



UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG  
CENTRO DE CIÊNCIAS COMPUTACIONAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO  
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

## **Mapeamento geométrico e estimativa de pose em operações com guindastes offshore**

Rafael Tavares Guthes

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande - FURG, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação

Orientador: Prof. Dr. Eduardo Nunes Borges  
Coorientadora: Profa. Dra. Silvia Silva da Costa Botelho

Rio Grande, 2025

Dados de catalogação na fonte:

colocar NOME DO BIBLIOTECÁRIO – CRB-colocar número do crb do bibliotecário  
Biblioteca Central – FURG

A999a Guthes, Rafael Tavares

Mapeamento geométrico e estimativa de pose em operações com guindastes offshore / Rafael Tavares Guthes. – Rio Grande, 2025. – 59 f: gráf. – Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG. Centro de Ciências Computacionais. Rio Grande, 2025. – Orientador Eduardo Nunes Borges; Coorientador Silvia Silva da Costa Botelho.

1. 3D. 2. Carga. 3. Câmera. 4. Matriz. I. Borges, Eduardo Nunes. II. Botelho, Silvia Silva da Costa. III. Título.

CDD: 999.9

## DISSERTAÇÃO DE MESTRADO

### Mapeamento Geométrico e Estimativa de Pose em Operações com Guindastes Offshore

RAFAEL TAVARES GUTHES

Banca examinadora:

Documento assinado digitalmente  
 **VAGNER SANTOS DA ROSA**  
Data: 05/04/2025 17:37:10-0300  
Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Vagner Santos da Rosa - FURG

Documento assinado digitalmente  
 **SAMUEL DA SILVA GOMES**  
Data: 05/04/2025 16:57:25-0300  
Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Samuel da Silva Gomes - IFRS

**FURG**  
UNIVERSIDADE FEDERAL  
DO RIO GRANDE  
**EMBRAPII**  
UE - ITec/FURG  
SISTEMAS ROBOTICOS E AUTOMAÇÃO

C=BR, O=CP-Brasil, OU=  
VideoConferência, OU=  
07215063000104, OU=Secretaria da  
Receita Federal do Brasil - RFB, OU=  
RFB e-CPF A3, OU=(em branco),  
CN=EDUARDO NUNES  
BORGES:00057035067  
Rio Grande - RS  
2025.04.05 18:22:03-03'00"

---

Prof. Dr. Eduardo Nunes Borges - FURG  
Orientador

*À minha querida avó, que sempre me ensinou  
que até podem tirar tudo de você, mas jamais sua sabedoria.  
Dedico este trabalho em honra à sua eterna inspiração.*

*"It's not who I am underneath, but what I do that defines me."  
— Bruce Wayne, Batman Begins (2005)*

## RESUMO

GUTHES, Rafael Tavares. **Mapeamento geométrico e estimativa de pose em operações com guindastes offshore**. 2025. 59 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

A segurança e a eficácia no manuseio de cargas durante operações de guindastes *offshore* são cruciais para prevenir danos a colaboradores e à infraestrutura. Um dos principais desafios nessas operações é garantir o transporte seguro das cargas, dado que os operadores enfrentam dificuldades em visualizar e prever os movimentos da carga. Os avanços recentes em visão computacional têm trazido grandes benefícios para aplicações de reconstrução 3D e mapeamento geométrico. Tais técnicas, combinadas com sensores ópticos, permitem o mapeamento preciso e a detecção das cargas transportadas. Neste contexto, este trabalho propõe a reconstrução 3D em escala real em um ambiente simulado e controlado, utilizando câmeras ópticas modeladas em um cenário virtual. A abordagem utiliza imagens não ordenadas e é implementada com bibliotecas de software de código aberto, em conjunto com o simulador ROS/Gazebo. Técnicas como *Structure from Motion* (SfM), *SIFT*, *Multi-View Stereo*, segmentação de instâncias e odometria foram aplicadas para melhorar o desempenho do sistema. Com a representação 3D do ambiente *offshore*, foi possível demonstrar visualmente as capacidades do sistema e facilitar a avaliação da viabilidade e eficácia da abordagem. Este trabalho destaca os principais métodos computacionais para mapeamento geométrico da posição das cargas, com o objetivo de aumentar a segurança e a eficiência nas operações de guindastes *offshore*. Ele também serve como base para futuros avanços em técnicas de mapeamento geométrico e estimativa de posição em ambientes desafiadores.

**Palavras-chave:** 3D, Carga, Câmera, Matriz.

## ABSTRACT

GUTHES, Rafael Tavares. **Geometric Mapping and Pose Estimation in Offshore Crane Operations**. 2025. 59 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

Safety and efficiency in handling loads during *offshore* crane operations are critical to preventing harm to personnel and damage to infrastructure. One of the primary challenges in these operations is ensuring the safe transport of loads, as operators often struggle to visualize and predict the movements of the cargo. Recent advancements in computer vision have brought significant benefits to applications involving 3D reconstruction and geometric mapping. These techniques, when combined with optical sensors, enable precise mapping and detection of transported loads. In this context, this work proposes real-scale 3D reconstruction in a simulated and controlled environment using optical cameras modeled in a virtual scenario. The approach leverages unordered images and is implemented using open-source software libraries, alongside the ROS/Gazebo simulator. Techniques such as *Structure from Motion (SfM)*, *SIFT*, *Multi-View Stereo*, instance segmentation, and odometry were applied to enhance the system's performance. With the 3D representation of the *offshore* environment, it was possible to visually demonstrate the system's capabilities and facilitate the evaluation of the approach's feasibility and effectiveness. This work highlights key computational methods for geometric mapping of load positions, aiming to increase safety and efficiency in *offshore* crane operations. It also serves as a foundation for future advancements in geometric mapping techniques and position estimation in challenging environments.

**Keywords:** 3D, Load, Camera, Matrix.

## LISTA DE FIGURAS

Figura 1	Pilares de atuação do projeto. . . . .	14
Figura 2	FPSO Anita Garibaldi na Bacia de Santos - Agência de Notícias (2023). 16	
Figura 3	Estrutura de um guindaste <i>offshore</i> - <i>American Petroleum Institute</i> (API) em 2020 [4]. . . . .	17
Figura 4	Arranjo geral de uma FPSO. . . . .	18
Figura 5	Reconstrução 3D da zona de carga e descarga. . . . .	18
Figura 6	Ambiente de trabalho do Gazebo . . . . .	20
Figura 7	Mapeamento de pixel em uma imagem RGB - Fonte: [34]. . . . .	21
Figura 8	Esquematisação do modelo de câmera pinhole - fonte: [9]. . . . .	22
Figura 9	Coordenadas do plano da imagem - Fonte: [57] . . . . .	26
Figura 10	Padrão de tabuleiro de xadrez com cantos internos marcados por círculos - fonte: [65]. . . . .	27
Figura 11	Imagens obtidas pela câmera - fonte: [69]. . . . .	27
Figura 12	A posição da imagem calibrada em relação à câmera - fonte: [36]. . . . .	28
Figura 13	<i>Epipolar geometry of stereo vision</i> - [11] . . . . .	29
Figura 14	<i>Feature detector</i> - fonte: [45]. . . . .	34
Figura 15	<i>A sparse 3D point cloud produced by SfM, means initialization</i> - fonte: [22] . . . . .	36
Figura 16	<i>The equation of a line that best fits a set of points can be estimated using RANSAC</i> - fonte: [40]. . . . .	37
Figura 17	<i>HSV color space</i> - Fonte: [28] . . . . .	39
Figura 18	Metodologia proposta. . . . .	40
Figura 19	Posição da câmera I no ambiente. . . . .	42
Figura 20	Posição da câmera II no ambiente. . . . .	42
Figura 21	Tabuleiro utilizado para calibração. . . . .	44
Figura 22	Pontos de correspondência entre um par de imagens. . . . .	45
Figura 23	Fluxograma - Odometria da câmeras. . . . .	45
Figura 24	Fluxograma - Etapa de rastreamento por cor. . . . .	46
Figura 25	Imagens obtidas pelas câmeras no ambiente simulado. . . . .	48
Figura 26	Vetores de posição da Câmera I em relação a Câmera II. . . . .	49
Figura 27	Câmera I - Rastreamento da carga alvo. . . . .	50
Figura 28	Câmera II - Rastreamento da carga alvo. . . . .	50
Figura 29	A arquitetura dos experimentos conduzidos. . . . .	51
Figura 30	Pose do objeto alvo em relação às duas câmeras. . . . .	52

## LISTA DE TABELAS

Tabela 1	Parâmetros intrínsecos calculados das câmeras utilizadas . . . . .	47
----------	--	----

## LISTA DE ABREVIATURAS E SIGLAS

<i>2D</i>	<i>Two-Dimensional</i>
<i>3D</i>	<i>Three-Dimensional</i>
<i>API</i>	<i>American Petroleum Institute</i>
<i>BA</i>	<i>Bundle Adjustment</i>
<i>DOG</i>	<i>Difference of Gaussian</i>
<i>FHD</i>	<i>Full High Definition</i>
<i>FPSO</i>	<i>Floating Production Storage and Offloading</i>
<i>GPU</i>	<i>Graphics Processing Unit</i>
<i>HSV</i>	<i>Hue, Saturation, Value</i>
<i>OpenCV</i>	<i>Open Source Computer Vision Library</i>
<i>ORB</i>	<i>Oriented FAST and Rotated BRIEF</i>
<i>RGB</i>	<i>Red, Green, Blue</i>
<i>ROS</i>	<i>Robot Operating System</i>
<i>SFM</i>	<i>Structure from Motion</i>
<i>SIFT</i>	<i>Scale-Invariant Feature Transform</i>
<i>SURF</i>	<i>Speeded-Up Robust Features</i>
<i>SVD</i>	<i>Singular Value Decomposition</i>

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>13</b>
1.1	Objetivos	15
1.2	Contribuição	15
1.3	Organização do texto	15
<b>2</b>	<b>Referencial Teórico</b>	<b>16</b>
2.1	Operação com cargas em uma FPSO	16
2.2	Segurança e tecnologia	19
2.3	Imagens	21
2.3.1	Formação das imagens	22
2.4	Modelagem da Câmera	23
2.4.1	Parâmetros Intrínsecos	23
2.4.2	Parâmetros Extrínsecos	24
2.4.3	Configuração da câmera	25
2.4.4	Cálculo de Matriz da pose da câmera	27
2.4.5	Geometria Epipolar	29
2.4.6	Matriz Fundamental	30
2.4.7	Matriz Essencial	30
2.4.8	Triangulação	31
2.4.9	<i>Bundle Adjustment</i>	32
2.5	Pontos de Interesse - <i>features</i>	33
2.5.1	Descritores e casamento de pontos	35
2.6	<i>Structure from Motion (SFM)</i>	35
2.7	Random Sample Consensus	36
2.8	Segmentação e Rastreamento Dinâmico em Sequências de Imagens	37
<b>3</b>	<b>Arquitetura e metodologia</b>	<b>40</b>
3.1	Ambiente Simulado	40
3.2	Comunicação	43
3.3	Desenvolvimento Computacional	43
3.3.1	Calibração das Câmeras	44
3.3.2	Odometria	44
3.4	<i>Output</i>	46
<b>4</b>	<b>Resultados</b>	<b>47</b>
4.1	Calibração das Câmeras	47
4.2	Cálculo da Odometria	48

<b>4.3</b> <i>Rastreamento e Detecção da Carga Alvo</i> . . . . .	49
<b>4.4</b> <i>Estimação da Pose</i> . . . . .	51
<b>5</b> <i>Conclusão</i>	<b>53</b>
<b>Referências</b>	<b>54</b>

# 1 INTRODUÇÃO

A visão desempenha um papel fundamental em uma ampla gama de campos, incluindo robótica, automação e visão computacional. No contexto das operações de guindastes *offshore*, a visão é crucial para garantir a segurança e a eficiência durante o manuseio de cargas pesadas. No entanto, este ambiente apresenta desafios únicos devido à sua natureza dinâmica e à complexidade das tarefas envolvidas [20].

Embarcações *offshore* equipadas com guindastes são comumente empregadas em diversas operações, como a instalação de módulos subaquáticos, turbinas eólicas offshore e carga e descarga de equipamentos. No entanto, essas operações são muitas vezes desafiadoras devido a fatores ambientais como vento, ondas e correntes. Uma tendência recente tem sido a utilização de embarcações com guindastes maiores, a fim de reduzir custos. No entanto, essa abordagem pode resultar em maior instabilidade no manuseio das cargas e controle do guindaste durante as operações devido a condições ambientais adversas [20]. Mais especificamente, a principal tarefa dos sistemas de guindaste é manter a carga útil estável em uma posição predefinida, com oscilação residual mínima ou zero, controlando cuidadosamente a lança e ajustando simultaneamente o comprimento do cabo [68].

Este trabalho está inserido em um projeto maior, que busca o desenvolvimento de tecnologias e soluções para aumentar a segurança e eficiência nas operações *offshore*. No diagrama da Figura 1, é possível visualizar os 3 pilares de desenvolvimento do projeto, este trabalho busca trabalhar dentro do pilar de Visão Computacional. Dentro desse escopo mais amplo, o presente estudo foca especificamente em métodos do estado da arte de visão computacional para estimativa de pose. As empresas de exploração e produção de óleo e gás cada vez mais estão buscando reduzir riscos e danos, tanto para seus colaboradores quanto suas estruturas no transporte de cargas e descarga com guindastes. No contexto de plataformas *Floating Production Storage and Offloading* (FPSO), onde o manuseio de transporte de carga e descarga dentro da plataforma representa um alto risco, a segurança não pode depender apenas do projeto. Além disso, é necessário considerar as habilidades humanas, equipes de manutenção e diversos parâmetros técnicos relacionados ao guindaste e sua operação no ambiente [5].

Ao conduzir estas operações, garantir a segurança requer uma abordagem holística. É

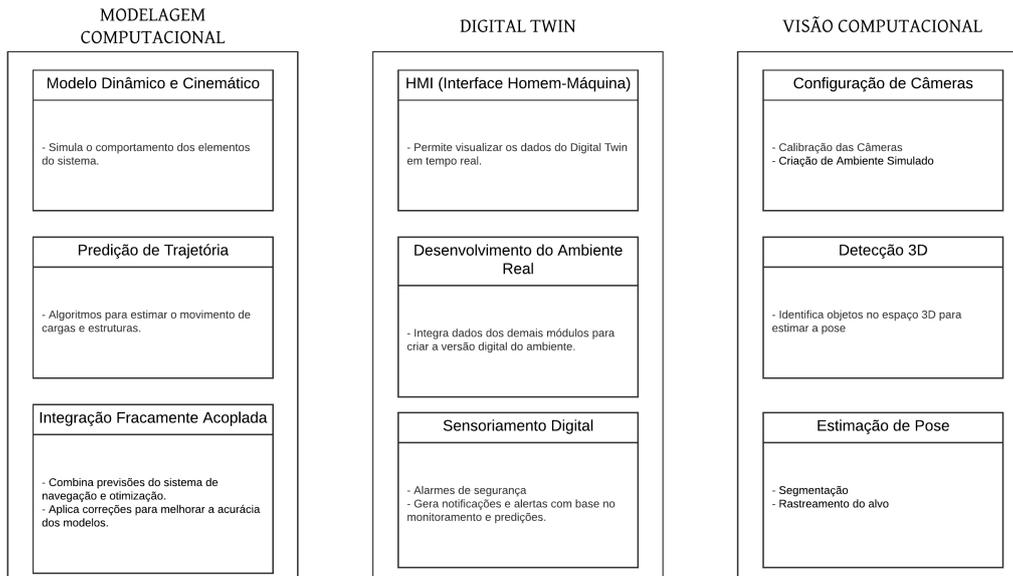


Figura 1: Pilares de atuação do projeto.

crucial combinar tecnologias avançadas de visão computacional, sensores sofisticados e conhecimento especializado dos operadores para criar um sistema robusto. A combinação desses componentes permite o monitoramento em tempo real das operações do guindaste, a identificação de riscos potenciais e o fornecimento de suporte aos operadores [66].

O foco deste trabalho é a utilização de *multiview* através de sensoriamento óptico, aliado a técnicas computacionais, para realizar o mapeamento geométrico e a estimativa de *pose*, com o objetivo de aumentar a segurança e a eficiência nas operações de guindastes *offshore* por meio de métodos computacionais. O aumento da segurança pode ser medido por diversos indicadores, como a redução de incidentes e acidentes registrados nas operações, o que reflete diretamente na eficácia do sistema em prevenir riscos [44]. Além disso, a análise de riscos utilizando ferramentas como a Análise Preliminar de Perigos (APP) e a Análise Quantitativa de Riscos (AQR) permite uma quantificação mais precisa da segurança, comparando os níveis de risco antes e depois da implementação do sistema [5].

Outro aspecto crucial é a confiabilidade e disponibilidade dos equipamentos, monitorada pelo tempo de inatividade e falhas, que diminui com a introdução de tecnologias como sensores ópticos e visão computacional. O tempo de resposta para detecção de falhas ou condições de risco é significativamente reduzido com o uso dessas tecnologias, proporcionando um monitoramento em tempo real das operações [44, 5]. Adicionalmente, a segurança também pode ser mensurada pela eficácia dos treinamentos e simulações aplicados aos operadores, os quais são aprimorados com sistemas de *feedback* e simulações mais realistas, garantindo que os operadores estejam bem preparados para situações críticas. O cumprimento de normas e certificações internacionais, como a *ISO 45001* e as diretrizes da *International Maritime Organization (IMO)*, também serve como

um indicador de que as melhorias implementadas aumentam a segurança em operações *offshore* [44].

## 1.1 Objetivos

O propósito deste trabalho consiste no desenvolvimento de algoritmos fundamentados em técnicas de visão computacional, aliado à criação de um ambiente simulado. Essa abordagem visa permitir a aplicação e teste dos processos implementados, com o objetivo final de gerar um mapeamento tridimensional do ambiente. Esse mapeamento 3D visa a estimação de pose junto a detecção de cargas, estruturas e colaboradores na área específica de operação do guindaste. A fim de alcançar o objetivo geral, os seguintes objetivos específicos foram definidos:

- Desenvolver um ambiente simulado a partir dos dados coletados, utilizando software de modelagem em conjunto com o simulador ROS/Gazebo.
- Implementar técnicas baseadas em cálculos geométricos e algoritmos baseado em visão computacional que auxiliem o mapeamento 3D e estimação de poses dos alvos detectados;
- Implementar um algoritmo capaz de visualizar os processos desenvolvidos.

## 1.2 Contribuição

Neste trabalho, foi abordado o desafio de estimar a pose de um objeto alvo dentro de um ambiente simulado usando quadros capturados por câmeras estrategicamente posicionadas em um cenário 3D construído com base em um ambiente operacional real. O objetivo deste cenário 3D é representar fielmente um ambiente operacional *offshore* real, no processo de transporte de carga manuseadas por guindastes.

## 1.3 Organização do texto

O restante desta dissertação está organizada da seguinte maneira. O Capítulo 2 apresenta o referencial teórico. Ele é dividido em oito seções que abrangem conceitos necessários para o entendimento de todo o trabalho. Em seguida, o Capítulo 3 apresenta a arquitetura do ambiente de simulação, a metodologia proposta e os algoritmos desenvolvidos. No Capítulo 4, os resultados obtidos são apresentados e discutidos. No Capítulo 5, são apresentadas as considerações finais desta dissertação, destacando resultados, contribuições realizadas e as perspectivas para trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

O presente Capítulo apresenta as contribuições teóricas disponíveis na literatura com relação aos temas considerados relevantes à realização desta pesquisa. Para tanto, foram selecionadas artigos científicos, dissertações, teses e livros a fim de subsidiar o referencial teórico. Os principais autores são apresentados no primeiro parágrafo de cada seção, o que possibilita ao leitor maior clareza no entendimento das fontes que nortearam esta investigação.

### 2.1 Operação com cargas em uma FPSO

A sigla FPSO é formada pelas iniciais das palavras em inglês *Floating, Production, Storage and Offloading*. Essas embarcações, em formato de navio, desempenham funções de produção, armazenamento e transferência de hidrocarbonetos [61]. A Figura 2 representa um navio FPSO, destacando os equipamentos de superfície no convés e dois guindastes nas bordas da embarcação.

□Imagens/fspo.jpg

Figura 2: FPSO Anita Garibaldi na Bacia de Santos - Agência de Notícias (2023).

Embora o escopo deste estudo de mestrado seja aplicável a qualquer tipo de unidade operacional, a necessidade surgiu durante o desenvolvimento do projeto de engenharia de uma FPSO, sendo esta dissertação vinculada a um projeto de pesquisa desenvolvido no âmbito da universidade. Essa embarcação enfrenta o desafio de manter uma grande quantidade de itens, espaços limitados e custos elevados com mão de obra de manutenção. Além disso, estima-se que, devido ao desenvolvimento das reservas encontradas no Pré-Sal [14], o Brasil demandará aproximadamente 28 FPSOs entre 2020 e 2030.

Para efetuar as operações de transporte de cargas dentro de uma embarcação FPSO, necessita-se do emprego de guindaste, similares ao exemplificado na Figura 3. A carga pode ser manipulada através do gancho principal, também chamado de moitão, ou utilizado a bola de peso ou gancho auxiliar. A distinção reside no fato de que a bola peso possui uma velocidade de movimentação superior, porém a capacidade de carga inferior. Normalmente são instalados dois guindastes, em bombordo ou boroeste, do lado oposto

das tubulações de chegada e saída da plataforma, conhecidas como *risers*. Esses guindastes são fixados na plataforma sobre um pedestal que é soldado ao casco. A área de atuação deles na plataforma é limitada ao que chamamos de raio de operação, distância ortogonal entre o centro do pedestal até ao moitão.

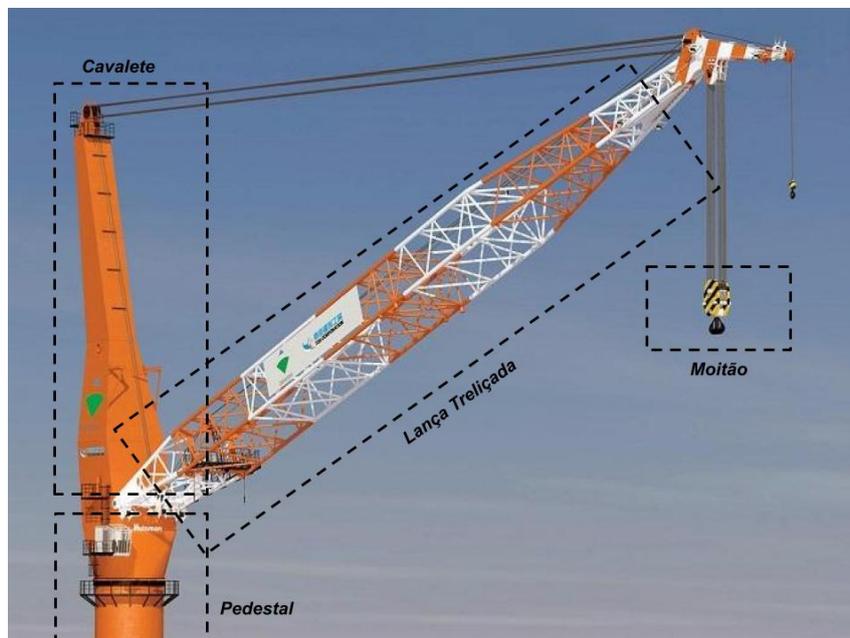


Figura 3: Estrutura de um guindaste *offshore* - American Petroleum Institute (API) em 2020 [4].

Na Figura 4, é apresentada uma planta que exemplifica um arranjo típico de uma FPSO, sendo que a região em verde geralmente abriga as tubulações responsáveis pelas interligações entre os diversos módulos, conhecida como *pipe rack* central. No primeiro piso da área da FPSO é possível visualizar uma zona de formato retangular em coloração vermelha, esta zona é conhecida como *laydown area* onde é realizado todas operações de carga e descarga atribuídas ao guindaste. Nessa representação, os raios de atuação dos guindastes também são mostrados, destacando que a área coberta por eles atende apenas a uma pequena parte da FPSO.

Os guindastes são projetados de forma a evitar colisões entre eles. Para facilitar as operações de carga e descarga dentro da embarcação, são empregadas áreas delimitadas chamadas *laydown*. Essas áreas específicas têm a função exclusiva de receber e retirar cargas manipuladas pelos guindastes e normalmente elas ficam localizadas próximas dos guindastes para facilitar a visão do operador visto que será uma região com alta frequência de movimentação. Na Figura 5, é possível observar a estrutura e os aspectos de uma zona de carga e descarga (*laydown*).

Mesmo diante das restrições para otimizar o arranjo visando a eficiência nas atividades de movimentação de carga, é importante ressaltar que há diversas oportunidades de aprimoramento. A grande quantidade de itens a serem movimentados demanda

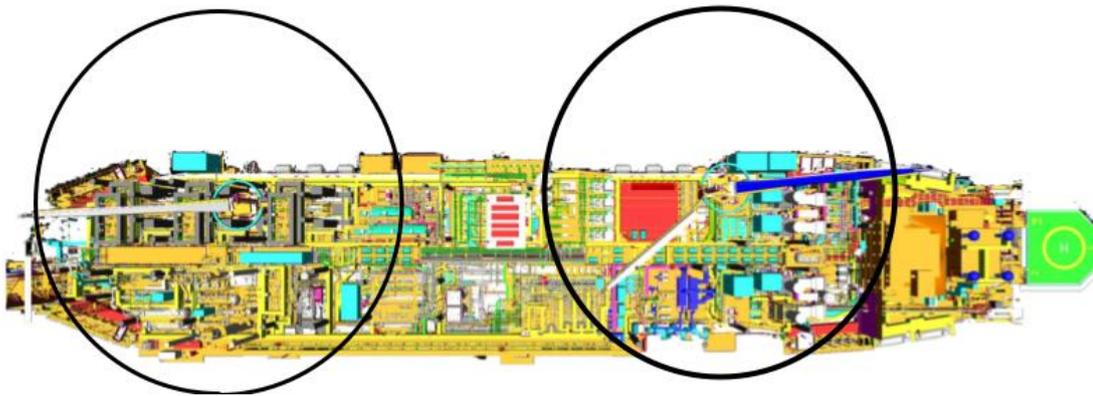


Figura 4: Arranjo geral de uma FPSO.

uma análise mais aprofundada na fase de projeto, proporcionando potenciais melhorias tanto em termos econômicos quanto de eficiência operacional. É crucial reconhecer que obstruções como passagens, colunas, escadas e equipamentos podem não apenas comprometer o movimento, mas também resultar em acidentes e falhas em diversas operações de movimentação de carga [53].

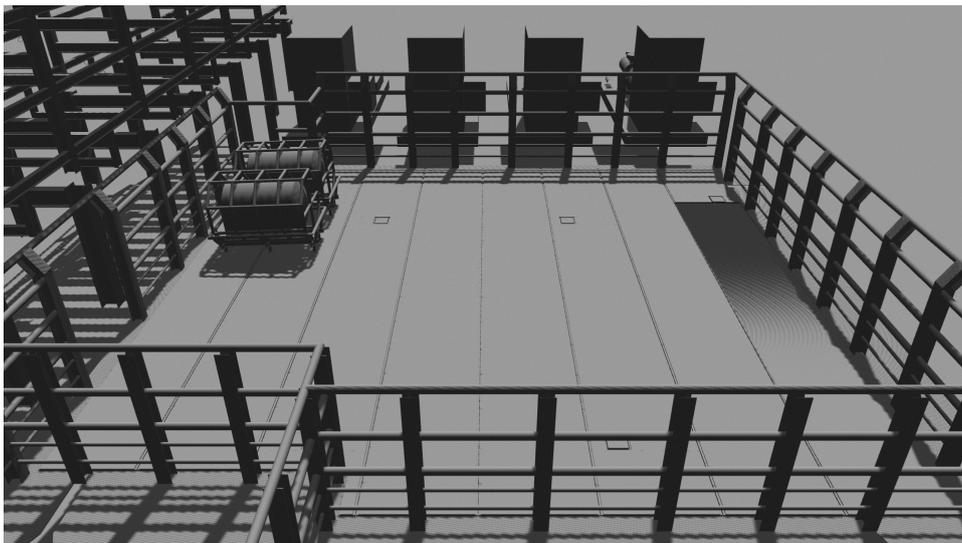


Figura 5: Reconstrução 3D da zona de carga e descarga.

## 2.2 Segurança e tecnologia

A segurança em ambientes *offshore*, como plataformas e FPSOs, é uma preocupação central, especialmente devido à complexidade das operações e aos riscos inerentes às atividades de extração de hidrocarbonetos [41]. A adoção de tecnologias avançadas, como a simulação em ambientes 3D e o uso de ferramentas como o Gazebo, permite uma melhor análise e gerenciamento desses riscos [16].

O Gazebo foi escolhido para simular o ambiente *offshore* por sua capacidade de criar cenários tridimensionais detalhados e interativos [32]. Ele é amplamente utilizado em projetos que exigem simulações realistas, fornecendo um ambiente robusto para testar e visualizar operações, como o transporte de cargas por guindastes, sem a necessidade de realizar essas operações no mundo real. O Gazebo também permite simular ambientes dinâmicos com a inclusão de fenômenos físicos realistas, como gravidade, colisões, vento, e interações fluido-estrutura, aspectos essenciais para o comportamento de plataformas e FPSOs em condições ambientais severas.

A plataforma Gazebo oferece uma variedade de plugins e módulos que permitem a integração com sensores, atuadores e sistemas de controle avançados. Isso possibilita que, em um ambiente de simulação, seja possível controlar e monitorar diversos componentes e sistemas da plataforma de forma integrada. A flexibilidade do Gazebo em suportar diferentes bibliotecas físicas, como ODE, Bullet, e Simbody, torna possível a escolha do motor de física mais apropriado para cada cenário, dependendo das necessidades de simulação [32]. Além disso, sua capacidade de trabalhar com simulações de larga escala, como múltiplas embarcações, plataformas e veículos submarinos operando em conjunto, faz com que seja a escolha ideal para cenários offshore.

Outro aspecto relevante do Gazebo é sua compatibilidade com o ROS (*Robot Operating System*), um middleware que facilita a comunicação entre os diversos sistemas da simulação. Isso permite a criação de sistemas distribuídos que podem ser controlados e monitorados remotamente, algo que reflete diretamente as práticas de controle em ambientes *offshore* reais. A possibilidade de integrar controladores robóticos autônomos também viabiliza o teste de tecnologias de automação, como robôs de inspeção e manutenção, o que aumenta a segurança operacional ao reduzir a necessidade de intervenção humana em áreas de risco.

A principal vantagem do uso de ambientes 3D está na possibilidade de prever e mitigar riscos antes que eles aconteçam, aumentando a segurança das operações [32]. A simulação pode abranger desde a disposição física dos elementos na plataforma, como guindastes, contêineres e maquinário, até a modelagem de operações específicas, como a movimentação de cargas, operações de içamento, ou o transporte de pessoal. Isso não só ajuda a otimizar o layout e o planejamento das operações, como também permite a realização de testes de segurança em cenários críticos, como falhas de equipamento ou

condições ambientais adversas. Na Figura 6, é possível visualizar o ambiente de desenvolvimento do Gazebo, um cenário para reconstruir e projetar simulações em tempo real.

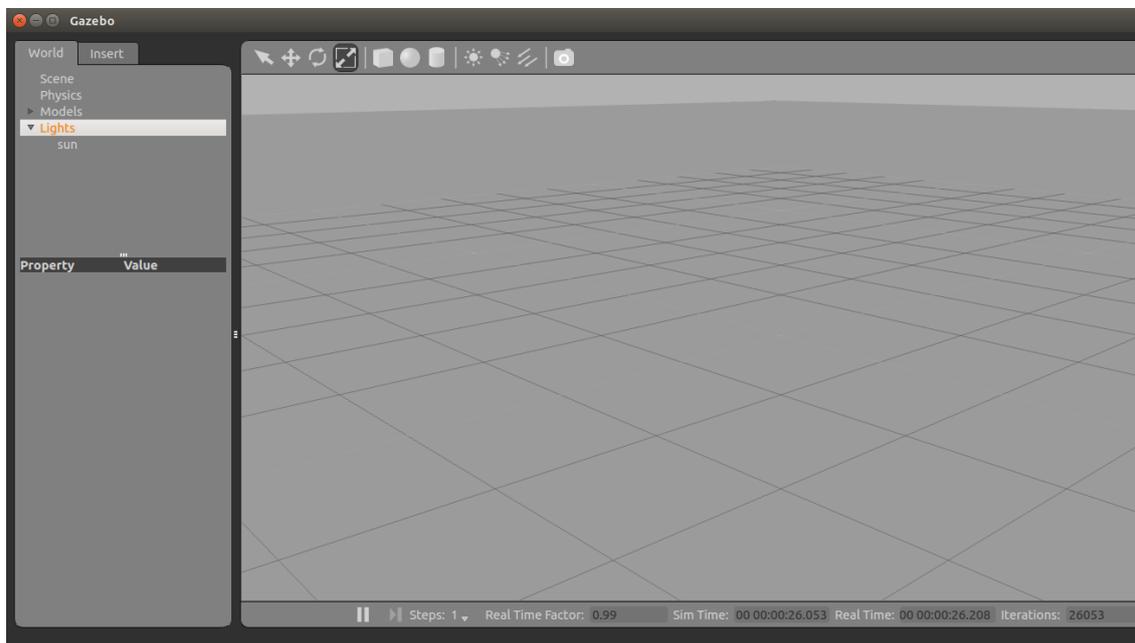


Figura 6: Ambiente de trabalho do Gazebo

*Python* foi a linguagem selecionada para controlar essas simulações devido à sua versatilidade e integração com diversas bibliotecas que facilitam a criação de algoritmos complexos para análise de dados e controle de dispositivos [63]. O uso de *Python* permite a implementação rápida de *scripts* para automação e o controle de eventos, além de sua vasta compatibilidade com bibliotecas como *ROS* e métodos de estimação de pose, como *SIFT*, foi essencial para implementar as funções de monitoramento no ambiente simulado [48].

No contexto de segurança, métricas como o índice de tempo médio entre falhas (MTBF), número de incidentes reportados e tempo de resposta a emergências são frequentemente usadas para avaliar a eficiência dos sistemas de segurança em ambientes *offshore*. Essas métricas são aplicadas para monitorar a eficácia dos procedimentos e equipamentos de segurança. A principal preocupação nesses ambientes é a proteção contra acidentes graves, como incêndios ou vazamentos de gás, que podem ser catastróficos. A capacidade de simular esses cenários em um ambiente 3D ajuda a preparar as equipes para ações rápidas e eficazes em caso de emergência [17].

No entanto, o ambiente 3D proposto não substitui completamente a necessidade de um especialista em segurança. Embora a simulação possa auxiliar em transporte de cargas realizadas por guindastes em uma plataforma petrolífera, as medidas de segurança recomendadas ainda dependem da interpretação de um profissional especializado. Esse especialista, ao visualizar o ambiente 3D, é capaz de fazer recomendações específicas, ajustadas ao contexto operacional da embarcação ou plataforma, garantindo que os pro-

cedimentos de segurança estejam em conformidade com os padrões regulamentares e prontos para lidar com as complexidades do ambiente *offshore*.

## 2.3 Imagens

As imagens desempenham um papel fundamental na reconstrução de modelos 3D. Elas contêm todas as informações necessárias para gerar o detalhamento necessário, em que uma imagem é interpretada pelo computador como uma matriz de números, representando a coloração da imagem.

Ao capturar uma imagem digital, a luz proveniente do objeto é capturada por um conjunto de lentes esféricas e projetada no fundo da câmera. A luz é então convertida em sinais elétricos por meio de um sensor semicondutor. Esses sinais são conduzidos até um chip que converte as informações eletrônicas recebidas em sinais digitais. Em outras palavras, a matriz de números interpretada pelo computador recebe valores discretos conforme a intensidade da luminosidade [34].

Na Figura 7, são ilustrados o processo de captura de imagem e a interpretação de suas informações digitais. À esquerda, temos uma imagem hiperespectral, que capta a luz em várias bandas de comprimentos de onda, proporcionando um espectro completo para cada pixel. No centro, os sensores RGB (azul, verde e vermelho) capturam faixas específicas do espectro visível, resultando em uma imagem RGB, mostrada à direita. Essa imagem RGB, composta por números digitais variando de 0 a 255, é uma matriz de pixels que representa a coloração da imagem. O processo de reconstrução espectral (SR), indicado pela seta na figura, usa os dados da imagem RGB para estimar e reconstruir as informações espectrais detalhadas da imagem hiperespectral original, proporcionando uma compreensão mais rica das características da imagem.

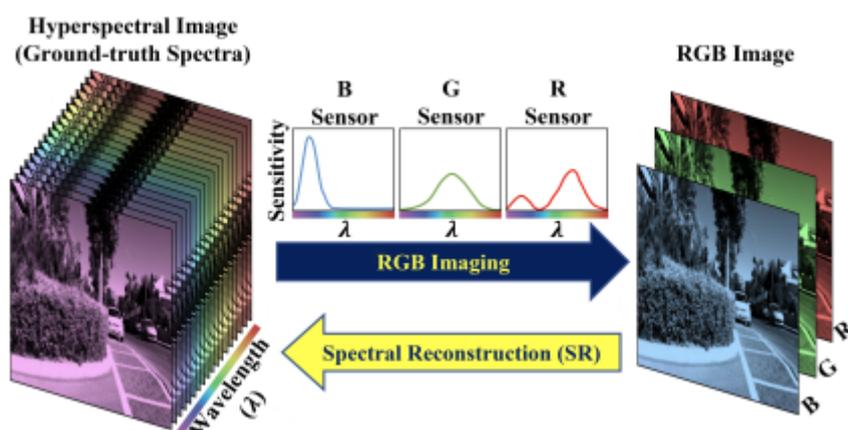


Figura 7: Mapeamento de pixel em uma imagem RGB - Fonte: [34].

Uma imagem pode ser conceituada como a transformação de uma cena tridimensional em um plano bidimensional. Nessa perspectiva, a imagem pode ser interpretada como

uma matriz bidimensional de brilho, como destacado por [39]. Cada elemento dessa matriz é referido como um *pixel*. A quantidade de *pixels* em cada imagem é determinada pela resolução adotada pela câmera. Imagens com maior resolução, que capturam mais detalhes, apresentam um número maior de *pixels*, contribuindo para uma representação mais precisa e nítida da cena tridimensional original.

Para *frames RGB*, cada pixel possui um vetor de três valores discretos, correspondentes a cor vermelha (*Red*), verde (*Green*) e azul (*Blue*). Um imenso conjunto de cores pode ser gerado pela combinação dessas 3 posições.

### 2.3.1 Formação das imagens

De acordo com [12] e [9] a câmera pinhole, também conhecida como câmera estenopeica, é um dispositivo fundamental na área de captura de imagens e visão computacional.

A essência da câmera pinhole reside em sua estrutura mínima, composta por uma caixa escura com um pequeno orifício, chamado de estêncil ou pinhole, em uma de suas faces. Esse orifício permite que a luz proveniente de uma cena externa penetre na caixa escura e projete uma imagem invertida no interior.

Durante o processo de captura de imagem, a luz que atravessa o pinhole segue trajetórias retilíneas até atingir um plano interno. O pinhole atua como uma lente única, eliminando a necessidade de lentes complexas encontradas em câmeras convencionais. Na Figura 8 demonstra o comportamento desse modelo.

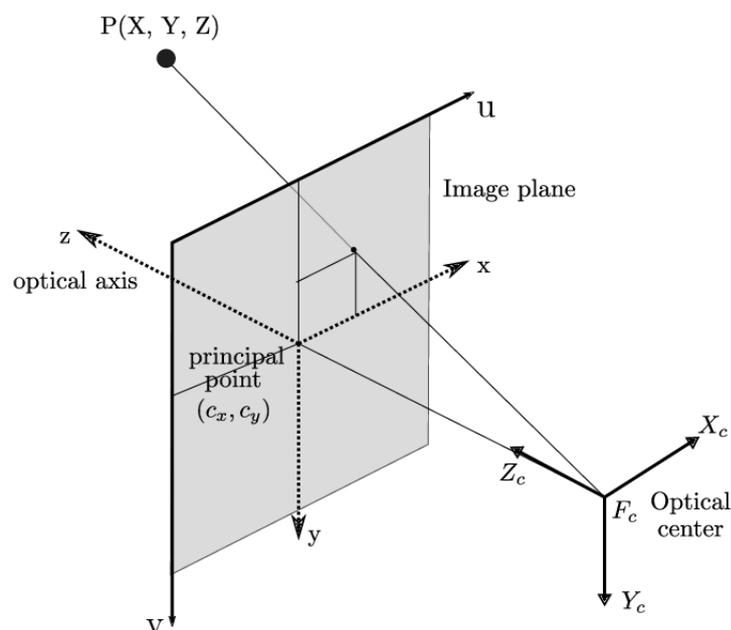


Figura 8: Esquemática do modelo de câmera pinhole - fonte: [9].

A lente permite captar objetos com menores luminosidades para a formação da imagem, no entanto, necessita de ajuste de foco para distâncias específicas. Ao realizar a captura, é gerado um plano de imagem (*image plane*) em frente a  $F_c$  (*optical center*).

Adotando  $F_c$  como origem, esse plano fica a uma distância no eixo de profundidade igual a distância focal da câmera ( $f$ ). A Equação 1 relaciona geometricamente um ponto de coordenadas  $P$  no espaço tridimensional com sua projeção  $p$  no plano da imagem, utilizando o centro óptico da câmera.

$$p = f \frac{P}{Z} \quad (1)$$

Essa equação é uma transformada projetista que possui as seguintes propriedades:

- linhas retas na cena são projetadas como linhas retas no plano da imagem;
- linhas paralelas na cena se interceptam no horizonte no plano da imagem, com exceção de linhas paralelas ao eixo  $x$  do plano da imagem;
- geometrias cônicas são projetadas para cônicas no plano da imagem, mas não necessariamente se mantêm iguais, por exemplo, um círculo pode ser projetado como uma elipse;
- o tamanho e o formato não são preservados;
- o mapeamento não é único, um ponto bidimensional não consegue definir sozinho um ponto tridimensional.

## 2.4 Modelagem da Câmera

A escolha do modelo da câmera influencia diretamente no final dos resultados de calibração, por isso existe uma importância ao selecionar o modelo de câmera apropriado. Esta seção demonstra como são realizados os cálculos dos parâmetros de uma câmera e a descrição da sua modelagem matemática, ou seja, como obter os parâmetros intrínsecos e extrínsecos. Por fim, é demonstrado como estimar os parâmetros intrínsecos, utilizando um processo de calibração através de imagens.

### 2.4.1 Parâmetros Intrínsecos

Os parâmetros intrínsecos de uma câmera são atributos específicos da mesma, tais como a resolução de suas imagens, a distorção causada pela lente e a distância focal. Em outras palavras, esses parâmetros revelam a perspectiva pela qual a câmera percebe o mundo ao seu redor [26].

A expressão que descreve a correspondência entre uma coordenada homogênea do plano de imagens  $(u, v, w)^T$  e as coordenadas de um ponto no mundo real  $(X, Y, Z)^T$  pode ser formalizada pela Equação 2, onde  $f$  representa a distância focal da imagem. O termo  $T$  denota a transposição do vetor, indicando que as coordenadas são representadas como vetores coluna em notação matricial. Nesta expressão,  $w$  representa o fator de escala,

$u$  é a coordenada no plano da imagem ao longo do eixo horizontal (em representação homogênea), e  $v$  é a coordenada no plano da imagem ao longo do eixo vertical (também em representação homogênea).

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2)$$

Esta equação é capaz de expressar uma transformação para câmeras que possuem semicondutores quadrados, implicando que a distância focal nos eixos  $x$  e  $y$  é igual. No entanto, é comum que esses sensores tenham uma configuração retangular, introduzindo uma discrepância na percepção entre os dois eixos. Para refletir esse comportamento, a distância focal é decomposta em duas componentes: distância focal no eixo  $X(f_x)$  e distância focal no eixo  $Y(f_y)$ . Dessa forma, a Equação 2 sofre uma leve modificação, resultando na Equação 3

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3)$$

Ao projetar uma imagem, um componente que sofre variação de acordo com a distorção causada pela lente é o centro óptico, representado nas variáveis  $f_x$  e  $f_y$  na Figura 8. Para obter a modelagem da transformação discreta do espaço 3D para 2D, o centro óptico deve ser deslocado para o centro da imagem e acrescido do deslocamento gerado pela distorção da lente. Isso resulta na Equação 4, onde a matriz obtida corresponde à matriz intrínseca  $K$  da câmera, onde  $P_x$  e  $P_y$  representam as coordenadas do centro óptico.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & P_x \\ 0 & f_y & P_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (4)$$

#### 2.4.2 Parâmetros Extrínsecos

Os parâmetros extrínsecos de uma câmera dependem de sua localização e orientação e nada têm a ver com seus parâmetros intrínsecos, como distância focal, campo de visão etc. Dado um *frame* de referência, as câmeras devem possuir a relação de sua orientação em relação a ele [58]. Essa relação é dada por uma matriz de rotação  $R$  e um vetor de translação  $t$  na Equação 5 é visualizada a matriz de pose onde representa as configurações da câmera. Para câmeras cujo *frame* é igual ao de referência, a rotação e a translação são nulas, como visto na Equação 6.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = K \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} fx & 0 & Px \\ 0 & fy & Py \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6)$$

A matriz extrínseca  $\zeta$  pode ser obtida pela Equação 7. Essa matriz representa a orientação da câmera em relação a um *frame* de referência e pode ser utilizada para projetar informações de um *frame* da câmera para outro *frame* desejado, utilizando sua inversa [33].

$$\zeta = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \quad (7)$$

Por fim, o centro óptico  $C$  da câmera é dado pela Equação 8. Com ele, é possível obter o ponto exato do sensor da câmera no momento da captura.

$$C = -R^T t \quad (8)$$

### 2.4.3 Configuração da câmera

Os parâmetros internos oferecem uma descrição das características ópticas e geométricas internas da câmera, como o centro da imagem, distância focal e distorção da lente, entre outros. Por sua vez, os parâmetros externos incluem a posição tridimensional e orientação do sistema de coordenadas da câmera em relação ao sistema de coordenadas global. O modelo pinhole, frequentemente utilizado [59], que desconsidera a espessura e distorção da lente, pode não representar com precisão a situação real. Na biblioteca multiplataforma para o desenvolvimento de software na área de visão computacional denominada *Open Source Computer Vision Library* (OpenCV)<sup>1</sup>, a calibração da câmera consiste no modelo pinhole e introduz tanto a distorção radial quanto a distorção tangencial [13]. Esse modelo reflete de maneira mais fiel a distorção real da lente em comparação com o modelo pinhole e o modelo de Tsai [21], que apenas introduz distorção radial de primeira ordem.

Suponha que  $P_w(X_w, Y_w, Z_w)$  seja um ponto nas coordenadas do sistema global,  $P_c(X_c, Y_c, Z_c)$  sejam as coordenadas desse ponto no sistema de coordenadas da câmera,  $P(x, y)$  sejam as coordenadas físicas desse ponto no sistema de coordenadas do plano da

<sup>1</sup><https://opencv.org>

imagem, e  $P(u, v)$  sejam as coordenadas de *pixel* no sistema de coordenadas do plano da imagem mostrado na Figura 9.

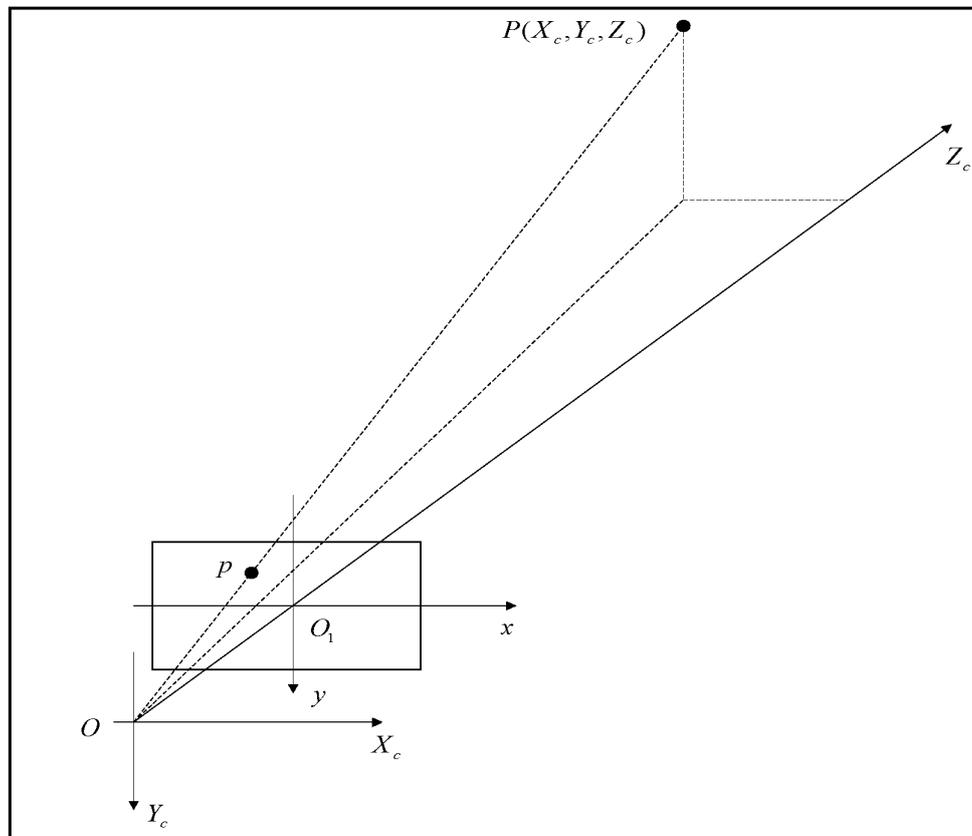


Figura 9: Coordenadas do plano da imagem - Fonte: [57]

As dimensões em *pixels* do objeto podem ser extraídas utilizando os pontos das extremidades do objeto na imagem, sendo  $dx$  o comprimento horizontal e  $dy$  o comprimento vertical. Com a distância  $dZ$  entre a câmera e o objeto, é possível obter as distâncias focais da câmera utilizando as Equações 9 e 10.

$$f_x = \frac{dx}{DX} \cdot dZ \quad (9)$$

$$f_y = \frac{dy}{DY} \cdot dZ \quad (10)$$

onde  $DX$  e  $DY$  são as dimensões do objeto no mundo real. Nessa calibração, é assumido que o ponto principal encontra-se no centro da imagem.

Um dos padrões mais utilizados é o tabuleiro de xadrez. Segundo [8] com seu formato geométrico plano e os formatos dos quadrados que podem ser identificados facilmente. Como os quadrados do tabuleiro possuem tamanhos fixos, seu valor pode ser definido *a priori*. A Figura 10 demonstra como esses padrões são obtidos em um tabuleiro.

Para realizar o método de calibração através da captura de *frames* do tabuleiro de xadrez pode ser utilizada uma implementação em Matlab incluído a biblioteca *OpenCV*

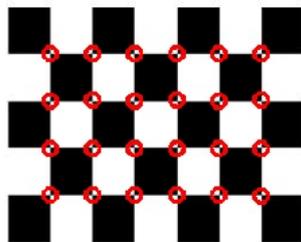


Figura 10: Padrão de tabuleiro de xadrez com cantos internos marcados por círculos - fonte: [65].

*open source* para determinar os parâmetros intrínsecos da câmara. Assim, permitindo calibrar uma câmara através de múltiplas imagens de posições diferentes quadriculada de um tabuleiro observada na Figura [11].

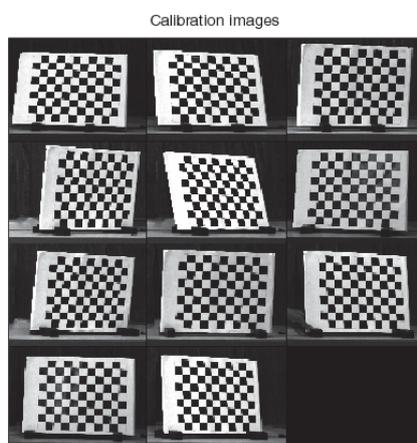


Figura 11: Imagens obtidas pela câmara - fonte: [69].

A partir da dimensão conhecida das quadrículas, o programa determina qual a posição exata no espaço tridimensional que cada ponto do tabuleiro ocupa. Na Figura [12] é demonstrado um método iterativo que vai ajustando os parâmetros intrínsecos da câmara de modo a minimizar o erro de projeção de cada uma das imagens.

#### 2.4.4 Cálculo de Matriz da pose da câmara

Para calcular a matriz de pose da câmara a partir de imagens 2D, são necessárias pelo menos duas figuras com alguma sobreposição. Nesse processo, adota-se o frame da primeira câmara como referência. Ao utilizar pontos correspondentes nas imagens, é possível estimar uma transformada projetiva com o auxílio da matriz fundamental, sem a necessidade de conhecer as características específicas da câmara. Essa transformação relaciona os epípolos das imagens e a matriz fundamental, no entanto, não leva em consideração ângulos e distâncias na cena [56].

Com os parâmetros intrínsecos da câmara em mãos, é viável empregar a matriz essencial para realizar uma transformação mais robusta, preservando as características essen-

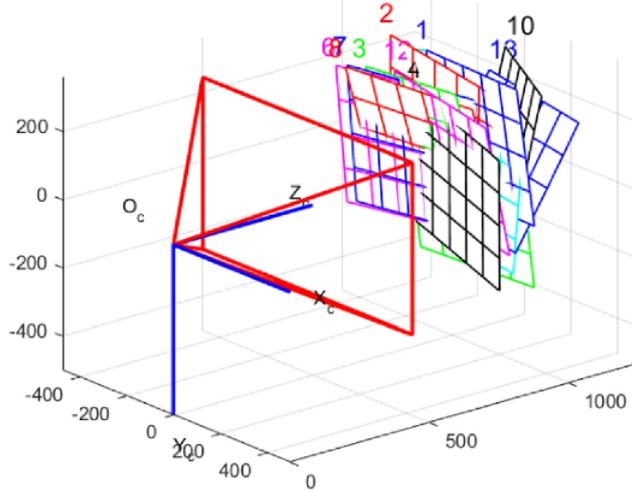


Figura 12: A posição da imagem calibrada em relação à câmera - fonte: [36].

ais da cena. No entanto, esse método não abrange a escala do cenário, pois não é possível determinar o tamanho real do objeto [26].

Ao utilizar os parâmetros derivados da decomposição em valores singulares (SVD) da matriz essencial  $E$  (Equação 20), torna-se possível obter a matriz de rotação  $R$  e o vetor de translação  $t$  da câmera através das Equações 11 e 12.

$R$  representa a orientação da câmera no espaço tridimensional, enquanto  $t$  define a direção do movimento da câmera. As matrizes  $U$  e  $V$  são matrizes ortogonais obtidas a partir da decomposição SVD da matriz essencial  $E$ , e  $U_3$  é a terceira coluna da matriz  $U$ , que fornece a direção da translação. A matriz  $W$ , definida na Equação 13, é uma matriz ortogonal específica usada para garantir que  $R$  seja uma matriz de rotação.

$$R = UWV^T \quad \text{ou} \quad R = UW^TV^T \quad (11)$$

$$t = U_3 \quad \text{ou} \quad t = -U_3 \quad (12)$$

Onde  $W$  corresponde a uma matriz ortogonal dada pela Equação 13.

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Como demonstrado nas Equações 11 e 12, é possível obter duas matrizes de rotação e dois vetores de translação, resultando em quatro soluções possíveis. Para identificar a solução correta, é necessário utilizar um ponto tridimensional e seus correspondentes nas imagens.

A matriz de pose da câmera pode ser representada pela matriz intrínseca  $K$ , que

contém informações como a distância focal e o ponto principal, além da matriz de extrínsecos, composta pela matriz de rotação  $R$  e o vetor de translação  $t$ . Essa relação é expressa na Equação [14].

$$P = K[R|t] \quad (14)$$

### 2.4.5 Geometria Epipolar

A Geometria Epipolar é a geometria que descreve as principais relações entre a perspectiva de duas câmeras. A relação entre duas imagens, baseada na posição dos pontos 3D, possui algumas restrições geométricas, destacando-se os resultados da orientação relativa das câmeras e suas propriedades. Essa relação geométrica é descrita como geometria epipolar, independente da estrutura da cena e considera apenas os parâmetros intrínsecos da câmera [26].

A Geometria Epipolar surge como resposta ao problema de, dadas as coordenadas de um ponto  $x$  em uma imagem e as coordenadas do ponto  $x_0$  correspondente ao mesmo ponto  $x$  no espaço em outra imagem, definir as relações entre os dois pontos.

A Figura [13] deixa claro que os raios retroprojetados (retas definidas pelas junções das imagens  $P_L$  e  $P_R$  com os centros de suas respectivas câmeras  $C_L$  e  $C_R$ ) se interceptam no ponto  $P$  e, portanto, são coplanares. Variando a posição do ponto  $P$ , obtém-se um conjunto de planos em forma de feixe [13], cuja interseção, ou eixo do feixe, é a reta que une os centros das duas câmeras. Ao plano definido pelos raios retroprojetados denomina-se plano epipolar. As retas de interseção do plano epipolar com os planos focais ( $e_L$  e  $e_R$ ) são chamadas de retas epipolares. Os pontos de interseção do eixo do feixe de planos com os planos focais  $e_L$  e  $e_R$  são chamados de epípolos. Todas as retas epipolares se interceptam no epípolo.

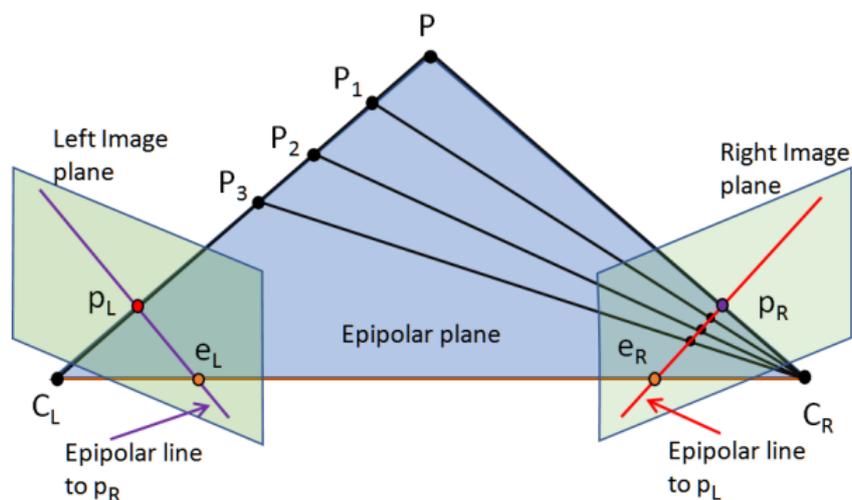


Figura 13: *Epipolar geometry of stereo vision* - [11]

### 2.4.6 Matriz Fundamental

A geometria epipolar pode ser representada algebricamente pela matriz fundamental  $F$ , que possui posto 2, ou seja, apenas duas de suas linhas são linearmente independentes. Essa matriz estabelece a relação entre pontos correspondentes em duas imagens, mapeando um ponto em uma imagem para a respectiva linha epipolar na outra [60]. A condição matemática que os pontos correspondentes devem satisfazer é dada pela Equação 15, onde  $x_1$  e  $x_2$  são as coordenadas homogêneas dos pontos correspondentes.

$$x_1^T F x_2 = 0 \quad (15)$$

Cada par de pontos correspondentes gera uma equação. Para determinar a matriz fundamental, são necessárias pelo menos sete correspondências. Dado um conjunto de pontos na primeira imagem  $[u_{i1}, v_{i1}, 1]^T$  e seus correspondentes na segunda imagem  $[u_{i2}, v_{i2}, 1]^T$ , a equação epipolar pode ser expressa na forma matricial como mostrado na Equação 16. Nela,  $f_{ij}$  representa os elementos da matriz fundamental  $F$ :

$$\begin{bmatrix} u_{11}u_{12} & u_{12}v_{11} & u_{12} & v_{11}u_{11} & v_{11}v_{11} & v_{11} & 1 & u_{11} & v_{11} \\ u_{21}u_{22} & u_{22}v_{21} & u_{22} & v_{21}u_{21} & v_{21}v_{21} & v_{21} & 1 & u_{21} & v_{21} \\ \vdots & \vdots \\ u_{n1}u_{n2} & u_{n2}v_{n1} & u_{n2} & v_{n1}u_{n1} & v_{n1}v_{n1} & v_{n1} & 1 & u_{n1} & v_{n1} \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0 \quad (16)$$

### 2.4.7 Matriz Essencial

Um aperfeiçoamento da matriz fundamental é a matriz essencial ( $E$ ). A matriz essencial é obtida quando as coordenadas das imagens são normalizadas, ou seja, a câmera já foi calibrada e os parâmetros intrínsecos são conhecidos. Isso ocorre porque a normalização das coordenadas remove os efeitos da matriz de calibração da câmera, garantindo que a matriz essencial contenha apenas informações sobre a posição e orientação relativa entre as câmeras e os pontos 3D correspondentes [19]. Dessa forma, a matriz essencial pode ser expressa pela Equação 17, onde  $K$  é a matriz de calibração da câmera.

$$E = K^{-T} F K^{-1} \quad (17)$$

A matriz essencial possui cinco graus de liberdade, pois depende da matriz de rotação e do vetor de translação da câmera. Isso é um número menor do que os sete graus de

liberdade da matriz fundamental. A matriz essencial pode ser obtida a partir da relação entre a translação relativa  $t$  e a rotação relativa  $R$  das câmeras, conforme demonstrado nas Equações [18] e [19]. Embora tanto a rotação quanto a translação possuam três graus de liberdade cada, a relação cruzada impõe restrições que reduzem um grau de liberdade adicional [67].

$$E = [t]_{\times} R \quad (18)$$

$$E = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} R \quad (19)$$

A matriz essencial pertence ao espaço  $\mathbb{R}^{3 \times 3}$  e possui dois autovalores iguais e um autovalor nulo. Isso permite que ela seja decomposta através da Decomposição em Valores Singulares (SVD), conforme mostrado na Equação [20], onde  $U$  e  $V$  são matrizes ortogonais, e  $\lambda$  representa os autovalores não nulos [39].

$$E = U \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T \quad (20)$$

#### 2.4.8 Triangulação

A triangulação é o processo responsável por converter correspondências de pontos entre imagens em coordenadas tridimensionais. Baseando-se em modelos matemáticos simplificados da câmera, a triangulação pode produzir resultados precisos, especialmente quando as matrizes de poses das câmeras são bem calibradas [42].

Dado que um ponto  $X$  é visível em duas imagens, o problema da triangulação consiste em determinar a interseção dos raios projetivos correspondentes a esses pontos no espaço tridimensional.

Para um par de imagens com matrizes de poses definidas como  $P_1$  e  $P_2$ , e pontos correspondentes  $x_1 = [u_1, v_1, 1]^T$  e  $x_2 = [u_2, v_2, 1]^T$  no espaço da imagem, a técnica de triangulação  $\tau$  pode determinar a posição tridimensional do ponto correspondente  $X = [X, Y, Z, 1]^T$ , conforme expresso na Equação [21].

$$X = \tau(x_1, x_2, P_1, P_2) \quad (21)$$

Na triangulação linear, realiza-se o produto vetorial entre cada ponto da imagem  $x_i$  e o vetor resultante da multiplicação  $P_j^T X$ , eliminando assim o fator de escala linear. Esse processo gera duas equações lineares independentes para cada imagem, conforme demonstrado na Equação [22], onde  $P_j^T$  representa a matriz de pose da câmera  $j$  [27].

$$A = \begin{bmatrix} x_1 P_1^{3^T} - P_1^{1^T} \\ y_1 P_1^{3^T} - P_1^{2^T} \\ x_2 P_2^{3^T} - P_2^{1^T} \\ y_2 P_2^{3^T} - P_2^{2^T} \end{bmatrix} X = 0 \quad (22)$$

#### 2.4.9 Bundle Adjustment

O *Bundle Adjustment* (BA) é o método preferido em muitas aplicações de fotogrametria e desempenha um papel fundamental na visão computacional para reconstrução 3D e estrutura a partir do movimento. Erros de medição nas imagens são comuns, podendo violar as relações epipolares. Assim, o *Bundle Adjustment* é um método unificado para estimar simultaneamente os parâmetros intrínsecos (2.4.1) e extrínsecos (2.4.2) das câmeras, além das coordenadas tridimensionais dos pontos da cena [1].

Dadas as imagens  $I = \{I_i | i = 1 \dots N_i\}$ , o BA geométrico [2] otimiza conjuntamente as poses das câmeras  $T = \{T_i | i = 1 \dots N_i\}$  e as coordenadas dos pontos tridimensionais da cena  $P = \{p_j | j = 1 \dots N_j\}$ , minimizando o erro de re-projeção:

$$X = \arg \min \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \|e_{i,j}^g(X)\| \quad (23)$$

onde a distância geométrica  $e_{i,j}^g(X) = \pi(T_i, p_j) - q_{i,j}$  mede a diferença entre um ponto projetado da cena e sua observação na imagem; a função  $\pi$  projeta os pontos da cena no espaço da imagem;  $q_{i,j} = [x_{i,j}, y_{i,j}, 1]$  representa a coordenada de pixel homogênea normalizada; e  $X = [T_1, T_2, \dots, T_{N_i}, p_1, p_2, \dots, p_{N_j}]$  contém todos os parâmetros das câmeras e dos pontos 3D.

A estratégia geral para minimizar a Equação 23 utiliza o método de Levenberg-Marquardt (LM) [49]. Em cada iteração, o método LM resolve uma atualização ótima  $\Delta X^*$ , minimizando a Equação 24.

$$\Delta X^* = \arg \min \|J(X)\Delta X + E(X)\| + \lambda \|D(X)\Delta X\| \quad (24)$$

onde  $E(X) = [e_{1,1}^g(X), e_{1,2}^g(X), \dots, e_{N_i, N_j}^g(X)]$ ,  $J(X)$  é a matriz Jacobiana de  $E(X)$  em relação a  $X$ , e  $D(X)$  é uma matriz diagonal não negativa, frequentemente definida como a raiz quadrada da diagonal do Hessiano aproximado  $J(X)^T J(X)$ . O parâmetro  $\lambda$  controla a regularização. A estrutura especial de  $J(X)^T J(X)$  motiva o uso do complemento de Schur [10].

O BA geométrico baseado no erro de re-projeção tem sido o padrão ouro para a estrutura a partir do movimento nas últimas duas décadas, mas apresenta duas limitações principais:

- Utiliza apenas informações de imagem associadas a características detectadas,

como cantos, *blobs* ou segmentos de linha.

- Exige correspondências de características entre imagens, um processo sujeito a valores discrepantes. Métodos de rejeição de *outliers*, como RANSAC, são necessários, mas não garantem um resultado completamente confiável.

Essas dificuldades motivaram o desenvolvimento de métodos diretos [52], que utilizam o *Bundle Adjustment* fotométrico para eliminar a correspondência de características e minimizar diretamente o erro fotométrico (diferença de intensidade de pixel) entre imagens alinhadas. O erro fotométrico é definido na Equação 25.

$$e_{i,j}^p(X) = I_i(\pi(T_i, d_j \cdot q_j)) - I_1(q_j) \quad (25)$$

onde  $d_j \in D = \{d_j | j = 1 \dots N_j\}$  representa a profundidade de um pixel  $q_j$  na imagem  $I_1$ , e  $d_j \cdot q_j$  transforma a coordenada de imagem do pixel  $q_j$  em sua posição tridimensional. Assim, o vetor de parâmetros a ser otimizado é  $X = [T_1, T_2, \dots, T_{N_i}, d_1, d_2, \dots, d_{N_j}]$ .

Os métodos diretos apresentam a vantagem de utilizar todos os *pixels* com gradiente de intensidade suficiente, oferecendo desempenho superior em cenas pouco texturizadas. No entanto, também possuem desvantagens:

- São sensíveis à inicialização [43, 50], pois o erro fotométrico introduz maior não convexidade na função de custo [35].
- São suscetíveis a variações de exposição e balanço de branco da câmera. Uma calibração fotométrica automática é necessária [35].
- São mais vulneráveis a *outliers*, como objetos em movimento dentro da cena.

## 2.5 Pontos de Interesse - *features*

Em *frames* de alta resolução, o rastreamento individual de cada *pixel* torna-se ineficiente, resultando em um custo computacional elevado [47]. Para mitigar esse problema, uma abordagem comum é a extração de pontos de interesse, que são regiões da imagem com características distintivas em relação às suas vizinhanças. Essas características podem incluir variações de textura, intensidade ou cor, tornando esses pontos facilmente identificáveis sob diferentes perspectivas e condições. Exemplos incluem cantos, pontos e *blobs* (regiões segmentadas da imagem). Uma propriedade essencial dos pontos de interesse é a repetibilidade, ou seja, a capacidade de serem detectados de maneira consistente sob diferentes condições de visualização [6].

Em 1988, Harris e Stephens [25] introduziram um método seminal conhecido como detector de cantos de Harris, que localiza pontos de interesse analisando gradientes de intensidade em diferentes direções e escalas [56]. De forma empírica, Schmid e Zisserman

[51] demonstraram que o erro de correspondência desse método geralmente é inferior a um *pixel*. A Figura 14 ilustra um exemplo do comportamento do detector de Harris.



Figura 14: *Feature detector* - fonte: [45].

A implementação computacional desse método foi descrita por [56]. Para um ponto no domínio da imagem, é definida uma matriz de estrutura da imagem  $M$ , que é positiva semi-definida e simétrica. Essa matriz é descrita na Equação 26.

$$M = \nabla I \nabla I^T = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (26)$$

onde  $\nabla I$  é o gradiente da imagem, contendo as derivadas parciais nas direções  $x$  ( $I_x$ ) e  $y$  ( $I_y$ ). A matriz  $M$  possui posto 1, com autovalores  $\lambda_1 = |\nabla I|^2$  e  $\lambda_2 = 0$ .

Para calcular a resposta de Harris, a matriz  $M$  é suavizada com um filtro gaussiano  $G_\sigma$ , resultando em  $\bar{M}$ , conforme a Equação 27:

$$\bar{M} = G_\sigma * M \quad (27)$$

O uso do filtro gaussiano reflete melhor as propriedades locais da imagem, tornando possível distinguir os autovalores da matriz  $M$  e da matriz resultante  $\bar{M}$ . Com isso, os autovalores indicam diferentes tipos de regiões na imagem:

- Ponto de interesse: ocorre quando  $\lambda_1$  e  $\lambda_2$  possuem valores elevados;
- Borda ou aresta: ocorre quando  $\lambda_1$  é alto e  $\lambda_2$  próximo de zero;
- Região uniforme: ocorre quando  $\lambda_1$  e  $\lambda_2$  são baixos.

A Equação 28 pode ser utilizada para identificar automaticamente pontos de interesse, sem a necessidade de calcular os autovalores explicitamente. Se o valor da resposta for superior a um limite  $th$ , o ponto é classificado como um canto.

$$R = \frac{\det(\bar{M})}{\text{Tr}(\bar{M})^2} \quad (28)$$

onde  $\det(\bar{M})$  é o determinante da matriz  $\bar{M}$  e  $\text{Tr}(\bar{M})$  é sua traça.

### 2.5.1 Descritores e casamento de pontos

Um dos princípios fundamentais na reconstrução 3D a partir de imagens 2D é a identificação de pontos correspondentes em diferentes imagens. Para atingir esse objetivo, diversos estudos foram conduzidos com o propósito de descrever os pontos de interesse [31, 37, 23], desenvolvendo descritores que operam em conjunto com os detectores de características. A interpretação dos pontos de interesse não é crítica; no entanto, seus descritores devem ser suficientemente robustos para localizar o mesmo ponto em imagens distintas.

Dado que as imagens analisadas no processo de reconstrução 3D podem sofrer transformações como rotação, translação e mudança de escala, a escolha de um detector de características do tipo *blobs* é preferível. Embora sejam menos precisos em termos de localização quando comparados a um detector de cantos, esses detectores são mais robustos a mudanças de formato e escala [19].

Em 1999, foi desenvolvida uma das metodologias mais abordadas para a obtenção de características e seus descritores. [37] introduziu um modelo invariante a iluminação, escala e rotação, denominado SIFT (*Scale Invariant Feature Transform*). Esse modelo utiliza uma técnica chamada DoG (*Difference of Gaussian*) para determinar as características da imagem.

O desfoque gaussiano, por não respeitar os limites naturais dos objetos, suaviza os detalhes e o ruído sem um tratamento específico, resultando na redução da precisão e distinção da localização [7]. Em contrapartida, o modelo denominado KAZE apresenta uma estratégia que proporciona melhor desempenho nessas circunstâncias. Esse modelo detecta e descreve recursos 2D em um espaço de escala não linear por meio de filtragem de difusão não linear, permitindo o desfoque ser realizado localmente, preservando os limites do objeto e reduzindo o ruído [3].

## 2.6 Structure from Motion (SFM)

*Structure From Motion* (SFM) é um processo de recuperação da estrutura 3D a partir de múltiplas imagens da mesma cena. Para cada imagem, é associada uma matriz de câmera, parametrizada por rotação, translação e distância focal. A correspondência entre as imagens é obtida utilizando seus pontos de interesse (*features*), juntamente com a aplicação da técnica *Random Sample Consensus* (RANSAC), responsável por filtrar os pontos consistentes com a matriz fundamental. As imagens são agrupadas em pares, e suas correspondências são usadas para criar a nuvem esparsa [30], é possível visualizar a técnica na Figura [15]. No entanto, difere fundamentalmente da fotogrametria convencional, na qual a geometria da cena, as posições da câmera e a orientação são resolvidas

automaticamente, sem a necessidade de especificar um conjunto predefinido de ‘controle de solo’ alvos/pontos, que possuem posições 3D conhecidas. Em vez disso, esses são resolvidos simultaneamente usando um método altamente redundante e iterativo de ajuste de pacote, baseado em um banco de dados de recursos extraídos automaticamente de um conjunto de sobreposições múltiplas de imagens [55].

Por outro lado, na fotogrametria tradicional, os parâmetros de orientação interior e exterior são definidos e determinados explicitamente, ou seja, as fotografias são coletadas em posições conhecidas com orientações e ângulos de incidência. A necessidade de um alto grau de sobreposição para cobrir toda a geometria do objeto ou cena de interesse dá origem ao nome: estrutura derivada de um sensor em movimento. A estrutura da cena refere-se tanto às posições e orientações das câmeras quanto à topografia. A abordagem é mais adequada para conjuntos de imagens com alto grau de sobreposição que capturam a estrutura 3D completa da cena visto de uma ampla variedade de posições ou, como o nome sugere, imagens derivadas de um sensor em movimento (*Motion*) [46].

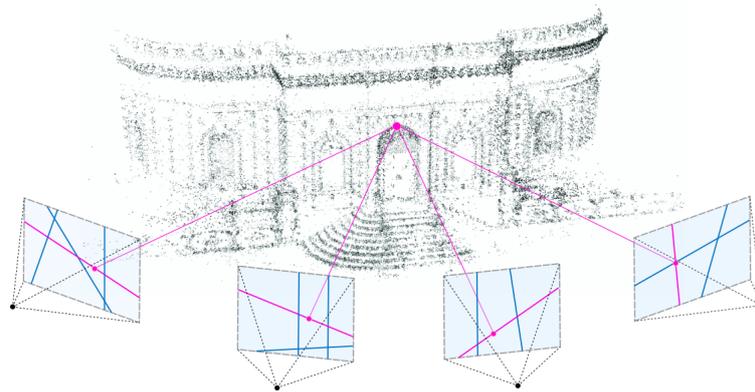


Figura 15: A sparse 3D point cloud produced by SfM, means initialization - fonte: [22]

## 2.7 Random Sample Consensus

O método *Random Sample Consensus* (RANSAC), proposto por Fischler e Bolles [18], é uma abordagem geral de estimação de parâmetros projetada para lidar com uma grande proporção de *outliers* nos dados de entrada. Ao contrário de muitas das técnicas comuns de estimativa robusta, como estimadores  $M$  e quadrados mínimos medianos, que foram adotadas pela comunidade de visão computacional a partir da literatura estatística, o RANSAC foi desenvolvido dentro da comunidade de visão computacional.

O RANSAC é uma técnica de re-amostragem que gera soluções candidatas usando o número mínimo de observações (pontos de dados) necessárias para estimar os parâmetros subjacentes do modelo. Conforme apontado por Fischler e Bolles [18], diferentemente de técnicas de amostragem que utilizam o máximo de dados possível para obter uma estimativa inicial da solução e, em seguida, procedem à remoção de *outliers*, o RANSAC

utiliza o menor conjunto possível e prossegue ampliando este conjunto com pontos de dados consistentes [18].

A implementação do método RANSAC pode ser organizado nas seguintes etapas:

1. Selecionar aleatoriamente um subconjunto do conjunto de dados;
2. Ajustar um modelo ao subconjunto selecionado;
3. Determinar o número de *outliers*;
4. Repetir as etapas de 1 a 3 por um número prescrito de iterações.

A equação de uma reta que melhor se ajustar a um conjunto de pontos pode ser estimada usando o RANSAC, tal processo é demonstrado na Figura 16.

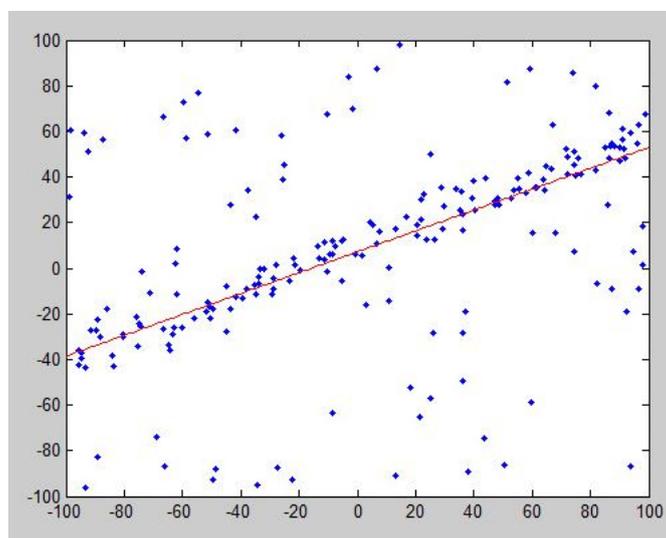


Figura 16: *The equation of a line that best fits a set of points can be estimated using RANSAC* - fonte: [40].

## 2.8 Segmentação e Rastreamento Dinâmico em Sequências de Imagens

De acordo com [54], uma representação tridimensional do espaço de cores HSV é um hexa-cone, com o eixo vertical central representando a intensidade. A matiz é definida como um ângulo no intervalo  $[0, 2\pi]$  em relação ao eixo vermelho, com vermelho no ângulo 0, verde em  $2\pi/3$ , azul em  $4\pi/3$  e vermelho novamente em  $2\pi$ . A saturação é a pureza da cor e é medida como uma distância radial a partir do eixo central, com valores entre 0 no centro e 1 na superfície externa. Qualquer cor no espaço HSV pode ser transformada em uma tonalidade de cinza reduzindo suficientemente a saturação. O valor da intensidade determina a tonalidade de cinza específica para a qual essa transformação converge. A saturação fornece uma ideia sobre a profundidade da cor e o olho humano é

menos sensível à sua variação em comparação com a variação na matiz ou intensidade. A saturação de um pixel é usada para determinar se a matiz ou a intensidade é mais relevante para a percepção visual humana da cor desse pixel, ignorando o valor real da saturação. Para baixa saturação, uma cor pode ser aproximada por um valor de cinza especificado pelo nível de intensidade, enquanto para alta saturação, a cor pode ser aproximada pela sua matiz. O limiar de saturação que determina essa transição é novamente dependente da intensidade. Para baixas intensidades, mesmo com alta saturação, uma cor está próxima do valor de cinza e vice-versa. Observa-se que para valores mais altos de intensidade, uma saturação de cerca de 0,2 diferencia entre a dominância da matiz e da intensidade. Assumindo que o valor máximo de intensidade é 255, a seguinte Equação 29 de limiar é usada para determinar se um pixel é representado pela matiz ou pela intensidade como sua característica dominante.

$$\text{th}_{\text{sat}}(V) = 1.0 - \frac{0.8V}{255} \quad (29)$$

Assim, cada pixel em uma imagem é tratado como um pixel de "cor verdadeira" (um pixel cuja saturação é maior que um determinado limiar  $\text{sat}(V)$  e cuja matiz é o componente dominante) ou como um pixel de "cor cinza" (um pixel cuja saturação é menor que  $\text{sat}(V)$  e cuja intensidade é o componente dominante). Este método de separação de *pixels* de cor verdadeira de *pixels* de cor cinza usando a saturação é um conceito inovador e alcança a segmentação de imagem que é útil para rastreamento de objetos.

A principal desvantagem da maioria das técnicas de rastreamento de objetos no domínio do pixel é a sensibilidade à variação de intensidade, que é bastante reduzida. Além disso, quadros de vídeo temporalmente próximos têm alta similaridade ao nível do objeto, exceto quando há uma mudança abrupta de cena. Uma representação ao nível do objeto dos quadros de vídeo oferece um método mais robusto para comparação de objetos para rastreamento [62]. Portanto, este método é semelhante à forma como os humanos percebem a presença de objetos e as mudanças de cena em vídeos. Os olhos humanos percebem uma mudança no movimento dos objetos apenas quando os objetos presentes em um quadro diferem consideravelmente do quadro anterior.

Na Figura 17, é demonstrado um especto convencional do método HSV.

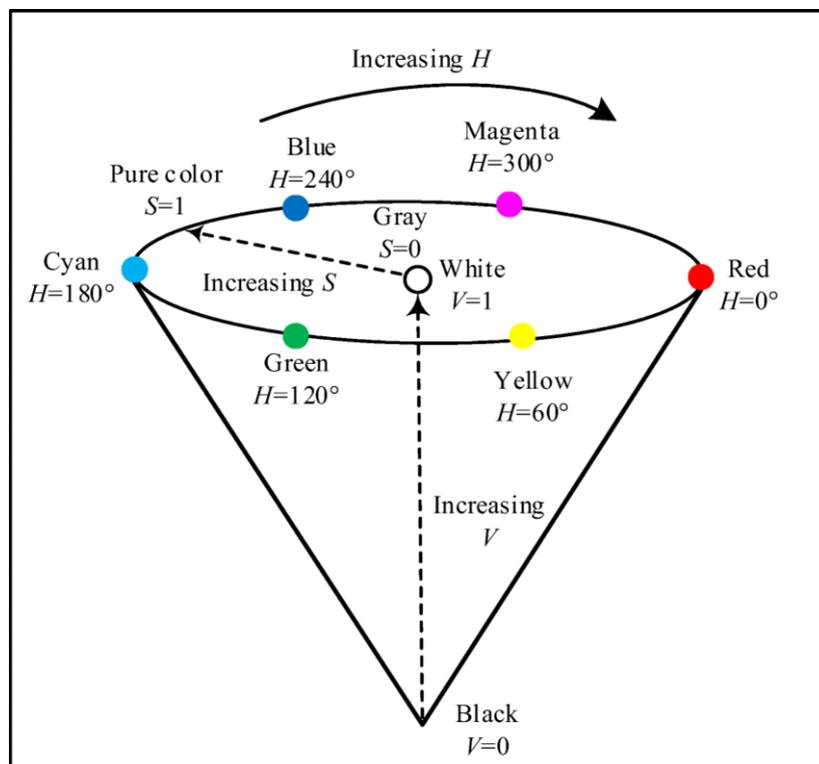


Figura 17: *HSV color space* - Fonte: [28]

### 3 ARQUITETURA E METODOLOGIA

Neste Capítulo é apresentada a metodologia utilizada para o presente estudo, explicando as etapas desenvolvidas de uma forma mais aplicada, assim como os algoritmos de apoio. O estudo é composto por quatro etapas: ambiente simulado, comunicação, desenvolvimento computacional (implementação) e a predição da estimativa da pose do alvo (*output*), conforme demonstra a Figura 18.

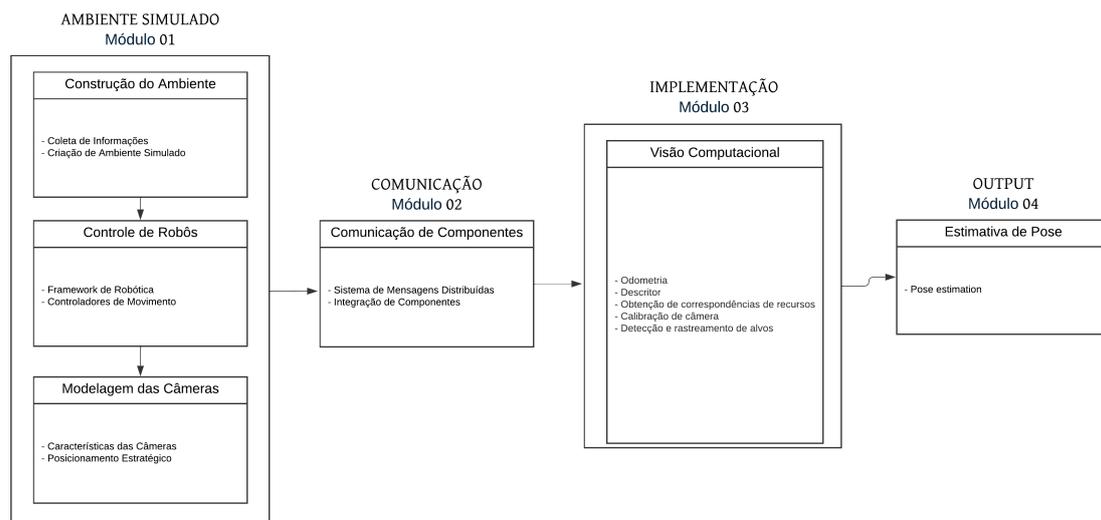


Figura 18: Metodologia proposta.

#### 3.1 Ambiente Simulado

Para projetar um gêmeo digital que se assemelhe a um ambiente operacional *offshore* real Módulo 01, foram pesquisadas informações públicas sobre áreas de carga e descarga em plataformas de petróleo. Usando modelos reais, um ambiente simulado foi criado dentro de um cenário do Gazebo. Abaixo é demonstrado uma descrição detalhada das técnicas usadas e sua implementação com ROS2<sup>1</sup>.

<sup>1</sup><https://www.ros.org>

- Simulação do Ambiente: O ambiente foi simulado usando o Gazebo<sup>2</sup>, um simulador de física 3D com bibliotecas de software de código aberto para aplicações de alto desempenho. O Gazebo permite a criação de um ambiente virtual com objetos, robôs e sensores. Modelos 3D dos robôs e do ambiente foram criados para replicar as condições do mundo real;
- Controle de Robôs: O controle do robô foi realizado usando o Sistema Operacional de Robôs (ROS2), uma estrutura de robótica de código aberto com bibliotecas de software e ferramentas para construir aplicações robóticas, incluindo *drivers*. Controladores de movimento foram implementados para permitir que o robô se mova e interaja com o ambiente.

Quanto às características de visão e dimensões físicas das câmeras usadas no ambiente simulado, a escolha das posições foi feita em conjunto com os colaboradores que atuam diretamente na plataforma offshore, levando em consideração a área de maior movimentação e relevância operacional. As câmeras foram selecionadas devido à sua capacidade de cobrir as áreas críticas a serem monitoradas. Ambas as câmeras possuem um campo de visão de 1,183 radianos (67,8 graus), oferecendo um ângulo de visão amplo que permite o monitoramento eficiente das operações.

A resolução de imagem foi configurada para 1080p (FHD), garantindo uma visualização clara e detalhada das atividades. As câmeras foram estrategicamente posicionadas no ambiente simulado do Gazebo para capturar as operações com o máximo de cobertura. Na Figura 19, a câmera I foi instalada no topo de uma estrutura usada como suporte para o lança treliçada (Figura 3) do guindaste, permitindo uma visão abrangente da movimentação de cargas. Já na Figura 20, a câmera II foi montada ao lado da área operacional, próxima ao pedestal do guindaste (Figura 3), para garantir uma visão detalhada das operações de içamento.

Essa abordagem colaborativa, que envolveu tanto engenheiros quanto operadores da plataforma, permitiu a escolha de câmeras compactas, adequadas para integração em sistemas robóticos e ambientes virtuais. Isso assegurou a captura de imagens de alta qualidade, essenciais para o estudo.

---

<sup>2</sup><https://gazebo.org>

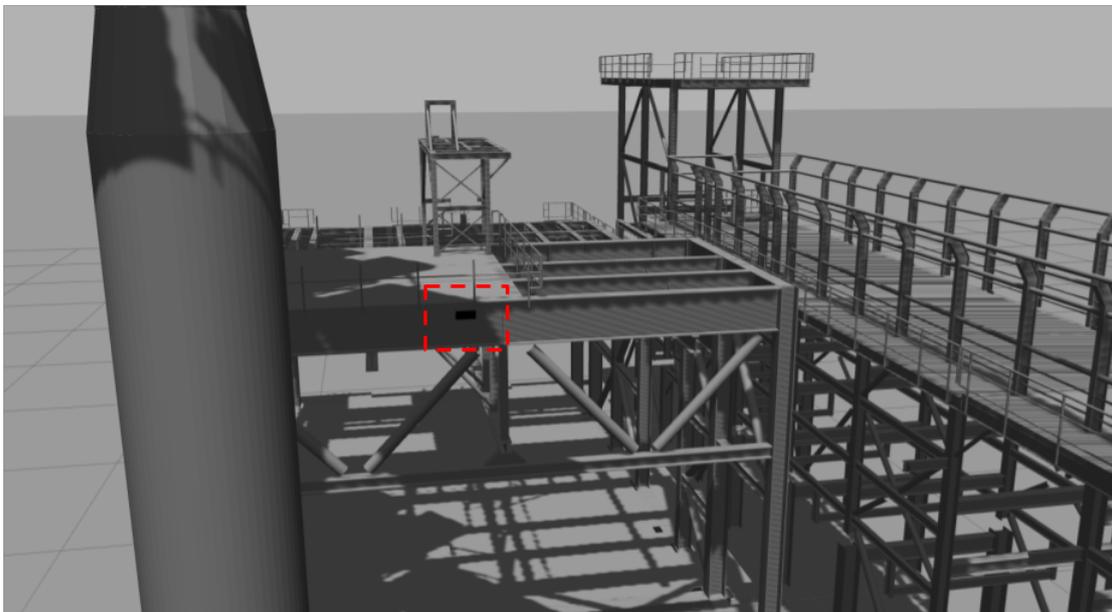


Figura 19: Posição da câmera I no ambiente.

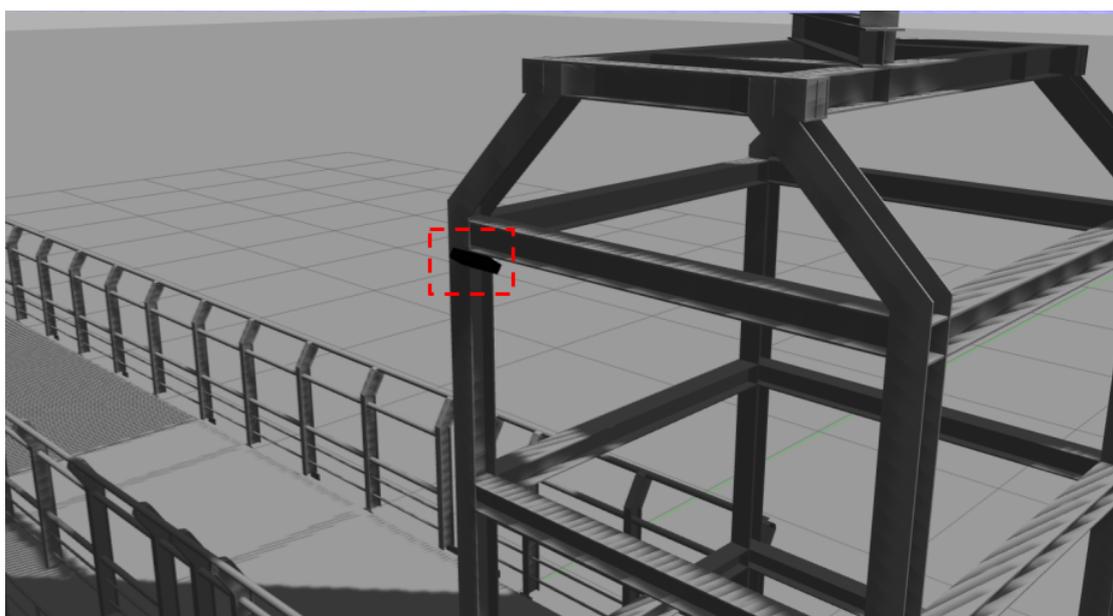


Figura 20: Posição da câmera II no ambiente.

## 3.2 Comunicação

No Módulo 02, o *ROS2* fornece uma infraestrutura de comunicação distribuída para facilitar a troca de mensagens entre os diferentes componentes do sistema, como o controlador do robô, os algoritmos de processamento de dados e outros módulos essenciais. O *ROS2* utiliza três principais mecanismos de comunicação: *tópicos*, *serviços* e *ações*, cada um desempenhando um papel fundamental na troca de informações e no controle do sistema.

- Tópicos são utilizados para a comunicação assíncrona e contínua de dados, ideal para fluxos de informações como leituras de sensores e comandos de controle. Nesse contexto, nós publicam e assinam tópicos para enviar e receber mensagens de maneira eficiente.
- Serviços oferecem uma comunicação síncrona, onde um nó pode solicitar uma ação específica de outro e aguardar uma resposta, útil para operações pontuais que exigem retorno imediato de informações ou execução.
- Ações são projetadas para tarefas que levam mais tempo, permitindo que um nó envie uma solicitação e receba *feedback* contínuo durante a execução, essencial para processos que demandam monitoramento em tempo real, como movimentos complexos de robôs.

A implementação desta etapa envolveu a criação de pacotes *ROS2* para organizar o código-fonte e facilitar a modularização, permitindo maior escalabilidade e manutenção. Foram desenvolvidos nós *ROS2* responsáveis por executar tarefas específicas, como a leitura de dados de sensores, controle dos atuadores e processamento de algoritmos. Além disso, a configuração dos parâmetros e arquivos de lançamento (*launch files*) no *ROS2* garantiu a integração adequada entre os componentes, facilitando o gerenciamento do sistema como um todo [15].

Testes e experimentos extensivos foram realizados no ambiente simulado para validar o desempenho e a funcionalidade do sistema. Sensores virtuais foram configurados para simular cenários do mundo real, e métricas-chave, como latência de comunicação, precisão dos sensores e eficiência dos algoritmos de controle, foram coletadas e analisadas.

## 3.3 Desenvolvimento Computacional

Nesta seção, serão abordados, em subseções, os processos e métodos utilizados no desenvolvimento computacional.

### 3.3.1 Calibração das Câmeras

No Módulo 03 é envolvido a estimativa de parâmetros intrínsecos e extrínsecos através da calibração das câmeras. Visto que todo o processo de captura é realizado com duas câmeras de mesma configuração, seus parâmetros intrínsecos terão o mesmo valor e podem ser estimados *a priori*. Para a calibração das câmeras e obter os parâmetros extrínsecos, é utilizado um tabuleiro de xadrez dentro do ambiente simulado no ROS/Gazebo, com quadrados e arestas de 20 milímetros, como apresentado na Figura 21. Para obter a calibração de uma câmera deve-se realizar a captura do tabuleiro de diferentes ângulos e seguir o processo descrito na 2.4.3.

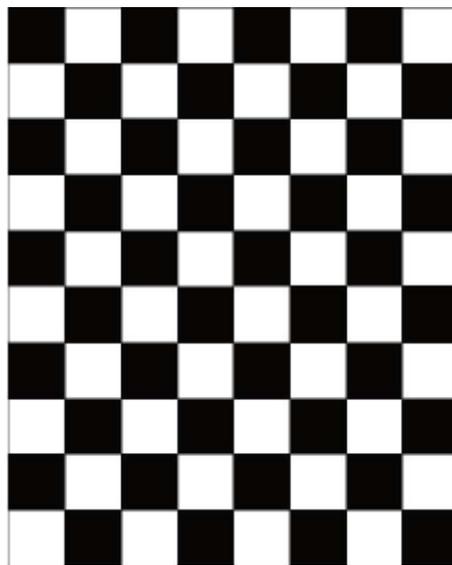


Figura 21: Tabuleiro utilizado para calibração.

### 3.3.2 Odometria

O desenvolvimento computacional utilizando odometria foi implementado utilizando o ambiente de desenvolvimento *Collaboratory* do *Google Research*. Esta plataforma aproveita recursos de *hardware* online, como *GPUs*, para processamento eficiente de algoritmos.

Em primeiro momento, deve-se obter a correspondência de pontos-chave (*matches*) entre as capturas de imagens das duas câmeras. Afim de estabelecer relações entre as imagens capturadas pelas câmeras os pontos-chave são pontos de interesse distintos que podem ser facilmente identificados em várias imagens. Modelos como *SIFT* [64], *SURF* [29] ou *ORB* [38] foram explorados, mas o *SIFT* foi escolhido para extrair e descrever os pontos-chave nas imagens da câmera.

Para cada par de imagens, faz-se uma varredura por pontos correspondentes entre elas. Essa varredura consiste em encontrar pontos nas duas imagens que possuam seus descritores semelhantes. De acordo com essas correspondências, é possível calcular a matriz

fundamental capaz de mapear os pontos de uma imagem na outra, conforme exemplificado na Seção 2.4.2. Em conjunto, o modelo *Random Sample Consensus* (RANSAC) foi usado para encontrar correspondências robustas entre os pontos-chave das câmeras. Este método iterativo estimou a matriz de rotação e o vetor de translação que melhor transformaram os pontos correspondentes de uma câmera para outra. O RANSAC lida com *outliers*, aumentando a robustez das correspondências encontradas.

A Figura 22 demonstra os *matches* obtidos em um conjunto de 1 par de imagens de cada câmera modelada no ambiente.

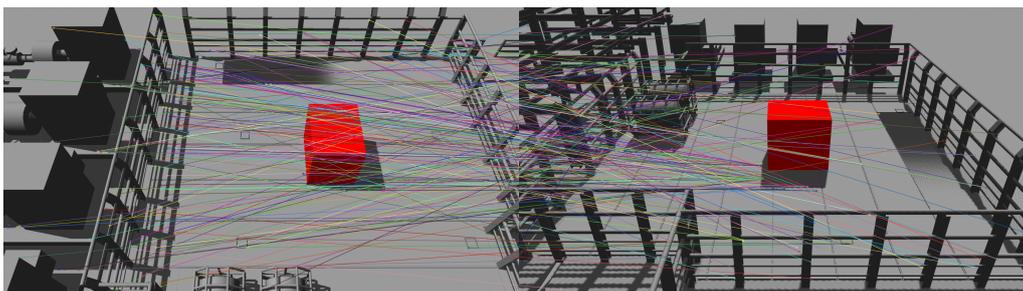


Figura 22: Pontos de correspondência entre um par de imagens.

A Figura 23 representa um fluxograma adotado para a estimação da odometria da câmera.

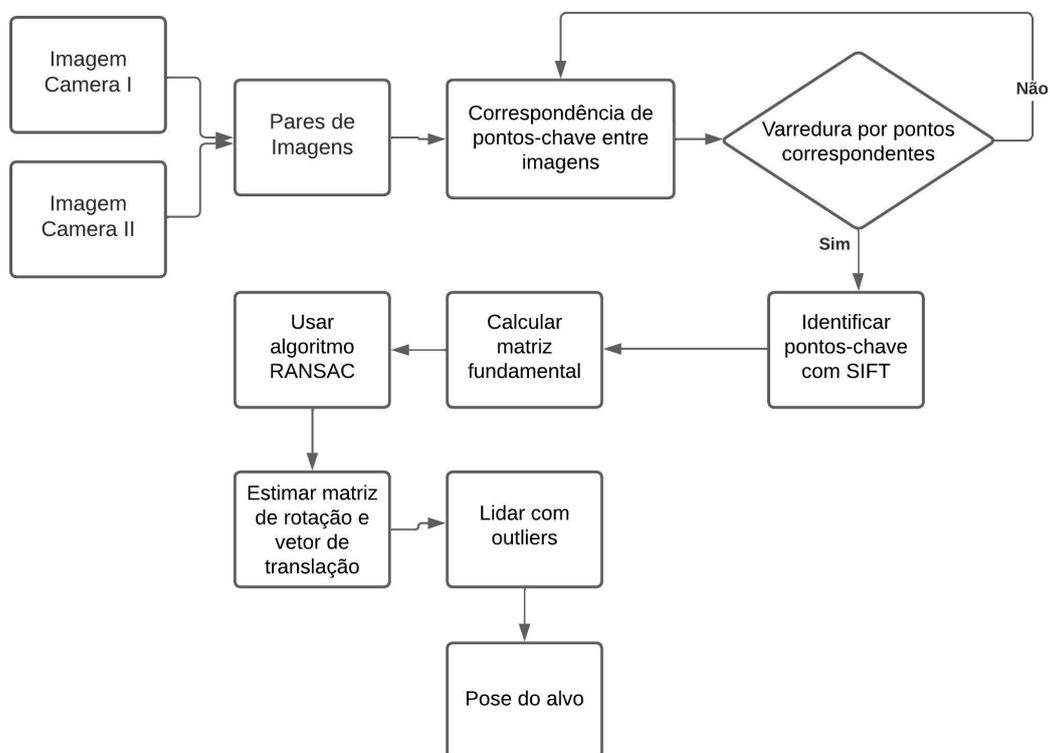


Figura 23: Fluxograma - Odometria da câmeras.

### 3.4 Output

Nesta última etapa no Módulo 04, o sistema tem como *Output* a estimativa da pose de um objeto encontrado na zona de carga e descarga onde o guindaste realiza as suas operações (*laydown area*) através das visadas de ambas câmeras.

Para melhorar a estimativa da pose do objeto, a caixa delimitadora também foi considerada em ambas as imagens das câmeras. A caixa delimitadora segmenta a região na imagem onde o objeto está presente, fornecendo informações adicionais para determinar sua posição e orientação.

De acordo com o estudo abordado na Seção 2.8, o modelo desenvolvido com base no HSV foi empregado para identificar a região colorida com objetivo de detectar os contornos da área do objeto. A extração das coordenadas da caixa delimitadora permitiu o rastreamento do objeto nas imagens da câmera. Essa abordagem integrada facilitou o desenvolvimento de um sistema robusto e eficiente de detecção e rastreamento de objetos.

Ao combinar as coordenadas da caixa delimitadora com as correspondências robustas encontradas e os parâmetros estimados, uma estimativa mais precisa da pose do objeto em relação às câmeras foi obtida.

A Figura 24 representa a etapa de rastreamento (*tracking*) por coloração do alvo mencionada nesta seção, adicionada ao fluxograma mostrado na Figura 23.

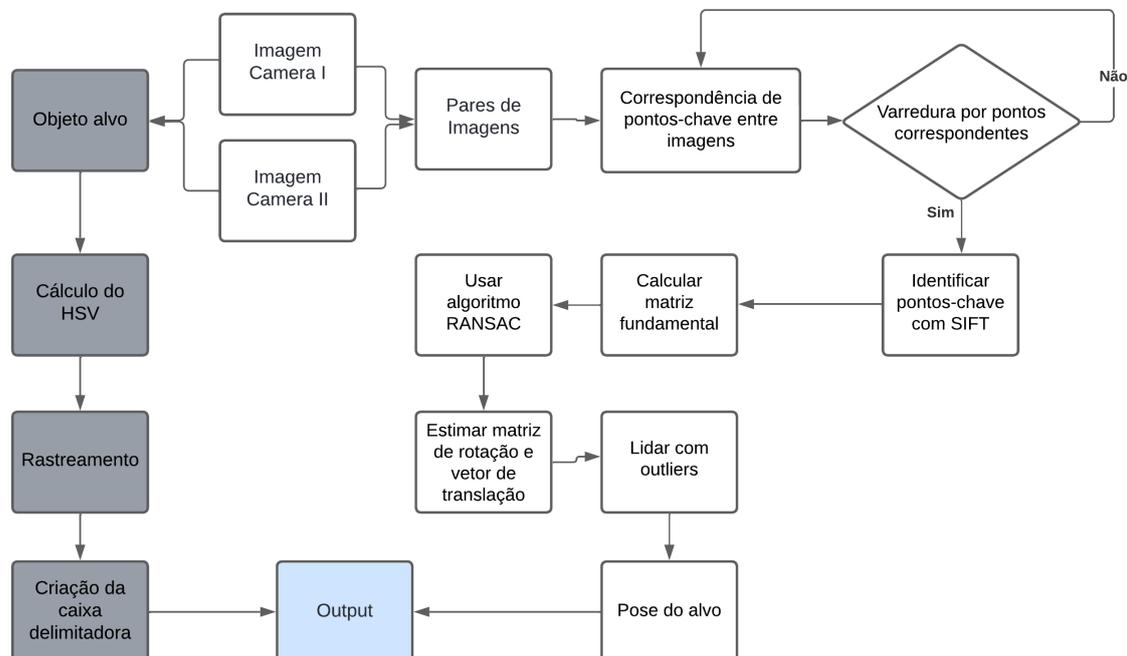


Figura 24: Fluxograma - Etapa de rastreamento por cor.

## 4 RESULTADOS

Nesse Capítulo serão apresentados os resultados obtidos durante o desenvolvimento dos processos que compõem o trabalho, demonstrando as evoluções de cada etapa. Para melhor entendimento do estudo e melhor apresentação esta seção foi dividida em quatro partes. A Seção 4.1 detalha a calibração das câmeras no ambiente simulado. Na Seção 4.2, foi modelado um cenário 3D de uma área de armazenamento **offshore** a partir de vídeos do YouTube, utilizando câmeras simuladas para capturar imagens e calcular a matriz de Pose, aplicando o método SIFT para extrair pontos-chave e triangulação. Na Seção 4.3, a conversão para o espaço de cor HSV foi utilizada para segmentar a carga alvo, seguida de operações morfológicas e aplicação de uma caixa delimitadora para destacar o alvo nas imagens. Por fim, na Seção 4.4, as técnicas anteriores são combinadas para estimar a pose da carga em relação ao campo de visão das câmeras, visualizando a pose do alvo como uma esfera nas imagens capturadas.

### 4.1 Calibração das Câmeras

O primeiro resultado parcial envolve a calibração das câmeras RGB seguindo os procedimentos abordados na Seção 3.3.1. Com esse propósito, foi realizado a captura de diferentes ângulos e vistas de um tabuleiro de xadrez simulado no ambiente ROS/Gazebo, semelhante a Figura 21 da Seção 3.1.

Os parâmetros obtidos durante a calibração podem ser observados na Tabela 1. Onde  $f_x$  e  $f_y$  podem ser calculados a partir do campo de visão horizontal e vertical respectivamente e  $c_x$  e  $c_y$  são as coordenadas do centro da imagem, que podem ser calculadas simplesmente como a metade da resolução da imagem.

$f_x$	$f_y$	$c_x$	$c_y$
1423.27	1423.27	960	540

Tabela 1: Parâmetros intrínsecos calculados das câmeras utilizadas

Como todas as capturas de imagens são obtidas por duas câmeras idênticas, os

parâmetros serão os mesmos para todas as imagens obtidas na operação.

## 4.2 Cálculo da Odometria

O cenário 3D, que simula a área de armazenamento de uma plataforma *offshore*, foi construído com base em vários vídeos disponíveis no YouTube. Usando diferentes ângulos, foi possível modelar este cenário para torná-lo o mais fiel possível ao ambiente operacional real. Na Figura 25, são visualizados os quadros capturados de cada uma das duas câmeras.

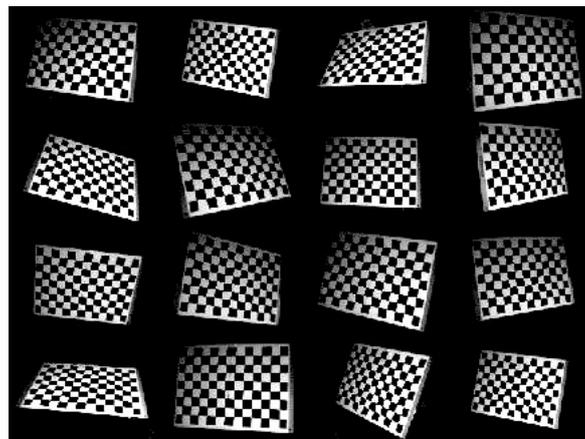


Figura 25: Imagens obtidas pelas câmeras no ambiente simulado.

Com os parâmetros intrínsecos das câmeras, foi realizado uma série de capturas de imagens da zona de carga e descarga com uma carga alvo ocupando diferentes pátios dentro das limitações da *laydown*.

Utilizando as técnicas abordadas na Seção 3.3, foi realizado o cálculo da matriz de Pose para cada imagem obtida. Assim, gerando os parâmetros extrínsecos.

Com os parâmetros obtidos das câmeras simuladas, o método computacional SIFT é aplicado para extrair pontos-chave e seus descritores entre os quadros capturados, demonstrado na Figura 22. Após realizar a correspondência de pontos-chave entre as duas imagens usando o BFMatcher, que é um modelo de correspondência baseado em força bruta, as correspondências são filtradas usando um critério de distância para reter apenas as correspondências satisfatórias. Este processo permite a extração da matriz fundamental e da matriz essencial, derivadas da matriz fundamental e das matrizes intrínsecas das câmeras.

Com o resultado obtido da aplicação do método SIFT e das matrizes derivadas, torna-se possível transformar pontos 3D no sistema de coordenadas da câmera. Esta transformação é crucial para projetar com precisão os pontos no plano da imagem da câmera e sobrepor os eixos da matriz de rotação e vetor de translação na imagem. Os eixos são projetados com base nas correspondências geradas a partir da Câmera I (Figura

[19] em relação à Câmera II (Figura [20]). Na Figura [26], apresenta os vetores de posição da Câmera I em relação a Câmera II, esse processo tem como objetivo obter uma triangulação das câmeras afim de obter uma melhor exatidão na estimação da pose do alvo, essa etapa é demonstrada na Seção [2.6].

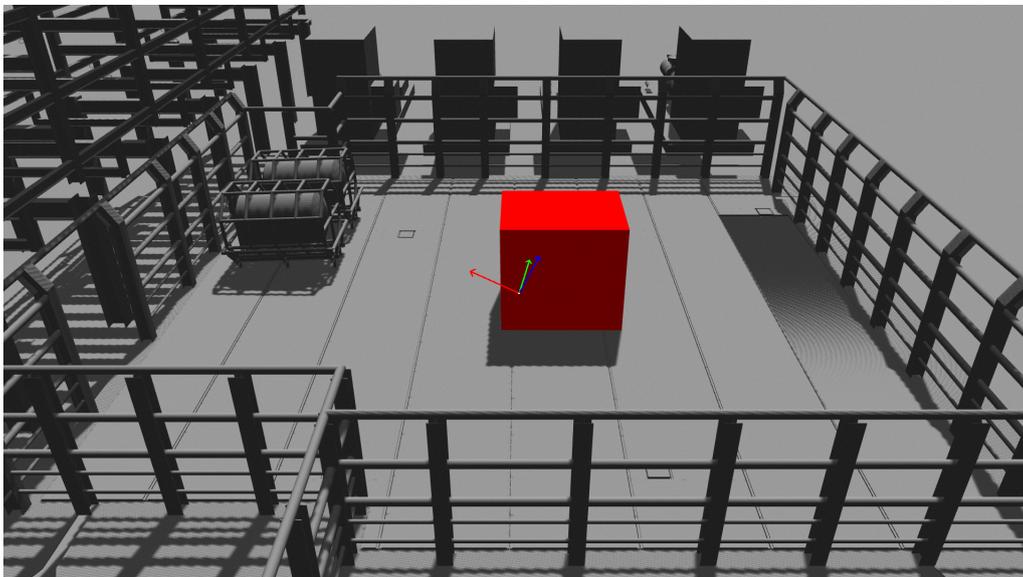


Figura 26: Vetores de posição da Câmera I em relação a Câmera II.

### 4.3 Rastreamento e Detecção da Carga Alvo

De acordo com a técnica descrita na Seção [2.8], foi inicialmente realizado em cada captura de imagem uma conversão do espaço de cor RGB para o espaço de cor HSV. A conversão para o espaço HSV é essencial porque permite segmentar a carga-alvo de forma robusta, isolando a matiz do vermelho, independentemente de variações de iluminação e sombras no ambiente. Além disso, o uso da saturação auxilia na remoção de ruídos causados por reflexos ou interferências de tons neutros na cena. Assim adquirindo as variáveis do espectro HSV da carga alvo alocada dentro da zona de operação. Em seguida foram definidos limiares específicos para a matiz e saturação que correspondem à cor do objeto alvo. Esses limiares foram determinados empiricamente através de uma série de testes preliminares.

Aplicamos esses limiares as capturas de imagens para segmentar o objeto alvo. *Pixels* cujas características de matiz e saturação estavam dentro dos limites definidos foram considerados parte do objeto alvo, enquanto os demais *pixels* foram descartados.

Para melhorar a precisão da segmentação, foram aplicadas operações morfológicas, como erosão e dilatação. Essas operações ajudaram a remover ruídos e pequenos artefatos, resultando em uma segmentação mais limpa e definida.

Por fim, com as coordenadas do alvo específico, foi realizando uma detecção onde para

melhor visualização foi aplicado uma caixa delimitadora (*Bounding Box*) ao redor da carga alvo.

As Figuras 27 e 28, demonstram a aplicação completa dessa etapa.

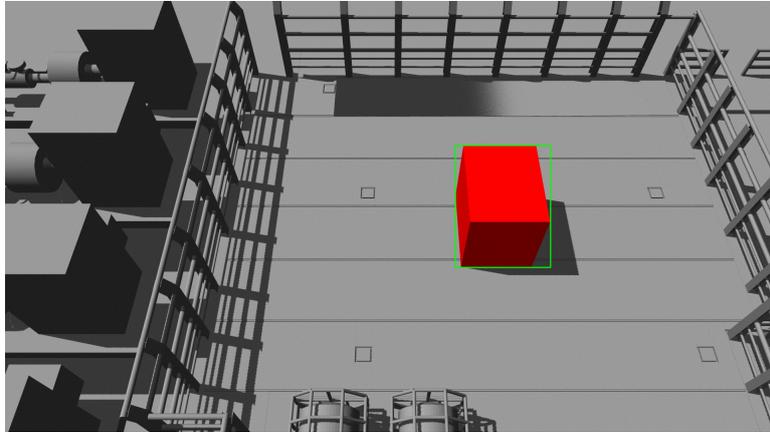


Figura 27: Câmera I - Rastreamento da carga alvo.

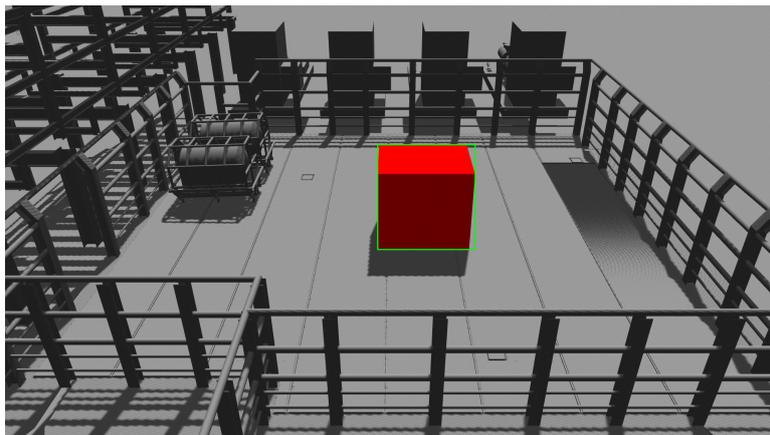


Figura 28: Câmera II - Rastreamento da carga alvo.

## 4.4 Estimação da Pose

As câmeras utilizadas estão integradas diretamente ao ambiente de simulação no *ROS2* com o Gazebo, onde elas são configuradas para capturar imagens de forma sincronizada e programada. Essas câmeras virtuais do Gazebo são responsáveis por simular a captura de imagens 2D a partir dos dois pontos de vista no ambiente simulado.

As imagens capturadas são enviadas para um local na nuvem, onde são armazenadas e processadas usando uma abordagem de cloud offloading. Nessa arquitetura, o processamento intensivo, como a estimativa de pose, é realizado na nuvem, aproveitando a escalabilidade e o poder de processamento remoto. Essa estratégia, conhecida como "edge-cloud architecture", otimiza o uso de recursos ao delegar o processamento para a nuvem, enquanto a captura de dados ocorre localmente no ambiente simulado.

O processamento de conversão das imagens ocorre na nuvem, onde os algoritmos responsáveis garantem a análise das imagens e a sua integração. Esse fluxo contínuo, desde a captura até a conversão e uso das imagens no ambiente de simulação, assegura que os dados visuais estejam sempre atualizados e disponíveis para monitoramento e análise das operações simuladas no Gazebo.

Na Figura 29, é possível visualizar um diagrama geral dos processos realizados no experimento. Os resultados parciais dos experimentos realizados foram publicados na conferência *Latin American Robotics Symposium (LARS)* [24].

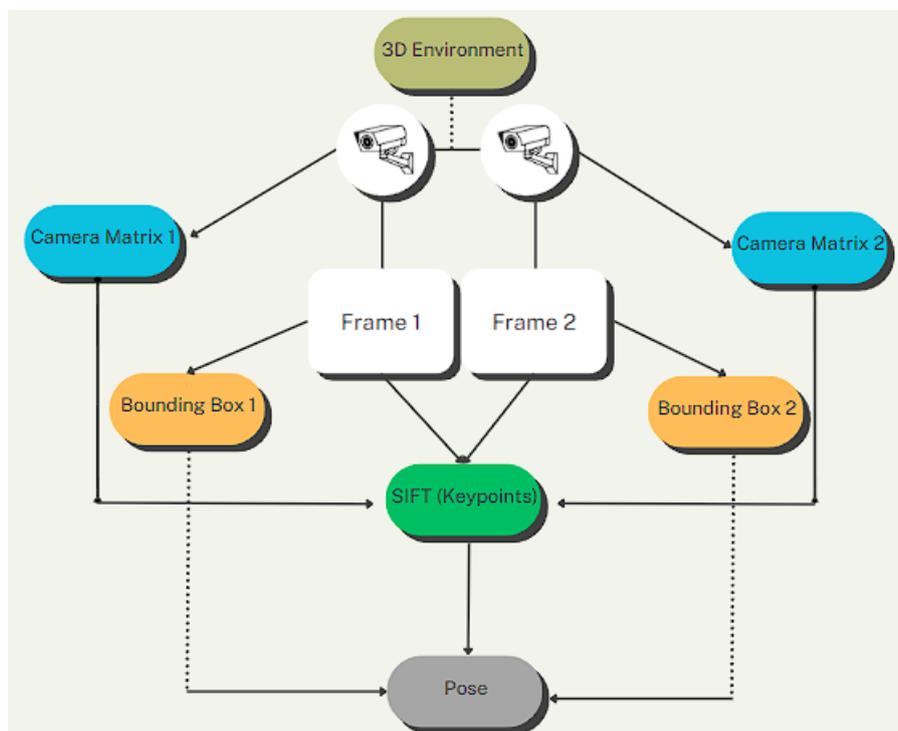


Figura 29: A arquitetura dos experimentos conduzidos.

Combinando as Seções 4.1, 4.2 e 4.3 é finalmente possível estimar a pose da carga em relação ao campo de visão de cada câmera. A Figura 30 é gerada no quadro de

captura da câmera II, onde demonstrada a pose da carga alvo em forma de uma esfera, a pose visualizada da câmera II é representado com uma esfera verde e a câmera II é representada com uma esfera azul.

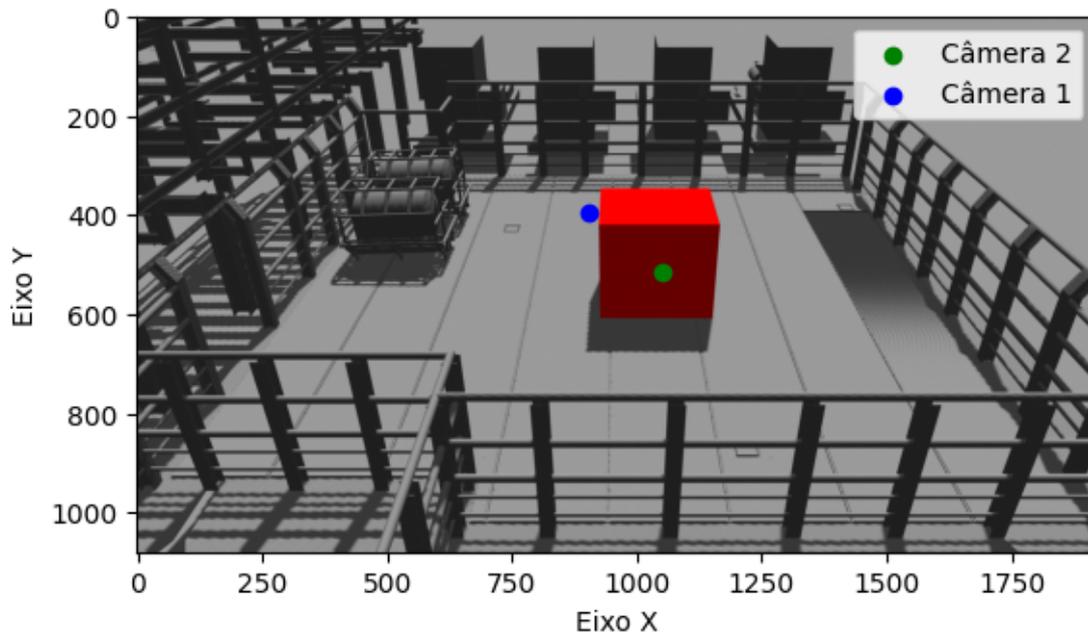


Figura 30: Pose do objeto alvo em relação às duas câmeras.

## 5 CONCLUSÃO

Nesta dissertação, abordamos o desafio de estimar a pose de uma *target load* em um ambiente simulado, utilizando imagens capturadas por câmeras estrategicamente posicionadas em um cenário 3D elaborado no ambiente *ROS2/Gazebo*. O ambiente foi construído com base em um contexto operacional real *offshore*, com o objetivo de representar de forma fiel parte do processo de transporte de cargas manuseadas por guindastes dentro de zonas delimitadas de carga e descarga, enfrentando desafios relacionados à segurança operacional e à proteção dos trabalhadores.

A metodologia desenvolvida combinou técnicas de visão computacional, incluindo rastreamento de alvos baseado em cores, correspondência de pontos-chave *SIFT* e o cálculo das matrizes *fundamental* e *essential* para a estimativa da pose em 3D. Os resultados obtidos demonstram o estudo e a eficácia dos métodos para a localização e o rastreamento dos objetos de interesse. A sobreposição de eixos da matriz de rotação e vetores de translação nas imagens das câmeras contribuiu para o aumento da precisão na estimativa da pose.

A relevância deste trabalho está não apenas no avanço científico relacionado à pesquisa em visão computacional e estimativa de pose, mas também em suas aplicações práticas no ambiente *offshore*. A metodologia proposta apresenta grande potencial para melhorar o monitoramento de cargas, a segurança operacional e a eficiência dos processos logísticos em operações *offshore*, especialmente na *Laydown Area*.

Entretanto, reconhecemos que a validação em cenários reais é indispensável para comprovar a aplicabilidade prática da abordagem. Assim, recomenda-se que futuros trabalhos envolvam a implementação desses algoritmos em tempo real e a realização de experimentos em campo, abrangendo diferentes condições ambientais.

Dessa forma, a metodologia proposta contribui para o avanço das operações *offshore*, estabelecendo uma ponte entre o ambiente simulado e a aplicação prática. O impacto potencial desta abordagem está diretamente relacionado ao aumento da segurança e à otimização dos processos logísticos. Espera-se que este trabalho possa servir como base para futuros estudos que explorem a integração entre tecnologia de visão computacional e operações *offshore*, promovendo avanços significativos no setor.

## REFERÊNCIAS

- [1] Agarwal, S., Snavely, N., Seitz, S. M., and Szeliski, R. (2010). Bundle adjustment in the large. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11*, pages 29–42. Springer.
- [2] Albl, C. and Pajdla, T. (2014). Global camera parameterization for bundle adjustment. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 3, pages 555–561.
- [3] Alcantarilla, P. F., Bartoli, A., and Davison, A. J. (2012). Kaze features. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 214–227. Springer.
- [4] American Petroleum Institute (2020). Energy and the 2020 election. Accessed: 2024-08-12.
- [5] Ancione, G., Paltrinieri, N., and Milazzo, M. F. (2020). Integrating real-time monitoring data in risk assessment for crane related offshore operations. *Journal of Marine Science and Engineering*, 8(7):532.
- [6] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359.
- [7] Bedruz, R. A. R., Fernando, A., Bandala, A., Sybingco, E., and Dadios, E. (2018). Vehicle classification using akaze and feature matching approach and artificial neural network. In *TENCON 2018-2018 IEEE Region 10 Conference*, pages 1824–1827. IEEE.
- [8] Bevilacqua, A., Gherardi, A., and Carozza, L. (2008). Automatic perspective camera calibration based on an incomplete set of chessboard markers. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 126–133. IEEE.

- [9] Breumier, S., Trudeau-Lalonde, F., Lafrance, T., Robert, E., Villani, A., Kermouche, G., and Lévesque, M. (2021). Controlled single and repeated impact testing for material plastic behaviour characterisation under high strain rates. *Strain*, 57(6):e12399.
- [10] Brown, M. and Lowe, D. G. (2005). Unsupervised 3d object recognition and reconstruction in unordered datasets. In *Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05)*, pages 56–63. IEEE.
- [11] Chotrov, D., Uzunova, Z., Yordanov, Y., and Maleshkov, S. (2018). Mixed-reality spatial configuration with a zed mini stereoscopic camera. In *Conf. Techno. Edu. Smart World*, volume 11.
- [12] Corke, P. and Robotics, V. (2017). control: Fundamental algorithms in matlab. *Springer Tracts in Advanced Robotics*, (Springer Berlin Heidelberg, 2011).
- [13] Dawson-Howe, K. M. and Vernon, D. (1994). Simple pinhole camera calibration. *International Journal of Imaging Systems and Technology*, 5(1):1–6.
- [14] EBPR (2020). Partilha de produção do pré-sal deve demandar 28 fpsos em dez anos, calcula ppsa. Acesso em: 8 nov. 2020.
- [15] Erős, E., Dahl, M., Bengtsson, K., Hanna, A., and Falkman, P. (2019). A ros2 based communication architecture for control in collaborative and intelligent automation systems. *Procedia Manufacturing*, 38:349–357.
- [16] Fairchild, C. and Harman, T. L. (2016). *ROS robotics by example*. Packt Publishing Ltd.
- [17] Feng, J. (2022). Reliability and safety in offshore engineering. In *Encyclopedia of Ocean Engineering*, pages 1446–1451. Springer.
- [18] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- [19] Förstner, W. and Wrobel, B. P. (2016). *Photogrammetric computer vision*. Springer.
- [20] Fotland, G., Haskins, C., and Rølvåg, T. (2020). Trade study to select best alternative for cable and pulley simulation for cranes on offshore vessels. *Systems Engineering*, 23(2):177–188.
- [21] Gee, T., Delmas, P., Stones-Havas, N., Sinclair, C., Van Der Mark, W., Li, W., Friedrich, H., and Gimel'farb, G. (2015). Tsai camera calibration enhanced. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 435–438. IEEE.

- [22] Geppert, M., Larsson, V., Speciale, P., Schönberger, J. L., and Pollefeys, M. (2020). Privacy preserving structure-from-motion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 333–350. Springer.
- [23] Grycuk, R., Scherer, R., and Gabryel, M. (2015). New image descriptor from edge detector and blob extractor. *Journal of Applied Mathematics and Computational Mechanics*, 14(4):31–39.
- [24] Guthes, R. T., Santos, M. M. D., Lemos, R., Azambuja, G., Santos, J. D., Borges, E. N., Drews Junior, P. L. J., and Botelho, S. S. C. (2023). Geometric mapping and pose estimation techniques for improving safety and efficiency in offshore crane operations. In *2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)*, pages 484–489.
- [25] Harris, C., Stephens, M., et al. (1988). A combined corner and edge detector. In *Alvey vision conference*, pages 147–152.
- [26] Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.
- [27] Hartley, R. I. and Sturm, P. (1997). Triangulation. *Computer vision and image understanding*, 68(2):146–157.
- [28] Hidayatullah, P. and Zuhdi, M. (2015). Color-texture based object tracking using hsv color space and local binary pattern. *International Journal on Electrical Engineering and Informatics*, 7(2):161.
- [29] Huijuan, Z. and Qiong, H. (2011). Fast image matching based-on improved surf algorithm. In *2011 International conference on electronics, communications and control (ICECC)*, pages 1460–1463. IEEE.
- [30] Iglhaut, J., Cabo, C., Puliti, S., Piermattei, L., O’Connor, J., and Rosette, J. (2019). Structure from motion photogrammetry in forestry: A review. *Current Forestry Reports*, 5:155–168.
- [31] Jiang, T. and Merickel, M. (1989). Identification and boundary extraction of blobs in complex imagery. *Computerized Medical Imaging and Graphics*, 13(5):369–382.
- [32] Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, pages 2149–2154. Ieee.

- [33] Li, G., Liu, Y., Dong, L., Cai, X., and Zhou, D. (2007). An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features. In *2007 IEEE/RSJ international conference on intelligent robots and systems*, pages 3854–3859. IEEE.
- [34] Lin, Y.-T. and Finlayson, G. D. (2023). A rehabilitation of pixel-based spectral reconstruction from rgb images. *Sensors*, 23(8):4155.
- [35] Liu, P., Geppert, M., Heng, L., Sattler, T., Geiger, A., and Pollefeys, M. (2018). Towards robust visual odometry with a multi-camera system. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1154–1161. IEEE.
- [36] Liu, W., Li, F., Jing, C., Wan, Y., Su, B., and Helali, M. (2021). Recognition and location of typical automotive parts based on the rgb-d camera. *Complex & Intelligent Systems*, 7:1759–1765.
- [37] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee.
- [38] Luo, C., Yang, W., Huang, P., and Zhou, J. (2019). Overview of image matching based on orb algorithm. *Journal of Physics: Conference Series*, 1237(3).
- [39] Ma, Y., Soatto, S., Košecák, J., and Sastry, S. (2004). *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer.
- [40] MathWorks (2024). Ransac. <https://fr.mathworks.com/discovery/ransac.html>. Accessed: 2024-06-04.
- [41] Metzger, B. H., Salmond, D., and Tilstone, A. (2010). Fpso environmental-risk management. *SPE Economics & Management*, 2(01):38–50.
- [42] Morris, D. D. and Kanade, T. (2000). Image-consistent surface triangulation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 332–338. IEEE.
- [43] Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163.
- [44] National Offshore Petroleum Safety and Environmental Management Authority (2021). Improving Offshore Crane Safety. Accessed: 2024-08-12.
- [45] OpenCV (2024). Feature matching. [https://docs.opencv.org/4.x/df/d0c/tutorial\\_py\\_fast.html](https://docs.opencv.org/4.x/df/d0c/tutorial_py_fast.html). Accessed: 2024-06-04.

- [46] Özyeşil, O., Voroninski, V., Basri, R., and Singer, A. (2017). A survey of structure from motion\*. *Acta Numerica*, 26:305–364.
- [47] Pareek, A. and Arora, N. (2019). Evaluation of feature detector-descriptor using ransac for visual tracking. In *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM)*, Amity University Rajasthan, Jaipur-India.
- [48] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., et al. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5.
- [49] Rupnik, E., Nex, F., and Remondino, F. (2013). Automatic orientation of large blocks of oblique images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40:299–304.
- [50] Sahoo, S. K., Tang, D., and Dang, C. (2017). Single-shot multispectral imaging with a monochromatic camera. *Optica*, 4(10):1209–1213.
- [51] Schmid, C. and Zisserman, A. (1998). The geometry and matching of curves in multiple views. In *Computer Vision—ECCV’98: 5th European Conference on Computer Vision Freiburg, Germany, June, 2–6, 1998 Proceedings, Volume I 5*, pages 394–409. Springer.
- [52] Schops, T., Sattler, T., and Pollefeys, M. (2019). Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 134–144.
- [53] Silva, R. M. d. (2010). Impactos da utilização de tecnologias logísticas na atividade de separação de pedidos no ambiente de armazenagem de produtos prontos: estudo de caso em uma empresa de serviços. Master’s thesis, PPG Engenharia de Produção e Sistemas, Universidade do Vale do Rio do Sinos.
- [54] Smith, A. R. (1978). Color gamut transform pairs. *ACM Siggraph Computer Graphics*, 12(3):12–19.
- [55] Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. ACM.
- [56] Solem, J. (2012). *Programming Computer Vision with Python: Tools and algorithms for analyzing images*. ”O’Reilly Media, Inc.”.
- [57] Song, H.-J., Chen, Y.-Z., and Gao, Y.-Y. (2013). Velocity calculation by automatic camera calibration based on homogenous fog weather condition. *International Journal of Automation and Computing*, 10(2):143–156.

- [58] Steinmann, P., Tardy, Y., and Hintermann, H. (1987). Adhesion testing by the scratch test method: the influence of intrinsic and extrinsic parameters on the critical load. *Thin solid films*, 154(1-2):333–349.
- [59] Strobl, K. H. and Hirzinger, G. (2011). More accurate pinhole camera calibration with imperfect planar target. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1068–1075. IEEE.
- [60] Szeliski, R. (2022). *Computer vision: algorithms and applications*. Springer Nature.
- [61] Ukooa, F. (2002). Design guidance notes for ukcs service. *RM Offshore LTD., Project Reviews LTD.*
- [62] Vadivel, A., Mohan, M., Sural, S., and Majumdar, A. K. (2005). Object level frame comparison for video shot detection. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1*, volume 2, pages 235–240. IEEE.
- [63] Van Rossum, G. and Drake, F. L. (2009). Python 2.6 reference manual.
- [64] Wu, J., Cui, Z., Sheng, V. S., Zhao, P., Su, D., and Gong, S. (2013). A comparative study of sift and its variants. *Measurement science review*, 13(3):122–131.
- [65] Xiao, W. S. X. Y. S. and Hu, W. (2008). Robust recognition of checkerboard pattern for deformable surface matching in multiple views. *HIGH PERFORMANCE COMPUTING & SIMULATION (HPCS 2008)*, page 265.
- [66] Xiong, G., Helo, P., Ekström, S., and Shen, Z. (2024). A service-oriented autonomous crane system. *IEEE Transactions on Computational Social Systems*.
- [67] Yang, J., Li, H., and Jia, Y. (2014). Optimal essential matrix estimation via inlier-set maximization. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I 13*, pages 111–126. Springer.
- [68] Yang, T., Sun, N., Chen, H., and Fang, Y. (2020). Swing suppression and accurate positioning control for underactuated offshore crane systems suffering from disturbances. *IEEE/CAA Journal of Automatica Sinica*, 7(3):892–900.
- [69] Yu, W., Xie, J., Wu, X., and Liu, K. (2017). Research of improved zhang’s calibration method. In *2017 Chinese Automation Congress (CAC)*, pages 1423–1427. IEEE.