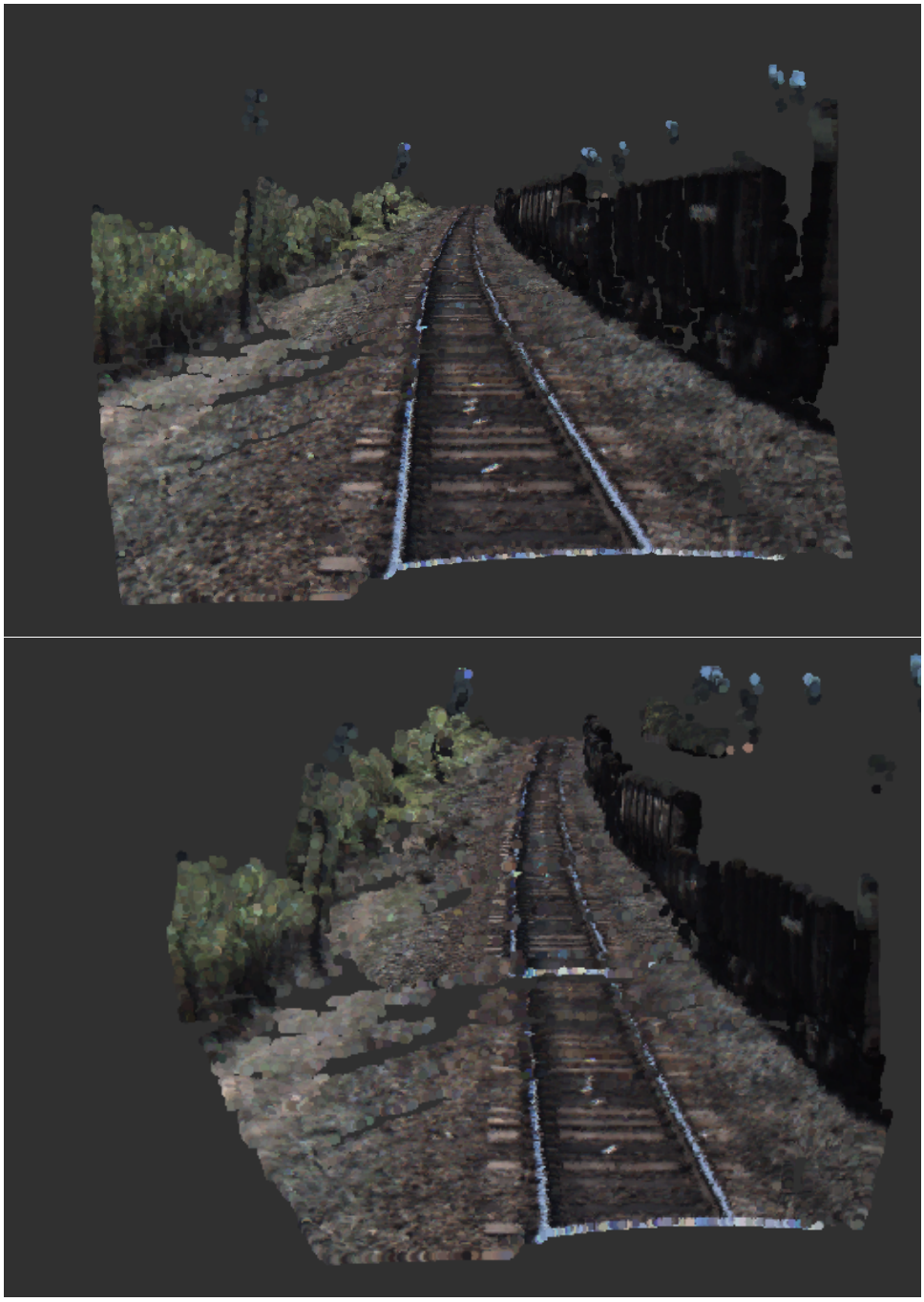


Figura 49 – Reconstrução de outro ponto de vista do laboratório.

Para as Figuras 50 e 51 a cena observada está em um ambiente externo rico em detalhes, em um dia ensolarado, sendo ele uma malha ferroviária com equipamentos a serem inspecionados. Um trecho de 50 metros do trilho é reconstruído, podendo notar os detalhes dos dormentes especificamente na Figura 51.

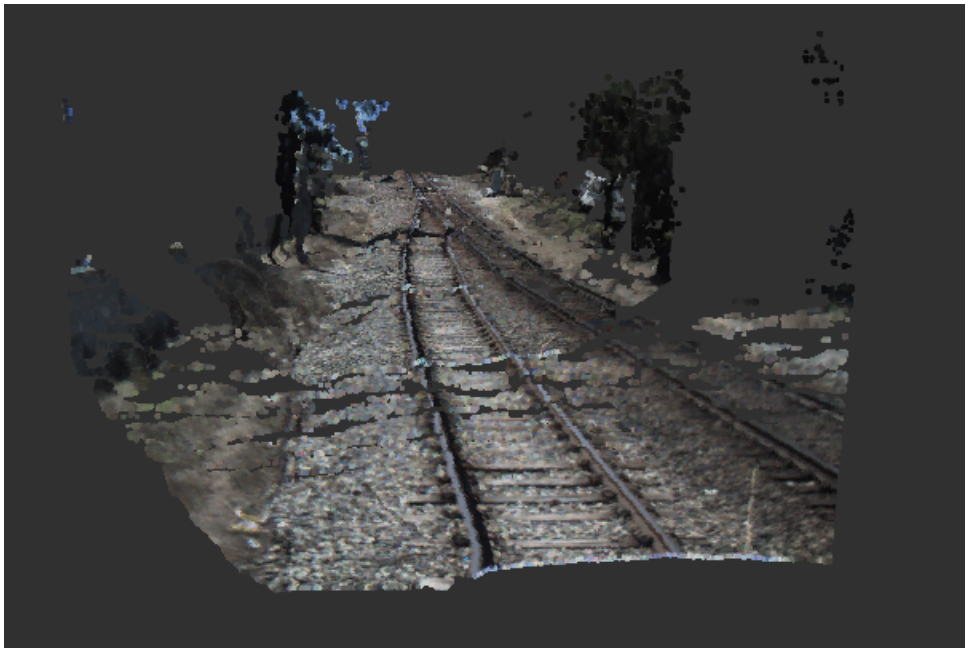
Figura 50 – Imagem obtida pela câmera esquerda do trilho durante inspeção e processo de reconstrução 3D.

Figura 51 – Modelo do trilho a partir de nuvens registradas. Dois pontos de vista podem ser vistos de um trecho de 50 metros.



Por fim, outro trecho do trilho com 30 metros é reconstruído e está exposto na Figura 52. Mesmo com pequenos buracos, a informação principal da cena em destaque é obtida na reconstrução. Vale destacar que informações desnecessárias são também retiradas aqui, como o céu e parte da vegetação, pois trazem confusão à profundidade da cena por estarem mais distantes.

Figura 52 – Trecho do trilho em um novo cenário, reconstruído a partir do registro de nuvens.

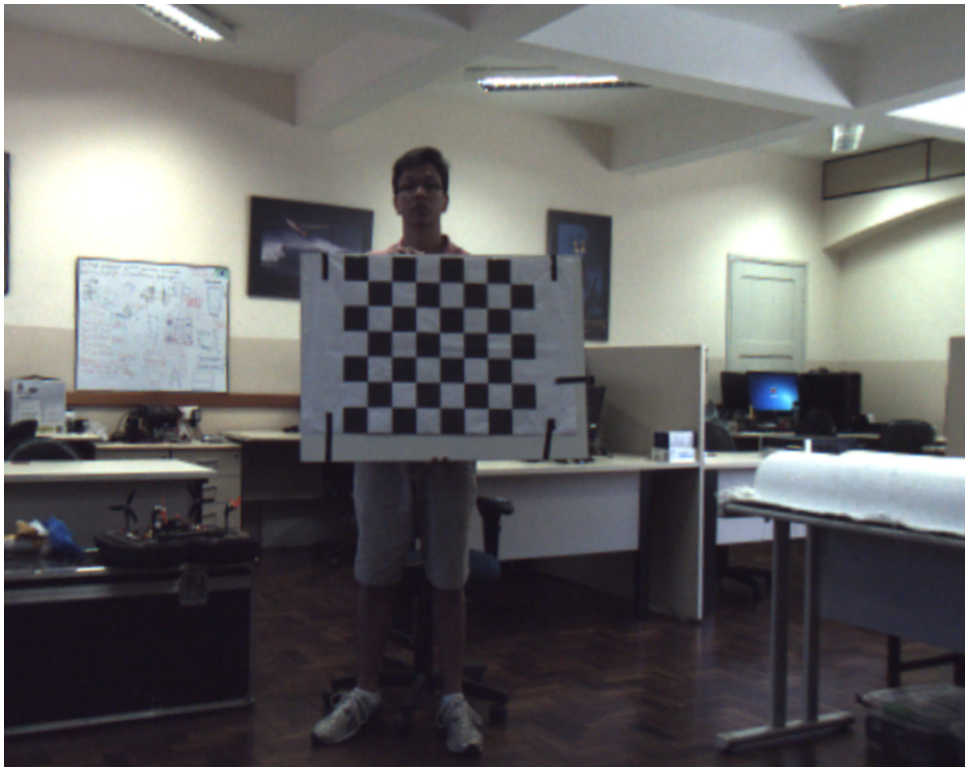


6.4 MEDIDAS REAIS SOBRE NUVENS DE PONTOS

Um dos principais motivos da reconstrução das cenas em 3D diz respeito à obtenção de medidas tridimensionais sobre o objeto reconstruído, desde uma reta no espaço entre dois pontos a cálculos de área e volume.

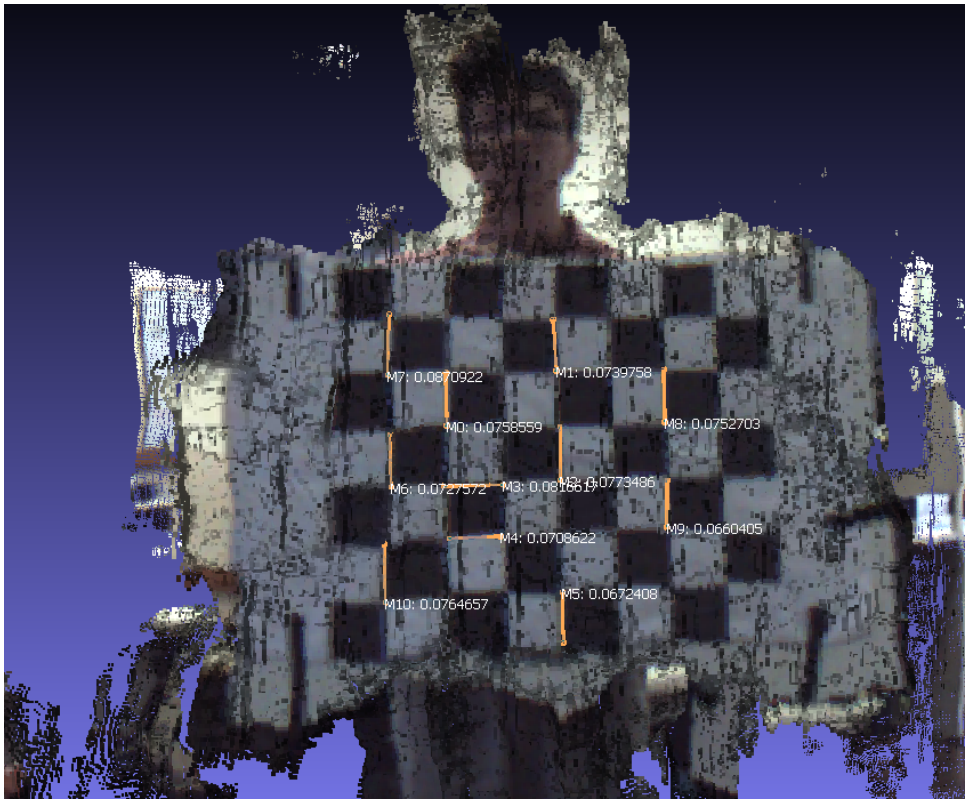
Para comprovar a efetividade do algoritmo sobre as distâncias resultantes na nuvem de pontos (e daí a confiabilidade da mesma sobre o objeto de interesse) foi observado um tabuleiro de xadrez, com comprimentos conhecidos para os lados dos quadrados. A Figura 53 mostra uma vista da câmera esquerda para o tabuleiro impresso, o qual em tese possui 75 milímetros para os lados dos quadrados.

Figura 53 – Tabuleiro observado pela câmera esquerda para aferir dimensões das nuvens de pontos.



Ao acumular nuvens sequenciais, um modelo final 3D da cena vista na Figura 53 foi salvo e utilizado para medir a distância obtida para os lados dos quadrados. Várias medidas aleatórias foram tomadas, a princípio sem grande cuidado sobre o ponto inicial e final, para demonstrar a confiabilidade da medição, o que está ilustrado na Figura 54.

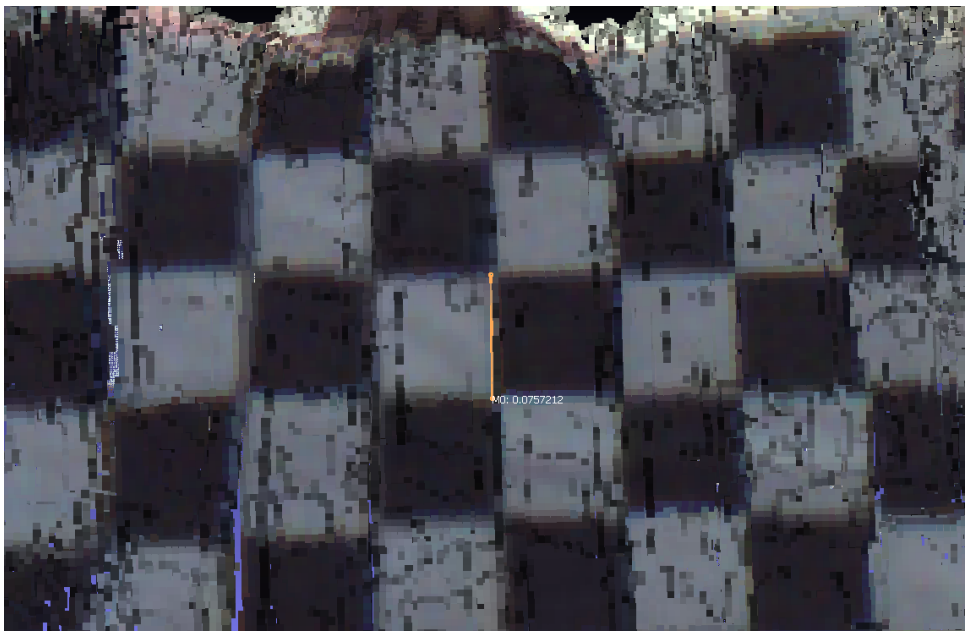
Figura 54 – Medidas aleatórias sobre o tabuleiro de xadrez.



Pode-se com essa imagem ter confiança na ordem de centímetros sobre uma medição linear em um modelo final obtido pelo algoritmo. No caso, a média de 74,2 diferiu 0,8 milímetros do valor esperado (o qual também pode apresentar certas imperfeições), com um desvio padrão de 5,6 milímetros.

Por fim, uma medida mais minuciosa de um dos lados do tabuleiro demonstra que o valor pode ser encontrado com proximidade da ordem de milímetros. A Figura 55 traz o resultado de uma medição mais acurada.

Figura 55 – Medição sobre o lado do tabuleiro de xadrez na nuvem de pontos, com o auxílio de zoom e cuidado na marcação dos pontos inicial e final.



Uma vez analisada a nuvem colorida RGB de forma qualitativa e quantitativa, é possível confirmar a efetividade dos algoritmos de odometria visual e reconstrução estéreo. Pode-se apresentar resultados sobre a reprojecao desses pontos da nuvem sobre a imagem térmica, analisando o modelo 3D térmico.

6.5 NUVENS DE PONTOS TÉRMICAS FINAIS

O último teste para o produto final do trabalho diz respeito à projeção das nuvens sobre a imagem térmica, a fim de obter um modelo 3D térmico do objeto a ser inspecionado.

A câmera térmica pode ser calibrada frente a um ponto de temperatura conhecida, o que foi feito frente ao corpo humano, considerando o mesmo medindo 36 graus Celsius. Segundo o fabricante, a incerteza da medição estaria em ± 5 graus Celsius. Além disso, a resolução da temperatura em *pixels* está na casa dos décimos de graus Celsius.

Frente ao observado na literatura em diferenças de temperatura causadas por falhas em equipamentos elétricos e pontos de estresse térmico, esse valor de incerteza e resolução permite a análise qualitativa sobre os dados térmicos na nuvem de pontos obtida de forma satisfatória.

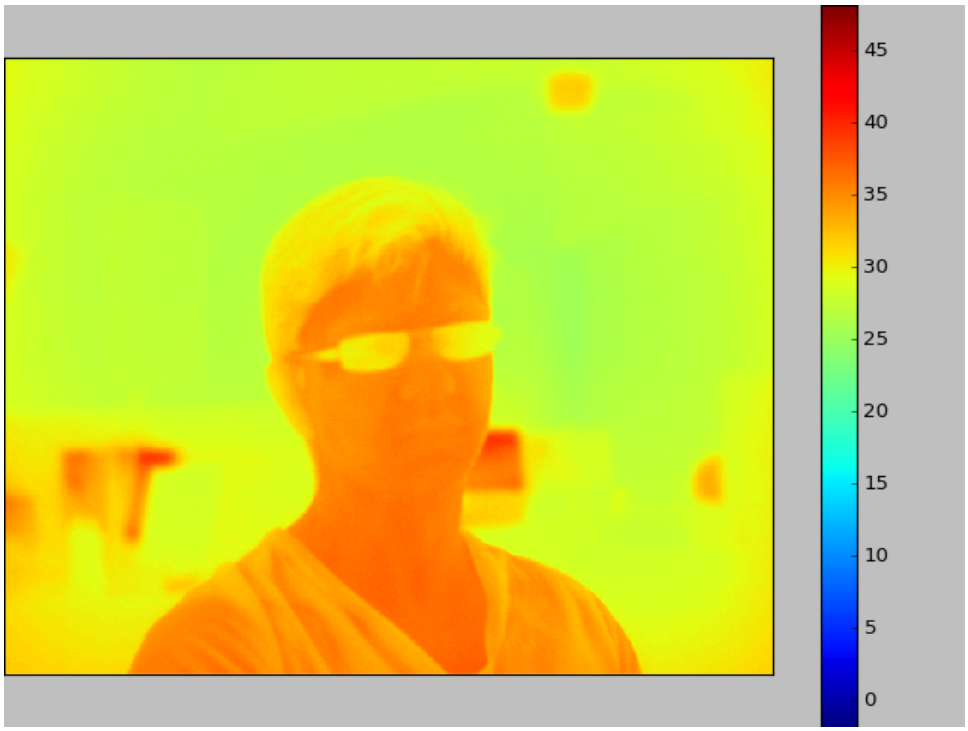
Tendo isso em mente, dois cenários são aqui apresentados para comprovação da técnica, descritos nas subseções a seguir.

6.5.1 **Cenário 1**

No primeiro cenário um teste foi realizado para acurácia da câmera quanto à temperatura, bem como análise de um objeto a princípio conhecido: o corpo humano.

A Figura 56 mostra a câmera visualizando um rosto, com uma escala ao lado direito para a temperatura lida no ponto. A paleta "*Fusion*" (escala de cores para diferenciar temperaturas) é usada aqui para visualizar a imagem.

Figura 56 – Imagem térmica com escala de temperatura ao lado.



Como pode ser visto comparando os pontos do rosto com a escala ao lado, a confiança na medição está inferior à separação da marcação de 5 graus, o que é suficiente para apontar defeitos críticos em equipamentos inspecionados [52].

Com a temperatura aferida, a projeção da nuvem na imagem foi realizada sobre uma pessoa (cujo modelo RGB está na Figura 57) para observar um objeto conhecido térmico em 3D, e o resultado da projeção está apresentado na Figura 58. Pode-se notar a presença da temperatura (em escala) maior para as regiões descobertas do corpo, bem como as mesmas maiores que o restante do ambiente.

Figura 57 – Nuvem de pontos RGB para a cena com uma pessoa e o corredor em profundidade.

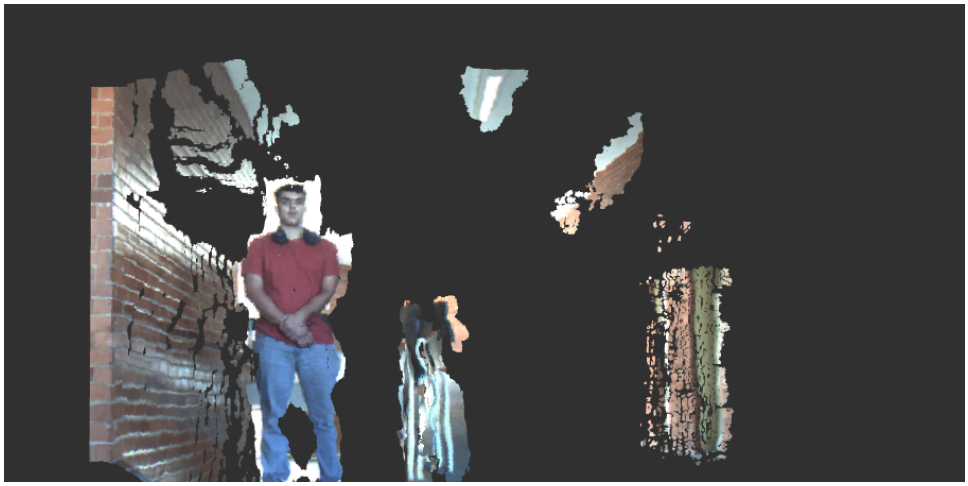
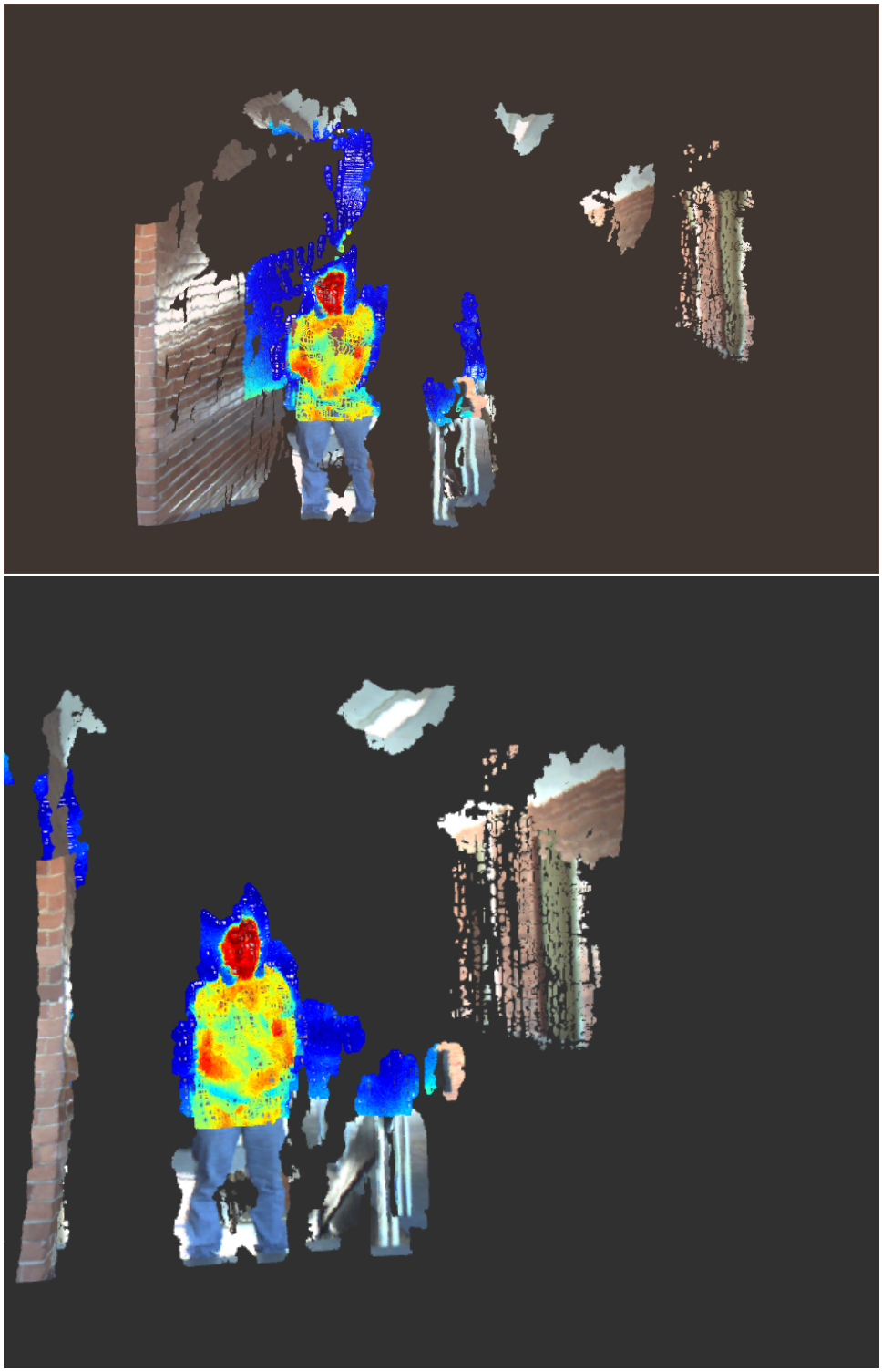


Figura 58 – Projeção da nuvem sobre a imagem térmica analisada de dois pontos de vista diferentes.



Vale ressaltar que o objeto não foi totalmente circulado por limitações físicas do teste, porém é possível interpretar o aspecto tridimensional da cena.

6.5.2 **Cenário 2**

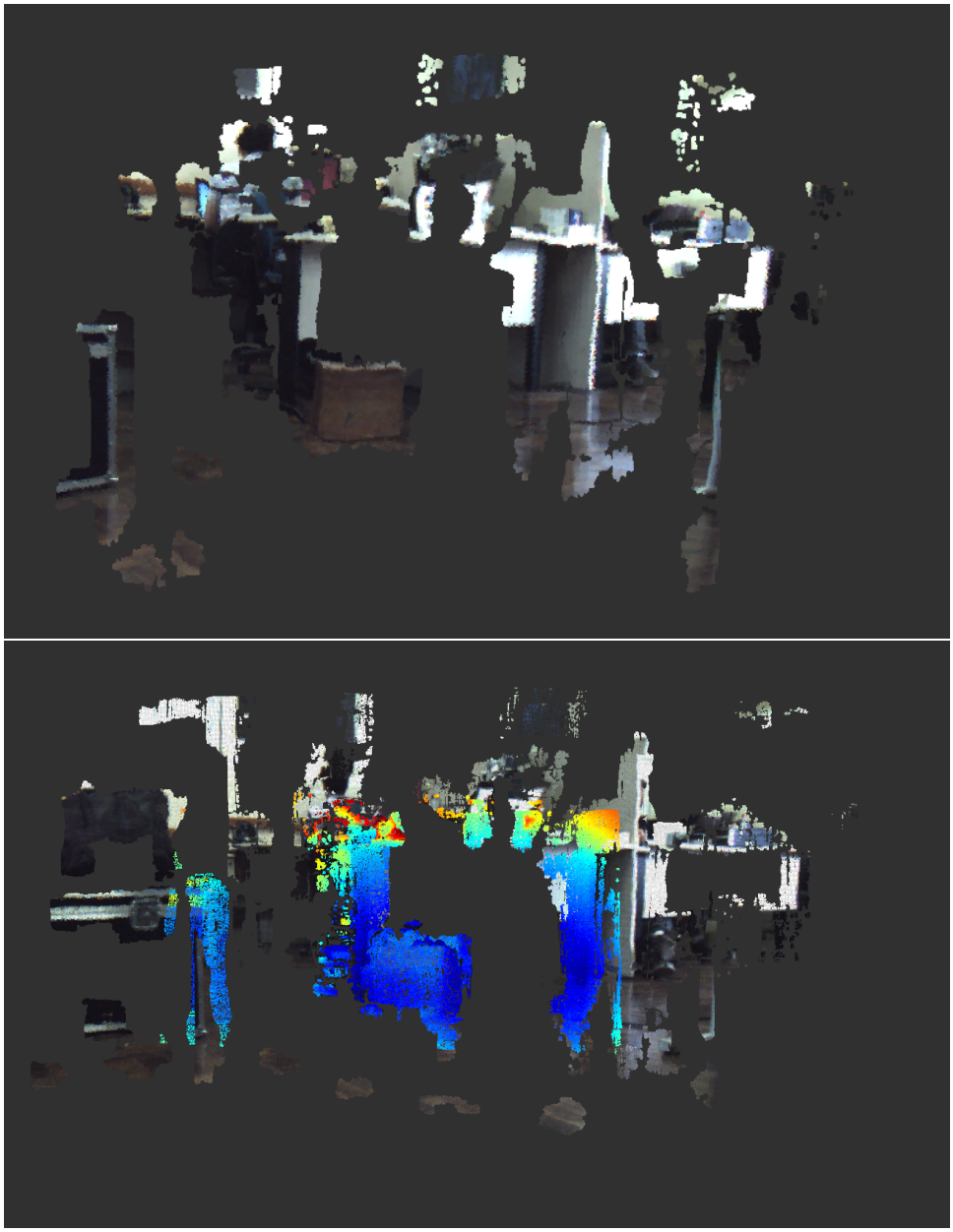
No segundo cenário, mais crítico para comprovar o produto final obtido pelo trabalho, uma caixa contendo uma lâmpada incandescente de 300 Watts (ilustrada na Figura 59) foi utilizada. A caixa simbolizaria um componente a ser inspecionado, com o seguinte objetivo: com a lâmpada desligada, nada deveria ser percebido, porém ao ligar a lâmpada a mesma iria se aquecer, e o ponto quente dentro da caixa deve ser detectado no modelo 3D final.

Figura 59 – Lâmpada incandescente utilizada no experimento.

Em ambiente interno, a primeira tomada da cena visualizou a caixa com a lâmpada desligada. A Figura 60 apresenta uma foto capturada pela câmera esquerda da cena em questão, dentro do laboratório, a qual apresenta clarões e relativamente poucas características frente a um ambiente externo. Sendo assim, a Figura 61 da nuvem RGB apresenta buracos, porém o objeto inspecionado está em destaque.

Figura 60 – Cenário do laboratório com a caixa fechada, visto pela câmera esquerda.

Figura 61 – Nuvem de pontos esparsa RGB do ambiente com a caixa acima, e nuvem térmica abaixo com a lâmpada desligada.



De posse dessa nuvem registrada final, a projeção sobre a imagem térmica aponta uma temperatura baixa e uniforme sobre toda a caixa, o que confirmaria ausência de problemas no componente.

Para a segunda tomada da mesma cena, realizando um movimento semelhante para obtenção dos dados, a lâmpada é ligada e aquecida. A Figura 62 traz o resultado da nuvem de pontos obtida após a projeção sobre a imagem térmica do ponto de vista capturado. O modelo é rotacionado para comprovação do aspecto tridimensional da caixa nessa situação (ilustrado na Figura 63). Por fim, deve-se salientar o ponto quente destacado no modelo, o qual seria interpretado como um defeito pontual no equipamento e foco de manutenção naquela região. É importante ressaltar aqui a utilidade do monitoramento térmico, visto

que em ambas as inspeções as imagens RGB trazem o mesmo resultado de inspeção.

Figura 62 – Nuvem de pontos com projeção da imagem térmica da caixa, com o ponto quente identificando a lâmpada acesa acima. Abaixo, a nuvem térmica isolada.

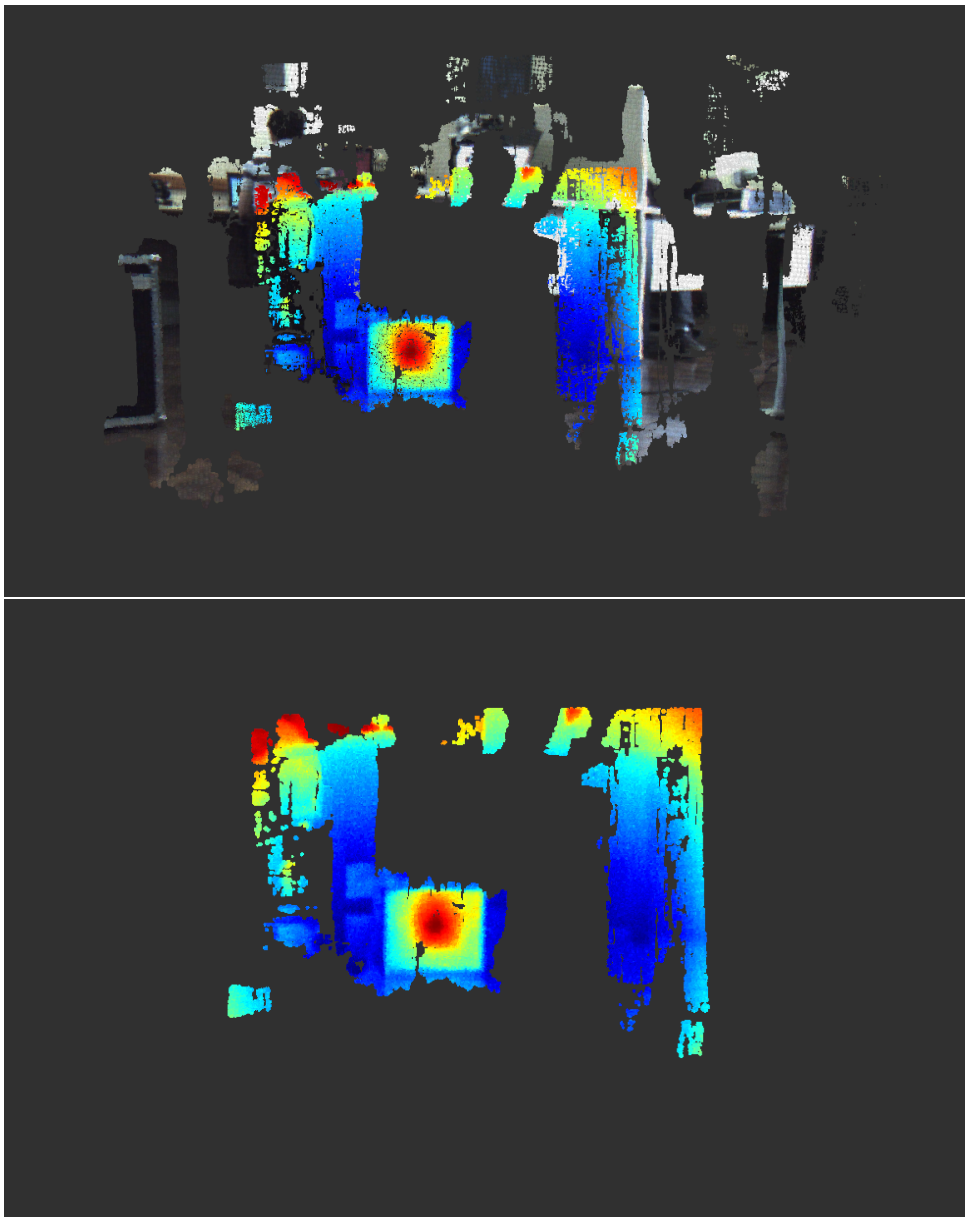
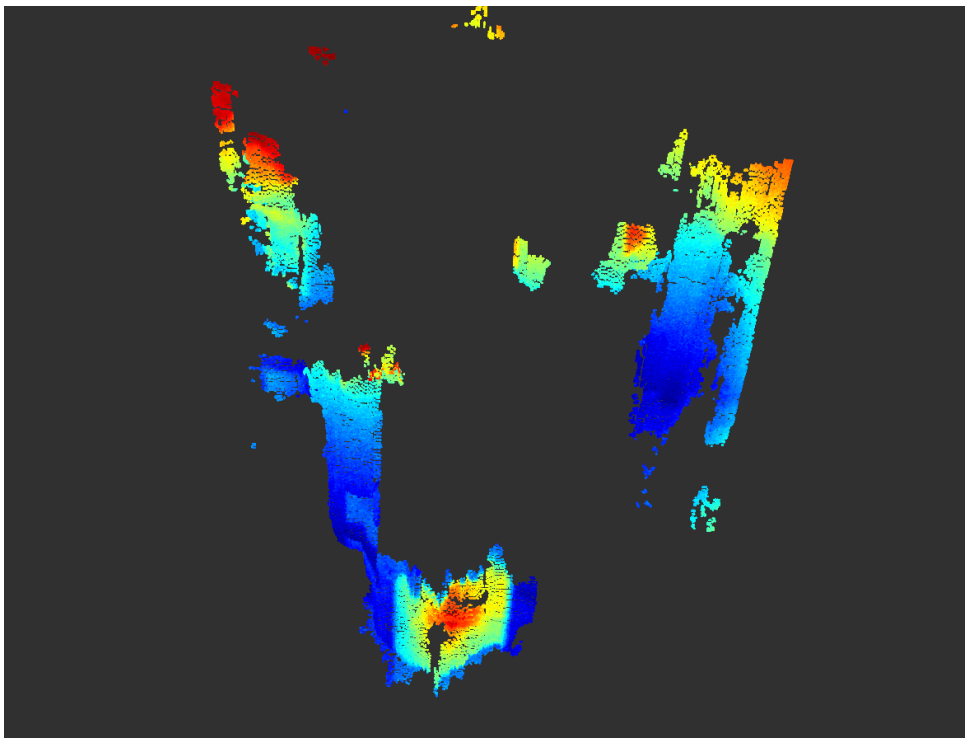


Figura 63 – Outro ponto de vista da nuvem térmica isolada, com destaque para o aspecto tridimensional obtido da caixa e o ponto quente isolado.



7 CONCLUSÃO

Este trabalho apresentou uma metodologia de reconstrução do ambiente real em 3D utilizando técnicas de visão computacional e um conjunto de câmeras estéreo e térmica. Com o resultado é possível obter dados de temperatura sobre cada ponto tridimensional do objeto, e com isso inspecionar frente a possíveis defeitos. Outro resultado é a inspeção do aspecto da cena reconstruída sem a necessidade de estar *in loco*.

Ao longo do trabalho as técnicas foram detalhadas para obtenção de um modelo final a partir do registro de nuvens de pontos. Todo o arcabouço matemático para localização das câmeras no ambiente e estudo de características da imagem é apresentado, a fim de explicar os pontos básicos do trabalho na etapa visual: como obter uma nuvem de pontos a partir de um par de câmeras estéreo, e como realizar a odometria visual a partir de imagens sequenciais sincronizadas. Com isso é possível captar o aspecto tridimensional por um ponto de vista, calcular a variação da pose da câmera para a próxima observação, e com isso unir nuvens de pontos da forma correta, criando modelos 3D.

Para a etapa de captação térmica é apresentada a termologia matemática envolvendo a geometria advinda do modelo da câmera, a qual permite projetar um ponto no mundo real em três dimensões sobre a imagem, em duas dimensões. Com isso, cada ponto do objeto inspecionado recebe um dado de temperatura obtido da câmera térmica, e o modelo final 3D térmico é formado.

Para reduzir o tamanho da memória destinada aos acumulados e repetição dos mesmos pontos foi introduzida a técnica de filtro por *overlap* entre as nuvens e cenas, o que garantiu melhor qualidade e mais eficiência no processo de registro de nuvens de pontos.

Os resultados mostraram confiabilidade da ordem de centímetros para a VO, e medições com erros da ordem de milímetros sobre a nuvem de pontos advinda do algoritmo estéreo, considerados satisfatórios para a aplicação. Quanto à leitura de temperatura, os resultados mostraram ser possível em um *range* de 50 graus obter certeza na escala visual da paleta de 5 graus Celsius, o que é suficiente para detectar anomalias, de forma qualitativa inclusive, sobre equipamentos defeituosos.

Trabalhos futuros são vislumbrados frente aos resultados apresentados, principalmente no que diz respeito à qualidade do modelo final em alguns aspectos. As falhas vistas como “buracos” graças a efeitos de iluminação ou características monocromáticas podem ser contornadas com a aplicação de técnicas como a interpolação de vizinhanças ou aproximações sucessivas. As regiões de busca por características e *match* entre os pares de imagens estéreo podem ser otimizadas em função das características do ambiente, para assim ter ainda mais fidelidade à profundidade dos objetos. Por fim, as bordas entre objetos de profundidades diferentes e entre objeto e fundo, advindas do problema de

oclusão, apresentam um defeito a ser aprimorado.

No tópico de Odometria Visual, possíveis trabalhos futuros podem envolver a fusão de sensores, como IMU e GPS RTK, para obter maior confiabilidade. Técnicas diferentes, fundamentadas em dados de intensidade direta da imagem, podem ser comparadas com a atual a fim de manter a acurácia obtida por uma distância maior de translação.

REFERÊNCIAS

[1] SZELISKI, R. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[2] CORKE, P. *Robotics, vision and control: Fundamental algorithms in matlab® second, completely revised*. Springer, 2017. v. 118.

[3] HARTLEY, R.; ZISSERMAN, A. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[4] LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998.

[5] KURADA, S.; BRADLEY, C. A review of machine vision sensors for tool condition monitoring. *Computers in industry*, v. 34, n. 1, p. 55–72, 1997.

[6] SILVA, L.; VIDAL, V.; SILVA, M.; SANTOS, M.; CARVALHO, A.; CERQUEIRA, A.; HONÓRIO, L.; REZENDE, H.; RIBEIRO, J.; PANCOTI, A. et al. Automatic recognition of electrical grid elements using convolutional neural networks. p. 822–826, 2018.

[7] MA, Y.; SOATTO, S.; KOSECKA, J.; SASTRY, S. S. *An invitation to 3-d vision: from images to geometric models*. Springer Science & Business Media, 2012. v. 26.

[8] ZHU, Z.; XU, G.; YANG, B.; SHI, D.; LIN, X. Visatram: A real-time vision system for automatic traffic monitoring. *Image and Vision Computing*, v. 18, n. 10, p. 781–794, 2000.

[9] JONES, G. A.; PARAGIOS, N.; REGAZZONI, C. S. *Video-based surveillance systems: computer vision and distributed processing*. Springer Science & Business Media, 2012.

[10] NI, K.; STEEDLY, D.; DELLAERT, F. Out-of-core bundle adjustment for large-scale 3d reconstruction. In: . c2007. p. 1–8.

[11] DELAGE, E.; LEE, H.; NG, A. Y. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In: . c2006. v. 2. p. 2418–2428.

[12] HARTLEY, R.; SCHAFFALITZKY, F. Powerfactorization: 3d reconstruction with missing or uncertain data. In: . c2003. v. 74. p. 76–85.

[13] FANWEN, M.; LUSHEN, W.; LIPING, L. 3d point clouds processing and precise surface reconstruction of the face. In: . c2010. p. 104–107.

[14] SHAHZAD, M.; ZHU, X. X.; BAMLER, R. Facade structure reconstruction using spaceborne tomosar point clouds. In: . c2012. p. 467–470.

[15] FORSTER, C.; PIZZOLI, M.; SCARAMUZZA, D. Svo: Fast semi-direct monocular visual odometry. In: . c2014. p. 15–22.

[16] HAMZAH, R. A.; HAMID, A. M. A.; SALIM, S. I. M. The solution of stereo correspondence problem using block matching algorithm in stereo vision mobile robot. In: . c2010. p. 733–737.

[17] MUR-ARTAL, R.; TARDÓS, J. D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, v. 33, n. 5, p. 1255–1262, 2017.

[18] MOHR, R.; QUAN, L.; VEILLON, F. Relative 3d reconstruction using multiple uncalibrated images. *The International Journal of Robotics Research*, v. 14, n. 6, p. 619–632, 1995.

[19] KIM, M.-H.; PARK, J.; KWEON, I. S. Rgbd sensor and mirrors: A practical setup for 3d reconstruction of dynamic objects. In: . c2014. p. 651–652.

[20] EYICE, K.; ÇULHA, O. Lrf assisted slam for airborne platforms. In: . c2018. p. 1131–1134.

[21] GUO, J.; XIAO, X.; PAN, P.; LUO, X. A design of multi-vision localization and navigation service robot system. In: . c2017. p. 787–790.

[22] KUMAR, V.; WANG, Q.; MINGHUA, W.; RIZWAN, S.; SHAIKH, S.; LIU, X. Computer vision based object grasping 6dof robotic arm using picamera. In: . c2018. p. 111–115.

[23] AYACHE, N. Medical computer vision, virtual reality and robotics. *Image and Vision Computing*, v. 13, n. 4, p. 295–313, 1995.

[24] MEIER, L.; TANSKANEN, P.; HENG, L.; LEE, G. H.; FRAUNDORFER, F.; POLLEFEYS, M. Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, v. 33, n. 1-2, p. 21–39, 2012.

[25] VIDAS, S. G. *Handheld 3d thermography using range sensing and computer vision*. 2014. Tese (Doutorado em Física) - Queensland University of Technology, 2014.

[26] BAGAVATHIAPPAN, S.; LAHIRI, B.; SARAVANAN, T.; PHILIP, J.; JAYAKUMAR, T. Infrared thermography for condition monitoring–a review. *Infrared Physics & Technology*, v. 60, p. 35–55, 2013.

[27] DURRANT-WHYTE, H.; BAILEY, T. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, v. 13, n. 2, p. 99–110, 2006.

[28] CADENA, C.; CARLONE, L.; CARRILLO, H.; LATIF, Y.; SCARAMUZZA, D.; NEIRA, J.; REID, I.; LEONARD, J. J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, v. 32, n. 6, p. 1309–1332, 2016.

[29] SHEN, D.; HUANG, Y.; WANG, Y.; ZHAO, C. Research and implementation of slam based on lidar for four-wheeled mobile robot. In: . c2018. p. 19–23.

[30] ADINANDRA, S.; ERFAWAN, D. Yuarm: A low cost android platform for vision based manipulators control. In: . c2016. p. 74–78.

[31] FORSTER, C.; CARLONE, L.; DELLAERT, F.; SCARAMUZZA, D. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, v. 33, n. 1, p. 1–21, 2017.

[32] CVIŠIĆ, I.; ĆESIĆ, J.; MARKOVIĆ, I.; PETROVIĆ, I. Soft-slam: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles. *Journal of Field Robotics*, v. 35, n. 4, p. 578–595, 2018.

[33] ENGEL, J.; SCHÖPS, T.; CREMERS, D. Lsd-slam: Large-scale direct monocular slam. In: . c2014. p. 834–849.

[34] POLLEFEYS, M.; NISTÉR, D.; FRAHM, J.-M.; AKBARZADEH, A.; MORDOHAI, P.; CLIPP, B.; ENGELS, C.; GALLUP, D.; KIM, S.-J.; MERRELL, P. et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, v. 78, n. 2-3, p. 143–167, 2008.

[35] NEWCOMBE, R. A.; IZADI, S.; HILLIGES, O.; MOLYNEAUX, D.; KIM, D.; DAVISON, A. J.; KOHI, P.; SHOTTON, J.; HODGES, S.; FITZGIBBON, A. Kinectfusion: Real-time dense surface mapping and tracking. In: . c2011. p. 127–136.

[36] WHELAN, T.; SALAS-MORENO, R. F.; GLOCKER, B.; DAVISON, A. J.; LEUTENEGGER, S. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, v. 35, n. 14, p. 1697–1716, 2016.

[37] ZHANG, Z.; WAN, W. Dovo: Mixed visual odometry based on direct method and orb feature. In: . c2018. p. 344–348.

[38] TENG, C.-H. Enhanced outlier removal for extended kalman filter based visual inertial odometry. In: . c2018. p. 74–77.

[39] CAI, Z.; YANG, M.; WANG, C.; WANG, B. Monocular visual-inertial odometry based on sparse feature selection with adaptive grid. In: . c2018. p. 1842–1847.

[40] GUI, J.; GU, D.; HU, H. Robust direct visual inertial odometry via entropy-based relative pose estimation. In: . c2015. p. 887–892.

[41] XU, J.; YU, H.; TENG, R. Visual-inertial odometry using iterated cubature kalman filter. In: . c2018. p. 3837–3841.

[42] LI, Y.; ZHONG, X.; TIAN, J.; ZOU, C.; PENG, X. Stereo visual inertial odometry using incremental smoothing. In: . c2018. p. 5334–5339.

[43] SNELL, J.; RENOWDEN, J. Improving results of thermographic inspections of electrical transmission and distribution lines. In: . c2000. p. 135–144.

[44] JADIN, M. S.; TAIB, S. Recent progress in diagnosing the reliability of electrical equipment by using infrared thermography. *Infrared Physics & Technology*, v. 55, n. 4, p. 236–245, 2012.

[45] GENUTIS, D. A. Infrared inspections and applications. `https://www.netaworld.org/sites/default/files/public/neta-journals/NWwtr06NoOutage.pdf`. Acessado: 10/12/2018.

[46] HURLEY, T. J. Infrared qualitative and quantitative inspections for electric utilities. In: . c1990. v. 1313. p. 6–25.

[47] DIAS, F. M. IntegraÇÃo de imagem infravermelha e visual para vistoria de equipamentos, 2017. Monografia, UFJF (Universidade Federal de Juiz de Fora), Juiz de Fora, MG, Brasil.

[48] WONG, W. K.; TAN, P. N.; LOO, C. K.; LIM, W. S. An effective surveillance system using thermal camera. In: . c2009. p. 13–17.

[49] ADÁN, A.; PRADO, T.; PRIETO, S.; QUINTANA, B. Fusion of thermal imagery and lidar data for generating tbim models. In: . c2017. p. 1–3.

[50] CAHYONO, B.; PHARMATRISANTI, A.; SIREGAR, R.; TAMSIR, Y. et al. Automatic thermal monitoring on power transformers. In: . c2008. p. 485–487.

[51] CAO, Y.; GU, X.-M.; JIN, Q. Infrared technology in the fault diagnosis of substation equipment. In: . c2008. p. 1–6.

[52] DRAGOMIR, A.; ADAM, M.; ANDRUÇCÂ, M.; MUNTEANU, A.; BOGHIU, E. Considerations regarding infrared thermal stresses monitoring of electrical equipment. In: . c2017. p. 100–103.

[53] HAM, Y.; GOLPARVAR-FARD, M. An automated vision-based method for rapid 3d energy performance modeling of existing buildings using thermal and digital imagery. *Advanced Engineering Informatics*, v. 27, n. 3, p. 395–409, 2013.

[54] RANGEL, J.; SOLDAN, S.; KROLL, A. 3d thermal imaging: Fusion of thermography and depth cameras. In: . c2014.

[55] LÓPEZ-FERNÁNDEZ, L.; LAGÜELA, S.; GONZÁLEZ-AGUILERA, D.; LORENZO, H. Thermographic and mobile indoor mapping for the computation of energy losses in buildings. *Indoor and Built Environment*, v. 26, n. 6, p. 771–784, 2017.

[56] BORRMANN, D.; NÜCHTER, A.; ĐAKULOVIĆ, M.; MAUROVIĆ, I.; PETROVIĆ, I.; OSMANKOVIĆ, D.; VELAGIĆ, J. A mobile robot based system for fully automated thermal 3d mapping. *Advanced Engineering Informatics*, v. 28, n. 4, p. 425–440, 2014.

[57] GONZALEZ, R. C.; WOODS, R. E. et al. Digital image processing, 2002.

[58] ROCHA, J. C. Cor luz, cor pigmento e os sistemas rgb e cmy. *Revista Belas Artes*, 2011.

[59] VIDAS, S.; MOGHADAM, P. Ad hoc radiometric calibration of a thermal-infrared camera. In: . c2013. p. 1–8.

[60] NORMAN, J. M.; BECKER, F. Terminology in thermal infrared remote sensing of natural surfaces. *Remote Sensing Reviews*, v. 12, n. 3-4, p. 159–173, 1995.

[61] BAJCSY, P.; KOOPER, R. Integration of data across disparate sensing systems over both time and space to design smart environments. In: *Sustainable Radio Frequency Identification Solutions*. InTech, 2010.

[62] SOLEM, J. E. *Programming computer vision with python: Tools and algorithms for analyzing images*. "O'Reilly Media, Inc.", 2012.

[63] LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, v. 60, n. 2, p. 91–110, 2004.

[64] MORAVEC, H. P. Rover visual obstacle avoidance. In: . c1981. p. 785–790.

[65] BAY, H.; TUYTELAARS, T.; VAN GOOL, L. Surf: Speeded up robust features. In: . c2006. p. 404–417.

[66] HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: . c1988. v. 15. p. 10–5244.

[67] ZHANG, Z.; DERICHE, R.; FAUGERAS, O.; LUONG, Q.-T. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence*, v. 78, n. 1-2, p. 87–119, 1995.

[68] JURIE, F.; SCHMID, C. Scale-invariant shape features for recognition of object categories. In: . c2004. v. 2. p. II–II.

[69] MINDRU, F.; TUYTELAARS, T.; VAN GOOL, L.; MOONS, T. Moment invariants for recognition under changing viewpoint and illumination. *Computer Vision and Image Understanding*, v. 94, n. 1-3, p. 3–27, 2004.

[70] WINDER, S.; HUA, G.; BROWN, M. A. Picking the best daisy. 2009.

[71] LOWE, D. G. Object recognition from local scale-invariant features. In: . c1999. v. 2. p. 1150–1157.

[72] MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, v. 27, n. 10, p. 1615–1630, 2005.

[73] CALONDER, M.; LEPETIT, V.; STRECHA, C.; FUA, P. Brief: Binary robust independent elementary features. In: . c2010. p. 778–792.

[74] GOLUB, G. H.; REINSCH, C. Singular value decomposition and least squares solutions. *Numerische mathematik*, v. 14, n. 5, p. 403–420, 1970.

[75] BRADSKI, G.; KAEHLER, A. *Learning opencv: Computer vision with the opencv library*. "O'Reilly Media, Inc.", 2008.

[76] BLACK, M. J.; ANANDAN, P. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, v. 63, n. 1, p. 75–104, 1996.

[77] SCHARSTEIN, D.; SZELISKI, R. Stereo matching with nonlinear diffusion. *International journal of computer vision*, v. 28, n. 2, p. 155–174, 1998.

[78] ZITNICK, C. L.; KANG, S. B.; UYTTENDAELE, M.; WINDER, S.; SZELISKI, R. High-quality video view interpolation using a layered representation. In: . c2004. v. 23. p. 600–608.

[79] TOLA, E.; LEPETIT, V.; FUA, P. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, v. 32, n. 5, p. 815–830, 2010.

[80] ZHANG, Z. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, v. 22, 2000.

[81] HILSENSTEIN, V. Surface reconstruction of water waves using thermographic stereo imaging. In: . c2005. v. 2.

[82] NG, H.; DU, R. et al. Acquisition of 3d surface temperature distribution of a car body. In: . c2005. p. 5–pp.

[83] BOUGUET, J. Y. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`. Acessado: 06/12/2018.

[84] WEISS, S.; SIEGWART, R. Y. Real-time metric state estimation for modular vision-inertial systems. In: . c2011. p. 4531–4537.

[85] NOURANI-VATANI, N.; BORGES, P. V. K. Correlation-based visual odometry for ground vehicles. *Journal of Field Robotics*, v. 28, n. 5, p. 742–768, 2011.

[86] JOHNSON, A. E.; GOLDBERG, S. B.; CHENG, Y.; MATTHIES, L. H. Robust and efficient stereo feature tracking for visual odometry. In: . c2008. p. 39–46.

[87] KASSIR, M. M.; PALHANG, M. Novel qualitative visual odometry for a ground: Vehicle based on funnel lane concept. In: . c2017. p. 182–187.

[88] ZHAO, Q.; LI, F.; LIU, X. Real-time visual odometry based on optical flow and depth learning. In: . c2018. p. 239–242.

[89] SUKVICHAI, K.; WONGSUWAN, K.; KAEWNARK, N.; WISANUVEJ, P. Implementation of visual odometry estimation for underwater robot on ros by using raspberrypi 2. In: . c2016. p. 1–4.

[90] KLEINSCHMIDT, S. P.; WAGNER, B. Visual multimodal odometry: Robust visual odometry in harsh environments. In: . c2018. p. 1–8.

[91] NISTÉR, D.; NARODITSKY, O.; BERGEN, J. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, v. 23, n. 1, p. 3–20, 2006.

[92] NI, K.; DELLAERT, F. Stereo tracking and three-point/one-point algorithms-a robust approach in visual odometry. In: . c2006. p. 2777–2780.

[93] GEIGER, A.; ZIEGLER, J.; STILLER, C. Stereoscan: Dense 3d reconstruction in real-time. In: . c2011. p. 963–968.

[94] HARALICK, B. M.; LEE, C.-N.; OTTENBERG, K.; NÖLLE, M. Review and analysis of solutions of the three point perspective pose estimation problem. *International journal of computer vision*, v. 13, n. 3, p. 331–356, 1994.

[95] FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, v. 24, n. 6, p. 381–395, 1981.

[96] MORÉ, J. J. The levenberg-marquardt algorithm: implementation and theory. In: *Numerical analysis*. Springer, 1978. p. 105–116.

[97] NEUBECK, A.; VAN GOOL, L. Efficient non-maximum suppression. In: . c2006. v. 3. p. 850–855.

[98] About ros. `http://www.ros.org/about-ros/`. Acessado: 13/11/2018.

[99] Ros features. `http://www.ros.org/core-components/`. Acessado: 13/11/2018.

[100] GEIGER, A.; LENZ, P.; STILLER, C.; URTASUN, R. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[101] GEIGER, A.; LENZ, P.; URTASUN, R. Are we ready for autonomous driving? the kitti vision benchmark suite, 2012.

[102] Stereolabs zed camera. `https://www.stereolabs.com/`. Acessado: 08/02/2019.

ANEXO A – Matrizes e parâmetros de calibração das câmeras

Para a câmera **esquerda**, têm-se os dados:

- Matriz intrínseca:

$$\mathbf{K}_l = \begin{bmatrix} 1787.501805657596 & 0 & 792.7050162594984 \\ 0 & 1729.308652099795 & 651.7856114841028 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

- Parâmetros de correção de distorção:

$$\mathbf{D}_l = \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix} = \begin{bmatrix} -0.3181630844613269 & -0.02543114076704484 & -0.001534776655405586 \\ -0.004933917202680872 & 0 & \end{bmatrix} \tag{A.2}$$

- Matriz de retificação:

$$\mathbf{Ret}_l = \begin{bmatrix} 0.9993473202626033 & 0.02221909072845398 & 0.02848237158585052 \\ -0.02241027218252644 & 0.9997283044914058 & 0.006410686347342406 \\ -0.02833219343183327 & -0.007044799921903466 & 0.9995737379550353 \end{bmatrix} \tag{A.3}$$

- Matriz da câmera:

$$\mathbf{P}_l = \begin{bmatrix} 1700.765763170159 & 0 & 709.6987609863281 & 0 \\ 0 & 1700.765763170159 & 630.5447998046875 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{A.4}$$

Para a câmera **direita**, têm-se os dados:

- Matriz intrínseca:

$$\mathbf{K}_r = \begin{bmatrix} 1787.156682643479 & 0 & 754.4216493603013 \\ 0 & 1728.594399744552 & 601.9903735106047 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.5}$$

- Parâmetros de correção de distorção:

$$\mathbf{D}_r = \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix} = \begin{bmatrix} -0.3210895890254977 & -0.08453864094388699 & -0.001115663267294536 \\ -0.002498937084831368 & 0 & \end{bmatrix} \tag{A.6}$$

- Matriz de retificação:

$$\mathbf{Ret}_r = \begin{bmatrix} 0.9995428161869873 & -0.02724945038782177 & 0.01310061306001749 \\ 0.02733699537065068 & 0.999604811705521 & -0.006550503742089108 \\ -0.01291693822435037 & 0.006905640356385523 & 0.9998927266653042 \end{bmatrix} \tag{A.7}$$

- Matriz da câmera:

$$\mathbf{P}_r = \begin{bmatrix} 1700.765763170159 & 0 & 709.6987609863281 & -171.5146569303692 \\ 0 & 1700.765763170159 & 630.5447998046875 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{A.8}$$

Para a câmera **térmica**, têm-se os dados:

- Matriz intrínseca:

$$\mathbf{K}_t = \begin{bmatrix} 1580 & 0 & 330 & -281.15 \\ 0 & 1580 & 270 & 228.16 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{A.9}$$

- Parâmetros de correção de distorção:

$$\begin{aligned} \mathbf{D}_t &= \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix} \\ &= \begin{bmatrix} -0.52910 & 2.240600 & 0.000000 & 0.0000000 & 0.000000 \end{bmatrix} \end{aligned} \tag{A.10}$$

- Matriz da câmera:

$$\mathbf{P}_t = \begin{bmatrix} 1580 & 0 & 330 & -281.15 \\ 0 & 1580 & 270 & 228.16 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{A.11}$$