

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

DIONATA DA SILVA NUNES

**Construção de nuvem de pontos a partir do movimento de uma câmera
monocular**

Dissertação apresentada como requisito parcial para
a obtenção do grau de Mestre em Microeletrônica.

Orientador: Prof. Dr. Altamiro Amadeu Susin

Porto Alegre
2024

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Nunes, Dionata da Silva

Construção de nuvem de pontos a partir do movimento de uma câmera monocular / Dionata da Silva Nunes. – Porto Alegre: PGMICRO da UFRGS, 2024.

150 f.:il.

Dissertação (Mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Microeletrônica. Porto Alegre, BR – RS, 2024. Orientador: Altamiro Amadeu Susin;

1. Câmeras monoculares. 2. Nuvem de pontos. 3. Reconstrução 3D. 4. Algoritmos de SfM. 5. Incerteza de medição. I. Susin, Amadeu Susin. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitor: Prof. Patricia Pranke

Pró-Reitor de Pós-Graduação: Prof^a. Ione Borcelli Gonçalves

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do PGMICRO: Prof. Cláudio Radtke

Bibliotecário-Chefe do Instituto de Informática: Alexander Borges Ribeiro

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão a todos que me apoiaram ao longo do processo de elaboração desta dissertação de mestrado. Em primeiro lugar, agradeço ao meu orientador, Professor Altamiro Amadeu Susin, por sua orientação, apoio e expertise inestimáveis. Sua dedicação e comprometimento com o meu desenvolvimento acadêmico e profissional foram muito importantes para moldar a direção desta pesquisa.

Agradeço também aos membros da minha banca examinadora por seus valiosos sugestões e críticas construtivas. Suas experiências contribuíram significativamente para o aprimoramento deste trabalho.

Minha gratidão estende-se à equipe e ao corpo docente da Universidade Federal do Rio Grande do Sul, cujos recursos, instalações e ambiente acadêmico proporcionaram uma excelente base para a realização desta pesquisa. O acesso aos equipamentos necessários e o suporte técnico foram muito importantes para os experimentos e análises de dados.

Agradeço aos meus colegas do Instituto Senai de Inovação e demais pesquisadores que forneceram apoio, discussões valiosas ao longo desta jornada. Suas contribuições e colaborações foram de grande importância para ampliar minha compreensão do assunto e aprimorar minha abordagem de pesquisa.

Finalmente, agradeço à minha família e amigos por seu apoio incondicional, incentivo constante e compreensão durante todo o processo. Seu amor e encorajamento foram uma motivação e bem-estar ao longo desta jornada acadêmica.

A todos que, de alguma forma, contribuíram para este trabalho, meu mais sincero obrigado.

RESUMO

O Processamento Digital de Imagens (PDI) e a captura de imagens evoluíram significativamente, permitindo sua aplicação em diversas áreas, como medicina, esporte, indústria e arte. A captura de imagens no espectro visível possibilita muitas aplicações, incluindo robótica, automação, controle de qualidade e monitoramento de ambientes. Técnicas de PDI também podem ser aplicadas a sinais de radar, ressonância magnética, imagens térmicas e hiperespectrais. Este trabalho explora as relações geométricas das projeções do espaço tridimensional para obter informações sobre a localização de pontos do cenário e a pose de uma câmera monocular. Utilizando algoritmos de identificação de detalhes (features) e sua correspondência em imagens capturadas de diferentes pontos de vista, o sistema determina as coordenadas desses pontos. Esses pontos formam uma nuvem que representa a estrutura do cenário, técnica conhecida como *Structure from Motion* (SfM). A parte experimental deste projeto foi realizada em laboratório, onde uma câmera foi acoplada a um braço robótico KUKA. O uso do braço robótico foi fundamental para garantir a repetibilidade dos movimentos, bem como a precisão na captura das imagens, assegurando que a câmera percorresse trajetórias programadas em um ambiente controlado. A automação proporcionada pelo braço robótico permitiu não apenas uma maior confiabilidade nos dados capturados, mas também flexibilidade para adaptar o experimento a diferentes peças e condições. Para avaliar o desempenho do sistema SfM, uma nuvem de pontos gerada por esse método foi comparada a uma nuvem de pontos da mesma peça, gerada por uma máquina de medição por coordenadas ZEISS Prismo Ultra, cujas medições possuem incerteza conhecida. Os resultados preliminares indicam que o sistema SfM apresentou medições lineares com desvios dentro de tolerâncias aceitáveis para diversas aplicações industriais, com desvios de até 2 mm em peças de formato escalonado. Experimentos complementares foram realizados em uma sala com esferas fixadas nas paredes, onde os desvios nas medições das esferas variaram de 4 mm a 9 mm, especialmente em distâncias maiores, quando comparadas a sistemas de referência como o *Laser Tracker LEICA*. Foram elaborados mapas de calor e histogramas para analisar a distribuição e densidade dos pontos. Gráficos de erros de reprojeção ao longo dos eixos X e Y mostraram a precisão do processo de projeção. Os resultados detalhados são apresentados no texto da dissertação.

Palavras-chave: *Structure from Motion*, monocular, visão computacional, reconstrução 3D, nuvem de pontos, fotogrametria

ABSTRACT

Digital Image Processing (DIP) and image capture have evolved significantly, enabling their application in various fields such as medicine, sports, industry, and art. Capturing images in the visible spectrum allows for many applications, including robotics, automation, quality control, and environmental monitoring. DIP techniques can also be applied to radar signals, magnetic resonance imaging, thermal and hyperspectral images. This work explores the geometric relationships of projections from three-dimensional space to obtain information about the location of points in the scene and the pose of a monocular camera. By using feature detection algorithms and their matching in images captured from different viewpoints, the system determines the coordinates of these points. These points form a cloud representing the structure of the scene, a technique known as Structure from Motion (SfM). The experimental part of this project was carried out in a laboratory, where a camera was mounted on a KUKA robotic arm. The use of the robotic arm was essential to ensure the repeatability of movements as well as the precision in image capture, ensuring that the camera followed programmed trajectories in a controlled environment. The automation provided by the robotic arm not only ensured greater reliability in the captured data but also flexibility to adapt the experiment to different parts and conditions. To evaluate the performance of the SfM system, a point cloud generated by this method was compared to a point cloud of the same part, generated by a ZEISS Prism Ultra coordinate measuring machine, whose measurements have known uncertainty. Preliminary results indicate that the SfM system showed linear measurements with deviations within acceptable tolerances for various industrial applications, with deviations of up to 2 mm in stair-shaped parts. Complementary experiments were conducted in a room with spheres fixed to the walls, where the measurement deviations of the spheres ranged from 4 mm to 9 mm, especially at greater distances, when compared to reference systems such as the LEICA Laser Tracker. Heat maps and histograms were generated to analyse the distribution and density of the points. Reprojection error graphs along the X and Y axes showed the accuracy of the projection process. Detailed results are presented in the dissertation text.

Keywords: Structure from Motion, monocular, computer vision, 3D reconstruction, point cloud, photogrammetry.

LISTA DE FIGURAS

Figura 1 - Coordenadas definidas para o movimento do braço robótico KUKA.....	18
Figura 2 - Movimento executado pelo braço robótico kuka.....	18
Figura 3 - Esquemático de projeção da câmera pinhole.....	24
Figura 4 - Coordenadas no sensor da câmera.....	25
Figura 5 - Linhas epipolares.....	26
Figura 6 - Perspective-n-Point (PnP).....	28
Figura 7 -Pontos encontrados do espaço tridimensional.....	29
Figura 8 - Técnicas mais citadas sobre SfM.....	41
Figura 9 - Visão geral da solução proposta.....	44
Figura 10 - Diagrama para gerar nuvem de pontos por SfM.....	45
Figura 11 - Fluxograma do algoritmo para detecção de pontos-chave e geração de descritores.....	46
Figura 12 - Imagens com diferentes resoluções.....	47
Figura 13 - Distribuição gaussiana bidimensional - $x=-3$ a 3 $y=-3$ a 3 $\sigma=0.707$ e - $x=-3$ a 3 $y=-3$ a 3 $\sigma=2.828$	48
Figura 14 - Aplicando o kernel nos pixels das extremidades da imagem.....	51
Figura 15 - Ajuste realizado na imagem para aplicação do kernel.....	51
Figura 16 - Imagem suavizada.....	51
Figura 17 - Organização das imagens DoG.....	52
Figura 18 - Imagem DoG.....	53
Figura 19 - Pontos-chave.....	54
Figura 20 - Definição as 16 sub-regiões com suas 8 direções pré-definidas.....	55
Figura 21 - Imagem (A) vetor resultante com os 16 histogramas concatenados (8 direções).....	56
Figura 22 - Função densidade de probabilidade típica para correspondência de pontos SIFT (Lowe, 2004).....	57
Figura 23 - Gráfico de correspondências – limiar 0.5.....	58
Figura 24 - Gráfico de correspondências – limiar 0.7.....	58
Figura 25 - Pontos-chave e suas correspondências (Imagem (A) algoritmo desenvolvido e imagem (B) SIFT OpenCV).....	59
Figura 26 - Correspondências em ambas as implementações (Imagem (A) algoritmo desenvolvido e imagem (B) SIFT OpenCV).....	59
Figura 27 - Imagem (A) descritor ponto de vista 1 e imagem (B) descritor ponto de vista 2 (algoritmo desenvolvido).....	60
Figura 28 - Imagem (C) descritor ponto de vista 1 e imagem (D) descritor ponto de vista 2 (SIFT OpenCV).....	60
Figura 29 - Imagem (E) sobreposição de descritores do algoritmo desenvolvido (Figura 27), imagem (F) referente ao SIFT do OpenCV (Figura 28).....	61
Figura 30 - Imagem (A) 58 x 58 pixels e imagem (B) com 56 x 56 pixels.....	61
Figura 31 - Fluxos para obter a matriz essencial (E).....	64
Figura 32 - Pontos correspondentes antes da normalização.....	65
Figura 33 - Pontos, média e desvio padrão (ponto de vista 1).....	66
Figura 34 - Pontos, média e desvio padrão (ponto de vista 2).....	67
Figura 35 - Pontos correspondentes normalizados.....	68
Figura 36 - Matriz A	69
Figura 37 - Matriz AAT	69
Figura 38 - Autovalores.....	70
Figura 39 - Autovetor para o autovalor $\lambda = 31,834$	70
Figura 40 - Matriz ATA	70
Figura 41 - Matriz VT	71
Figura 42 - Matriz S	71
Figura 43 - Matriz ajustada.....	71
Figura 44 - Matriz essencial (E).....	72

Figura 45 – Matriz essencial (E) resultante.	72
Figura 46 - Ponto dos pontos de vista 1.	72
Figura 47 - Cálculo da linha epipolar.	72
Figura 48 - Linha epipolar normalizada.	73
Figura 49 – Avaliação da matriz essencial (E) para um determinado ponto.	73
Figura 50 - Linha epipolar traçada.	74
Figura 51 - Decomposição da matriz essencial (E).	77
Figura 52 – Matriz de calibração.	78
Figura 53 - Matriz identidade.	78
Figura 54 - Matrizes de projeção.	78
Figura 55 - Matriz A calculada.	79
Figura 56 - Coordenadas homogêneas.	79
Figura 57 - Transformação para coordenadas cartesianas.	79
Figura 58 - Ponto 3D.	79
Figura 59 - Marcador ArUco.	84
Figura 60 - Marcadores sendo detectados.	84
Figura 61 - Imagens capturadas durante o movimento do braço robótico KUKA.	86
Figura 62 - Nuvem de pontos gerada.	87
Figura 63 - Mapa de calor objeto de frente.	88
Figura 64 - Histograma da nuvem de pontos.	89
Figura 65 - Desvio padrão da nuvem de pontos gerada.	90
Figura 66 - Nuvem de pontos gerada (nova posição do objeto).	90
Figura 67 - Mapa de colar objeto segunda posição.	91
Figura 68 – Histograma.	91
Figura 69 - Desvio padrão.	92
Figura 70 – Cenário externo de planta química.	93
Figura 71 - Nuvem de pontos gerada pelo sistema por SfM.	93
Figura 72 – Densificação da nuvem gerada por SfM utilizando a biblioteca OpenMVS.	94
Figura 73 - Planos utilizados.	95
Figura 74 - Modelo de referência.	95
Figura 75 - Marcação das medições.	95
Figura 76 - Planos definidos no software da ZEISS Prismo ultra.	95
Figura 77 - Destaque apalpador.	96
Figura 78 - Medições sendo realizadas pelo ZEISS Prismo ultra.	96
Figura 79 - Braço robótico KUKA.	98
Figura 80 - Frames do objeto padrão.	98
Figura 81 - Nuvem de pontos gerada.	99
Figura 82 - Interseção entre planos.	100
Figura 83 - Todo os planos utilizados nas medições.	101
Figura 84 - Definição da interseção entre planos.	101
Figura 85 - SfM X Zeiss Prismo Ultra.	102
Figura 86 - Setup do ensaio para reconstrução 3D.	103
Figura 87 - Esfera utilizada no ensaio.	103
Figura 88 - Apalpador realizando as medidas na esfera.	104
Figura 89 - Nuvem de pontos gerado por SfM.	104
Figura 90 - Resultado das medições das 9 esferas.	105
Figura 91 - Resultado das medições entre esferas.	106
Figura 92 - Gráfico de comparação das medições realizadas nos dois ensaios.	107
Figura 93 - Medições entre a parede A e B.	107
Figura 94 – Gráfico dos resultados das medições entre a parede A e B.	108
Figura 95 - Medições entre a parede A e C.	108
Figura 96 - Gráfico dos resultados das medições entre a parede A e C.	109
Figura 97 - Gráfico comparativo das medições realizadas entre a parede A-B e parede A-C.	109
Figura 98 - Erro reprojeção eixo X.	110

Figura 99 - Erro de reprojeção eixo Y.	111
---	-----

LISTA DE TABELAS

Tabela 1 - Configuração das câmeras utilizadas.....	16
Tabela 2 - Cenários utilizados nos experimentos.....	19
Tabela 3 - Equipamentos utilizados como referência nos experimentos	20
Tabela 4 - Tabela Comparativa entre os Trabalhos	39
Tabela 5 - Valores de sigmas utilizados.....	49
Tabela 6 - Comparativo entre algoritmo desenvolvido e SIFT (resolução 1840 x 4080 pixels).	62
Tabela 7 - Comparativo entre algoritmo desenvolvido e SIFT (resolução 368 x 816 pixels).	62
Tabela 8 - Resultado objeto padrão (software ZEISS Prismo). (Obs.: medidas estão milímetro).....	97
Tabela 9 - Resultados das medições. (Obs.: medidas estão milímetro)	102

LISTA DE ABREVIATURAS E SIGLAS

A/D	Analógico para digital
BA	<i>Bundle Adjustment</i>
CCD	<i>Charged Coupled Device</i>
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
DOE	<i>Design of Experiments</i>
FDP	<i>Função Densidade de Probabilidade</i>
ISI	<i>Instituto Senai de Inovação</i>
GUIs	<i>Graphical User Interfaces</i>
LMEDS	<i>Least Median of Squares</i>
M_{ext}	<i>Matriz extrínseca</i>
M_{int}	<i>Matriz intrínseca</i>
OLS	<i>Ordinary Least Squares</i>
OpenMVS	<i>Open Multiple View Stereo Vision</i>
ORB	<i>Oriented FAST and Rotated BRIEF</i>
PCD	<i>Point Cloud Data</i>
PCL	<i>Point Cloud Library</i>
RANSAC	<i>Random Sample Consensus</i>
SfM	<i>Structure from Motion</i>
SIFT	<i>Scale-Invariant Feature Transform</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
SURF	<i>Speeded Up Robust Features</i>
SVD	<i>Decomposição em Valores Singulares</i>
8POINT	<i>8-point algorithm</i>

SUMÁRIO

1. INTRODUÇÃO	12
1.1 Motivação	13
1.2 Objetivos	13
1.2.1 Objetivo Geral.....	13
1.2.1 Objetivos Específicos.....	13
1.3 Contribuições	14
1.4 Organização	14
2. MATERIAIS E MÉTODOS	15
2.1 Câmeras digitais utilizadas e movimentação das câmeras	16
2.2 Cenários calibrados para a validação do sistema	19
2.3 Plataforma de desenvolvimento	21
2.4 Operações matemáticas para projeção e determinação da pose da câmera	22
2.4.1 Projeção (geometria projetiva).....	22
2.4.2 Triangulação (geometria epipolar).....	26
2.4.3. Estimção de posição e orientação	27
2.5 Calibração da câmera	28
2.6 Aquisição das imagens	30
2.7 Identificação, codificação e rastreamento de detalhes	31
2.8 Refinamento das coordenadas dos detalhes e exclusão outliers	32
2.8.1 RANSAC	32
2.8.2 Menor mediana de quadrados	33
2.9 Exibição e exportação da nuvem de pontos	35
3. TRABALHOS CORRELATOS	36
4. ALGORITMO SFM PROPOSTO	43
4.1 Captura de imagens para geração da nuvem de pontos	45
4.2 Correspondência entre descritores	46
4.4 Recuperação da pose	63
4.4.1 Recuperação da pose inicial.....	63
4.4.2 Recuperação da pose nas imagens subsequentes	75
4.5 Construção da nuvem de pontos	77
4.6 Ajuste erro de reprojeção	82
4.6.1 Ajuste de pacote	82
4.7 Recuperação de escala	84
5. EXPERIMENTOS E RESULTADOS	86
5.1 Análise dos pontos gerados	86
5.2 Incerteza de medição	94
5.2.1 Medições da peça de referência com a Zeiss Prismo Ultra	94
5.2.2 Medições da peça de referência com SfM	97
5.2.3 Avaliação de incerteza de medição de uma sala	102
5.3 Análise dos erros de reprojeção	110
6. CONCLUSÕES E TRABALHOS FUTUROS	112
REFERÊNCIAS	114
APÊNDICES	120
ANEXOS	139

1. INTRODUÇÃO

A visão computacional é uma ferramenta importante na indústria, especialmente no controle de qualidade, permitindo inspeções rápidas que não interferem no processo de produção. Nesse contexto, este trabalho apresenta um sistema que realiza a construção de uma nuvem de pontos utilizando uma câmera monocular por meio da técnica de *Structure from Motion* (SfM). Essa técnica identifica os principais pontos de interesse em uma imagem e os relaciona com os mesmos pontos em uma imagem capturada pela câmera em outra posição. A partir dessas correspondências, é possível construir uma única nuvem de pontos.

Para avaliar o desempenho do sistema, uma câmera foi acoplada a um braço robótico KUKA em um ambiente controlado, com o objetivo de gerar uma nuvem de pontos de uma peça. A escolha do braço robótico é motivada pela sua capacidade de garantir a repetibilidade nos movimentos, além de controlar a posição da câmera durante a aquisição das imagens. O uso do braço também permite a automação do processo de captura e oferece flexibilidade para a adaptação do experimento a diferentes objetos e condições. Posteriormente, uma máquina de medição por coordenadas ZEISS Prismo Ultra, que possui incerteza conhecida, também gerou uma nuvem de pontos da mesma peça. Dessa forma, foi possível determinar o erro e quantificar a incerteza do sistema projetado.

Os resultados preliminares indicam que o sistema SfM apresentou medições lineares com desvios dentro das tolerâncias aceitáveis para diversas aplicações industriais. Por exemplo, nas medições realizadas com o braço robótico em uma peça com formato de escada, utilizada para determinar a incerteza do sistema, os desvios foram de até 2 mm. Em experimentos complementares realizados em uma sala com esferas fixadas nas paredes para as medições, os desvios nas medições das esferas ficaram na faixa de 5 mm. Já nas medições de distâncias entre centros de esferas no mesmo plano, as incertezas variaram entre 4 mm e 9 mm. Comparado às referências de sistemas de medição como a ZEISS Prismo Ultra e o *Laser Tracker LEICA*.

Essa abordagem experimental permitiu compreender e quantificar a precisão do processo implementado de construção de nuvem de pontos. Ao combinar a técnica de SfM com um ambiente controlado e uma referência de máquina de medição confiável, foi possível realizar uma análise do sistema e fornecer informações sobre sua precisão e incerteza. Esse

conhecimento é importante para garantir a confiabilidade das aplicações industriais que dependem da construção precisa de nuvens de pontos.

1.1 Motivação

A visão computacional desempenha um papel importante na indústria ao empregar algoritmos avançados para interpretar e extrair informações a partir de dados visuais. Essa tecnologia é contribui para inspeções automatizadas de maior qualidade, detecção de falhas em linhas de produção, monitoramento de segurança por meio de câmeras inteligentes e otimização de processos industriais. Ao integrar capacidades de visão computacional, as indústrias podem aumentar a eficiência operacional, reduzir custos e aprimorar a tomada de decisões.

Nesse contexto, este trabalho apresenta um algoritmo que utiliza uma câmera monocular e a técnica de SfM para a construção de nuvens de pontos. Essa abordagem visa obter informações geométricas sobre objetos tridimensionais, atendendo às necessidades das aplicações que dependem da reconstrução dimensional de objetos e ambientes.

Para avaliar o desempenho do sistema proposto, uma câmera é acoplada a um braço robótico em um ambiente controlado, permitindo a geração da nuvem de pontos de uma peça específica. Além disso, uma máquina de medição de referência com incerteza conhecida gera a nuvens de pontos da mesma peça. Essa metodologia proporciona uma análise do sistema e a modelagem do erro, permitindo uma avaliação de seu desempenho.

1.2 Objetivos

O levantamento dimensional por visão computacional (captura e processamento de imagens) permite medida rápida e sem contato facilitando a instalação e operação. A captura de várias vistas do objeto pode ser feita pelo movimento da câmera ou do objeto.

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um sistema para a determinação das coordenadas espaciais de pontos específicos de um objeto tridimensional por meio do processamento de imagens do objeto sob diversos pontos de vista.

1.2.1 Objetivos Específicos

- Desenvolver um sistema para a construção de nuvens de pontos utilizando uma câmera monocular e a técnica de SfM.

- Realizar a captura de imagens por meio da câmera acoplada a um braço robótico em um ambiente controlado.
- Gerar uma nuvem de pontos da peça específica por meio do algoritmo de construção.
- Comparar os resultados obtidos com a nuvem de pontos gerada pela máquina de medição por coordenadas ZEISS Prismo Ultra.
- Determinar o erro e realizar uma análise detalhada da incerteza do sistema projetado.
- Fornecer informações sobre a precisão e a incerteza do sistema, contribuindo para a confiabilidade das aplicações que dependem da construção precisa de nuvens de pontos.

1.3 Contribuições

Este trabalho tem como contribuição principal o desenvolvimento de um sistema para reconstrução 3D utilizando imagens tendo como base a técnica de SfM. No desenvolvimento do sistema foram utilizadas várias técnicas que compuseram o sistema que merecem destaque:

As contribuições deste texto são:

- Desenvolvimento de um sistema para a construção de nuvens de pontos utilizando uma câmera monocular e a técnica de SfM.
- Descrição da configuração experimental, em que uma câmera é acoplada a um braço robótico (KUKA) em um ambiente controlado para gerar uma nuvem de pontos de uma peça específica. Além disso, uma máquina de medição por coordenadas ZEISS Prismo Ultra também gera uma nuvem de pontos da mesma peça, permitindo uma comparação e análise do desempenho do sistema.
- Entendimento dos limites práticos das técnicas de SfM.

1.4 Organização

A dissertação está organizada em 6 capítulos. Nesta Introdução foram apresentados a motivação, os objetivos gerais e específicos, as contribuições esperadas e a organização do trabalho. O capítulo de materiais e métodos detalha os procedimentos e técnicas empregados, como a captura de imagens, geometria projetiva e epipolar, calibração e detecção de detalhes. A implementação do sistema é discutida em profundidade, cobrindo desde a calibração da câmera até a recuperação da pose e triangulação 3D. Resultados e discussões analisam os dados obtidos e avaliam a precisão e incertezas das medições. A dissertação conclui com as considerações finais e sugestões para trabalhos futuros.

2. MATERIAIS E MÉTODOS

As imagens de um cenário tridimensional, capturadas por uma câmera, apresentam a projeção de um ponto de vista deste cenário num plano, perdendo-se uma dimensão. Com duas imagens tomadas em pontos de vista distintos, nas quais elementos possam ser identificados em ambas as imagens, é possível determinar a dimensão perdida, a menos da escala e também a pose da câmera. Este trabalho apresenta a implementação de um sistema de processamento de imagens para a determinação das coordenadas de pontos de um cenário tridimensional e da pose da câmera, a partir de imagens deste cenário, capturadas em pontos de vista diversos. A determinação da correspondência de pontos em imagens distintas é feita por algoritmos de processamento de imagens para a seleção e codificação de pontos de interesse para que possam ser rastreados nas diversas imagens. Ao conjunto dos pontos assim determinadas chama-se "nuvem de pontos". Em procedimentos independentes pode-se realizar o processo de calibração da(s) câmera(s) e de recuperação da escala.

Neste capítulo são apresentados os equipamentos, procedimentos, algoritmos desenvolvidos e bibliotecas de software utilizadas nas diversas etapas. As operações realizadas podem ser organizadas nos seguintes procedimentos do processo:

- 1- Câmeras digitais utilizadas e movimentação das câmeras;
- 2- Cenários calibrados para a validação do sistema;
- 3- Plataforma de desenvolvimento; linguagem Python e bibliotecas de acesso livre.
- 4- Operações matemáticas para projeção e determinação da pose da câmera;
- 5- Calibração da câmera;
- 6- Aquisição das imagens;
- 7- Identificação, codificação e rastreamento de detalhes;
- 8- Refinamento das coordenadas dos detalhes e exclusão outliers;
- 9- Exibição e exportação da nuvem de pontos.

Os experimentos foram realizados nos espaços do Instituto Senai [site] que dispõe dos equipamentos e laboratórios adequados.

2.1 Câmeras digitais utilizadas e movimentação das câmeras

A captura de imagens é feita por câmeras digitais. Foram utilizadas as câmeras *DFK AFU050-L34 Color Camera* e *Canon D200* as características técnicas das câmeras utilizada está listado na Tabela 1.

Tabela 1 - Configuração das câmeras utilizadas.

COMPORTAMENTO GERAL	DFK AFU050-L34 Color	Nikon D200
Imagem da câmera		
Formatos de vídeo @ taxa de quadros (máximo)	2,592×1,944 (5 MP) MJPG @ 15 fps 1,920×1,080 (2.1 MP) MJPG @ 30 fps 1,280×720 (0.9 MP) MJPG @ 60 fps	Não suporta gravação de vídeo
Tipo de sensor	CMOS	CCD, 10,2 MP, formato DX ¹
Obturador	<i>rolling shutter</i>	<i>global shutter</i>
Tamanho dos pixels	1,96 µm	6,05 µm
Distância focal	3.4 mm	Compatível com lentes de montagem Nikon F
Interface	USB 2.0 (compatível com versões futuras)	USB, suporte a GPS (opcional)
Tensão de alimentação	4.5 VDC à 5.5 VDC	Alimentada por bateria recarregável EN-EL3e
Dimensões	A: 36 mm, L: 36 mm, C: 18 mm	147 mm x 113 mm x 74 mm
Massa	65 g	830 g
Ganho	50 dB - ISO 3200 dB	ISO de 100 a 1600, expansível até ISO 3200

¹ O termo **DX** refere-se a um formato de sensor desenvolvido pela Nikon para suas câmeras DSLR. O sensor **DX** tem um tamanho menor do que o sensor full-frame (também chamado de formato **FX**). Especificamente, o sensor DX tem dimensões de aproximadamente 24 x 16 mm, enquanto o sensor FX (full-frame) tem 36 x 24 mm, equivalente ao tamanho do filme 35 mm.

Processamento de imagem	modos: nitidez, brilho, contraste	Espaços de cor sRGB e Adobe RGB suportados
Processamento de cores	modos: saturação, matiz	Configurações personalizáveis de cores como vivo, normal, retrato, suave, entre outras

Fonte: autoria própria

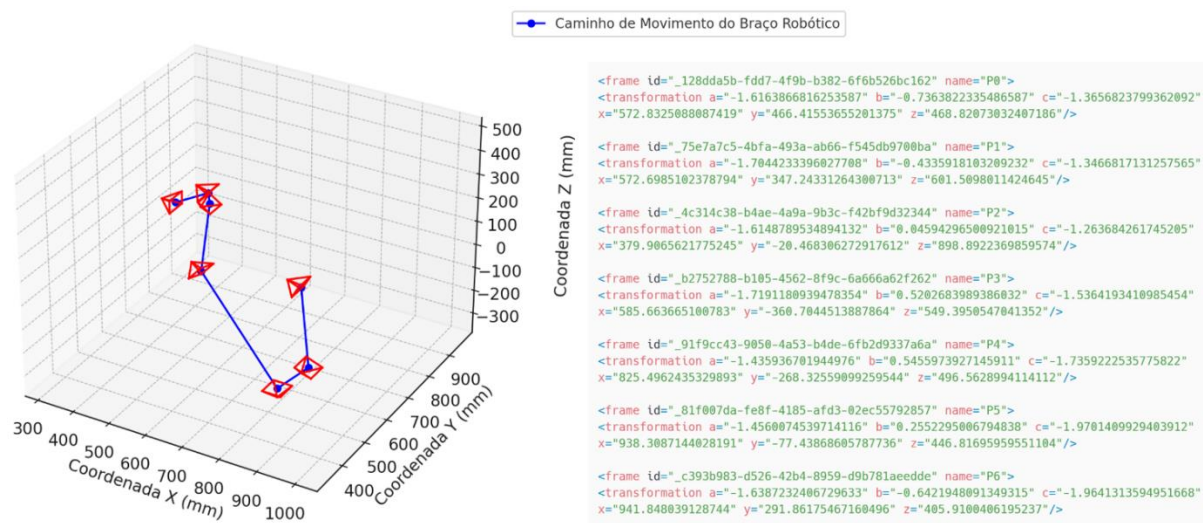
A movimentação das câmeras foi realizada de duas formas: a) acoplada a um braço robótico e b) conduzida manualmente.

Os experimentos que utilizaram a movimentação a) foram realizados em um ambiente controlado, onde um braço robótico KUKA² (Anexo B) foi programado para transportar a câmera FU050-L34 Color Camera ao longo de uma trajetória pré-definida ao redor de uma peça fabricada especificamente para o experimento. A velocidade de captura das imagens foi ajustada, levando em consideração a velocidade de movimento do braço robótico, frame rate da câmera e obturador garantindo a sobreposição adequada entre as imagens. Essa sobreposição foi necessária para a aplicação do método de SfM, conforme descrito no cálculo da seção 4.1. A configuração permitiu a repetição dos experimentos, viabilizando a avaliação da incerteza associada às medições realizadas sobre o objeto.

O uso do braço robótico KUKA traz diversas vantagens, como a repetibilidade das capturas e a redução do tempo total de operação. A capacidade de programar e reutilizar as trajetórias permite que o sistema seja facilmente adaptado para diferentes objetos e cenários de análise. O braço robótico foi programado utilizando o software KUKA *Sunrise Workbench* em conjunto com a linguagem de programação Java. As coordenadas de movimento foram calculadas com base nas características geométricas de cada objeto e nos requisitos específicos de cada análise, garantindo que todas as superfícies relevantes fossem capturadas. A Figura 1 mostra as coordenadas de movimento geradas para o escaneamento.

² Robô modelo da linha KUKA LBR iiwa (*Intelligent Industrial Work Assistant*), robô colaborativo de 7 eixos. Projetado para realizar tarefas em ambientes colaborativos, permitindo a interação segura com humanos e outras máquinas. O LBR iiwa tem uma repetibilidade de $\pm 0,1$ mm. Além disso, ele é equipado com sensores de torque em cada junta, permitindo detectar forças e ajustar seus movimentos em tempo real.

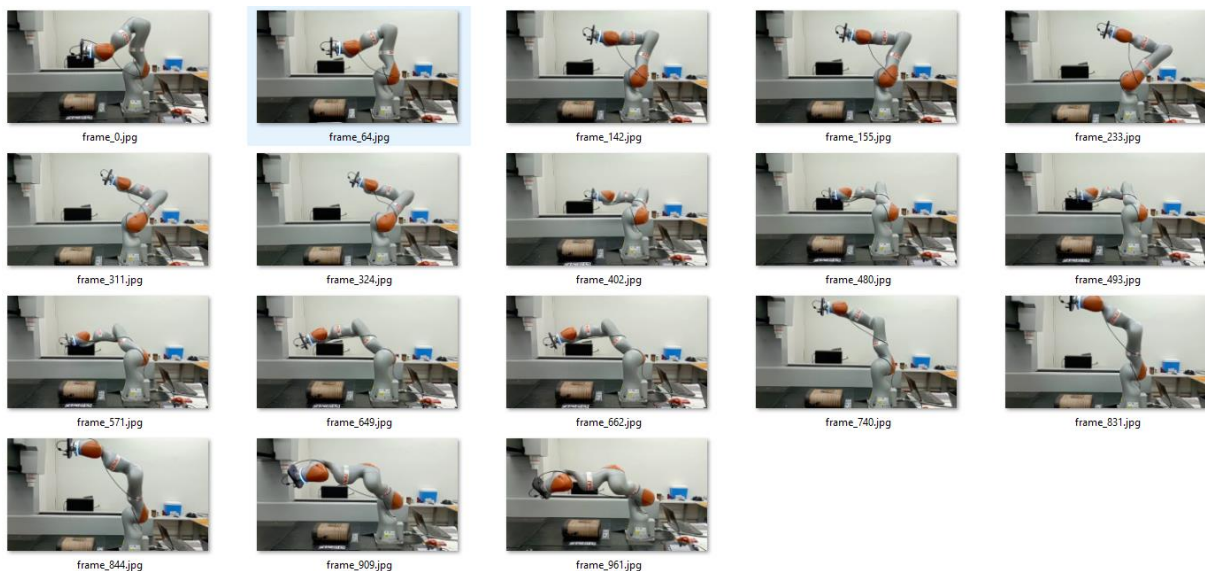
Figura 1 - Coordenadas definidas para o movimento do braço robótico KUKA.



Fonte: autoria própria

As trajetórias do braço foram planejadas para garantir a coleta de dados em diferentes posições e ângulos ao longo de sua movimentação. A captura das imagens foi realizada em pontos específicos da trajetória do braço robótico, proporcionando múltiplas visões do objeto (conforme ilustrado nas imagens da Figura 2, que mostram os diferentes frames capturados durante o experimento).

Figura 2 - Movimento executado pelo braço robótico kuka.



Fonte: autoria própria

Além da peça fabricada, um segundo objeto, disponível comercialmente, foi utilizado com outro objetivo: em vez de avaliar as dimensões do objeto, o foco deste experimento foi a análise da integridade da nuvem de pontos gerada a partir das imagens capturadas. O estudo

com esse objeto teve como finalidade verificar a consistência do processo de geração de nuvens de pontos.



Esses experimentos foram planejados para validar o processo de captura e medição, bem como explorar possíveis aplicações futuras, como a inspeção dimensional automatizada e a engenharia reversa. O uso do braço robótico KUKA nessas áreas pode ser integrado a sistemas de controle de qualidade e manufatura, permitindo uma coleta automatizada de dados tridimensionais.

No caso de estudo b), dois outros cenários foram utilizados enquanto a câmera Nikon D200 era conduzida manualmente: uma 10 m x 10 m com marcadores afixados nas paredes e uma maquete de instalação química em ambiente externo com as seguintes dimensões: 15 m x 7 m x 5.2 m (largura, comprimento, altura).

2.2 Cenários calibrados para a validação do sistema

Para realizar os experimentos são utilizados quatro cenário para geração da nuvem de pontos uma fabricada para o experimento (Tabela 2 (A)) aferidas com a máquina Zeiss Prismo Ultra (Tabela 3 (B)), uma madeira disponível no mercado (Tabela 2 (C)), uma sala com objetos afixados nas paredes (Tabela 2 (D)) cujas dimensões foram aferidas com o equipamento a laser Laser Tracker LEICA (Tabela 3 (A)) e uma maquete de planta química (Tabela 3 (B)).

Tabela 2 - Cenários utilizados nos experimentos.

(A) Peça fabricada	(B) Maquete de planta química
	
(C) Madeira	(D) Sala com objetos fixados


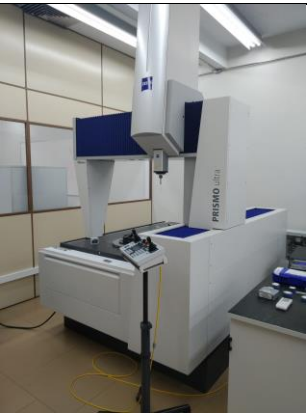


Fonte: autoria própria

Os equipamentos de medição utilizados para aferição incluem o Leica Absolute Tracker AT960, um dispositivo de medição a laser, com um alcance de até 160 metros e precisão linear de até 15 micrômetros + 6 micrômetros por metro. O equipamento é capaz de rastrear alvos móveis em até 1000 Hz. Quando utilizado em conjunto com o acessório T-Scan, o AT960 possibilita a digitalização de superfícies complexas em 3D.

Por sua vez, o Zeiss Prismo Ultra é uma máquina de medição por coordenadas projetada para oferecer medições extremamente precisas em processos de controle dimensional. O Prismo Ultra possui uma precisão inferior a 0,5 micrômetros. Seu design robusto, aliado a um avançado sistema de controle térmico, garante estabilidade e precisão, mesmo em condições ambientais variáveis.

Tabela 3 - Equipamentos utilizados como referência nos experimentos

(A) Leica Absolute Tracker AT960 + T-SCAN	(B) Zeiss Prismo Ultra
	

Fonte: autoria própria

Além disso, foram realizados experimentos complementares, descritos no Apêndice X, onde alguns elementos de uma sala foram medidos utilizando o FARO Focus S (Anexo A), um scanner a laser para captura tridimensional de ambientes e objetos. Com alcance de até 350 metros e precisão de até ± 1 mm, o Focus S foi escolhido devido à sua portabilidade, permitindo a realização de medições precisas de maneira prática e eficiente.

2.3 Plataforma de desenvolvimento

A principal linguagem de programação utilizada no desenvolvimento deste projeto é o Python, devido à sua versatilidade e à ampla gama de bibliotecas disponíveis para processamento de imagens, visão computacional e visualização de dados. As bibliotecas mais utilizadas incluem:

- OpenCV: para captura e processamento de imagens, além de funções de calibração de câmera.
- NumPy: para manipulação de *arrays* e cálculos numéricos eficientes.
- Matplotlib: para visualização de dados e criação de gráficos.
- SCP: para operações matemáticas avançadas, incluindo funções de otimização.
- CloudCompare: utilizado para realizar medições, gerar histogramas de densidade volumétrica de nuvens de pontos e definir planos por meio do método dos mínimos quadrados.
- Tkinter: para criação de interfaces gráficas de usuário (GUIs), facilitando a interação com as funcionalidades do projeto.
- Open3D: utilizado para manipulação e visualização de dados 3D, como nuvens de pontos, malhas e volumes, além de exportar as nuvens de pontos geradas no formato PCD (*Point Cloud Data*).

Essas ferramentas permitiram o processamento, análise, visualização e exportação dos dados em formatos adequados, otimizando o desenvolvimento das atividades relacionadas ao projeto.

2.4 Operações matemáticas para projeção e determinação da pose da câmera

O processo fotográfico é baseado na projeção de um cenário tridimensional numa superfície plana. A formulação matemática usa o modelo de uma câmera ideal chamada "pin hole". Nesta câmera todos os raios luminosos que adentram a câmara escura, passam por este único ponto. Parte deles irão projetar-se sobre o anteparo que contém os elementos sensíveis à luz. Os raios que atingem o plano de projeção formam a imagem da região do universo "visível" pela câmera. O modelo teórico permite equacionar a projeção dos pontos com coordenadas 3D numa superfície bidimensional.

2.4.1 Projeção (geometria projetiva)

A geometria projetiva é uma área da matemática que estuda as propriedades das figuras geométricas que mantêm sua forma quando vistas de diferentes perspectivas ou projeções. Ela estabelece as relações entre figuras e suas imagens projetadas em um plano ou em espaços de dimensões mais elevadas [20].

Alguns dos conceitos importantes na geometria projetiva incluem [44]:

- Ponto: Um ponto é uma entidade fundamental que pode ser entendida como a interseção de duas linhas em um espaço projetivo.
- Reta: Uma reta é uma coleção de pontos alinhados, estendendo-se infinitamente em ambas as direções.
- Plano: Um plano é uma superfície bidimensional infinita composta por uma coleção infinita de pontos e retas.
- Projeção: Uma projeção é uma transformação que mapeia pontos de um espaço projetivo para um plano projetivo, preservando a incidência de retas. Essa operação mantém a colinearidade, ou seja, pontos que estão alinhados em uma reta no espaço projetivo permanecem alinhados em uma reta correspondente no plano projetivo.
- Desenho perspectivo: É uma técnica de representação gráfica que modela a projeção de pontos tridimensionais em um plano bidimensional, usando transformações projetivas para preservar as relações proporcionais e angulares entre os objetos. Isso é feito através do uso de pontos de fuga e linhas convergentes.

A geometria projetiva permite uma variedade mais ampla de transformações em comparação com a geometria euclidiana. No entanto, sua aplicação generalizada é limitada porque grandezas como distância, ângulo e paralelismo não são preservadas. Exemplos de propriedades preservadas em transformações projetivas incluem colinearidade, incidência e a razão dupla. Transformações projetivas típicas incluem rotação, translação, escala uniforme e projeção de perspectiva.

A geometria projetiva utiliza coordenadas homogêneas, adicionando uma dimensão extra ao espaço euclidiano. Na prática, um ponto no espaço tridimensional projetivo é representado por um vetor com quatro componentes. Essa dimensão adicional permite que as coordenadas sejam escaladas proporcionalmente, sem alterar a posição do ponto no espaço. Para coordenadas homogêneas, pontos em duas e três dimensões podem ser representados conforme as equações (1) e (2).

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} kx_1 \\ kx_2 \\ k \end{bmatrix} \quad (1)$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} = \begin{bmatrix} kx_1 \\ kx_2 \\ kx_3 \\ k \end{bmatrix} \quad (2)$$

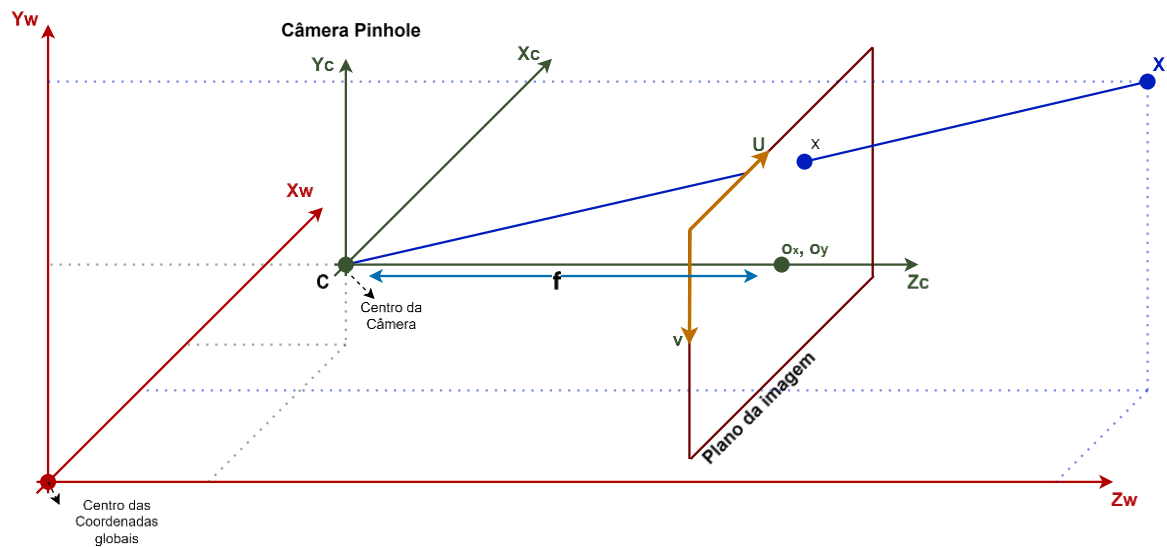
Por meio dessas transformações, chegamos às equações de projeção (equações 3 e 4). Nelas, m_x e m_y correspondem ao tamanho físico do pixel (por exemplo, milímetros por pixel) no sensor da câmera [2]. A distância focal f da câmera e os parâmetros o_x e o_y , que indicam os centros óticos, também são considerados. Caso o tamanho do pixel e a distância focal sejam desconhecidos, eles podem ser combinados em f_x e f_y . As coordenadas da imagem no sensor, chamadas de pixels, são expressas no plano da imagem e medidas em milímetros, conforme mostrado nas equações (3) e (4):

$$u = m_x \cdot \frac{f \cdot X_c}{Z_c} + o_x \rightarrow u = f_x \cdot \frac{X_c}{Z_c} + o_x \quad (3)$$

$$v = m_y \cdot \frac{f \cdot Y_c}{Z_c} + o_y \rightarrow v = f_y \cdot \frac{Y_c}{Z_c} + o_y \quad (4)$$

Essas equações descrevem a relação entre as coordenadas tridimensionais de um ponto no espaço e suas coordenadas projetadas na imagem [3].

Figura 3 - Esquemático de projeção da câmera pinhole.



Fonte: autoria própria

O processo de projeção é frequentemente representado por equações matriciais. A matriz extrínseca M_{ext} (equação 5) realiza a conversão das coordenadas do sistema de referência global para o sistema de coordenadas da câmera, por meio de matrizes de rotação e translação [4]. Esses parâmetros são conhecidos como parâmetros extrínsecos, que descrevem a posição e orientação da câmera em relação ao mundo.

A matriz intrínseca K (ou matriz M_{int}) (equação 6), por outro lado, define como as coordenadas tridimensionais de um ponto são projetadas no sensor da câmera [5]. Ela inclui parâmetros como a distância focal f , o ponto principal (centro da imagem) O_x e O_y , e possíveis distorções da lente. Esses são chamados de parâmetros intrínsecos da câmera e determinam diretamente como a imagem é formada no sensor.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & O_x & 0 \\ 0 & f_y & O_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (5)$$

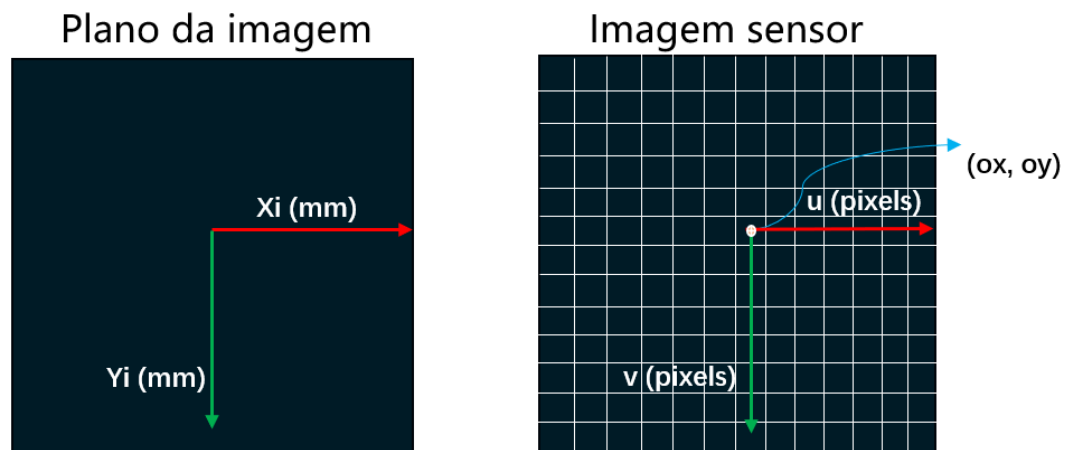
$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (6)$$

As equações (5) e (6) mostram como as coordenadas do mundo tridimensional x_w, y_w, z_w são convertidas em coordenadas da câmera x_c, y_c, z_c por meio da matriz extrínseca M_{ext} , e como essas coordenadas são posteriormente projetadas no plano da imagem em pixels u e v , (equação 7) através da matriz intrínseca K .

$$\mathbf{pw} = \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \xrightarrow{M_{ext}} \mathbf{pc} = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \xrightarrow{K} \mathbf{ps} = \begin{bmatrix} u \\ v \end{bmatrix} \quad (7)$$

Após a projeção tridimensional no plano da imagem, o ponto projetado \mathbf{x} tem suas coordenadas medidas em milímetros, como x_i e y_i . No entanto, ao ser convertido para o sensor da câmera, as coordenadas u e v , são medidas em pixels. Essa conversão é importante, pois o sensor da câmera utiliza pixels para representar a imagem digital capturada.

Figura 4 - Coordenadas no sensor da câmera.



Fonte: autoria própria

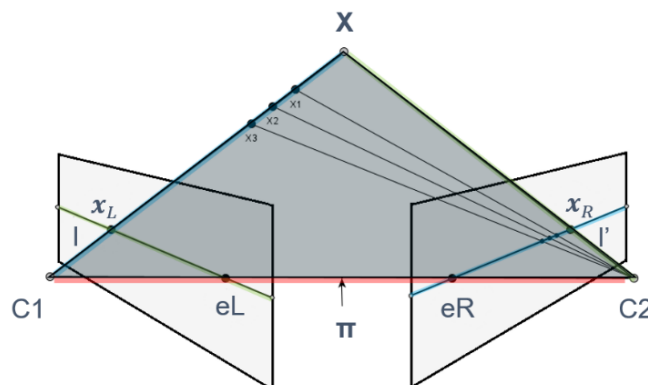
A Figura 4 acima ilustra como a projeção no plano da imagem é medida em milímetros (x_i , y_i), e como, no sensor da câmera, essas coordenadas são convertidas para pixels (u , v) com a adição dos centros óticos O_x e O_y . Essa diferença reflete a transformação de coordenadas do espaço físico tridimensional para a imagem digitalizada no sensor.

O ponto \mathbf{x} no espaço tridimensional possui coordenadas globais $\mathbf{pw} = [x_w, y_w, z_w]$. Utilizando a matriz extrínseca M_{ext} , essas coordenadas são transformadas em coordenadas no sistema de referência da câmera $\mathbf{pc} = [x_c, y_c, z_c]$ [7]. Posteriormente, aplicamos a matriz intrínseca K , que projeta essas coordenadas no plano da imagem, resultando em $\mathbf{ps} = [u, v]$. A distância focal f é a distância entre o pinhole da câmera e o plano da imagem. O ponto \mathbf{ps} é a projeção de \mathbf{x} na imagem, representando a conversão da cena tridimensional em uma imagem bidimensional capturada pela câmera.

2.4.2 Triangulação (geometria epipolar)

A geometria epipolar permite determinar a posição de um ponto do cenário projetado em duas imagens capturadas a partir de pontos de vista diferentes. Considere um ponto \mathbf{x} no espaço tridimensional que é projetado em duas imagens por câmeras posicionadas em locais distintos. O ponto \mathbf{x} , juntamente com os centros ópticos das câmeras em duas posições diferentes ($C1$ e $C2$), define um plano no espaço denominado plano epipolar (π). A interseção desse plano com os planos de imagem das câmeras gera duas linhas, conhecidas como linhas epipolares (l e l'). As projeções do ponto \mathbf{x} nas imagens são representadas como \mathbf{x}_L e \mathbf{x}_R , e os epípólos e_L e e_R são os pontos de interseção das linhas epipolares com as imagens, conectando os centros ópticos das câmeras. A Figura 5 ilustra esse conceito com o ponto \mathbf{x} projetado em duas imagens distintas, mostrando o plano epipolar (π) e as linhas epipolares (l e l'), juntamente com os epípólos e_L e e_R .

Figura 5 - Linhas epipolares.



Fonte: autoria própria

A partir da perspectiva da câmera na posição $C2$, é possível observar o ponto \mathbf{x} como \mathbf{x}_1 , \mathbf{x}_2 e \mathbf{x}_3 . Contudo, na perspectiva da câmera $C1$, esses pontos se confundem em \mathbf{x}_L , devido à posição relativa das câmeras e do ponto no espaço tridimensional. Esse fenômeno está diretamente relacionado à geometria epipolar, que restringe onde o ponto correspondente a \mathbf{x}_L pode ser encontrado na outra imagem.

A geometria epipolar é descrita por duas matrizes: a matriz fundamental e a matriz essencial. A matriz fundamental \mathbf{F} é uma matriz de 3×3 que descreve a relação geométrica entre duas imagens, sem exigir a calibração das câmeras. Ela impõe a condição $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$, garantindo que o ponto \mathbf{x} em uma imagem corresponda a uma linha epipolar na outra imagem.

A matriz fundamental tem sete graus de liberdade e pode ser obtida por correspondências de pontos entre as duas vistas [26].

Já a matriz essencial (\mathbf{E}) é derivada da matriz fundamental e é aplicável quando as câmeras estão calibradas. Isso significa que os parâmetros intrínsecos das câmeras, como a distância focal e o centro ótico, são conhecidos. A relação entre a matriz essencial e a fundamental é dada por $\mathbf{E} = \mathbf{K}'^T \mathbf{F} \mathbf{K} = 0$, onde \mathbf{K} e \mathbf{K}' são as matrizes de calibração das câmeras. A matriz essencial possui cinco graus de liberdade, e além da condição de posto dois, ela possui a restrição de que dois de seus valores singulares são iguais e o terceiro é zero. Assim, ela captura a relação entre a rotação e a translação entre as câmeras[26].

A calibração das câmeras envolve a determinação desses parâmetros intrínsecos, que permitem a normalização das coordenadas das imagens. A matriz essencial, nesse caso, permite calcular a pose relativa entre as câmeras (rotação e translação), possibilitando uma análise mais precisa da geometria entre as vistas [26].

A matriz essencial é uma forma específica da matriz fundamental e é utilizada quando se conhece a calibração das câmeras. Enquanto a matriz fundamental descreve a geometria epipolar de forma geral, aplicável a qualquer par de imagens, a matriz essencial é restrita ao caso de câmeras calibradas. Essas matrizes, em conjunto, são essenciais para a reconstrução tridimensional de cenas e para a determinação da relação geométrica entre múltiplas vistas [26].

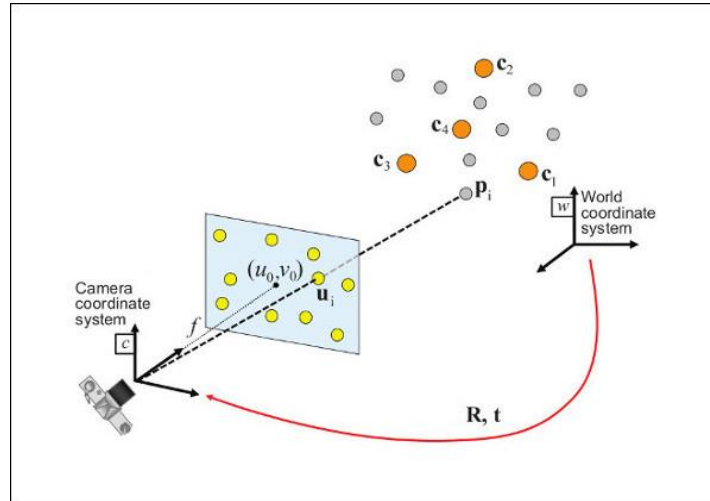
2.4.3. Estimação de posição e orientação

O problema que o *Perspective-n-Point* (PnP) se propõe a resolver refere-se à tarefa de estimar a pose (posição e orientação) de um objeto tridimensional a partir das projeções de alguns pontos conhecidos desse objeto em uma imagem bidimensional. A solução envolve encontrar uma transformação (rotação e translação) que alinhe esses pontos tridimensionais com suas correspondências bidimensionais na imagem.

Na Figura 6, pode se observar a relação entre os pontos no espaço tridimensional e sua projeção no plano da câmera. Os pontos c_1, c_2, c_3, c_4 , que representam pontos no sistema de coordenadas do mundo, são projetados para o plano da imagem bidimensional da câmera, resultando em coordenadas da imagem u_i . A câmera captura a projeção de um ponto 3D P_i em sua correspondente coordenada 2D (u_0, u_0) . A transformação entre os sistemas de coordenadas

da câmera e do mundo é dada por uma matriz de rotação \mathbf{R} e um vetor de translação \mathbf{t} , que constituem a pose da câmera em relação ao mundo.

Figura 6 - Perspective-n-Point (PnP).



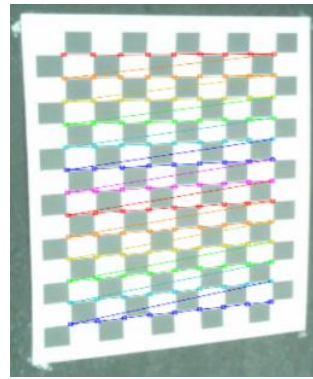
Fonte - https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html

O algoritmo *PnP* resolve essa relação projetiva, mas pode ser sensível a ruídos nos pontos de correspondência, levando a soluções imprecisas. Para mitigar esse problema, o algoritmo *RANSAC* (*Random Sample Consensus*) é frequentemente utilizado em conjunto com o *PnP*. O *solvePnP**Ransac*, por exemplo, é uma implementação que ignora os pontos de correspondência errôneos ou fora do padrão, fornecendo uma solução mais robusta ao estimar a pose da câmera. A combinação de *PnP* com *RANSAC* é utilizada em cenários onde há outliers nas correspondências entre os pontos 3D e 2D.

2.5 Calibração da câmera

Para obter a matriz da câmera, utilizamos a função *calibrateCamera()* da biblioteca OpenCV. Essa função utiliza um tabuleiro de xadrez específico, conhecido como padrão de calibração, para encontrar pontos no mundo real. Na Figura 7, é mostrado um exemplo dos pontos encontrados em diferentes posições definidas pela câmera.

Figura 7 -Pontos encontrados do espaço tridimensional.



Fonte: autoria própria

A função *calibrateCamera()* é responsável por encontrar os parâmetros intrínsecos e extrínsecos da câmera. Ela funciona identificando pontos de um padrão conhecido (como o tabuleiro de xadrez) em várias imagens e utiliza esses pontos para calcular a matriz intrínseca da câmera, os vetores de rotação e translação, os coeficientes de distorção e o erro de reprojeção. A saída inclui diversos parâmetros importantes para a calibração da câmera. Entre eles, destacam-se a matriz da câmera (intrínseca), o vetor de rotação, o vetor de translação, os coeficientes de distorção e o erro de reprojeção.

Alguns parâmetros foram definidos para realizar a calibração, como o tamanho do tabuleiro de xadrez (*chessboard_size*), o tamanho da janela utilizada para refinar a localização dos cantos (*winSize*) e o tamanho do quadro (*frameSize*).

A matriz da câmera e outros parâmetros de calibração, é possível corrigir distorções nas imagens, realizar reconstrução 3D e obter medidas precisas.

Os parâmetros f_x, f_y, c_x, c_y de uma matriz de câmera (equação 8) são usados para a transformação de coordenadas 3D do espaço da câmera em coordenadas 2D na imagem e f_x e f_y são os fatores de escala nas direções x e y, representando a distância focal da câmera em termos de pixels e relacionados ao campo de visão horizontal e vertical e c_x e c_y indicam as coordenadas do centro óptico da câmera, o ponto onde o eixo óptico intercepta o plano da imagem. A matriz de câmera intrínseca \mathbf{K} , que incorpora esses parâmetros, é usada para transformar coordenadas 3D no sistema de coordenadas da câmera para coordenadas 2D no plano da imagem.

$$\mathbf{A} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (8)$$

2.6 Aquisição das imagens

O procedimento de captura de vídeo ou sequência de imagens foi realizado a partir de diferentes pontos de vista, assegurando que elementos do cenário aparecessem em mais de uma imagem. Com isso, alguns detalhes são registrados em duas ou mais imagens, permitindo o rastreamento deles na etapa seguinte.

Os experimentos envolveram tanto a captura automática quanto a manual. Nos experimentos com captura automática, utilizando o braço robótico, foi implementado um código para capturar as imagens. Para isso, foi utilizada a biblioteca OpenCV, responsável por abrir a fonte de vídeo, ler os frames e exibi-los em uma janela. A cada 30 frames capturados pela câmera, que opera a 15 frames por segundo, uma imagem é salva, enquanto as outras 29 são descartadas. Esse processo de discretização é realizado de forma paralela, evitando interrupções na captura contínua de novas imagens e garantindo que apenas imagens com intervalos significativos de movimento do braço robótico sejam armazenadas.

A velocidade do braço robótico foi reduzida de sua capacidade máxima de 2 m/s para 0,4 m/s, com o objetivo de minimizar os efeitos de distorção do *rolling shutter* e do *motion blur*, que podem ocorrer devido ao movimento rápido da câmera. O *rolling shutter* causa distorções em câmeras onde diferentes partes da imagem são capturadas em momentos ligeiramente diferentes, enquanto o *motion blur* resulta em borrões devido ao movimento relativo entre a câmera e os objetos na cena. Ao limitar a captura a um frame a cada 30, assegura-se a obtenção de imagens mais nítidas e com menos distorções, otimizando a eficiência computacional e a qualidade das imagens utilizadas. Esse procedimento também inclui um mecanismo de controle para o número total de imagens a serem capturadas, garantindo cobertura espacial adequada e evitando redundância de dados desnecessária.

Nos experimentos realizados em ambientes fechados, as imagens foram capturadas em metade de uma sala, cobrindo três paredes dispostas em forma de U. A câmera foi conduzida manualmente ao longo dessas três paredes, mantendo uma sobreposição de aproximadamente 70 a 80% entre as fotos [67]. A captura começou no canto esquerdo da sala, movendo-se lateralmente ao longo da primeira parede, e seguiu para as outras duas paredes em uma trajetória contínua, sempre com a câmera sendo mantida a distâncias variadas para coletar diferentes ângulos. Em determinados pontos, detalhes específicos, como texturas das paredes e objetos presentes, foram capturados mais de perto, conforme recomendações anteriores sobre a importância de registrar essas feições para o correto alinhamento das imagens durante a

reconstrução [68]. A iluminação utilizada era difusa, minimizando sombras e variações bruscas de luz, conforme recomendado para evitar distorções no processo de reconstrução [69].

Nos experimentos realizados em ambiente aberto, as imagens foram capturadas em uma planta química. A câmera foi conduzida manualmente ao redor das principais estruturas, como tanques e tubulações, garantindo uma sobreposição de 60 a 70% entre as fotos [70]. O percurso começou ao redor do tanque central, movendo-se ao longo das principais estruturas da planta, com capturas em diferentes alturas, variando entre 1 e 2 metros. Foram feitas capturas circulares ao redor das estruturas, garantindo que cada lado fosse devidamente registrado em ângulos variados [71]. A iluminação natural foi aproveitada no início da manhã, minimizando sombras intensas e reflexos indesejados. O uso de drones foi considerado para cobrir áreas mais altas e inacessíveis, seguindo as recomendações da literatura para ambientes complexos como plantas industriais [22]. Esses procedimentos ajudaram a cobrir as grandes estruturas da planta e a coletar as imagens necessárias para a reconstrução do ambiente.

2.7 Identificação, codificação e rastreamento de detalhes

A identificação de detalhes em imagens envolve a descoberta de diferentes elementos, realizada pelo processamento de imagens para identificar regiões com variações significativas no valor dos pixels. Um método conhecido para isso é a Transformada de Característica Invariante à Escala, com o algoritmo SIFT (*Scale Invariant Feature Transform*), desenvolvido por David Lowe em 2004. O SIFT é um detector e descritor de detalhes locais que extrai pontos de interesse de objetos em imagens, mesmo sob diferentes perspectivas, permitindo correspondências em alterações de escala, rotação ou iluminação [43]. Para definir esses detalhes, o algoritmo gera vetores que descrevem numericamente as propriedades dos pontos identificados. Em seguida, a extração da matriz de gradiente representa as variações no nível de intensidade dos pixels, o que possibilita identificar bordas e detalhes. Posteriormente, descritores são calculados para todos os pontos coletados, e um algoritmo de correspondência utiliza esses descritores para encontrar pontos correspondentes entre duas imagens, filtrando as correspondências falsas [6,11,12].

Outro algoritmo utilizado é o SURF (*Speeded Up Robust Features*), desenvolvido como uma versão mais rápida e robusta do SIFT, mantendo a invariância à escala e rotação. O SURF converte a imagem em uma imagem integral e calcula a matriz Hessiana, o que reduz o tempo de processamento. A detecção de pontos de interesse usa o operador Haar, que, com base em uma transformada wavelet, decompõe a imagem em diferentes escalas e orientações. O SURF

identifica os pontos de interesse onde há uma diferença máxima entre regiões, utilizando informações de intensidade local e gradientes ao redor do ponto. Os pontos redundantes são removidos para otimizar o processo [14,15].

O algoritmo ORB (*Oriented FAST and Rotated BRIEF*) foi desenvolvido a partir dos algoritmos FAST e BRIEF. O FAST realiza a busca por pontos característicos ao comparar intensidades de pixels adjacentes, enquanto o BRIEF extrai detalhes representando-as como sequências binárias. O ORB rotaciona a janela de comparação em diferentes ângulos, o que aumenta a robustez do descritor em relação à rotação da imagem [16,17].

O detector de detalhes Shi-Tomasi é outro método que se baseia no cálculo da matriz de autovalores dos gradientes da imagem. Os pontos de interesse são aqueles onde o menor autovalor é elevado, indicando uma variação significativa no gradiente [18,19].

Foi desenvolvido um detector de detalhes próprio, inspirado no SIFT, mas com melhorias para atender às necessidades específicas do projeto. O detector utiliza uma abordagem semelhante para identificar pontos de interesse e gerar descritores, mantendo a robustez frente a variações de escala, rotação e iluminação, o que permite a correspondência dos detalhes em diferentes imagens.

2.8 Refinamento das coordenadas dos detalhes e exclusão outliers

No processo de correspondência de detalhes, cujo objetivo é encontrar pontos semelhantes entre imagens de um ponto específico no espaço, podem ocorrer erros nessa associação. Em outras palavras, pontos identificados como semelhantes podem, na verdade, não representar o mesmo ponto no espaço. Para lidar com isso, é necessário utilizar algoritmos que filtrem esses pontos incorretos, conhecidos como *outliers*, garantindo uma correspondência mais precisa e confiável.

2.8.1 RANSAC

Um dos algoritmos que tenta resolver esse tipo de falha é o Random Sample Consensus (RANSAC). Trata-se de um algoritmo iterativo que ajusta um modelo a partir de pontos selecionados aleatoriamente dentro de um conjunto de dados. Isso pode ser feito, por exemplo, por meio de regressão linear, ajuste polinomial, transformações geométricas, entre outros. O RANSAC, portanto, inclui os pontos que estão de acordo com o modelo e remove os pontos que não estão de acordo com esse modelo.

As falsas correspondências entre detalhes são encontradas se houver pontos suficientes no conjunto que se encaixem no modelo. Caso o modelo não seja satisfatório, novos pontos são sorteados. Por exemplo, suponhamos que o algoritmo seja configurado com um número máximo de 36 iterações, visando assegurar uma probabilidade de 90% de identificar um subconjunto de pontos sem outliers. Se, após atingir esse limite de iterações, o algoritmo não conseguir obter um modelo válido, ele encerra com falha [10][11].

Existem vários fatores que podem resultar em falha no processo de validação de um modelo. Um deles é quando um determinado par de imagens é capturado com uma diferença muito pequena entre seus pontos de vista. Outro fator é quando a segunda imagem do par é muito diferente da primeira imagem. Para evitar esses problemas, é necessário levar em consideração algumas restrições fotogramétricas que permitem obter imagens com distâncias consideradas ideais [9].

Entre essas restrições, destacam-se:

- Relação base/altura (B/H) ideal: Deve ser mantida dentro de um intervalo específico (geralmente de 0,3 a 0,5) [47, 51].
- Sobreposição de imagens adequada: Com uma sobreposição longitudinal de 60% a 80% e lateral de 20% a 40% [47].
- Angulação controlada das imagens: Para evitar distorções [48].
- Qualidade das imagens: Devem ser altas, livres de distorções significativas e com iluminação uniforme [49].
- Geometria de captura consistente: A distância focal da câmera deve ser mantida constante [47, 50].

Seguir essas restrições ajuda a minimizar erros e a garantir que as imagens obtidas sejam adequadas para a validação e o processamento fotogramétrico [11].

2.8.2 Menor mediana de quadrados

LMEDS (*Least-Median of Squares*) é um método estatístico robusto utilizado para estimar modelos matemáticos a partir de dados ruidosos e com *outliers*. O método consiste em escolher um conjunto mínimo de pontos de correspondência e, em seguida, calcular o modelo que melhor se ajusta a esse conjunto de pontos. O erro residual de cada ponto em relação ao

modelo é então calculado. Em vez de usar a soma dos quadrados dos erros como critério de ajuste, como no método dos mínimos quadrados, o LMEDS usa a mediana dos erros residuais. O uso da mediana, em vez da média, torna o LMEDS mais robusto a outliers, ou seja, pontos que estão distantes da distribuição dos demais pontos.

Um ponto é considerado um outlier quando ele se desvia significativamente das tendências ou padrões observados na maioria dos outros pontos no conjunto de dados. Existem várias maneiras de identificar outliers, como o critério do desvio padrão, onde um ponto está a mais de 2 ou 3 desvios padrão da média, e o método dos quartis e intervalo interquartil (IQR), onde pontos que estão a uma certa distância além do primeiro ou terceiro quartil são considerados outliers.

Os pontos que têm um erro residual maior que um limite definido são considerados outliers e são removidos da amostra. Esse processo é repetido várias vezes, escolhendo diferentes conjuntos mínimos de pontos de correspondência aleatoriamente e selecionando o modelo que tem a menor mediana de erros residuais [25,26].

O LMEDS é capaz de lidar com uma porcentagem significativa de outliers e, portanto, é amplamente utilizado em aplicações onde há muitos pontos ruidosos ou com outliers. É importante notar que o método não garante a estimativa do modelo exato, mas sim uma boa aproximação que é robusta a outliers. A maioria dos métodos tradicionais de ajuste de modelos, como o método dos mínimos quadrados (OLS - *Ordinary Least Squares*), também não garante a estimativa do modelo exato na presença de outliers. Esses métodos assumem que os dados seguem uma distribuição normal e que os erros são aleatórios e independentes. Quando há outliers, o OLS pode ser significativamente influenciado por esses pontos atípicos, resultando em uma estimativa de modelo que pode ser distorcida ou incorreta.

Portanto, a diferença principal entre métodos como o LMEDS e o OLS é a robustez frente a outliers. O LMEDS é projetado para minimizar a influência dos outliers, proporcionando uma aproximação do modelo que é mais robusta em cenários com dados ruidosos ou contaminados por outliers. Assim, nenhum método garante a estimativa exata do modelo em todos os casos, mas métodos robustos como o LMEDS são mais eficazes em fornecer boas aproximações na presença de dados problemáticos [25,26].

2.9 Exibição e exportação da nuvem de pontos

O processo de exibição e exportação da nuvem de pontos gerada durante os experimentos. Para realizar essa tarefa, foi desenvolvida uma função específica em Python, denominada `viz_3d_v3`, utilizando as bibliotecas Open3D e Mayavi.

O código da função tem como objetivo gerar uma nuvem de pontos tridimensional com base nas coordenadas 3D e nas cores correspondentes para cada ponto. Para isso, a função começa inicializando uma instância da classe `PointCloud` da biblioteca Open3D, que será utilizada para armazenar os pontos e suas respectivas cores.

A cada iteração, os pontos 3D são adicionados ao objeto de nuvem de pontos, e suas cores são normalizadas para o intervalo entre 0 e 1, permitindo a visualização correta no espaço tridimensional. Além disso, a função utiliza a biblioteca Mayavi para renderizar a nuvem de pontos em tempo real, com o método `points3d` desenhando cada ponto individualmente no espaço.

Após a visualização, a nuvem de pontos é exportada no formato PCD (*Point Cloud Data*), amplamente utilizado para armazenar dados de nuvens de pontos. O arquivo gerado é salvo no diretório especificado como argumento da função. Ao final do processo, a visualização da nuvem de pontos é exibida e o arquivo PCD é armazenado, garantindo que os dados possam ser posteriormente utilizados em análises e comparações adicionais.

O processo de exibição e exportação da nuvem de pontos é um passo essencial para a verificação da qualidade dos dados adquiridos e a sua compatibilidade com outros sistemas de processamento de nuvens de pontos.

Este capítulo apresentou a estrutura geral do sistema de processamento de imagens desenvolvido para capturar e processar informações tridimensionais a partir de imagens bidimensionais. Foram descritos os equipamentos utilizados, os cenários de validação e as ferramentas de software empregadas no desenvolvimento. Os procedimentos para projeção, determinação da pose da câmera, calibração, aquisição de imagens e processamento dos pontos de interesse foram organizados em etapas claras, abordando desde a movimentação das câmeras até a geração e exportação da nuvem de pontos. O uso dos laboratórios do Instituto Senai de Inovação permitiu a realização dos experimentos com os recursos adequados para a execução do trabalho.

3. TRABALHOS CORRELATOS

Nos últimos anos, tem havido um progresso significativo no desenvolvimento de abordagens para resolver o problema de construção de nuvens de pontos de cenas e objetos. Diversas técnicas computacionais têm sido aplicadas nesse contexto, incluindo programação estocástica, visão computacional, redes neurais, entre outras.

A metodologia *Structure from Motion* (SfM) pressupõe a existência de movimentos da câmera ou da cena, enquanto a Reconstrução Estéreo não requer essa verificação de movimento. Ambas as abordagens possuem aplicações e vantagens específicas, dependendo do contexto e dos requisitos do problema [1].

Um dos trabalhos pesquisados [2] apresenta uma visão abrangente de uma pilha de código aberto para a reconstrução tridimensional baseada em movimento (SfM). O objetivo principal do artigo é fornecer um conjunto de ferramentas e algoritmos eficientes para a reconstrução 3D. A ênfase na abordagem de código aberto permite acesso livre e contribui para o desenvolvimento contínuo da pilha de código. Ao final, são apresentados estudos de caso que demonstram a eficácia em diferentes cenários com dados reais.

Outro estudo [27] visa desenvolver um algoritmo recursivo que estima o movimento da câmera e a estrutura da cena, combinando as estimativas de dois quadros por meio de técnicas de aproximação estocástica. Embora seja possível realizar a reconstrução 3D com apenas duas imagens, é difícil quantificar o erro na estimativa do movimento da câmera. Para superar essa limitação, o trabalho propõe adquirir múltiplos pares de imagens e combinar as estimativas.

O artigo [28] apresenta uma revisão aprofundada do método *Bundle Adjustment* (BA). O BA envolve a otimização conjunta da posição da câmera e da estrutura 3D da cena para minimizar o erro de projeção entre imagens e pontos tridimensionais. O estudo discute os principais desafios enfrentados, como ruído, falta de pontos correspondentes, presença de outliers e grandes volumes de dados. Também são abordadas estratégias de paralelização para acelerar o processamento e integrar dados adicionais, como informações inerciais e de calibração [3].

O artigo [29] apresenta um sistema escalável e distribuído para *Bundle Adjustment* em sistemas SLAM, que envolve a otimização simultânea dos parâmetros da câmera e a reconstrução 3D da cena. Para lidar com a complexidade computacional, o estudo propõe o uso

do framework SLAMBench 2.0, que divide o processo em várias etapas, escalando-o para grandes conjuntos de dados.

Outro estudo recente [38] foca na otimização de sistemas de visão computacional que utilizam o processo de *Bundle Adjustment* em ambientes distribuídos. A abordagem propõe dividir o problema em subproblemas menores e distribuí-los em diferentes nós de processamento, otimizando o tempo de execução e o desempenho do sistema.

O artigo [55] apresenta o NeRF, uma abordagem baseada em redes neurais que aprende a reconstruir uma cena 3D a partir de um conjunto de imagens 2D. NeRF representa cenas tridimensionais como campos de radiação contínuos, e, ao ser treinado, pode gerar renderizações foto realistas de novos pontos de vista da cena. Essa técnica é essencial, pois combina a precisão geométrica do SfM com a capacidade de síntese de imagens altamente detalhadas, mesmo em ambientes de baixa textura e condições de iluminação variáveis. O NeRF demonstra uma melhoria substancial na qualidade visual e precisão da reconstrução 3D.

O estudo do artigo [56] explora como técnicas de aprendizado profundo podem ser integradas ao pipeline tradicional de SfM, resultando em melhor desempenho e robustez. O *DeepSfM* utiliza redes neurais para identificar feições e correspondências entre imagens, o que é especialmente útil em cenários onde as técnicas clássicas de detecção de feições falham, como em superfícies de baixa textura ou cenas com iluminação irregular. A abordagem é fundamental para lidar com ambientes complexos, e suas vantagens incluem uma redução significativa no erro acumulado durante a estimativa de movimento.

No artigo [57] GRAF expande a ideia de campos de radiação neurais aplicando redes generativas adversariais (GANs) para criar campos de radiação tridimensionais. Ao gerar imagens conscientes de profundidade, o GRAF permite a síntese de imagens 3D com um nível de detalhe superior e oferece soluções robustas para áreas de oclusão, onde a reconstrução SfM tradicional pode falhar. O uso de GANs é essencial para a melhoria da qualidade da imagem e o preenchimento automático de áreas com informações faltantes.

O artigo [58] combina o *Bundle Adjustment* tradicional com NeRFs para otimizar a estimativa de parâmetros da câmera e da cena simultaneamente. O BARF utiliza a robustez geométrica do BA e a capacidade do NeRF de representar campos de radiação contínuos para gerar reconstruções 3D de alta precisão. Esta abordagem é essencial para garantir a

convergência e a precisão da reconstrução, especialmente em cenas complexas e dinâmicas, onde as técnicas tradicionais podem não ser suficientes.

No trabalho [59] os autores introduzem uma técnica inovadora para representar cenas dinâmicas com volumetria tridimensional usando redes neurais. Em vez de lidar apenas com cenas estáticas, o *Neural Volumes* permite a reconstrução 3D de ambientes dinâmicos, capturando o movimento e a mudança de forma. Esta abordagem é essencial para áreas como realidade aumentada e captura de movimento, onde a representação precisa de cenas em movimento é crucial para a qualidade da experiência 3D.

No trabalho do artigo [60] PixelNeRF aprimora o NeRF original ao permitir reconstruções 3D a partir de apenas uma ou poucas imagens. Esta técnica é fundamental para cenários onde a aquisição de múltiplas imagens não é viável, como em ambientes móveis ou aplicações de vigilância. Utilizando aprendizado profundo, o PixelNeRF consegue gerar reconstruções tridimensionais detalhadas a partir de dados limitados, proporcionando uma solução eficiente e de alta qualidade para problemas onde o acesso a grandes volumes de imagens não é possível [15].

O artigo [61] propôs a integração de dados LiDAR com SfM para melhorar a qualidade da reconstrução 3D em ambientes de baixa textura, como interiores de edifícios. A fusão das duas fontes de dados foi feita através de um modelo híbrido que combinou correspondências 2D com pontos 3D. Esta técnica é fundamental para melhorar a precisão em reconstruções 3D em ambientes onde as técnicas tradicionais de SfM falham devido à ausência de texturas ricas, e demonstrou alta eficiência em cenários complexos [4].

No artigo [62], os autores focaram na eficiência computacional do SfM em grandes conjuntos de dados urbanos. Utilizando uma abordagem distribuída, o artigo demonstra uma redução de 30% no tempo de processamento ao dividir o processamento entre várias máquinas, o que permite a aplicação de SfM em grandes volumes de dados urbanos de maneira eficiente. Esta contribuição é essencial para a escalabilidade das técnicas de reconstrução 3D em grandes cidades.

No trabalho [63] foi investigado o uso de correspondências automáticas em ambientes com baixa iluminação e grande variação de perspectiva. A técnica proposta melhora a precisão da correspondência entre imagens e, por consequência, a reconstrução 3D. Embora não utilize

IA, o artigo propõe avanços importantes na precisão da reconstrução para ambientes adversos, como espaços industriais mal iluminados e com grandes obstáculos.

O estudo [64] focou na melhoria da estimativa de movimento em ambientes aéreos utilizando aprendizado profundo para refinar a correspondência de feições entre imagens. Testes realizados em cenários de mapeamento aéreo demonstraram um desempenho superior em relação às técnicas tradicionais, com maior precisão na estimativa de movimento e redução de erros na reconstrução de grandes áreas geográficas.

No artigo [65] foi abordado a aplicação de SfM em ambientes subaquáticos, onde as distorções causadas pela refração da luz representam um desafio significativo. O artigo propõe um método de calibração de refração que melhora substancialmente a precisão da reconstrução 3D de estruturas submersas, como recifes de corais, superando as limitações das abordagens tradicionais.

No estudo [66] apresenta um pipeline de SfM otimizado para câmeras de baixo custo, aplicando técnicas de redução de ruído e otimização de movimento. O pipeline é fundamental para permitir que dispositivos com hardware limitado, como smartphones, possam ser utilizados em reconstruções 3D com alta precisão, abrindo novas possibilidades para a democratização da tecnologia de modelagem 3D.

Na Tabela 4 apresenta uma comparação entre os trabalhos citados.

Tabela 4 - Tabela Comparativa entre os Trabalhos

Artigo	Ano	Abordagem Principal	Técnica/ Conceito	Aplicação	Desafios Abordados	Resultados
[2]	2020	Pilha de código aberto	SfM	Reconstrução 3D	Acesso aberto a ferramentas	Estudos de caso com dados reais
[27]	2020	Algoritmo recursivo	Aproximação estocástica	Estimativa de movimento	Combinação de múltiplos pares de imagens	Melhora na qualidade da reconstrução
[28]	2021	Bundle Adjustment	Otimização conjunta	AR, mapeamento 3D	Ruído, outliers, paralelização	Uso de regularização e filtragem
[29]	2021	SLAMBench 2.0	Distributed Bundle Adjustment	SLAM	Escalabilidade e distribuição de dados	Eficiência em grandes

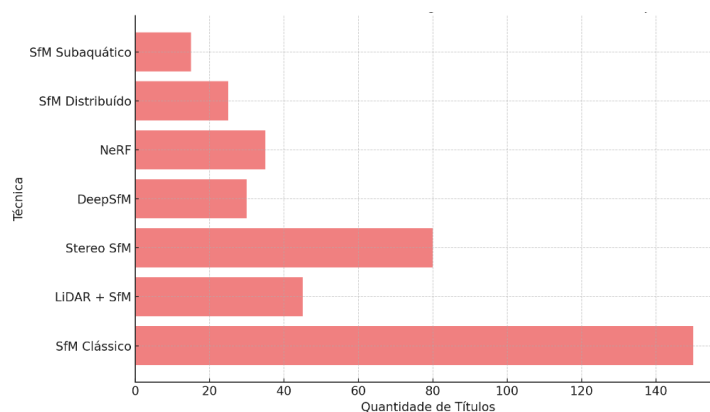
						conjuntos de dados
[38]	2021	Ambientes distribuídos	Distribuição de subproblemas	Visão computacional	Sincronização e comunicação entre nós	Melhora no desempenho e tempo de processamento
[55]	2020	Redes neurais	Neural Radiance Fields	Renderização e reconstrução	Ambientes de baixa textura e iluminação	Alta qualidade em renderizações 3D fotorrealistas
[56]	2021	Aprendizado profundo	Detecção de feições robustas	Reconstrução 3D em baixa textura	Correspondências em cenas complexas	Melhor precisão em cenas dinâmicas e desafiadoras
[57]	2020	GANs	Campos de radiação generativos	Síntese de imagens 3D	Oclusão e preenchimento de áreas faltantes	Melhoria na qualidade de reconstrução
[58]	2021	NeRF + BA	Bundle Adjustment + NeRF	Reconstrução otimizada	Convergência e precisão	Combina otimizações geométricas e neurais
[59]	2019	Redes neurais	Representação volumétrica neural	Cenas dinâmicas	Reconstrução de cenas em movimento	Melhor representação de cenas dinâmicas
[60]	2021	NeRF	Reconstrução com poucas imagens	Cenários limitados em dados	Poucas imagens	Reconstrução a partir de uma única ou poucas imagens
[61]	2021	Híbrido LiDAR + SfM	Fusão de dados	Ambientes interiores	Baixa textura em imagens	Reconstrução mais detalhada em ambientes complexos
[62]	2021	SfM distribuído	Processamento distribuído	Cidades	Grandes datasets urbanos	Redução de 30% no tempo de processamento
[63]	2022	Correspondência robusta	Correspondência de imagens	Baixa iluminação	Variação de perspectiva	Melhora na precisão em

						condições desafiadoras
[64]	2022	Aprendizado profundo	Correspondência de feições	Mapeamento aéreo	Refino de correspondência	Melhor desempenho comparado a técnicas tradicionais
[65]	2020	SfM subaquático	Calibração de refração	Ambientes subaquáticos	Distorção pela luz na água	Reconstrução precisa de ambientes submersos
[66]	2020	Pipeline otimizado	SfM para dispositivos móveis	Dispositivos móveis	Câmeras de baixo custo	Alta precisão com dispositivos acessíveis

Fonte: autoria própria

No gráfico da Figura 8, ilustra as técnicas mais citadas nas pesquisas realizadas relacionadas ao SfM. A técnica de SfM Clássico destaca-se como a mais utilizada, com 150 menções, refletindo sua aplicação em fotogrametria, modelagem para realidade aumentada e mapeamento. A combinação de LiDAR + SfM tem ganhado relevância, especialmente em ambientes com baixa textura ou áreas internas, com 45 menções. Já o Stereo SfM, mencionado em 80 artigos, continua a ser amplamente empregado na reconstrução tridimensional com câmeras estéreo, em especial para mapeamento aéreo.

Figura 8 - Técnicas mais citadas sobre SfM



Fonte: autoria própria

Técnicas baseadas em inteligência artificial, como DeepSfM e NeRF, têm emergido como alternativas inovadoras às abordagens clássicas. DeepSfM foi citado em 30 artigos e tem

se mostrado eficaz na correspondência de imagens em condições desafiadoras, como cenas de baixa textura ou iluminação. NeRF (*Neural Radiance Fields*), com 35 menções, vem se destacando pela sua capacidade de gerar renderizações foto realistas a partir de visualizações esparsas, oferecendo alta qualidade nas reconstruções 3D. O SfM Distribuído aparece com 25 menções, focado em lidar com grandes volumes de dados em cenários como mapeamento urbano ou modelagem com drones. O SfM Subaquático, com 15 citações, lida com os desafios da reconstrução em ambientes submersos, sendo aplicado em estudos de ecossistemas marinhos e arqueologia subaquática.

A análise abrange artigos publicados entre 2015 e 2022, período que demonstra um crescimento significativo no uso de aprendizado profundo e técnicas híbridas de reconstrução. A partir de 2020, técnicas como DeepSfM e NeRF ganharam destaque, refletindo a crescente integração de inteligência artificial nas soluções de visão computacional, enquanto as abordagens tradicionais como SfM Clássico e Stereo SfM continuam sendo amplamente utilizadas em diversas áreas de aplicação.

4. ALGORITMO SfM PROPOSTO

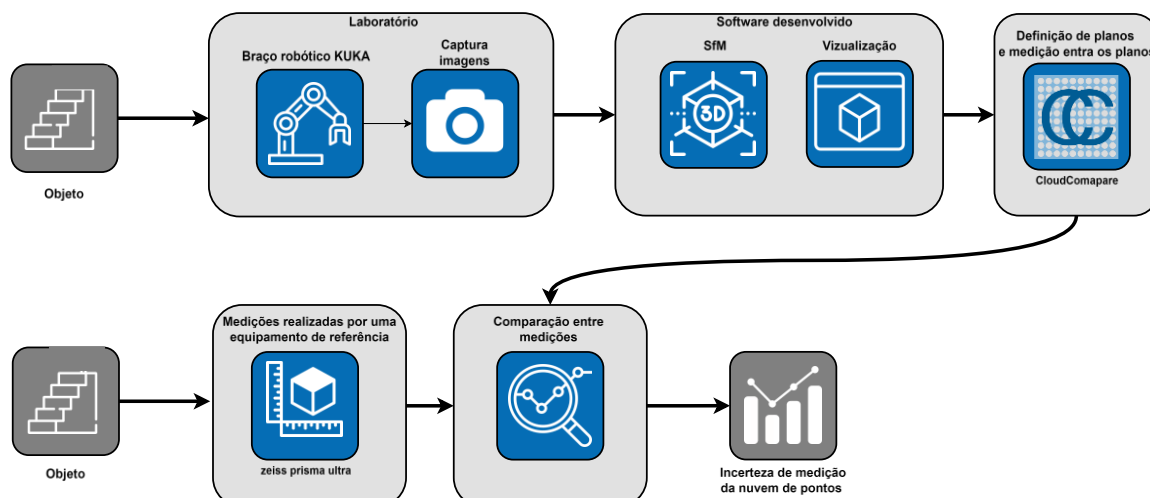
Neste capítulo, detalharemos a solução proposta e a realização dos ensaios no laboratório de metrologia do Instituto Senai de Inovação (ISI), abordando as tecnologias utilizadas, equipamentos empregados, e o fluxo geral do sistema. A solução é voltada para a geração de nuvens de pontos de objetos, utilizando métodos passivos de captura de imagem e técnicas de visão computacional.

Para a realização dos ensaios e obtenção dos dados necessários, foram utilizados diversos equipamentos, destacando-se a Câmera DFK AFU050-L34 Color, responsável pela captura das imagens para a geração da nuvem de pontos, o Braço Robótico KUKA, que posiciona a câmera em diferentes ângulos para obter múltiplas visões do objeto, e a Máquina de Medição por Coordenadas ZEISS Prismo Ultra, utilizada para gerar uma nuvem de pontos de alta precisão da peça, servindo como referência para a modelagem do erro e análise da incerteza. Os ensaios foram realizados em um laboratório de metrologia, onde parâmetros como temperatura, vibração e umidade foram rigorosamente controlados para garantir a precisão e repetibilidade dos resultados.

A Figura 9 apresenta a visão geral da solução proposta. O processo começa com uma câmera que é manuseada por um braço robótico KUKA no laboratório para a captura de imagens. Essas imagens são então processadas por um software desenvolvido que utiliza técnicas de SfM para criar nuvens de pontos dos objetos. Posteriormente, essas nuvens de pontos são usadas no CloudCompare para a definição de planos e medições entre os planos.

Simultaneamente, o mesmo objeto passa por medições realizadas por um equipamento de referência, especificamente o Zeiss Prismo Ultra. Os resultados dessas medições são comparados com os obtidos pelo software desenvolvido. A comparação entre as medições permite determinar a incerteza da medição da nuvem de pontos, proporcionando uma análise das técnicas de captura e processamento de imagens utilizadas.

Figura 9 - Visão geral da solução proposta.



Fonte: autoria própria

O sistema desenvolvido é composto por várias etapas principais, descritas a seguir. Primeiro, realiza-se a calibração da câmera para extrair os seus parâmetros internos e externos. Em seguida, passa-se para a aquisição de imagens, onde as fotos da peça são capturadas e armazenadas de diferentes ângulos.

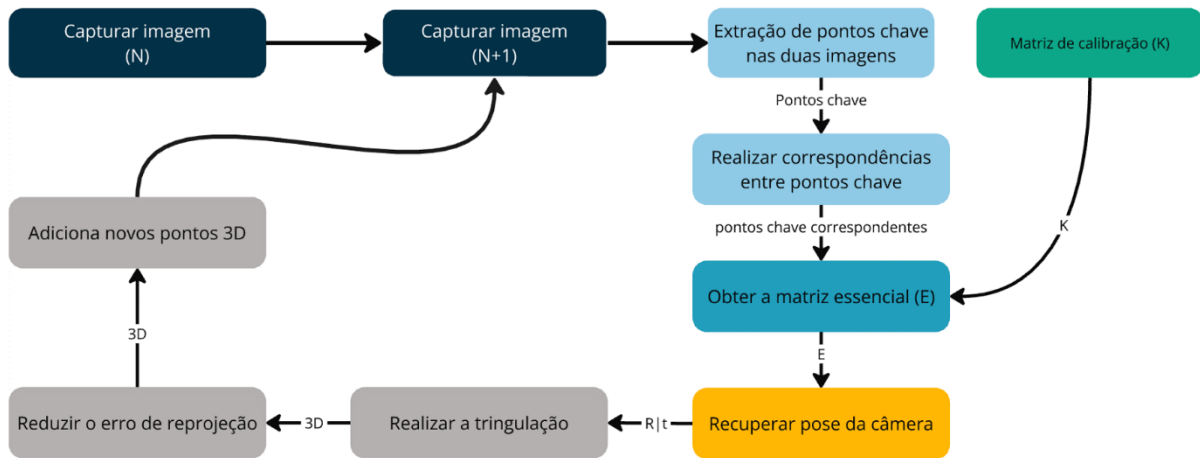
Na etapa de correspondência de descritores, utiliza-se o método SIFT para detectar pontos-chave, extrair detalhes e fazer a correspondência entre os descritores. Posteriormente, na fase de recuperação da pose, a pose inicial da câmera é estimada a partir da matriz essencial e, então, atualizada com base em imagens subsequentes.

A construção da nuvem de pontos é realizada por meio da técnica de triangulação linear, que gera coordenadas tridimensionais dos pontos da cena. Para melhorar a qualidade da estrutura 3D estimada, aplica-se o ajuste do erro de reprojeção, utilizando o ajuste de pacotes.

Finalmente, na etapa de recuperação de escala, são utilizados marcadores ArUco, que são padrões visuais bidimensionais em formato de quadrados com um padrão único de bits, permitindo a sua identificação individual. Esses marcadores são empregados para recuperar informações de escala na geração da nuvem de pontos.

A Figura 10 apresenta o fluxograma do processo de construção da nuvem de pontos utilizando a técnica de SfM

Figura 10 - Diagrama para gerar nuvem de pontos por SfM.



Fonte: autoria própria

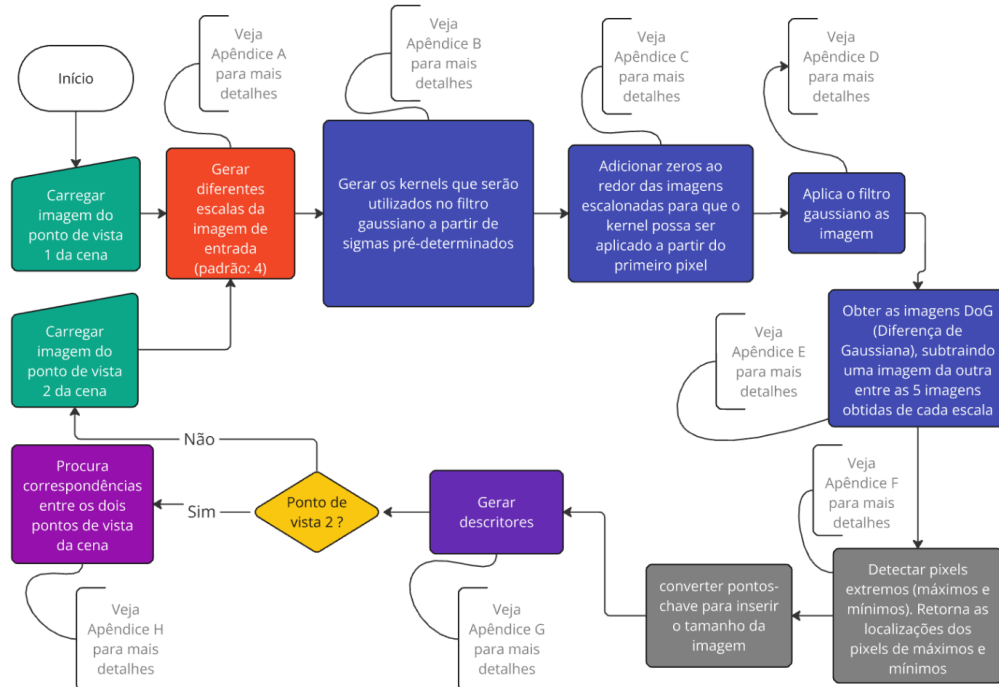
Vale ressaltar que a geração de nuvem de pontos por meio de métodos passivos apresenta vantagens em relação aos métodos ativos em diversas situações. Métodos passivos utilizam luz ambiente ou fontes de luz naturais para capturar imagens e dados, ao contrário dos métodos ativos, que dependem de fontes de luz artificiais, como lasers ou projeções estruturadas. Esses métodos passivos são especialmente eficazes ao lidar com objetos brilhantes, em longas distâncias ou em cenários com movimento. Objetos brilhantes referem-se a superfícies reflexivas que podem dificultar a captura de dados devido ao reflexo da luz.

4.1 Captura de imagens para geração da nuvem de pontos

A captura das imagens é realizada pela câmera acoplada no braço robótico. Para automatizar o processo, foi desenvolvido um código que permite visualizar e salvar os frames do fluxo contínuo de vídeo (explicado na sessão materiais e métodos (2.6)), para o processamento posterior.

4.2 Correspondência entre descritores

Figura 11 - Fluxograma do algoritmo para detecção de pontos-chave e geração de descritores.

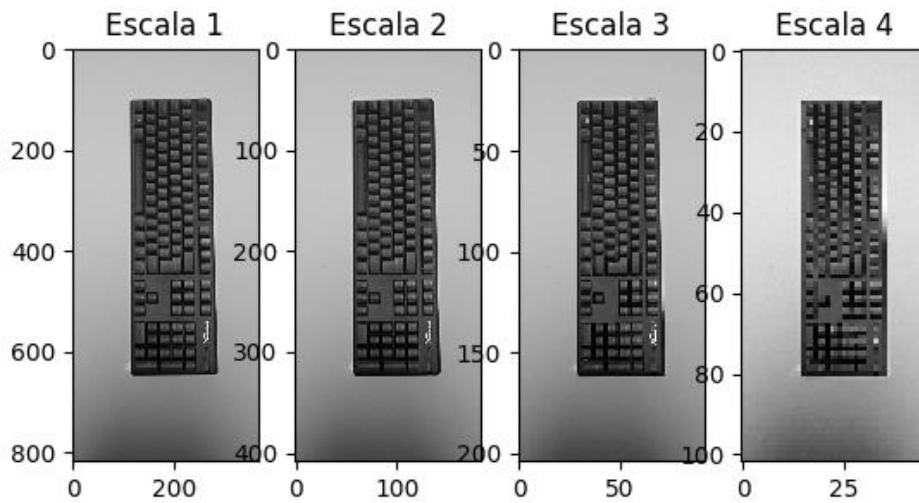


Fonte: autoria própria

Para realizar a correspondência de detalhes de diferentes visões de um objeto ou cena, foi desenvolvido um algoritmo (Figura 11) baseado no método utilizado no algoritmo SIFT. A implementação divide-se em três etapas: detecção de pontos-chave, extração de detalhes e correspondência entre descritores. A seguir, será descrito cada uma dessas etapas.

No processo de detecção de pontos-chave de uma imagem, é inicialmente necessário gerar várias escalas dessa imagem. Isso é feito para garantir a detecção de pontos-chave invariantes à escala nas etapas seguintes. Na Figura 12, é mostrado o resultado do ensaio, apresentando as quatro escalas geradas do objeto (816px x 368px, 408px x 184px, 204px x 92px e 102px x 46px). O pseudocódigo para realizar o redimensionamento da imagem pode ser encontrado no Apêndice A.

Figura 12 - Imagens com diferentes resoluções.



Fonte: autoria própria

Com as imagens em diferentes escalas definidas, o próximo passo é gerar os conjuntos de kernels que serão utilizados para a aplicação dos filtros gaussianos em cada imagem. O filtro Gaussiano suaviza as imagens, reduzindo o ruído e detalhes finos, o que torna os detalhes detectados mais robustos e menos suscetíveis a pequenas variações.

Além disso, o filtro Gaussiano é importante para a criação da pirâmide de escalas, onde a imagem é suavizada em diferentes níveis, permitindo a detecção de detalhes em múltiplas resoluções. Esta suavização facilita a identificação de pontos chave que são estáveis em diversas escalas e invariantes à rotação, através da diferença entre imagens suavizadas consecutivas (Diferença de Gaussiano, ou DoG).

Ao reduzir detalhes irrelevantes e o ruído, o filtro Gaussiano ajuda o algoritmo a focar nos elementos estruturais significativos da imagem, importante para a correspondência de detalhes e reconhecimento de objetos. A equação 9 foi utilizada para calcular os valores dos kernels.

$$(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-x^2+y^2}{2\sigma^2}\right) \quad (9)$$

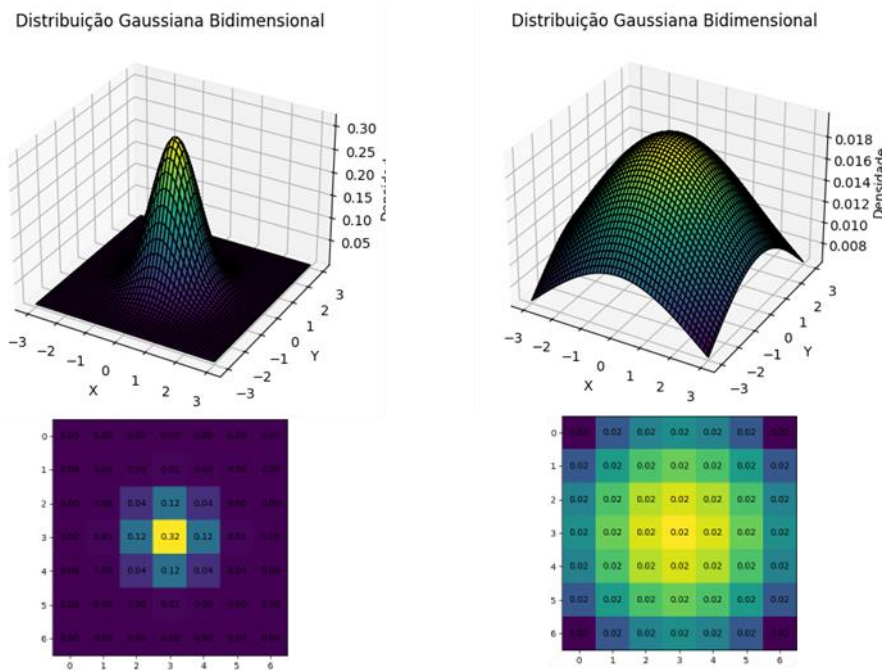
Nestas equações, destaca-se as seguintes partes:

- $\frac{1}{2\pi\sigma^2}$ normaliza a distribuição gaussiana bidimensional. Isso garante que a soma total da probabilidade seja igual a 1, ou seja, a probabilidade de todas as possíveis combinações de x e y na distribuição.

- $\exp\left(\frac{-x^2+y^2}{2\sigma^2}\right)$ descreve a forma da distribuição gaussiana. Os termos x e y representam as coordenadas espaciais, o desvio padrão σ que controla a largura da distribuição. A equação calcula os valores dos pixels para um kernel gaussiano em diferentes intervalos gerando valores ponderados para cada posição no kernel gaussiano.

A Figura 13 apresenta duas distribuições gaussianas bidimensionais, ambas definidas no intervalo de -3 a 3 para as variáveis x e y, utilizadas como exemplos dos diferentes kernels. A distribuição à esquerda, com um desvio padrão $\sigma=0.707$, exibe um pico central acentuado, demonstrando como valores de probabilidade são altamente concentrados no centro e decrescem rapidamente nas bordas. Por outro lado, a distribuição à direita, com um desvio padrão maior de $\sigma=2.828$, resulta em uma superfície mais ampla e suave, indicando uma densidade de probabilidade menos concentrada e espalhada de forma mais homogênea. Esses exemplos destacam o impacto significativo que diferentes valores de desvio padrão têm sobre os kernels, afetando diretamente a forma e a dispersão das distribuições gaussianas e, conseqüentemente, a suavização dos dados representados.

Figura 13 - Distribuição gaussiana bidimensional
- $x=-3$ a 3 $y=-3$ a 3 $\sigma=0.707$ e - $x=-3$ a 3 $y=-3$ a 3 $\sigma=2.828$.



Fonte: autoria própria

Para cada imagem em diferentes escalas, serão aplicados cinco filtros gaussianos, resultando em um total de 20 novas imagens. Foram utilizados diferentes valores de σ para cada imagem; valores maiores de σ resultam em um desfoque mais forte, o que ajuda na

detecção de estruturas ou detalhes maiores, enquanto valores menores de σ capturam detalhes mais finos.

Na Tabela 5, os sigmas aplicados a cada imagem estão listados. Para imagens com resoluções maiores, utilizamos sigmas menores para aproveitar a maior quantidade de informações devido ao maior número de pixels.

Tabela 5 - Valores de sigmas utilizados.

Imagem	Resolução da imagem do experimento	Valores de σ
Imagem escala 1	816 x 368 pixels	0.707
		1
		1.414
		2
		2.828
Imagem escala 2	408 x 184 pixels	1.414
		2
		2.828
		4
		5.656
Imagem escala 3	204 x 92 pixels	2.828
		4
		5.657
		8
		11.314
Imagem escala 4	102 x 46 pixels	5.657
		8
		11.314
		16
		22.627

Fonte: autoria própria

Os sigmas foram escolhidos com base nos seguintes critérios: valores maiores de σ produzem um desfoque mais acentuado, útil para detectar estruturas maiores, enquanto valores menores de σ capturam detalhes mais finos. Esses valores aumentam exponencialmente com a redução da resolução da imagem em cada escala subsequente. O objetivo é garantir uma suavização progressiva, capturando detalhes em múltiplas resoluções. Por exemplo, na primeira escala, os valores de σ variam de 0.707 a 2.828, proporcionando uma suavização gradual na mesma imagem. Essa variação de σ gera cinco novas imagens suavizadas. À medida

que a resolução da imagem diminui, os valores de σ aumentam proporcionalmente, contribuindo para a consistência na detecção de detalhes invariantes à escala e à rotação

O número de imagens suavizadas por escala de imagem, geralmente entre 3 e 5, é escolhido para equilibrar a precisão na detecção de detalhes e a eficiência computacional. Esse intervalo permite capturar pontos-chave consistentes em diferentes níveis de suavização, garantindo robustez contra ruídos e variações. A utilização de 3 imagens suavizadas normalmente está associada à necessidade de reduzir o tempo de processamento e o uso de memória, o que torna essa abordagem adequada para sistemas com recursos limitados ou para aplicações em que a detecção de detalhes não requer um nível extremamente alto de precisão. Por outro lado, optar por 5 imagens suavizadas é mais comum em cenários que exigem maior robustez na detecção de detalhes, especialmente em ambientes com ruídos ou variações significativas de escala e rotação. Embora o aumento no número de imagens suavizadas possa melhorar a precisão ao permitir uma análise mais detalhada e capturar pontos-chave de forma mais consistente, ele também resulta em maior uso de recursos computacionais. O pseudocódigo para gerar os kernels gaussianos pode ser encontrado no Apêndice B.

Os kernels gerados foram definidos como uma matriz de 7×7 , o que exige um ajuste nas imagens para possibilitar a aplicação do filtro gaussiano nos pixels nas extremidades da imagem. Na Figura 14, é apresentado um exemplo de um kernel 7×7 aplicado a toda a imagem, inclusive nas extremidades. Para isso, é necessário adicionar um número de bordas de pixels, com valor zero, ao redor da imagem para que seja possível aplicar o kernel nos pixels das extremidades da imagem. Foram realizados testes com vários tamanhos de matrizes, e o kernel 7×7 foi o que apresentou o resultado, proporcionando um equilíbrio entre suavização e preservação de detalhes da imagem. Esse tamanho de kernel é suficientemente para realizar uma suavização significativa, ajudando a reduzir o ruído e as variações de intensidade. Ao mesmo tempo, não é tão grande a ponto de desfocar ou perder detalhes importantes da imagem.

Tomando o pixel de valor 3 como exemplo. A borda adicionada permite que o filtro gaussiano seja aplicado corretamente, mesmo nas extremidades, garantindo que todos os pixels da imagem original sejam processados de maneira uniforme.

Figura 14 - Aplicando o kernel nos pixels das extremidades da imagem.

Kernel 7 x 7							Imagem original				
0.00	0.00	0.00	0.00	0.00	0.00	0.00	3	32	222	236	100
0.00	0.00	0.00	0.00	0.00	0.00	0.00	164	25	36	25	25
0.00	0.00	0.04	0.12	0.04	0.00	0.00	46	82	49	35	52
0.00	0.00	0.12	0.32	0.12	0.00	0.00	144	7	56	92	45
0.00	0.00	0.04	0.12	0.04	0.00	0.00	32	56	6	4	85
0.00	0.00	0.00	0.00	0.00	0.00	0.00	3	2	65	12	45
0.00	0.00	0.00	0.00	0.00	0.00	0.00					

Fonte: autoria própria

A imagem resultante, com a adição dos zeros, é mostrada na Figura 15. O pseudocódigo para ajustar as imagens com escalas diferentes pode ser encontrado no Apêndice C.

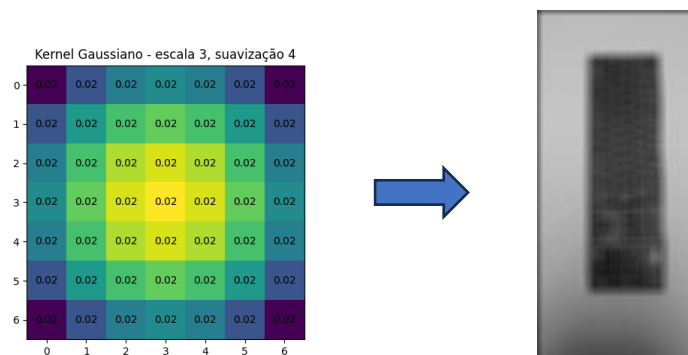
Figura 15 - Ajuste realizado na imagem para aplicação do kernel.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	3	32	222	236	100	0	0	0	0
0	0	0	164	25	36	25	25	0	0	0	0
0	0	0	46	82	49	35	52	0	0	0	0
0	0	0	144	7	56	92	45	0	0	0	0
0	0	0	32	56	6	4	85	0	0	0	0
0	0	0	3	2	65	12	45	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Fonte: autoria própria

Com as imagens ajustadas em diferentes escalas, os filtros gaussianos definidos são aplicados. Na Figura 16, é mostrado um exemplo onde o kernel com $\sigma=8$ é aplicado na terceira escala da imagem de entrada. O resultado é uma imagem suavizada. O pseudocódigo para aplicar os filtros gaussianos nas imagens com diferentes escalas pode ser encontrado no Apêndice D.

Figura 16 - Imagem suavizada.

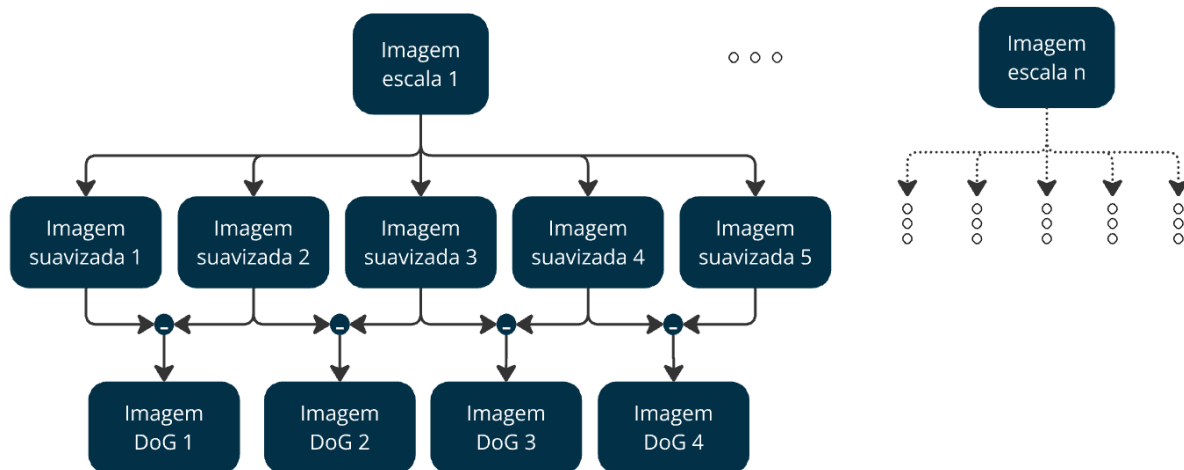


Fonte: autoria própria

A partir de quatro resoluções diferentes da mesma imagem original, foram aplicados cinco filtros gaussianos distintos a cada resolução, gerando cinco versões suavizadas para cada

uma delas. Assim, de cada imagem inicial (em cada resolução), obtemos cinco imagens suavizadas, resultando em um total de 20 imagens suavizadas (4 resoluções x 5 filtros gaussianos = 20 imagens suavizadas). Cada um dos filtros gaussianos foi aplicado a todas as imagens nas diferentes resoluções. Na Figura 17 é apresentado como fica a organização das imagens DoG.

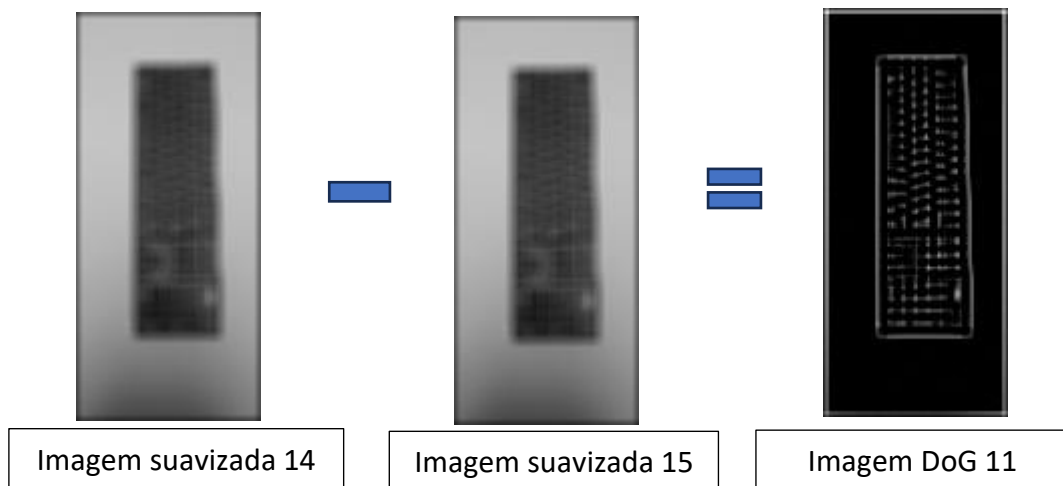
Figura 17 – Organização das imagens DoG.



Fonte: autoria própria

Após a geração dessas 20 imagens suavizadas, o próximo passo é extrair as imagens DoG (*Difference of Gaussians* - Diferença de Gaussianas). As imagens DoG são obtidas subtraindo-se, pixel a pixel, uma imagem suavizada de outra imagem suavizada adjacente (dentro da mesma resolução). Dessa forma, a partir das 20 imagens suavizadas, geramos 16 imagens DoG. Na Figura 18, por exemplo, é ilustrada a subtração da imagem suavizada 14 pela imagem suavizada 15, resultando na imagem DoG 11. O pseudocódigo para gerar as imagens DoG pode ser encontrado no Apêndice E.

Figura 18 - Imagem DoG.

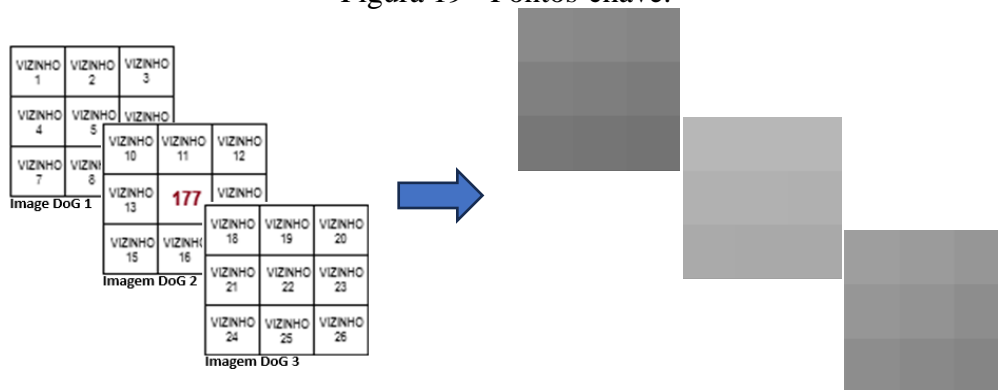


Fonte: autoria própria

Com as imagens DoG, podem ser detectados os pontos-chave. Para isso, é realizada uma busca pixel por pixel. Para que um determinado pixel seja definido como um ponto-chave, ele deve ser maior ou menor em intensidade (valor do pixel) em relação a todos os seus vizinhos, ou seja, deve se destacar em relação aos pixels ao redor do pixel que está em análise. Além disso, é feita uma comparação, pixel a pixel, com a imagem DoG anterior e a imagem DoG posterior à sua, verificando se esses pixels também são maiores ou menores. A "imagem DoG anterior" e "posterior" são definidas com base na escala de σ , que controla o grau de suavização aplicado. A imagem DoG anterior refere-se àquela gerada com um valor de σ ligeiramente menor, enquanto a posterior refere-se à gerada com um valor de σ maior. Portanto, se o pixel que está sendo avaliado for maior ou menor que seus 26 vizinhos (considerando tanto os vizinhos na mesma escala quanto os das escalas anterior e posterior), ele é considerado um ponto-chave e, posteriormente, será gerado um descritor da região ao redor dos pontos-chave.

Na Figura 19, é ilustrada a sobreposição das imagens DoG para avaliação do pixel com valor 177 com seus 26 vizinhos. O pseudocódigo para encontrar os pontos-chave pode ser encontrado no Apêndice F.

Figura 19 - Pontos-chave.



Fonte: autoria própria

Para cada ponto-chave, é gerado um descritor de sua região. Os descritores são formados computando as magnitudes e orientações dos gradientes na região ao redor da localização do ponto-chave. O tamanho da região é definido com base na escala do ponto-chave, ou seja, varia de acordo com a escala em que o ponto-chave foi detectado na pirâmide de escalas. Em geral, o tamanho da região é proporcional à escala do ponto-chave multiplicada por um fator constante, normalmente em torno de 3 vezes o tamanho da vizinhança local em torno do ponto-chave. O fator 3 é definido empiricamente para garantir uma captura adequada da vizinhança ao redor do ponto-chave, equilibrando detalhe e robustez nas descrições de detalhes. A fórmula básica para o tamanho da região está definida na equação 10.

$$T = k \times \text{escala do ponto} - \text{chave} \quad (10)$$

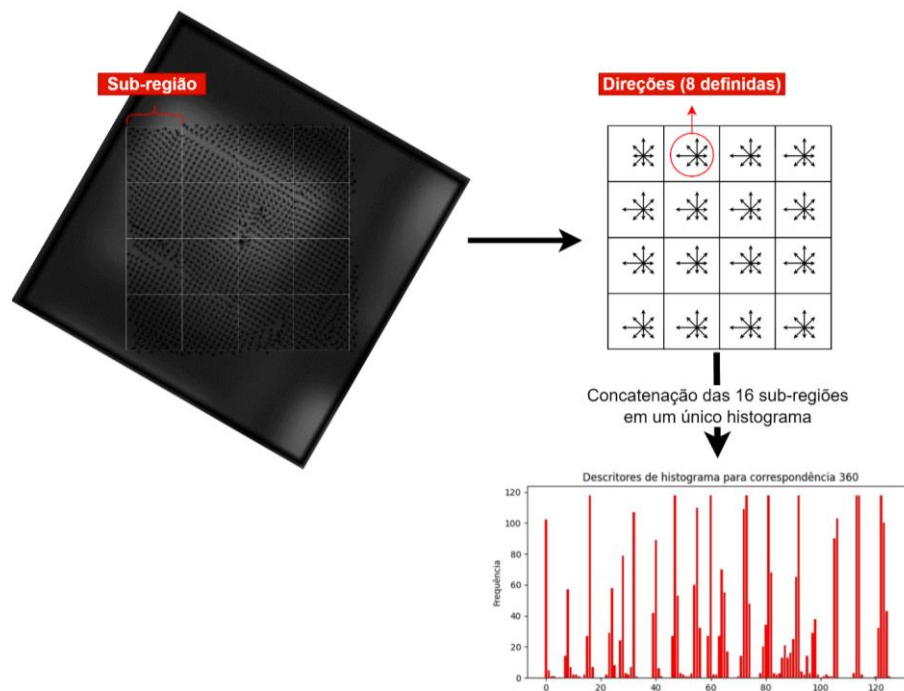
Onde k é um fator de ajuste, comumente definido como 3. Nessa região, são definidas 16 sub-regiões de amostragem, e, para cada sub-região, são pré-definidas 8 direções com base nas magnitudes dos pixels. Essas direções pré-definidas serão utilizadas como acumuladores das magnitudes dos gradientes.

Para cada pixel dentro da sub-região, a magnitude $m(x, y)$ do gradiente é acumulada no bin³ correspondente à orientação $\theta(x, y)$, criando um histograma de gradientes. Ao final, cada uma das 16 sub-regiões contribui com 8 direções, resultando em um histograma por sub-região. Todos os 16 histogramas são então concatenados, formando um descritor de 128 índices (16 sub-regiões x 8 direções).

³ Um bin é uma subdivisão usada para acumular dados dentro de intervalos, como direções de gradientes em um histograma.

Na Figura 20, é apresentado um exemplo de um descritor com seus vetores de direção demarcados e suas 16 sub-regiões. O pseudocódigo para gerar os descritores pode ser encontrado no Apêndice G.

Figura 20 - Definição as 16 sub-regiões com suas 8 direções pré-definidas.



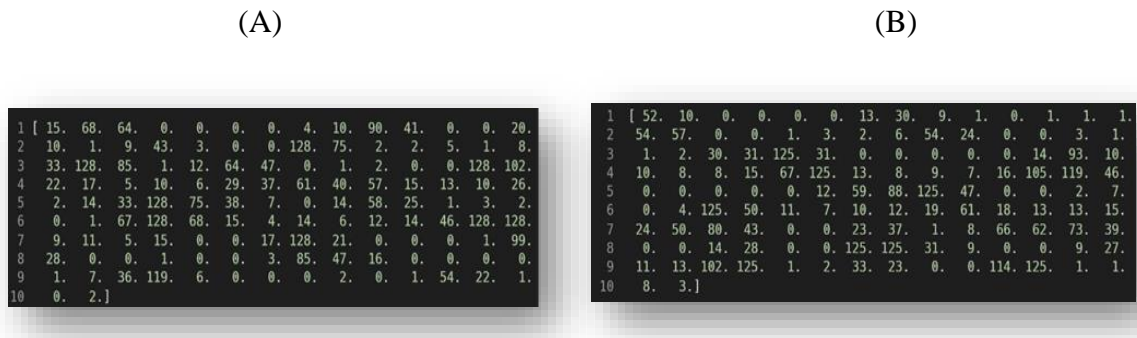
Fonte: autoria própria

Portanto, ao final desta etapa, haverá diversos descritores para cada ponto-chave, que serão utilizados para realizar a correspondência entre imagens. Na etapa de correspondência, é necessário agora encontrar os pares de descritores similares entre as imagens. Para isso, foi utilizada a distância euclidiana (equação 11) para medir essa similaridade.

$$d(v1, v2) = \sqrt{\sum_{i=1}^n (v1i - v2i)^2} \quad (11)$$

Para exemplificar, os descritores apresentados na Figura 21 serão utilizados para os cálculos.

Figura 21 - Imagem (A) vetor resultante com os 16 histogramas concatenados (8 direções).



Fonte: autoria própria

Ao aplicar o cálculo da distância euclidiana, o valor resultante será aproximadamente 93.98, indicando que os dois descritores são similares.

$$d(v_1, v_2) = \sqrt{(15 - 52)^2 + (68 - 10)^2 + (64 - 0)^2 + (0 - 0)^2 + \dots + (2 - 8)^2 + (0 - 3)^2}$$

$$d(v_1, v_2) \approx 93.98$$

O cálculo da distância euclidiana é então aplicado ao conjunto de descritores da imagem do ponto de vista 1 e ao conjunto de descritores da imagem do ponto de vista 2, permitindo determinar os pares de pontos-chave que têm a menor distância entre si.

Nem todos os pares determinados podem ser considerados correspondências corretas por vários motivos, que vão desde a confusão entre o fundo da imagem e o objeto até a não detecção de pontos.

Para resolver esse problema, foi aplicada uma medida comparativa baseada na relação entre a menor e a segunda menor distância dos pontos-chave, de modo a descartar pares inadequados e obter boas correspondências. Segundo David G. Lowe [43], as correspondências corretas precisam ter o vizinho mais próximo significativamente mais próximo do que a correspondência incorreta mais próxima para garantir confiabilidade. Portanto, uma correspondência correta deve apresentar uma distinção clara entre a menor distância (vizinho mais próximo) e a segunda menor distância (segundo vizinho mais próximo). A equação 12 descreve essa relação.

$$razão = \frac{\text{menor distância}}{\text{segunda menor distância}} \quad (12)$$

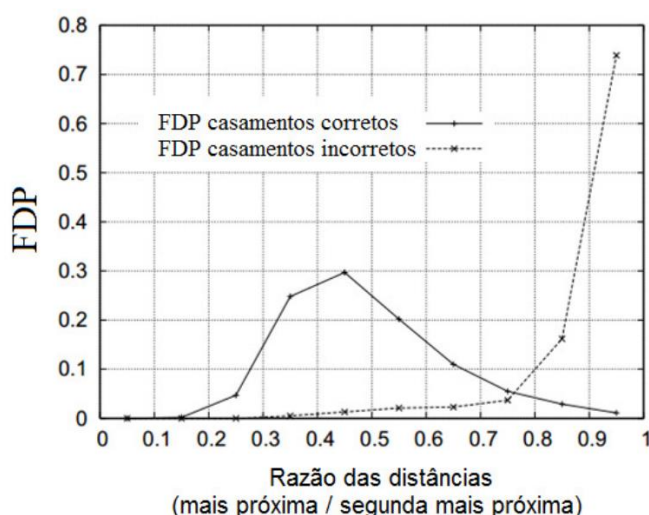
Essa razão é comparada a um limiar predefinido, geralmente em torno de 0.7 ou 0.8. Se a razão for menor que o limiar, a correspondência é considerada válida; caso contrário, ela é

descartada como falsa. Dessa forma, o processo de correspondência pode descartar pares de pontos-chave inadequados e melhorar a precisão dos resultados.

O gráfico da Figura 22⁴, mostra que, a partir de um determinado valor da razão das distâncias, as correspondências corretas diminuem enquanto as incorretas aumentam exponencialmente. Portanto, ao definir um limiar baixo, reduz-se a quantidade de outliers e correspondências errôneas. No entanto, isso também diminui o número de correspondências corretas potencialmente válidas, evidenciando um trade-off entre a redução de falsos positivos e a perda de verdadeiros positivos.

Para otimizar os resultados, é necessário encontrar um equilíbrio no ajuste do limiar. Esse equilíbrio deve ser capaz de minimizar os erros sem comprometer a precisão das correspondências.

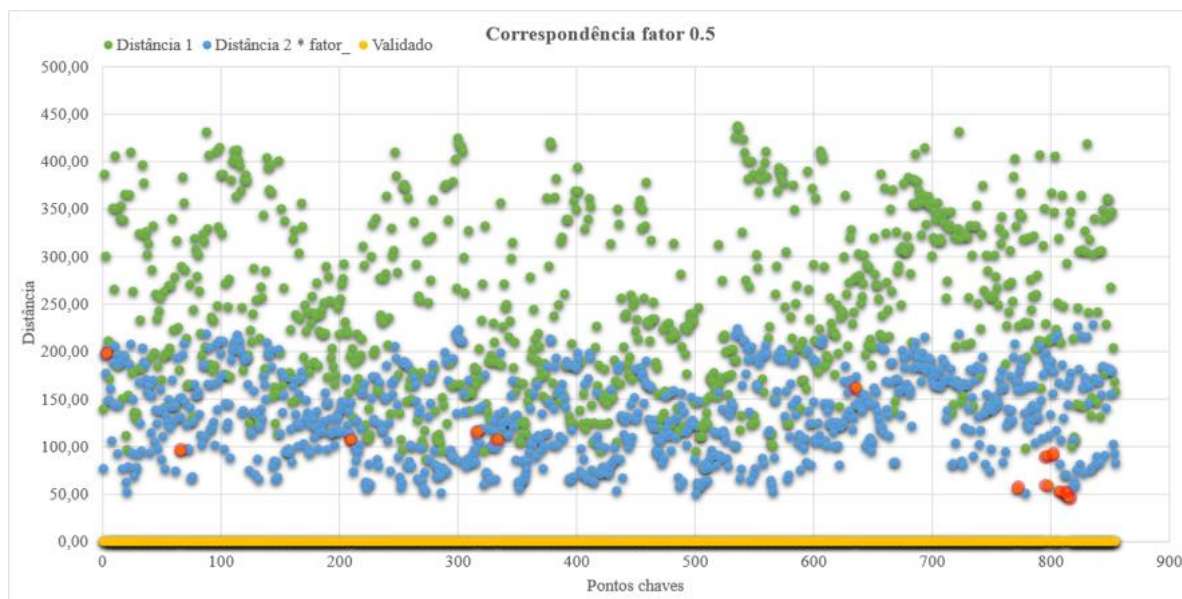
Figura 22 - Função densidade de probabilidade típica para correspondência de pontos SIFT (Lowe, 2004).



Considerando nosso experimento com a imagem do teclado, encontramos um equilíbrio entre boas correspondências e poucos outliers utilizando um limiar de 0,5. O gráfico da Figura 23 apresenta o resultado para o limiar de 0,5. Os pontos verdes representam a menor distância entre os pares de pontos-chave, enquanto os pontos azuis representam a segunda menor distância encontrada vezes o limiar. Os pontos amarelos são as correspondências consideradas válidas. O pseudocódigo para gerar as correspondências pode ser encontrado no Apêndice H.

⁴ Foram utilizados 40 mil pontos chave neste experimento de David G. Lowe.

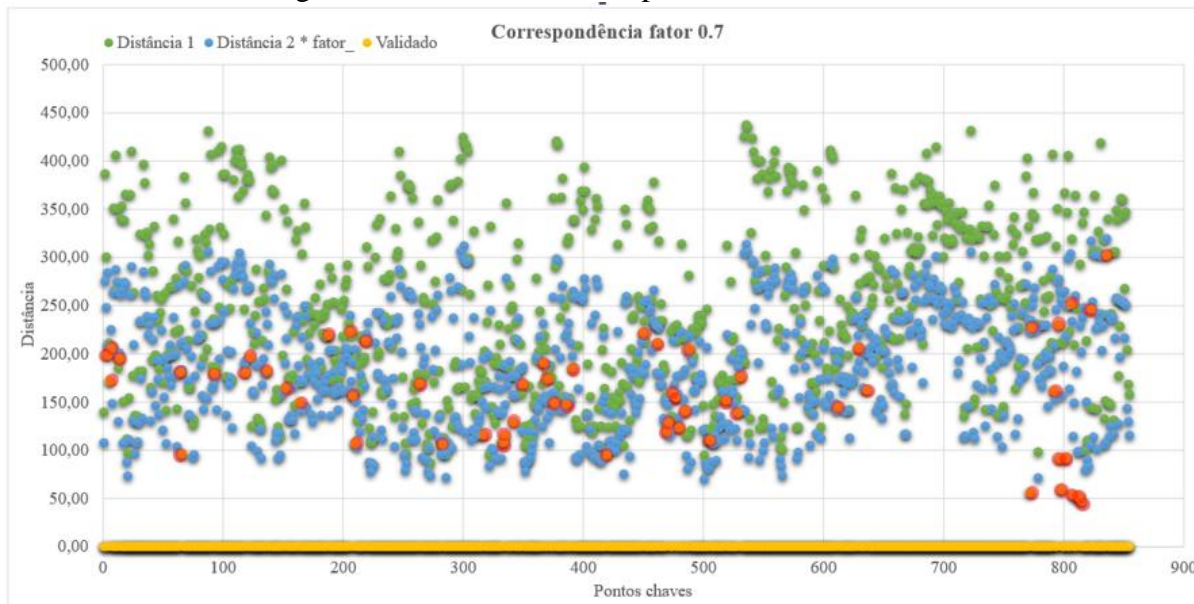
Figura 23 - Gráfico de correspondências – limiar 0.5.



Fonte: autoria própria

No gráfico da Figura 24, para fins de comparação, o limiar foi elevado para 0,7. O número de correspondências aumenta significativamente, mas o número de outliers também é elevado, conforme pode ser visto nas Tabela 6 e Tabela 7.

Figura 24 - Gráfico de correspondências – limiar 0.7

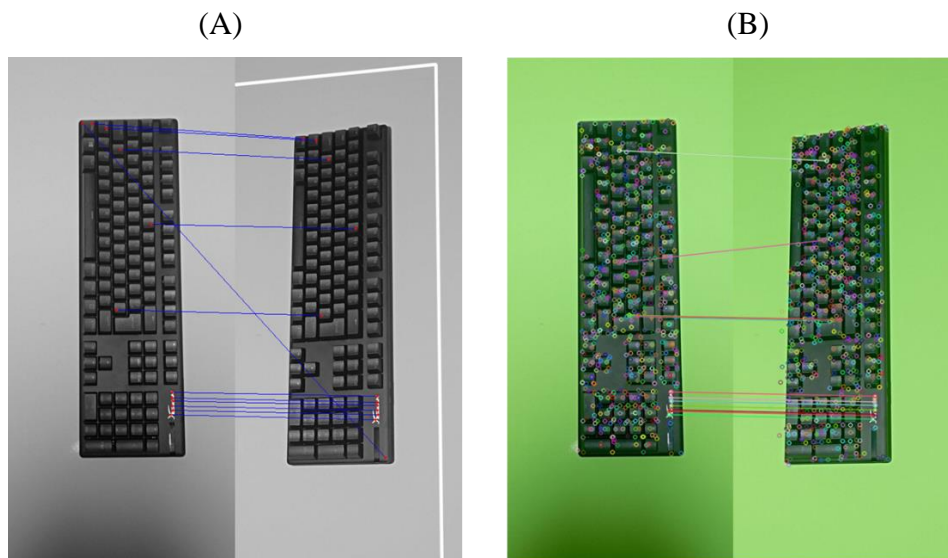


Fonte: autoria própria

Foi realizada uma comparação entre o algoritmo de detecção de pontos-chave desenvolvido e a função SIFT do OpenCV. Embora os pontos-chave encontrados por ambas as implementações não sejam exatamente os mesmos, eles são próximos. Na Figura 25,

apresenta-se o resultado do algoritmo desenvolvido e do OpenCV, juntamente com suas correspondências.

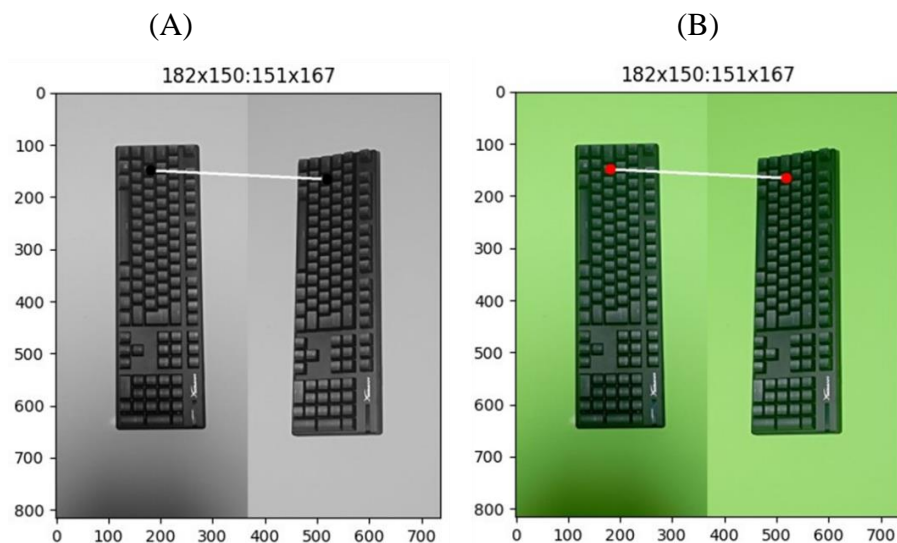
Figura 25 - Pontos-chave e suas correspondências
(Imagem (A) algoritmo desenvolvido e imagem (B) SIFT OpenCV).



Fonte: autoria própria

Na Figura 26, é apresentada uma comparação das correspondências feitas entre dois descritores obtidos pelo algoritmo desenvolvido e pelo OpenCV. Na Tabela 7, é apresentado o número de correspondências encontradas e os outliers. Foram escolhidas correspondências cujos pontos-chave estão em localizações próximas em ambas as implementações, para que seja possível comparar os resultados encontrados nas duas implementações.

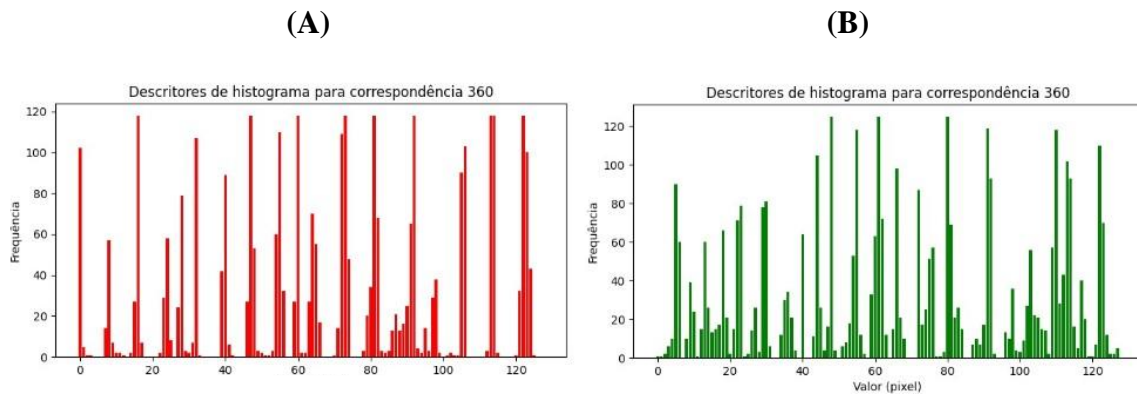
Figura 26 - Correspondências em ambas as implementações
(Imagem (A) algoritmo desenvolvido e imagem (B) SIFT OpenCV).



Fonte: autoria própria

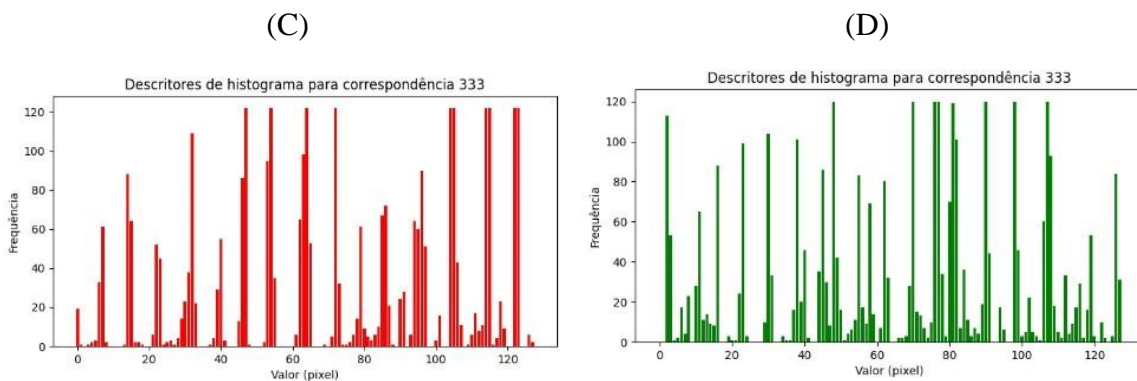
Na Figura 27 e Figura 28, são apresentados os histogramas de ambas as implementações. Os dois histogramas correspondem à mesma região e são considerados como descritores correspondentes, ainda que sejam ligeiramente diferentes.

Figura 27 - Imagem (A) descritor ponto de vista 1 e imagem (B) descritor ponto de vista 2 (algoritmo desenvolvido).



Fonte: autoria própria

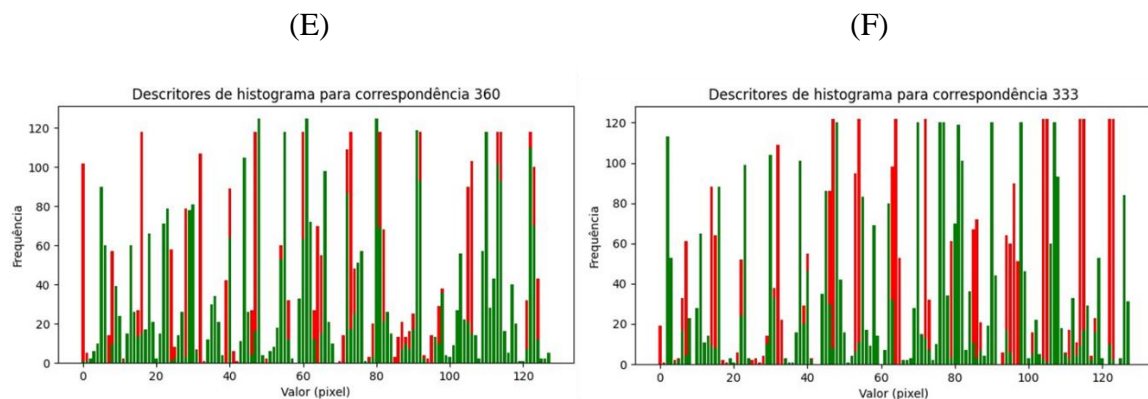
Figura 28 - Imagem (C) descritor ponto de vista 1 e imagem (D) descritor ponto de vista 2 (SIFT OpenCV).



Fonte: autoria própria

Na Figura 29, é apresentada a sobreposição dos descritores considerados correspondentes, apesar de não serem idênticos. Tanto no algoritmo desenvolvido quanto no SIFT do OpenCV, esses descritores são considerados correspondências válidas porque, no cálculo da distância euclidiana entre eles, os valores ficaram abaixo do limiar especificado.

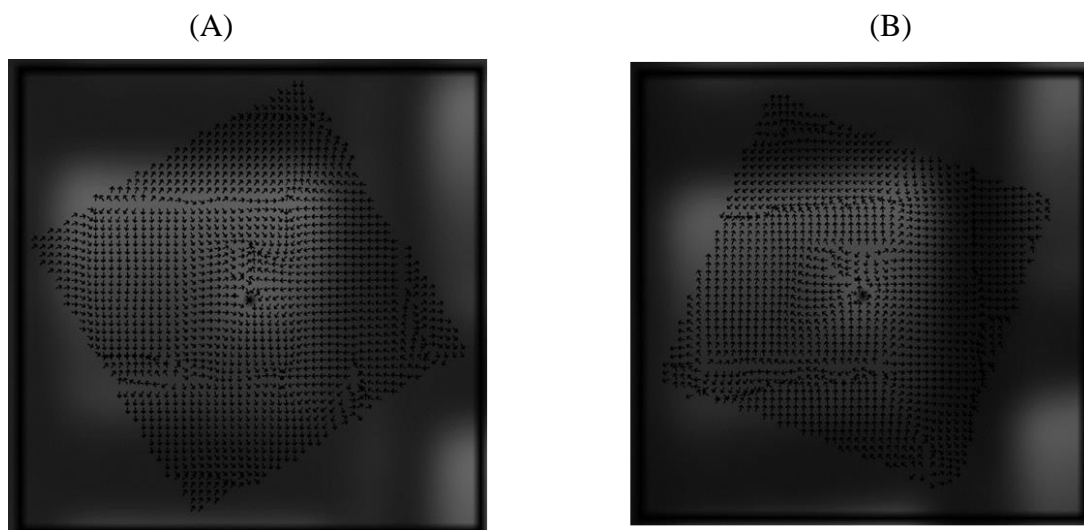
Figura 29 - Imagem (E) sobreposição de descritores do algoritmo desenvolvido (Figura 27), imagem (F) referente ao SIFT do OpenCV (Figura 28).



Fonte: autoria própria

Na Figura 30, são apresentados os vetores de gradientes do algoritmo desenvolvido. Além disso, também são mostrados os valores de magnitude de ambos os descritores.

Figura 30 - Imagem (A) 58 x 58 pixels e imagem (B) com 56 x 56 pixels.



Fonte: autoria própria

A análise apresentada nas Tabela 6 e Tabela 7 compara o desempenho do algoritmo desenvolvido e do SIFT do OpenCV em diferentes resoluções e limiares de correspondência. Na Tabela 6, as imagens de alta resolução (1840 x 4080 pixels) foram avaliadas com limiares de 0,5 e 0,7. Observou-se um aumento no número de correspondências corretas com o limiar de 0,7, porém acompanhado de um aumento proporcional nos outliers. De forma consistente, mesmo com um número elevado de pontos-chave detectados, como 2507 ou 3193, o número de correspondências válidas permanece relativamente baixo. Isso pode ser explicado pelo rigor do limiar de correspondência, que impõe critérios mais estritos, e pelas distorções geométricas ou mudanças de perspectiva entre as imagens, dificultando o alinhamento dos pontos. A

sensibilidade do algoritmo a variações de iluminação, escala e rotação também afeta o processo de correspondência.

Na Tabela 7, o ensaio foi realizado com imagens de menor resolução (368 x 816 pixels), resultando em uma diminuição no número de pontos-chave e correspondências, embora o comportamento em relação aos limiares se mantenha: limiares mais altos geram mais correspondências, mas também mais outliers. Assim, conclui-se que imagens de maior resolução capturam mais detalhes, gerando mais pontos-chave e correspondências, mas também mais outliers. Essa tendência, observada tanto com limiares mais baixos (0,5) quanto mais altos (0,7), demonstra que o problema não está relacionado a um caso isolado, mas sim ao comportamento geral do algoritmo e à sua capacidade de lidar com mudanças nas imagens. Portanto, é necessário equilibrar a escolha do limiar e a resolução da imagem para otimizar os resultados, minimizando outliers e maximizando as correspondências válidas.

Tabela 6 - Comparativo entre algoritmo desenvolvido e SIFT
(resolução 1840 x 4080 pixels).

Algoritmo	Limiar	Resolução (pixel)	Ponto de vista	Pontos chave	Tempo de execução (segundos)	Correspondências	Outliers	Tempo de execução total (segundos)
Trabalho	0,5	1840 x 4080	Ponto de vista 1	2507	267,455	38	2	557,230
			Ponto de vista 2	3193	289,448			
OpenCV (SIFT)	0,5	1840 x 4080	Ponto de vista 1	2376	0,674	42	1	1,353
			Ponto de vista 2	2898	0,655			
Trabalho	0,75	1840 x 4080	Ponto de vista 1	2507	285,092	142	58	603,030
			Ponto de vista 2	3193	317,611			
OpenCV (SIFT)	0,75	1840 x 4080	Ponto de vista 1	2376	0,665	172	79	1,393
			Ponto de vista 2	2898	0,682			

Fonte: autoria própria

Tabela 7 - Comparativo entre algoritmo desenvolvido e SIFT
(resolução 368 x 816 pixels).

Algoritmo	Limiar	Resolução (pixel)	Ponto de vista	Pontos chave	Tempo de execução (segundos)	Correspondências	Outliers	Tempo de execução total (segundos)
Trabalho	0,5	368 x 816	Ponto de vista 1	855	37,690	14	4	74,815
			Ponto de vista 2	852	37,092			
OpenCV (SIFT)	0,5	368 x 816	Ponto de vista 1	931	0,030	16	1	0,073
			Ponto de vista 2	999	0,032			
Trabalho	0,75	368 x 816	Ponto de vista 1	855	37,368	78	33	73,000
			Ponto de vista 2	852	35,600			
OpenCV (SIFT)	0,75	368 x 816	Ponto de vista 1	931	0,034	88	39	0,084
			Ponto de vista 2	999	0,030			

Fonte: autoria própria

4.4 Recuperação da pose

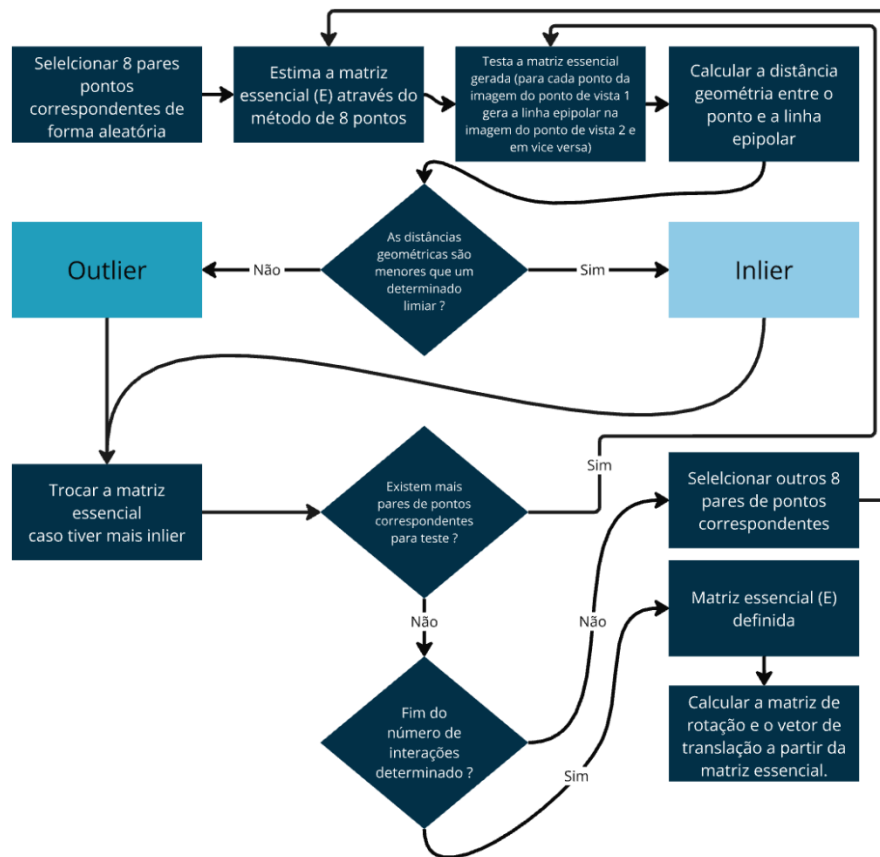
A recuperação da pose da câmera é um processo que consiste em determinar a posição e a orientação da câmera em relação ao ambiente observado, a partir da correspondência de pontos entre imagens capturadas de diferentes pontos de vista. Para realizar esse cálculo, utiliza-se a matriz essencial, que encapsula as informações de rotação e translação da câmera. O método dos 8 pontos, juntamente com o algoritmo RANSAC, é comumente empregado para estimar essa matriz e eliminar outliers.

4.4.1 Recuperação da pose inicial

Para recuperar a pose da câmera, é necessário obter a matriz essencial (**E**). Para isso, foi utilizado o método dos 8 pontos e o algoritmo RANSAC para a exclusão de outliers. A Figura 31 ilustra o fluxograma para obtenção da matriz essencial.

O processo começa com a seleção aleatória de 8 pares de pontos correspondentes, seguida pela estimativa da matriz essencial (**E**) através do método dos 8 pontos. A matriz essencial gerada é então testada para verificar a consistência dos pontos entre as duas imagens, calculando-se a distância geométrica entre os pontos e suas respectivas linhas epipolares. Se as distâncias geométricas forem menores que um determinado limiar, os pontos são considerados inliers.

O processo verifica se há mais pares de pontos para teste, iterando conforme necessário. Caso as distâncias não atendam ao limiar, a matriz essencial é substituída. O processo continua até que o número de iterações determinado seja atingido. Após a validação dos pontos, a matriz essencial final é definida e, a partir dela, são calculados a matriz de rotação e o vetor de translação, completando a recuperação da pose da câmera.

Figura 31 - Fluxos para obter a matriz essencial (**E**).

Fonte: autoria própria

O método dos 8 pontos inicia com a normalização dos pontos T_1 e T_2 (pontos correspondentes na imagem do ponto de vista 1 e ponto de vista 2). Em seguida, constrói-se a matriz A com base nas coordenadas dos pontos correspondentes, seguindo a fórmula $A[i] = [x_i' x_i \ x_i' y_i \ x_i \ y_i' x_i \ y_i' y_i \ 1]$. O próximo passo é resolver o sistema linear $A \cdot e = 0$ utilizando Decomposição em Valores Singulares (SVD).

Depois, aplica-se a restrição de posto 2 na matriz essencial (E), definida como $E = U \cdot S_{adj} \cdot V^T$. Finalmente, desnormaliza-se a matriz essencial com a fórmula $E = T_2^T \cdot T \cdot E \cdot T_1$, concluindo o processo de obtenção da matriz essencial.

Para exemplificar processo é apresentado na Figura 32 os pontos correspondentes ao experimento com a imagem do teclado antes da normalização.

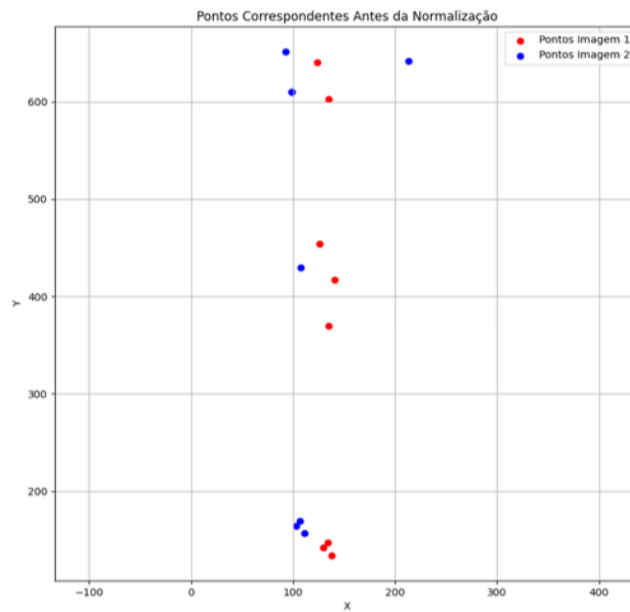
Figura 32 - Pontos correspondentes antes da normalização.

Pontos do pontos de vista 1:

X:123 Y:640
 X:126 Y:454
 X:129 Y:142
 X:133 Y:146
 X:134 Y:369
 X:134 Y:602
 X:137 Y:133
 X:140 Y:417

Pontos do pontos de vista 2:

X:92 Y:651
 X:213 Y:641
 X:103 Y:164
 X:107 Y:169
 X:98 Y:610
 X:98 Y:610
 X:110 Y:156
 X:107 Y:429



Fonte: autoria própria

Agora é necessário realizar a normalização desses pontos para obter uma distribuição uniforme, tipicamente em torno da origem (0,0) com um desvio padrão de 1. Esse processo envolve o deslocamento dos pontos para que seu centro de massa esteja na origem e a escala dos pontos para que a média das distâncias ao centro de massa seja a mesma.

A normalização dos pontos é feita para melhorar a estabilidade numérica dos algoritmos que envolvem cálculos matriciais, como a construção da matriz **A** e a subsequente decomposição em valores singulares (SVD). Quando os pontos não são normalizados, podem ocorrer problemas de escala que tornam os cálculos menos precisos e podem levar a erros significativos nos resultados. A normalização ajuda a mitigar esses problemas ao garantir que todos os pontos tenham uma distribuição semelhante, o que facilita os cálculos numéricos.

Para normalizar os pontos, primeiro é necessário calcular o desvio padrão, a médias dos pontos em X e Y e centro de massa. Para cálculo do desvio padrão foi utilizado a equação 13, onde:

- N é o número de pontos
- Xi é cada ponto na dimensão X ou Y
- μ é a média dos pontos na dimensão X ou Y

$$\text{desvio padrão} = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_i - \mu)^2} \quad (13)$$

Os resultados para os pontos do ponto de vista 1 são os seguintes:

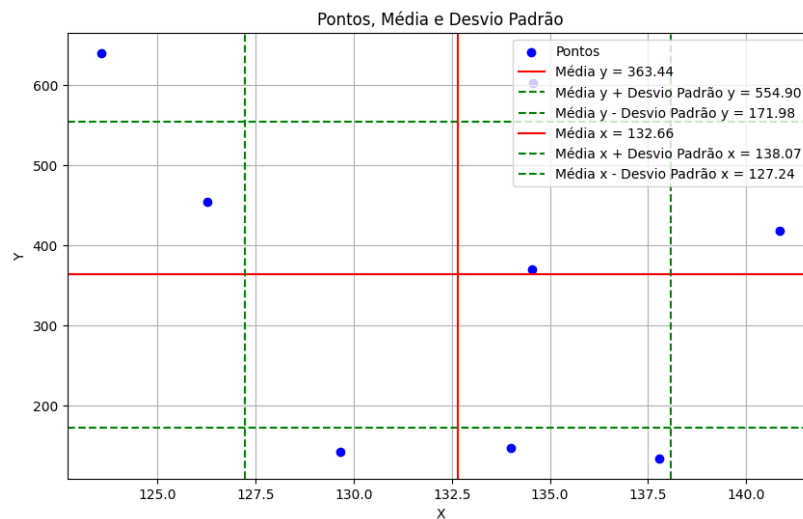
- Média dos valores em x: 132,78
- Média dos valores em y: 375,68

Portanto, o centro de massa desses pontos está localizado em [132,78 375,68].

- Desvio padrão dos valores em x: 5,41
- Desvio padrão dos valores em y: 187,22

Esses resultados são ilustrados na Figura 33.

Figura 33 - Pontos, média e desvio padrão (ponto de vista 1).



Fonte: autoria própria

Os resultados para os pontos do ponto de vista 2 são os seguintes:

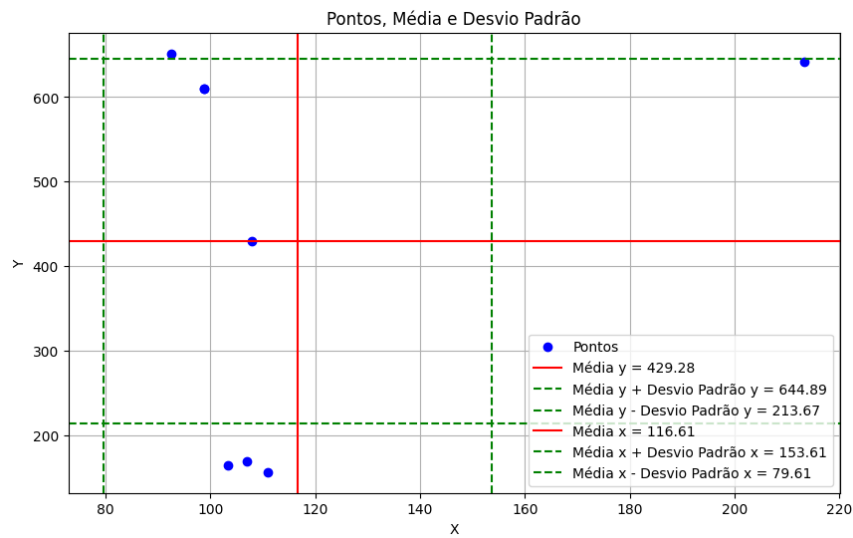
- Média dos valores em x: 116,73
- Média dos valores em y: 429,78

Portanto, o centro de massa desses pontos está localizado em [116,73 429,78].

- Desvio padrão dos valores em x: 38,72
- Desvio padrão dos valores em y: 224,20

Esses resultados são ilustrados na Figura 34.

Figura 34 - Pontos, média e desvio padrão (ponto de vista 2).



Fonte: autoria própria

Com as médias calculadas, é aplicada a equação de centralização, que consiste em subtrair a média dos pontos de cada ponto individual, resultando nos pontos centralizados (equação 14).

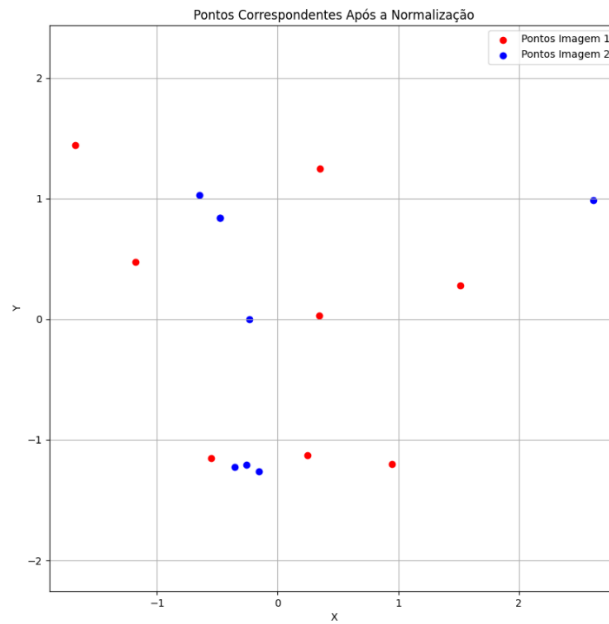
$$pontos_{centralizados} = pontos - media\ dos\ pontos \quad (14)$$

Em seguida, é aplicada a equação de normalização sobre os pontos centralizados, dividindo-os pelo desvio padrão dos pontos (equação 15).

$$pontos_{normalizados} = \frac{pontos_{centralizados}}{desvio\ padrao} \quad (15)$$

Esse processo é realizado separadamente para cada um dos dois pontos de vista (Figura 35).

Figura 35 - Pontos correspondentes normalizados.



Fonte: autoria própria

Pode ser utilizado a matriz de transformação para normalizar os pontos (equação 16).

$$\mathbf{T} = \begin{bmatrix} \frac{1}{\sigma_x} & 0 & -\frac{\mu_x}{\sigma_x} \\ 0 & \frac{1}{\sigma_y} & -\frac{\mu_y}{\sigma_y} \\ 0 & 0 & 0 \end{bmatrix} \quad (16)$$

Onde μ_x e μ_y são as médias dos pontos nas dimensões x e y, respectivamente, e σ_x e σ_y são os desvios padrão dos pontos nas dimensões x e y, respectivamente.

A matriz essencial (\mathbf{E}) possui 9 elementos (é uma matriz 3×3). No entanto, devido à sua definição, ela tem apenas 8 graus de liberdade. Cada correspondência de ponto entre duas imagens fornece uma equação linear. Para determinar os 8 graus de liberdade da matriz essencial \mathbf{E} , é necessário ter pelo menos 8 correspondências de pontos para formar um sistema de equações suficiente (equação 17).

$$\mathbf{A}[i] = [x_i' x_i \quad x_i' y_i \quad x_i' \quad y_i' x_i \quad y_i' y_i \quad y_i' \quad x_i \quad y_i \quad 1] \quad (17)$$

Sendo que x_i e y_i são os pontos do ponto de vista 1 normalizados e x_i' e y_i' são os pontos do ponto de vista 2 normalizados a Figura 36 mostra os valores calculados.

Figura 36 - Matriz **A**.

$$A = \begin{bmatrix} 2,0596 & -1,7097 & -1,2098 & -1,9613 & 1,6291 & 1,1516 & -1,7028 & 1,4133 & 1 \\ -1,5887 & -0,5540 & 1,3199 & -1,3214 & -0,4600 & 1,0962 & -1,2027 & 0,4193 & 1 \\ 0,5188 & 1,1222 & -0,9009 & 0,6706 & 1,4498 & -1,1649 & -0,5760 & -1,2458 & 1 \\ -0,1795 & 0,9764 & -0,7991 & -0,2557 & 1,3909 & -1,1387 & 0,2247 & -1,2219 & 1 \\ -0,3338 & -0,0333 & -1,0319 & 0,3359 & 0,0333 & 1,0386 & 0,3237 & -0,0320 & 1 \\ -0,3412 & 1,2788 & -1,0319 & 0,3431 & -1,2852 & 1,0386 & 0,3305 & 1,2112 & 1 \\ -0,6547 & 0,9156 & -0,7091 & 1,0935 & -1,5287 & -1,1888 & 0,9229 & -1,2914 & 1 \\ -1,4478 & -0,1925 & -0,8620 & 0,3760 & 0,0495 & -0,1676 & 1,6797 & 0,2232 & 1 \end{bmatrix}$$

Para descobrir a matriz essencial, é preciso resolver o sistema linear $\mathbf{A} \cdot \mathbf{e} = 0$. Para isso, será utilizada a decomposição em valores singulares (SVD). A decomposição SVD envolve a decomposição da matriz **A** (equação 18) e a identificação do vetor correspondente ao menor valor singular. Esse vetor será a solução para a matriz essencial **E**.

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T \quad (18)$$

Onde:

- **U** é uma matriz ortogonal de dimensão 8×8 .
- **S** é uma matriz diagonal de dimensão 8×9 contendo os valores singulares.
- v^x é a transposta de uma matriz ortogonal de dimensão 9×9 .

Para encontrar a matriz **U**, é preciso encontrar os autovalores e os autovetores da matriz $\mathbf{A}\mathbf{A}^T$ (equação 19). A Figura 37 mostra o resultado desse processo.

$$\mathbf{A}\mathbf{A}^T = \mathbf{A} \cdot \mathbf{A}^T \quad (19)$$

Figura 37 - Matriz $\mathbf{A}\mathbf{A}^T$.

$$\mathbf{A}\mathbf{A}^T = \begin{bmatrix} 18,132 & -2,878 & -3,196 & -4,065 & -0,277 & -6,107 & -0,615 & -1,726 \\ -2,878 & 10,917 & -5,461 & -4,653 & -2,424 & -1,722 & -5,153 & -2,325 \\ -3,196 & -5,461 & 11,559 & -0,591 & 2,158 & 3,315 & -0,058 & 1,899 \\ -4,065 & -4,653 & -0,591 & 10,996 & -3,182 & -0,192 & -0,970 & -0,764 \\ -0,277 & -2,424 & 2,158 & -3,182 & 3,262 & -1,746 & 1,682 & -0,474 \\ -6,107 & -1,722 & 3,315 & -0,192 & -1,746 & 12,493 & 0,307 & -6,349 \\ -0,615 & -5,153 & -0,058 & -0,970 & 1,682 & 0,307 & 6,487 & -1,679 \\ -1,726 & -2,325 & 1,899 & -0,764 & -0,474 & -6,349 & -1,679 & 8,000 \end{bmatrix}$$

Os autovalores de $\mathbf{A}\mathbf{A}^T$ (equação 20) são os valores da lista resultante. Eles representam o fator de escalonamento aplicado aos autovetores. Na Figura 38, é apresentado o resultado.

$$\det(\mathbf{A}\mathbf{A}^T - \lambda I) = 0 \quad (20)$$

Figura 38 - Autovalores.

$$\det \left(\begin{bmatrix} 18,132 & -2,878 & -3,196 & -4,065 & -0,277 & -6,107 & -0,615 & -1,726 \\ -2,878 & 10,917 & -5,461 & -4,653 & -2,424 & -1,722 & -5,153 & -2,325 \\ -3,196 & -5,461 & 11,559 & -0,591 & 2,158 & 3,315 & -0,058 & 1,899 \\ -4,065 & -4,653 & -0,591 & 10,996 & -3,182 & -0,192 & -0,970 & -0,764 \\ -0,277 & -2,424 & 2,158 & -3,182 & 3,262 & -1,746 & 1,682 & -0,474 \\ -6,107 & -1,722 & 3,315 & -0,192 & -1,746 & 12,493 & 0,307 & -6,349 \\ -0,615 & -5,153 & -0,058 & -0,970 & 1,682 & 0,307 & 6,487 & -1,679 \\ -1,726 & -2,325 & 1,899 & -0,764 & -0,474 & -6,349 & -1,679 & 8,000 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right) = 0 \quad \lambda_U = \begin{bmatrix} 31,834 \\ 18,606 \\ 13,006 \\ 7,301 \\ 4,285 \\ 1,866 \\ 0,973 \\ 0,304 \end{bmatrix}$$

Para realizar o cálculo dos autovetores (v), é utilizado a equação 21.

$$(AA^T - \lambda I)v = 0 \quad (21)$$

Para exemplificar o cálculo de um autovetor, será calculado especificamente o autovalor $\lambda = 31,834$. O resultado está demarcado em azul na Figura 39.

Figura 39 - Autovetor para o autovalor $\lambda = 31,834$.

$$AA^T = \begin{bmatrix} 22,352 & 2,823 & 0,165 & -0,726 & 1,613 & -1,062 & -10,457 & -4,004 \\ 2,823 & 10,355 & -4,295 & -2,643 & 0,463 & 0,858 & -3,099 & -0,361 \\ 0,165 & -4,295 & 9,133 & 7,287 & 0,636 & -1,355 & 3,306 & 0,083 \\ -0,726 & -2,643 & 7,287 & 7,464 & 0,742 & -1,329 & 3,311 & 2,029 \\ 1,613 & 0,463 & 0,636 & 0,742 & 3,476 & 3,355 & 1,342 & 2,870 \\ -1,062 & 0,858 & -1,355 & -1,329 & 3,355 & 6,518 & -2,520 & 0,625 \\ -10,457 & -3,099 & 3,306 & 3,311 & 1,342 & -2,520 & 16,779 & 5,064 \\ -4,004 & -0,361 & 0,083 & 2,029 & 2,870 & 0,625 & 5,064 & 7,190 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad v = \begin{bmatrix} 0,68397 \\ 0,21841 \\ -0,17164 \\ -0,18414 \\ -0,01211 \\ 0,04642 \\ -0,60315 \\ -0,23456 \end{bmatrix}$$

$$U = \begin{bmatrix} 0,684 & 0,475 & -0,314 & -0,249 & 0,308 & -0,077 & -0,191 & 0,085 \\ 0,218 & -0,353 & -0,430 & -0,355 & -0,699 & -0,130 & -0,040 & -0,092 \\ -0,172 & 0,602 & 0,049 & 0,163 & -0,340 & -0,202 & -0,161 & -0,629 \\ -0,181 & 0,491 & -0,103 & 0,114 & -0,432 & 0,195 & 0,111 & 0,682 \\ -0,021 & 0,049 & -0,498 & 0,269 & 0,146 & -0,160 & 0,780 & -0,147 \\ 0,046 & -0,192 & -0,284 & 0,682 & 0,017 & -0,411 & -0,454 & 0,196 \\ -0,604 & 0,084 & -0,300 & -0,478 & 0,271 & -0,441 & -0,158 & 0,134 \\ -0,235 & -0,025 & -0,533 & 0,081 & 0,136 & 0,713 & -0,289 & -0,209 \end{bmatrix}$$

O próximo passo é encontrar a matriz V^T . Para isso, é preciso encontrar a matriz $A^T A$ (Figura 40) e seus autovalores (equação 22) e autovetores (equação 23).

Figura 40 - Matriz $A^T A$.

$$A^T A = A^T \cdot A$$

$$A^T A = \begin{bmatrix} 9,820 & -2,980 & -2,504 & -3,036 & 5,945 & 0,550 & -5,192 & 1,937 & -1,967 \\ -2,980 & 7,954 & -2,223 & 5,945 & -2,599 & -4,758 & 4,084 & -4,915 & 1,804 \\ -2,504 & -2,223 & 8,031 & -1,571 & -2,662 & 0,857 & -1,965 & 0,449 & -5,225 \\ -3,036 & 5,945 & -1,571 & 7,676 & -4,054 & -4,855 & 6,348 & -4,772 & -0,719 \\ 5,945 & -2,599 & -2,662 & -4,054 & 10,894 & -1,392 & -4,485 & -0,969 & 1,279 \\ 0,550 & -4,758 & 0,857 & -4,855 & -1,392 & 8,780 & -3,563 & 7,652 & 0,665 \\ -5,192 & 4,084 & -1,965 & 6,348 & -4,485 & -3,563 & 8,615 & -2,895 & 0,000 \\ 1,937 & -4,915 & 0,449 & -4,772 & -0,969 & 7,652 & -2,895 & 8,404 & -0,524 \\ -1,967 & 1,804 & -5,225 & -0,719 & 1,279 & 0,665 & 0,000 & -0,524 & 8,000 \end{bmatrix}$$

$$(A^T A - \lambda I) = 0 \quad (22)$$

$$(A^T A - \lambda I)v = 0 \quad (23)$$

Na Figura 41 é apresentado a matriz V^T calculada.

Figura 41 - Matriz V^T .

$$V^T = \begin{pmatrix} 0,328 & -0,402 & 0,197 & -0,581 & -0,272 & -0,236 & 0,459 & -0,116 & 0,037 \\ -0,411 & -0,133 & -0,090 & -0,029 & -0,571 & 0,591 & 0,068 & -0,218 & 0,278 \\ 0,068 & 0,404 & 0,534 & 0,382 & -0,099 & 0,120 & 0,597 & 0,145 & 0,016 \\ -0,447 & -0,014 & 0,053 & -0,334 & -0,056 & 0,100 & 0,073 & 0,578 & -0,577 \\ 0,270 & -0,605 & -0,103 & 0,224 & 0,395 & 0,556 & 0,075 & 0,110 & -0,136 \\ 0,359 & 0,363 & -0,348 & -0,106 & -0,081 & 0,304 & 0,131 & -0,396 & -0,578 \\ -0,426 & 0,116 & -0,124 & -0,282 & 0,635 & 0,066 & 0,422 & -0,301 & 0,181 \\ 0,365 & 0,320 & -0,270 & -0,335 & 0,067 & 0,314 & 0,008 & 0,523 & 0,451 \\ -0,043 & -0,212 & -0,667 & 0,390 & -0,129 & -0,258 & 0,472 & 0,225 & 0,005 \end{pmatrix}$$

Agora, é necessário agora encontrar a matriz S , que contém os valores singulares. Para isso, é necessário obter as raízes quadradas dos elementos da diagonal de $A^T A$ (equação 24).

$$\sigma_i = \sqrt{\lambda_i} \quad (24)$$

O resultado é apresentado na Figura 42.

Figura 42 - Matriz S .

$$S_{\text{adj}} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad S_{\text{adj}} = \begin{bmatrix} 5,6422 & 0 & 0 \\ 0 & 4,3135 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad S_{\text{adj}} = \begin{bmatrix} 5,6422 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4,3135 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A matriz essencial (E) deve ter exatamente dois valores singulares não-nulos, ambos iguais, e o terceiro sendo zero. Isso é importante para garantir que E seja uma matriz de transformação epipolar válida entre duas vistas em geometria estéreo (Figura 43).

Figura 43 - Matriz ajustada.

$$S = \begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_n \end{bmatrix} = \begin{bmatrix} 5,642 \\ 4,314 \\ 3,606 \\ 2,702 \\ 2,070 \\ 1,366 \\ 0,986 \\ 0,552 \end{bmatrix}$$

Com a equação 25, pode-se agora calcular a matriz essencial.

$$\mathbf{E} = \mathbf{U} \cdot \mathbf{S}_{adj} \cdot \mathbf{V}^T \quad (25)$$

Na Figura 44, é apresentada a matriz essencial calculada.

Figura 44 - Matriz essencial (\mathbf{E}).

$$E = \begin{bmatrix} -1,039 & 1,411 & -0,645 \\ 1,634 & -2,261 & 1,032 \\ -0,735 & 0,973 & -0,423 \end{bmatrix}$$

Para restaurar a matriz essencial (\mathbf{E}) ao seu sistema de coordenadas original, é necessário primeiro calcular a média dos valores x e y para cada conjunto de pontos. Após determinar essas médias, deve-se calcular a escala necessária para que a distância média dos pontos normalizados até a origem seja igual a $\sqrt{2}$. Esse processo é importante para ajustar a matriz à escala apropriada, permitindo uma análise mais precisa dos dados (equação 26).

$$\mathbf{E} = \mathbf{T}_2^T \cdot \mathbf{T} \cdot \mathbf{E}_{normalizada} \cdot \mathbf{T}_1 \quad (26)$$

Figura 45 – Matriz essencial (\mathbf{E}) resultante.

$$E = \begin{bmatrix} 4,79886767e-05 & 9,12381743e-06 & -1,57199599e-01 \\ 2,93956440e-05 & -7,14041942e-08 & -7,44861236e-03 \\ 1,20956153e-01 & -2,30723135e-03 & 9,53914273e-01 \end{bmatrix}$$

Se tivermos a matriz de calibração é possível extrair a matriz fundamental F (equação 27).

$$\mathbf{F} = \mathbf{K}^{-T} \cdot \mathbf{E} \cdot \mathbf{K}^{-1} \quad (27)$$

Para validar a matriz essencial encontrada, é necessário calcular as linhas epipolares. Para exemplificar, será utilizado um ponto P do nosso exemplo do teclado do ponto de vista 1 (Figura 46).

Figura 46 - Ponto dos pontos de vista 1.

$$P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 123 \\ 567 \\ 1 \end{bmatrix}$$

No ponto de vista 2, utilizando a matriz essencial (\mathbf{E}), será calculado a linha epipolar l' , dado por $l' = E.P$ (Figura 47).

Figura 47 - Cálculo da linha epipolar.

$$l' = \begin{bmatrix} 4,79886767 \times 10^{-5} & 9,12381743 \times 10^{-6} & -1,57199599 \times 10^{-1} \\ 2,93956440 \times 10^{-5} & -7,14041942 \times 10^{-8} & -7,44861236 \times 10^{-3} \\ 1,20956153 \times 10^{-1} & -2,30723135 \times 10^{-3} & 9,53914273 \times 10^{-1} \end{bmatrix} \begin{bmatrix} 123 \\ 567 \\ 1 \end{bmatrix} = \begin{bmatrix} -0,145 \\ -0,004 \\ 14,420 \end{bmatrix}$$

Normalizando l' (Figura 48).

Figura 48 - Linha epipolar normalizada.

$$\|I'_{xy}\| = \sqrt{(-0,145431)^2 + (-0,003866)^2} = 0,14550$$

$$I' = \frac{I'}{\|I'_{xy}\|} = \begin{bmatrix} \frac{-0,145431}{0,14550} \\ \frac{-0,003866}{0,14550} \\ \frac{14,4199}{0,14550} \end{bmatrix} = \begin{bmatrix} -0,9995 \\ -0,0266 \\ 99,1135 \end{bmatrix}$$

Agora, obtém-se, portanto, a equação da linha epipolar no ponto de vista 2 (equação 28)

$$-0,9995x - 0,0266y + 99,1135 = 0 \quad (28)$$

Para validar a matriz essencial será utilizado ponto correspondente no ponto de vista 2 que neste caso é: (x= 92 e y=651) e substituir da equação da linha epipolar (equação 29).

$$-0,9995x - 0,0266y + 99,1135 = 0 \quad (29)$$

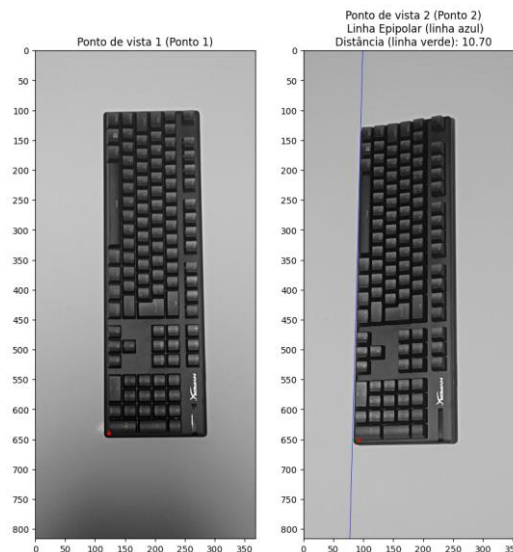
A resposta do cálculo está na Figura 49.

Figura 49 – Avaliação da matriz essencial (**E**) para um determinado ponto.

$$\begin{aligned} &= -0,9995 \times 92,528709 - 0,0266 \times 651,524963 + 99,1135 \\ &= -92,4824 - 17,3354 + 99,1135 \\ &= -109,8178 + 99,1135 \\ &= -10,7043 \end{aligned}$$

Como o resultado não é próximo de zero, a equação da linha epipolar não é satisfeita exatamente por esse ponto. Na Figura 50, é apresentado o ponto 1 na primeira imagem e seu ponto correspondente na segunda imagem, com a linha epipolar traçada em azul. Também é mostrada a linha verdadeira entre o ponto e a linha epipolar, bem como a distância entre eles.

Figura 50 - Linha epipolar traçada.



Fonte: autoria própria

A partir da definição da matriz essencial, é possível realizar a recuperação da pose da câmera. No Algorithm 1, é mostrada a função desenvolvida para recuperar a pose inicial da câmera.

Algorithm 1: Recuperar a pose inicial da câmera

```

1 Função recuperar_pose_inicial(K, p1, p2):
2   // Parâmetros:
3   // - K: matriz de calibração da câmera
4   // - p1: pontos correspondentes na imagem 1
5   // - p2: pontos correspondentes na imagem 2
6   // Cálculo dos parâmetros da câmera
7   comprimento_focal <- 0.5 * (K[0, 0] + K[1, 1])
8   ponto_principal <- (K[0, 2], K[1, 2])
9
10  // Encontrar a matriz essencial e a máscara
11  E, mascara <- encontrarMatrizEssencial(p1, p2, comprimento_focal, ponto_principal, RANSAC, 0.999, 1.0)
12
13  // Criar a matriz de câmera
14  matriz_camera <- [[comprimento_focal, 0, ponto_principal[0]], [0, comprimento_focal, ponto_principal[1]], [0, 0,
15  1]]
16  // Recuperar a pose (rotação e translação) da câmera
17  quantidade_passos, R, T, mascara <- recuperarPose(E, p1, p2, matriz_camera, mascara)
18
19  Retornar R, T, mascara
20 Fim da Função

```

Para melhorar a performance, foi utilizado as funções da biblioteca OpenCV para calcular a matriz essencial. Detalhando o código, primeiramente, o comprimento focal da câmera é calculado a partir da matriz de calibração \mathbf{K} como a média dos valores diagonais, enquanto o ponto principal é extraído da terceira coluna dessa matriz.

A função `cv.findEssentialMat` é usada para estimar a matriz essencial \mathbf{E} entre dois conjuntos de pontos correspondentes (p_1 e p_2). Este método normaliza os pontos, estima a matriz fundamental usando o método dos 8 pontos e filtra correspondências incorretas

utilizando o método de rejeição RANSAC. A matriz essencial é então refinada minimizando a distância entre os pontos correspondentes e suas projeções epipolares.

Em seguida, uma matriz de câmera é criada a partir do comprimento focal e do ponto principal, formando uma matriz 3x3 que define as propriedades intrínsecas da câmera.

A função *cv.recoverPose* utiliza a matriz essencial \mathbf{E} para recuperar a pose da câmera em relação ao objeto 3D. Primeiramente, os pontos inliers são filtrados usando a máscara retornada pelo *findEssentialMat*. A matriz de projeção da primeira imagem é diretamente estimada a partir da matriz de calibração, enquanto a da segunda imagem é derivada da matriz essencial e da calibração. Finalmente, a matriz de rotação \mathbf{R} e o vetor de translação \mathbf{t} são estimados, representando a transformação entre as duas câmeras. O resultado inclui o número de pontos inliers, as matrizes \mathbf{R} e \mathbf{t} , e uma máscara atualizada para filtrar correspondências incorretas.

4.4.2 Recuperação da pose nas imagens subsequentes

Após obter a pose inicial da câmera é necessário atualizar a cada nova imagem, para isso foi utilizado a função *solvePnP*. Antes de executar a função *solvePnP* é necessário obter os pontos 3D e os pontos 2D do objeto, para isso foi desenvolvida a função *get_objpoints_and_imgpoints* (Algorithm 2) que recebe quatro listas de dados: *correspondencias*, *estrutura_indices*, *estrutura* e *pontos_chave*. Em seguida, ela itera sobre cada correspondência em *correspondencias* e realiza as seguintes etapas:

- Obtém os índices da correspondência (*consulta_idx* e *treino_idx*).
- Obtém o índice da estrutura correspondente (*estrutura_idx*) a partir de *estrutura_indices*.
- Se *estrutura_indices* for menor que 0, continua para a próxima iteração.
- Adiciona o ponto 3D correspondente em *estrutura [int(estrutura_idx)]* à lista *pontos_objetos*.
- Adiciona o ponto 2D correspondente em *key_points[treino_idx].pt* à lista *pontos_imagem*.

Algorithm 2: Algoritmo que salva os pontos 3D e 2D

```

1 Função Salva_pontos3D_pontos2D_vetores(correspondencias, estrutura_indices, estrutura,
2 pontos_objeto):
3     // - correspondencias: lista de correspondências entre pontos-chave
4     // - estrutura_indices: índices das estruturas correspondentes
5     // - estrutura: pontos 3D da estrutura
6     // - pontos_chaves: pontos-chave da imagem
7
8     pontos_objeto <- []
9     pontos_imagem <- []
10
11     Para cada correspondencia em correspondencias:
12         consulta_idx <- correspondencia.consultaIdx
13         treino_idx <- correspondencia.trainoIdx
14         estrutura_idx <- estrutura_indices[consulta_idx]
15
16         Se estrutura_idx < 0, então
17             Continue para a próxima iteração
18
19         pontos_objeto.adicionar(estrutura[int(estrutura_idx)])
20         pontos_imagem.adicionar(pontos_chaves[treino_idx].pt)
21
22     Retornar array(pontos_objeto), array(pontos_imagem)
23 Fim da Função

```

A função *get_objpoints_and_imgpoints* desempenha um papel crucial na extração dos pontos tridimensionais e bidimensionais correspondentes de uma correspondência de pontos-chave entre duas imagens. A função percorre cada correspondência na lista *correspondências* e, para cada uma delas, executa as seguintes etapas:

- **Obtenção dos Índices:** Para cada correspondência, obtém o índice de consulta (*consulta_indice*) e o índice de treinamento (*treino_indice*).
- **Verificação de Validade:** Verifica se o índice de estrutura correspondente é válido (ou seja, não é negativo).
- **Adição de Pontos:** Se o índice for válido, adiciona o ponto tridimensional correspondente da estrutura à lista de pontos tridimensionais (*pontos_objeto*) e o ponto bidimensional correspondente da imagem à lista de pontos bidimensionais (*pontos_imagem*).

No final, a função retorna um array numpy contendo os pontos tridimensionais (*pontos_objeto*) e outro array numpy contendo os pontos bidimensionais (*pontos_imagem*) (Algorithm 3).

Algorithm 3: Recupera a pose da câmera

```

1 Função recupera_pose_camera(pontos_objeto, pontos_imagem, K):
2   // Parâmetros de entrada:
3   //   - pontos_objeto: pontos 3D do objeto
4   //   - pontos_imagem: pontos 2D da imagem correspondentes
5   //   - K: matriz de calibração da câmera
6
7   // Inicialização da estimativa da pose
8   MelhorPose <- NULL
9   MelhorContagem <- 0
10  NúmeroMáximoIterações <- 1000
11  Tolerância <- 0.5
12
13  Para i de 1 até NúmeroMáximoIterações:
14    // Selecionar um subconjunto aleatório de pontos correspondentes
15    PontosAleatórios <- AmostraAleatória(pontos_objeto, pontos_imagem)
16
17    // Estimar a pose usando os pontos selecionados
18    PoseEstimada <- EstimarPose(PontosAleatórios, K)
19
20    // Calcular a projeção dos pontos 3D estimados na imagem
21    PontosProjetados <- ProjetarPontos3D(PoseEstimada, pontos_objeto, K)
22
23    // Calcular a contagem de pontos dentro da tolerância
24    ContagemDentroTolerância <- ContarPontosDentroTolerância(PontosProjetados, pontos_imagem,
25 Tolerância)
26    Se ContagemDentroTolerância > MelhorContagem:
27      // Atualizar a melhor estimativa de pose
28      MelhorPose <- PoseEstimada
29      MelhorContagem <- ContagemDentroTolerância
30
31    Se ContagemDentroTolerância >= LimiteMínimo:
32      // Parar o loop se o número mínimo de pontos dentro da tolerância for atingido
33      PararLoop
34
35    // Refinar a pose usando todos os pontos correspondentes e a melhor estimativa obtida
36    PoseFinal <- RefinarPose(MelhorPose, pontos_objeto, pontos_imagem, K)
37
38  Retornar PoseFinal
39 Fim da Função
40

```

4.5 Construção da nuvem de pontos

Para obter a nuvem de pontos, utilizamos a técnica de triangulação linear, que permite a recuperação das coordenadas tridimensionais dos pontos de uma cena a partir de suas imagens capturadas de diferentes pontos de vista. Detalhando os cálculos realizados, primeiramente, é obtido as matrizes de rotação (\mathbf{R}) e translação (\mathbf{t}) através da decomposição da matriz essencial. (Figura 51).

Figura 51 - Decomposição da matriz essencial (\mathbf{E}).

$$T = U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} U^T \quad R = U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T \quad \text{ou} \quad U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T$$

Com as matrizes de rotação e translação, é possível agora obter as matrizes de projeção (Figura 54). Para isso, será considerado a matriz identidade da câmera da Figura 53 e a matriz de calibração da Figura 52.

Figura 52 – Matriz de calibração.

$$K = \begin{bmatrix} 2405.07 & 0 & 1239.30 \\ 0 & 2418.33 & 1066.47 \\ 0 & 0 & 1 \end{bmatrix}$$

Figura 53 - Matriz identidade.

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

A matriz de projeção \mathbf{P} é calculada usando a matriz identidade \mathbf{I} e um vetor de translação zero porque, por convenção, à primeira vista é considerada como estando na origem do sistema de coordenadas. Isso simplifica os cálculos, pois define a posição e orientação da vista de maneira padrão, permitindo que outras vistas sejam descritas em relação a essa referência.

Figura 54 - Matrizes de projeção.

$$P_1 = K \cdot [I | 0]$$

$$P_1 = \begin{bmatrix} 0,684 & 0,218 & -0,172 & -0,184 \\ -0,012 & 0,046 & -0,603 & -0,235 \\ 0,475 & -0,353 & 0,602 & 0,490 \end{bmatrix}$$

$$P_2 = K \cdot [R | T]$$

$$P_2 = \begin{bmatrix} 5.1452e + 02 & -1.2677e + 02 & -2.6532e + 03 & -1.1736e + 02 \\ 1.0360e + 03 & 2.3964e + 03 & -4.1174e + 02 & 2.4152e + 03 \\ 9.5980e - 01 & -1.8031e - 02 & -2.8011e - 01 & -2.2342e - 04 \end{bmatrix}$$

Para cada par de pontos correspondentes x_1, y_1 para o primeiro ponto de vista e x_2, y_2 para o segundo ponto de vista, é montado um sistema de equações lineares. A matriz de projeção \mathbf{P}_1 é usada para o primeiro ponto de vista e a matriz de projeção \mathbf{P}_2 é usada para o segundo ponto de vista. O sistema de equações é apresentado na equação 30.

$$\mathbf{A} = \begin{bmatrix} x_1 \cdot P_1[2, :] & - & P_1[0, :] \\ y_1 \cdot P_1[2, :] & - & P_1[1, :] \\ x_2 \cdot P_2[2, :] & - & P_2[0, :] \\ y_2 \cdot P_2[2, :] & - & P_2[1, :] \end{bmatrix} \quad (30)$$

Para exemplificar, será calculado o ponto tridimensional a partir das projeções P_1 e P_2 para o ponto ($x=123$ $y=640$) no primeiro ponto de vista da imagem do teclado e ($x=92$ $y=651$) no segundo ponto de vista. Na Figura 55 é apresentada a matriz \mathbf{A} calculada.

Figura 55 - Matriz A calculada.

$$\mathbf{A} = \begin{bmatrix} 57,741 & -43,637 & 74,218 & 60,454 \\ 304,012 & -225,966 & 385,563 & 313,835 \\ -425,814 & 125,111 & 2627,830 & 117,339 \\ -411,240 & -2408,136 & 229,248 & -2415,345 \end{bmatrix}$$

Para encontrar o ponto tridimensional, é preciso resolver a equação $\mathbf{AX}=0$ usando SVD, onde o vetor \mathbf{x} corresponde ao menor valor singular da matriz \mathbf{A} , representando o ponto tridimensional em coordenadas homogêneas normalizadas (Figura 56).

Figura 56 - Coordenadas homogêneas

$$X_h = [-1,340 \quad -0,795 \quad -0,224 \quad 1]$$

Por fim é necessário realizar a transformação de coordenadas homogêneas em coordenadas cartesianas isso é feito dividindo cada elemento do vetor pelo último componente (Figura 57).

Figura 57 - Transformação para coordenadas cartesianas.

$$x = \frac{-1,340}{1} = -1,340$$

$$y = \frac{-0,795}{1} = -0,795$$

$$z = \frac{-0,224}{1} = -0,224$$

Portanto, as coordenadas cartesianas do ponto 3D são apresentadas na Figura 58.

Figura 58 - Ponto 3D.

$$(x \quad y \quad z) = (-1,340 \quad -0,795 \quad -0,224)$$

Os cálculos para realiza triangulação foi utilizada a função *cv2.triangulatePoints* do OpenCV. Esta função recebe como entrada as matrizes de projeção das câmeras e as coordenadas dos pontos correspondentes nas imagens, e retorna as coordenadas 3D dos pontos. No Algorithm 4 apresenta o pseudocódigo para que implementa esta rotina.

Algorithm 4: Realiza a triangulação para obter os pontos 3D

```

1 Função triangulacao_obter_pontos3D(K, R1, T1, R2, T2, p1, p2):
2 // Parâmetros:
3 // - K: matriz de calibração da câmera
4 // - R1, R2: matrizes de rotação
5 // - T1, T2: matrizes de translação
6 // - p1, p2: pontos correspondentes na imagem
7
8 proj1 <- matriz 3x4 de zeros
9 proj2 <- matriz 3x4 de zeros
10
11 proj1[0:3, 0:3] <- converter para float32(R1)
12 proj1[:, 3] <- converter para float32(T1.T)
13
14 proj2[0:3, 0:3] <- converter para float32(R2)
15 proj2[:, 3] <- converter para float32(T2.T)
16
17 fk <- converter para float32(K)
18
19 proj1 <- multiplicação de matrizes fk * proj1
20 proj2 <- multiplicação de matrizes fk * proj2
21
22 points_3D <- triangulatePoints(proj1, proj2, transposta(p1),
23 transposta(p2))
24 estrutura <- lista vazia
25
26 Para i de 0 até o tamanho de points_3D[0]:
27     col <- coluna i de points_3D
28     col /= col[3]
29
30     estrutura.adicionar([col[0], col[1], col[2]])
31
32 Retornar estrutura
33 Fim da Função
34

```

A cada nova imagem de ponto de vista diferente será gerado mais pontos 3D da cena o pseudocódigo do Algorithm 5 realiza a fusão de todas as estruturas 3D encontradas.

Algorithm 5: Realiza a fusão das estruturas 3D

```

1 Função fusao_estruturas3D(correspondencias, estrutura_indices, proxima_estrutura_indices, estrutura, proxima_estrutura,
2 cores, proximas_cores):
3 // Parâmetros:
4 // - correspondencias: lista de correspondências
5 // - estrutura_indices: índices das estruturas
6 // - proxima_estrutura_indices: índices das próximas estruturas
7 // - estrutura: estrutura atual
8 // - proxima_estrutura: próxima estrutura
9 // - cores: cores da estrutura atual
10 // - proximas_cores: cores da próxima estrutura
11
12 Para i de 0 até o tamanho de correspondencias:
13     correspondencia <- correspondencias[i]
14     consulta_idx <- correspondencia.consultaIdx
15     traino_idx <- correspondencia.trainIdx
16     estrutura_idx <- estrutura_indices[consulta_idx]
17
18     Se estrutura_idx >= 0, então
19         proxima_estrutura_indices[traino_idx] <- estrutura_idx
20         Continue para a próxima iteração
21
22     estrutura <- concatenar estrutura e [proxima_estrutura[i]] ao longo do eixo 0
23     cores <- concatenar cores e [proximas_cores[i]] ao longo do eixo 0
24
25     estrutura_indices[consulta_idx] <- proxima_estrutura_indices[traino_idx] <- tamanho de estrutura - 1
26
27 Retornar estrutura_indices, proxima_estrutura_indices, estrutura, cores
28 Fim da Função

```

A função *fusion_structure* desempenha um papel importante na fusão de estruturas tridimensionais de múltiplas imagens, combinando os pontos tridimensionais e as informações de cor de duas estruturas em uma única estrutura unificada.

O código recebe os seguintes parâmetros:

- *correspondencias*: uma lista de correspondências de pontos-chave entre duas estruturas.
- *estrutura_indices*: uma lista de índices de pontos tridimensionais correspondentes à primeira estrutura.
- *proxima_estrutura_indices*: uma lista de índices de pontos tridimensionais correspondentes à segunda estrutura.
- *estrutura*: uma matriz contendo os pontos tridimensionais da primeira estrutura.
- *proxima_estrutura*: uma matriz contendo os pontos tridimensionais da segunda estrutura.
- *cores*: uma matriz contendo as informações de cor da primeira estrutura.
- *proxima_cores*: uma matriz contendo as informações de cor da segunda estrutura.

O código percorre cada correspondência na lista correspondências e obtém o índice de consulta (*consulta_indice*) e o índice de treinamento (*treino_indice*). Em seguida, verifica se o índice de estrutura correspondente na primeira estrutura é válido (ou seja, não é negativo). Se for válido, o código atualiza o índice de estrutura correspondente na segunda estrutura (*proxima_estrutura_indices[treino_idx]*) para o mesmo índice encontrado na primeira estrutura (*estrutura_idx*) e continua para a próxima correspondência.

Caso o índice de estrutura correspondente na primeira estrutura não seja válido, o código adiciona o ponto tridimensional correspondente e as informações de cor da segunda estrutura para as matrizes *structure* e *colors*, respectivamente. Em seguida, atualiza tanto o índice de estrutura correspondente na primeira estrutura (*estrutura_indices[consulta_idx]*) quanto o índice de estrutura correspondente na segunda estrutura (*proxima_estrutura_indices[treino_idx]*) para o novo índice do ponto adicionado na estrutura unificada.

No final, a função retorna as seguintes informações atualizadas:

- *estrutura_indices*: a lista de índices de pontos tridimensionais correspondentes à primeira estrutura, possivelmente atualizada.

- *proxima_estrutura_indices*: a lista de índices de pontos tridimensionais correspondentes à segunda estrutura, possivelmente atualizada.
- *estrutura*: a matriz de pontos tridimensionais da estrutura unificada.
- *cores*: a matriz de informações de cor da estrutura unificada.

4.6 Ajuste erro de reprojeção

Os erros de reprojeção são medidas críticas na reconstrução 3D usando SfM, pois ajudam a quantificar o quão bem a reconstrução se ajusta às imagens reais e permitem otimizar os parâmetros da câmera e a localização dos pontos 3D para obter uma reconstrução precisa.

4.6.1 Ajuste de pacote

A função `bundle_adjustment` ou ajuste de pacote é responsável por realizar o ajuste dos pacotes de dados em um pipeline de reconstrução 3D. Essa etapa é essencial para melhorar a qualidade da estrutura tridimensional estimada, refinando as posições dos pontos 3D com base em informações adicionais disponíveis. O pseudocódigo do Algorithm 6 recebe os seguintes parâmetros:

- *rotacoes*: uma lista de matrizes de rotação que representam as orientações das câmeras.
- *translacoes*: uma lista de vetores de translação que representam as posições das câmeras.
- **K**: a matriz de calibração intrínseca da câmera.
- *estruturas_correspondentes_idx*: uma lista de índices correspondentes aos pontos 3D na estrutura.
- *pontos_chave_para_todos*: uma lista de listas contendo os pontos-chave correspondentes às imagens.
- *structure*: uma matriz que armazena os pontos 3D estimados.

O código inicia convertendo as matrizes de rotação para o formato adequado usando a função `converter_para_matriz_de_rotacao`. Essa conversão é necessária para facilitar as operações futuras.

Em seguida, o código itera sobre os índices correspondentes aos pontos 3D na estrutura. Para cada índice, é obtida a lista de pontos-chave correspondentes às imagens e as matrizes de rotação e translação relacionadas à câmera correspondente.

Dentro do loop, o código itera sobre os índices dos pontos 3D e verifica se o ponto possui um ID válido (maior ou igual a zero). Caso seja válido, chama a função *get_3dpos* para obter uma nova estimativa da posição do ponto 3D, usando as informações disponíveis (posição anterior estimada, coordenadas do ponto-chave, matrizes de rotação e translação e matriz de calibração intrínseca).

Ao final do processo, a estrutura é atualizada com as novas estimativas dos pontos 3D. O retorno da função é a estrutura atualizada com os pontos 3D refinados.

Algorithm 7: Ajuste de pacotes

```

1 Função ajuste_pacote(rotacoes, translacoes, K, estruturas_correspondentes_idx, pontos_chave_para_todos,
2 estrutura_parametros:
3   // - rotacoes: lista de rotações
4   // - translacoes: lista de translações
5   // - K: matriz de calibração da câmera
6   // - estruturas_correspondentes_idx: índices das estruturas correspondentes
7   // - pontos_chave_para_todos: lista de pontos-chave para todas as imagens
8   // - estrutura: pontos 3D da estrutura
9
10  Para i de 0 até o tamanho de rotacoes:
11    r <- Converter_para_matriz_de_rotacao(rotacoes[i])
12    rotacoes[i] <- r
13
14  Para i de 0 até o tamanho de estruturas_correspondentes_idx:
15    pontos3d_ids <- estruturas_correspondentes_idx[i]
16    pontos_chave <- pontos_chave_para_todos[i]
17    r <- rotacoes[i]
18    t <- translacoes[i]
19
20    Para j de 0 até o tamanho de pontos3d_ids:
21      pontos3d_id <- ConverterParaInteiro(pontos3d_ids[j])
22
23      Se pontos3d_id < 0, então
24        Continue para a próxima iteração
25
26      novo_ponto <- get_3dpos(estrutura[pontos3d_id], pontos_chave[j].pt, r, t, K)
27      estrutura[pontos3d_id] <- novo_ponto
28
29  Retornar estrutura
30 Fim da Função
31

```

A função *get_3dpos* é responsável por realizar o cálculo de reprojeção dos pontos em um sistema tridimensional. Essa função é importante na análise de pontos e sua validação é determinada por meio de um limiar pré-estabelecido. Se o erro de reprojeção no eixo x for maior que 0,5 pixels ou se o erro de reprojeção no eixo y for maior que 1 pixels, o ponto será considerado inválido e, portanto, descartado.

4.7 Recuperação de escala

Para recuperar informações de escala na reconstrução 3D, foram utilizados marcadores ArUco (Figura 59). Primeiramente, são detectados os marcadores adicionados na cena (Fonte: autoria própria

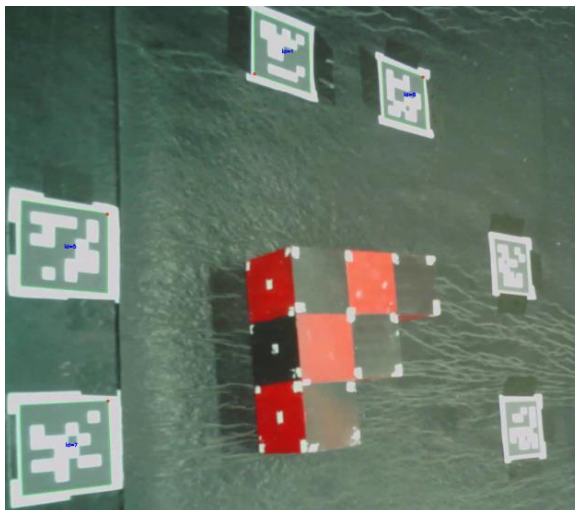
Figura 60) usando um algoritmo de detecção de marcadores disponível na biblioteca OpenCV. Os marcadores ArUco são padrões quadrados binários que codificam um ID único, permitindo sua fácil identificação e cálculo de posição e orientação em 3D.

Figura 59 - Marcador ArUco.



Fonte: autoria própria

Figura 60 - Marcadores sendo detectados.



Fonte: autoria própria

Essa função permite identificar a posição dos marcadores na imagem. Após a detecção dos marcadores, foi utilizado a matriz de calibração da câmera e a matriz de transformação do marcador para estimar suas coordenadas 3D na cena. Isso fornecerá informações sobre a posição tridimensional dos marcadores. Para calcular a escala real, foi utilizado o tamanho real dos marcadores (5 centímetros) e a distância em relação a câmera na cena, com isso foi aplicar a equação 42.

$$escala\ real = \frac{tamanho\ real\ do\ marcador}{distância\ do\ marcador\ à\ câmera} \quad (42)$$

É importante ressaltar que a precisão da reconstrução 3D depende da qualidade da calibração da câmera, da precisão na detecção dos marcadores ArUco e da disponibilidade de informações de escala confiáveis. A distância em relação à câmera é a distância entre a posição tridimensional do marcador detectado na cena e o ponto de referência da câmera. Para encontrar a distância do marcador à câmera, você precisa da posição tridimensional do marcador e a posição da câmera. Essa posição é determinada usando a matriz de calibração da câmera e a matriz de transformação do marcador.

O ponto da câmera que é utilizado como referência geralmente é o centro óptico da câmera, também conhecido como o ponto principal. Esse ponto é determinado durante o processo de calibração da câmera e é representado na matriz de calibração (intrínseca).

O procedimento geral começa com a detecção dos marcadores. Usando o algoritmo de detecção de marcadores da biblioteca OpenCV, os marcadores ArUco são detectados na imagem. Com a matriz de calibração e a matriz de transformação do marcador, as coordenadas 3D dos marcadores são estimadas. Finalmente, a distância do marcador à câmera é calculada usando as coordenadas 3D do marcador e a posição do ponto principal da câmera.

5. EXPERIMENTOS E RESULTADOS

Neste capítulo, serão apresentados os resultados obtidos a partir dos experimentos realizados com o sistema de SfM desenvolvido. O objetivo dos experimentos foi avaliar o desempenho do sistema, tanto em termos de geração de nuvens de pontos, quanto na análise da incerteza de medição das geometrias reconstruídas. Os subcapítulos a seguir abordam a análise dos pontos gerados, a avaliação da incerteza de medição utilizando um objeto de referência, e os experimentos complementares realizados em ambiente externo.

5.1 Análise dos pontos gerados

Na análise da nuvem de pontos gerada pelo sistema desenvolvido, foram utilizados outros objetos para fins de estudo e avaliação. Para facilitar o processo de digitalização desses objetos, o braço robótico KUKA foi empregado. A utilização do braço robótico KUKA oferece uma série de benefícios na digitalização de objetos para análise em nuvens de pontos. Sua capacidade de movimentação controlada e precisa permitem que o sensor de escaneamento capture os pontos necessários em diferentes ângulos e posições, garantindo uma cobertura abrangente do objeto.

Na Figura 61, é possível visualizar as imagens capturadas durante o movimento do braço robótico KUKA. Essas imagens representam etapas intermediárias no processo de geração da nuvem de pontos, utilizando o sistema desenvolvido.

Figura 61 - Imagens capturadas durante o movimento do braço robótico KUKA.

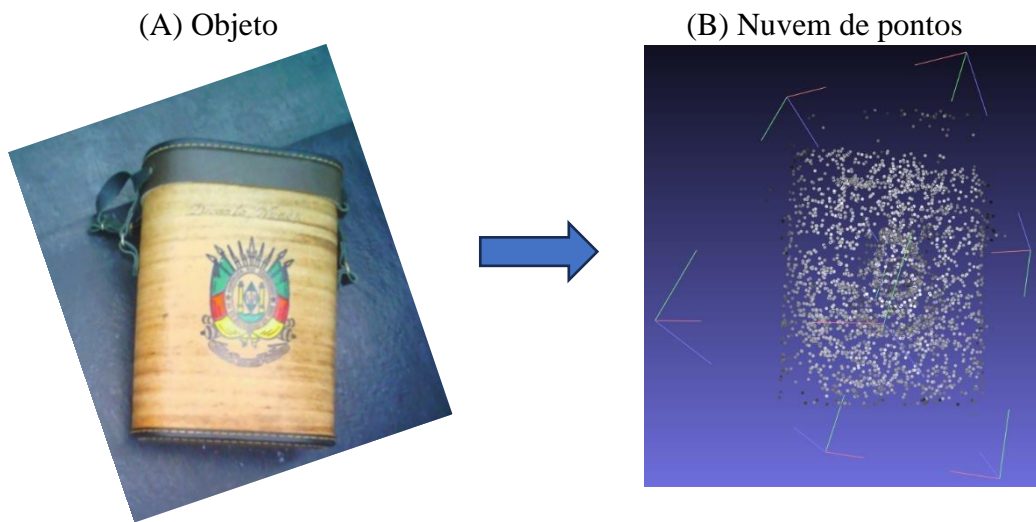


Fonte: autoria própria

A Figura 62 (B) apresenta a nuvem de pontos resultante do sistema após a utilização das imagens capturadas. O processo de SfM demonstra que o sistema conseguiu reconstruir a

geometria tridimensional do objeto de forma coerente, com a nuvem de pontos seguindo o contorno e a forma geral do objeto. A distribuição de pontos é densa nas áreas principais. Entretanto, algumas inconsistências são visíveis, como a dispersão irregular em certas regiões, o que pode indicar ruídos na reconstrução ou limitações na captura das imagens, como ângulos e sobreposição inadequados.

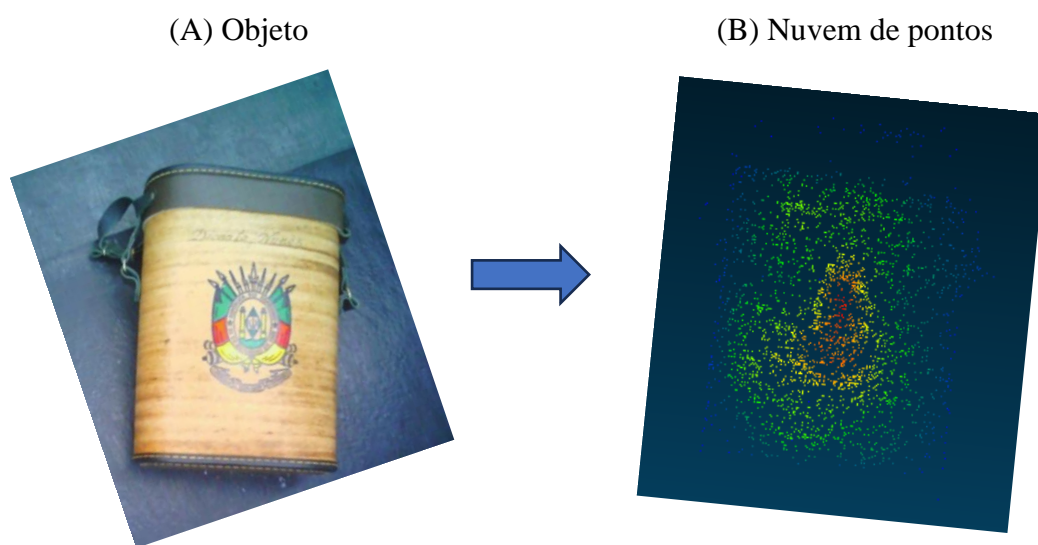
Figura 62 - Nuvem de pontos gerada.



Fonte: autoria própria

A análise da nuvem de pontos gerada foi complementada com a criação de um mapa de calor, representado na Figura 63 (B). Esse mapa de calor destaca a concentração mais alta de pontos (indicada pela cor mais quente, como o vermelho), enquanto as áreas mais afastadas do centro têm menos pontos (indicadas pela cor mais fria, como o azul). Isso sugere que a região central possui mais detalhes que são únicas ou distintivas como: contraste elevado, texturas complexas e cantos e junções, enquanto as regiões periféricas são mais homogêneas ou menos texturizadas. A criação desse mapa de calor foi possível graças à utilização do algoritmo SIFT (*Scale-Invariant Feature Transform*) durante a captura dos pontos na nuvem.

Figura 63 - Mapa de calor objeto de frente.



Fonte: autoria própria

Além disso, outra análise realizada foi a geração de um histograma que reflete a distribuição da densidade volumétrica, conforme ilustrado nas Figura 64 e Figura 65. No eixo horizontal do histograma, observa-se a densidade volumétrica com um fator de escala definido em $r=3$, enquanto o eixo vertical apresenta a contagem de valores correspondentes a cada faixa de densidade.

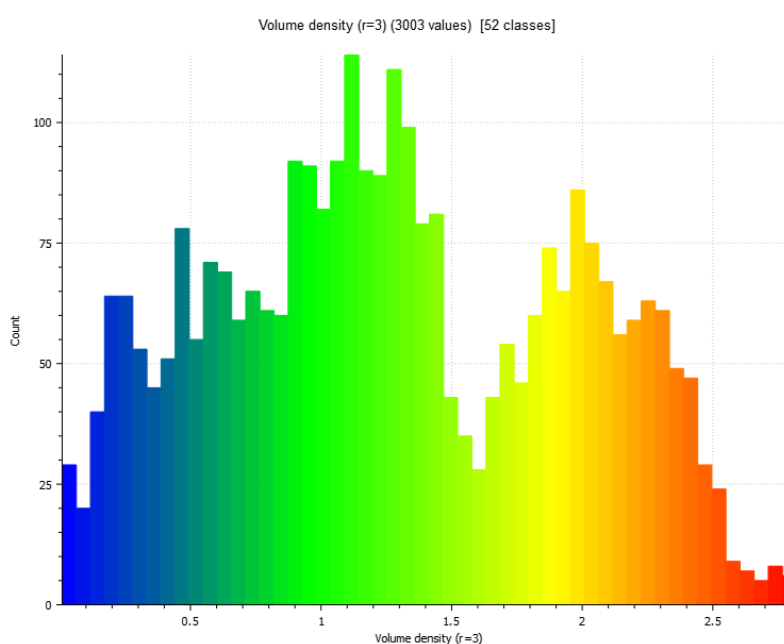
O histograma revela três picos principais. O primeiro, localizado na região azul, corresponde a uma densidade volumétrica baixa e sugere uma quantidade considerável de valores nessa classe de densidade. O segundo pico, que é o mais alto e está na região verde, indica que a maioria dos valores se concentra em densidades médias. Por fim, o pico na região amarela indica uma quantidade menor de valores em densidades mais altas, mas que ainda possui relevância na distribuição total.

Ao analisar o histograma da nuvem de pontos, é possível observar como os pontos estão distribuídos ao longo do objeto escaneado. Faixas com valores mais altos indicam uma maior concentração de pontos em uma determinada região do espaço, enquanto as faixas com valores mais baixos apontam áreas com menor densidade de pontos.

Essa análise de distribuição de densidade volumétrica é útil para obter informações sobre a quantidade de detalhes capturados no objeto. Em áreas com maior densidade, é possível identificar regiões com mais informações, o que pode representar um ganho para avaliar uniformidade, detectar irregularidades ou verificar o preenchimento de áreas vazias no modelo tridimensional.

A densidade volumétrica observada no histograma também fornece uma visão sobre a qualidade e a quantidade de dados capturados pelo sistema de escaneamento. Quando a densidade é alta, isso reflete uma maior quantidade de pontos, o que pode trazer ganho para representar o objeto de forma mais detalhada. Por outro lado, densidades mais baixas indicam áreas com menor número de pontos, o que pode impactar negativamente a precisão e a confiabilidade da análise nessas regiões. Essas informações podem ser úteis para direcionar esforços em áreas que exigem mais atenção ou ajustes na captura, com o objetivo de otimizar a coleta de dados e melhorar a qualidade do modelo.

Figura 64 - Histograma da nuvem de pontos.



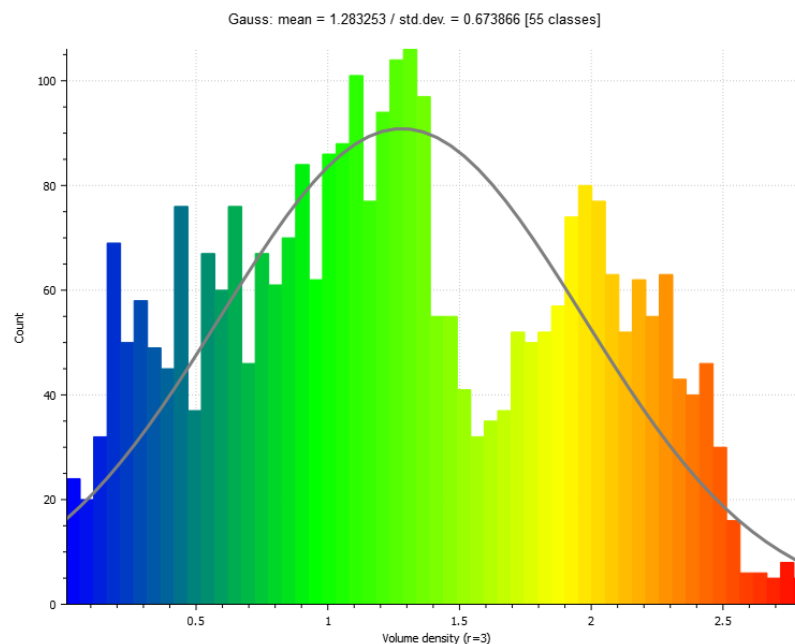
Fonte: autoria própria

Além disso, foi realizado o cálculo do desvio padrão do histograma da nuvem de pontos gerada. O desvio padrão é uma medida estatística que indica a magnitude da dispersão dos valores em relação à média. No contexto da nuvem de pontos, o desvio padrão do histograma fornece uma avaliação da variação na distribuição dos pontos em relação ao raio de análise.

Ao calcular o desvio padrão do histograma, é possível quantificar a dispersão dos pontos em torno da média, fornecendo uma medida da variabilidade dos valores dentro do raio considerado. Valores mais baixos de desvio padrão indicam uma distribuição mais uniforme dos pontos, enquanto valores mais altos indicam uma maior variabilidade na distribuição (Figura 65).

Por outro lado, valores mais altos de desvio padrão podem indicar áreas com maior variação na densidade de pontos, o que pode ser útil para identificar regiões com maior complexidade ou irregularidades. Essas informações podem ser relevantes em análises que requerem a detecção de anomalias ou detalhes distintivos.

Figura 65 - Desvio padrão da nuvem de pontos gerada.



Fonte: autoria própria

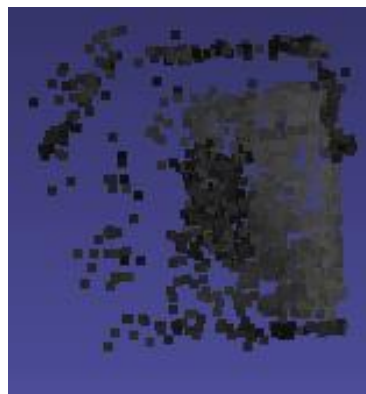
Realizou-se uma segunda avaliação do mesmo objeto, variando apenas a sua posição, enquanto o movimento do braço robótico permaneceu inalterado. Os resultados dessa avaliação são apresentados na Figura 66.

Figura 66 - Nuvem de pontos gerada (nova posição do objeto).

(A) Objeto



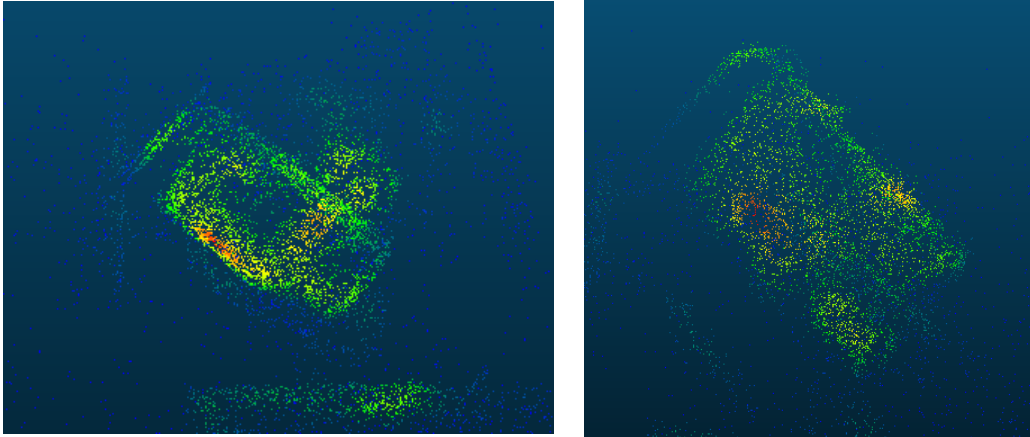
(B) Nuvem de pontos



Fonte: autoria própria

Além disso, foi gerado um mapa de calor a partir da nuvem de pontos obtida (Figura 67). Nesse mapa, é possível observar que houve dificuldade em gerar pontos na lateral do objeto, que estava apoiada na mesa. A concentração de pontos também foi menor em comparação com a posição anterior do objeto.

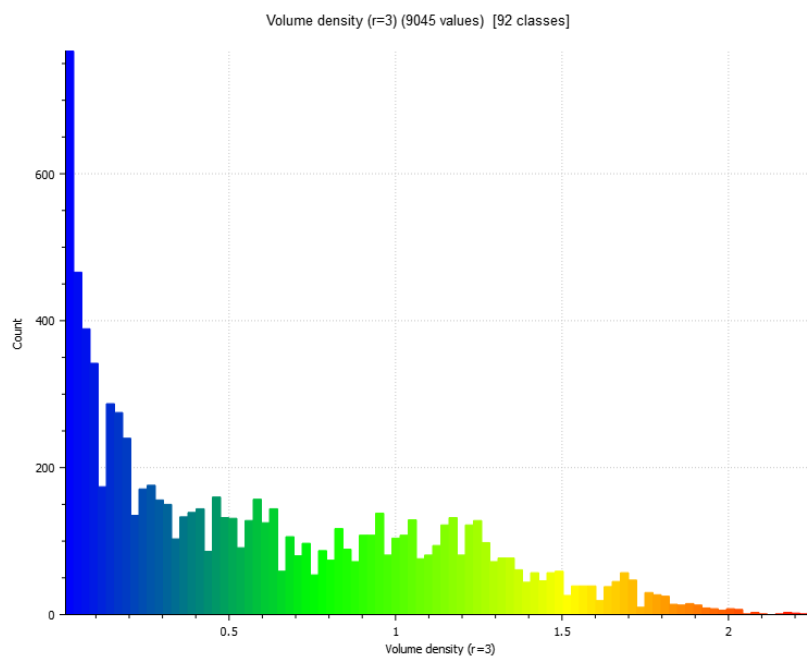
Figura 67 - Mapa de calor objeto segunda posição.



Fonte: autoria própria

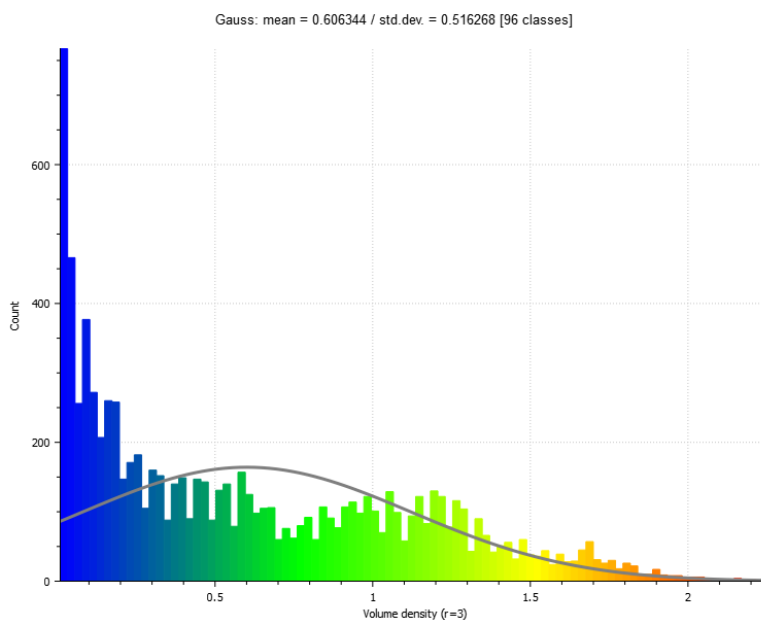
Os histogramas gerados confirmam uma densidade menor de pontos em um raio de 3 cm (Figura 68 e Figura 69). Essa redução na densidade pode ser atribuída a fatores como oclusões, variações na iluminação e sombras decorrentes da nova posição do objeto.

Figura 68 – Histograma.



Fonte: autoria própria

Figura 69 - Desvio padrão.



Fonte: autoria própria

Essa análise mais avançada dos resultados permite identificar padrões e variações nos dados coletados. Compreender as possíveis causas das diferenças observadas, como o impacto da posição do objeto, oclusões e condições de iluminação, é importante para garantir a qualidade e a consistência das análises tridimensionais. Essas informações são utilizadas para a tomada de decisões em projetos que envolvem a digitalização e a reconstrução tridimensional de objetos.

Outro experimento realizado consistiu em produzir uma nuvem de pontos em um ambiente externo. Nesse experimento, foi utilizada uma câmera Nikon D200, com a captura das imagens realizada manualmente, conduzindo a câmera ao redor da cena de interesse, que se tratava de uma planta química. As capturas das imagens foram feitas levando em consideração a sobreposição necessária entre elas para possibilitar a reconstrução 3D. Na Figura 70, é apresentada a cena em que as imagens foram capturadas.

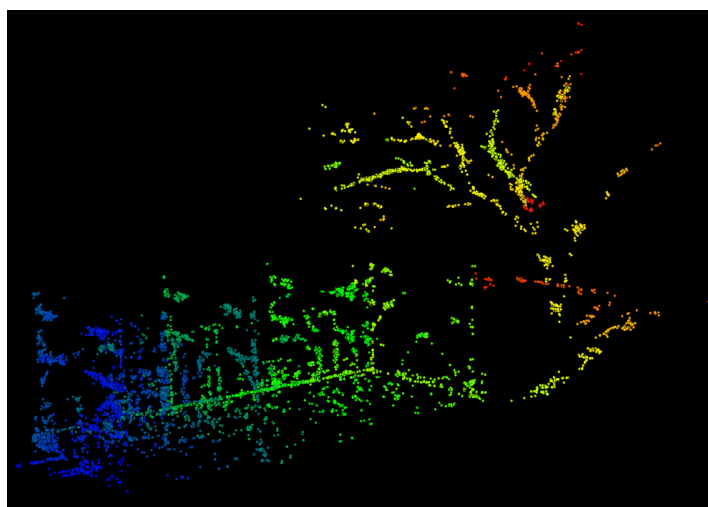
Figura 70 – Cenário externo de planta química.



Fonte: autoria própria

Na Figura 71, temos o resultado da nuvem de pontos gerada pelo sistema desenvolvido, que apresenta uma nuvem de pontos esparsa da cena.

Figura 71 - Nuvem de pontos gerada pelo sistema por SfM.

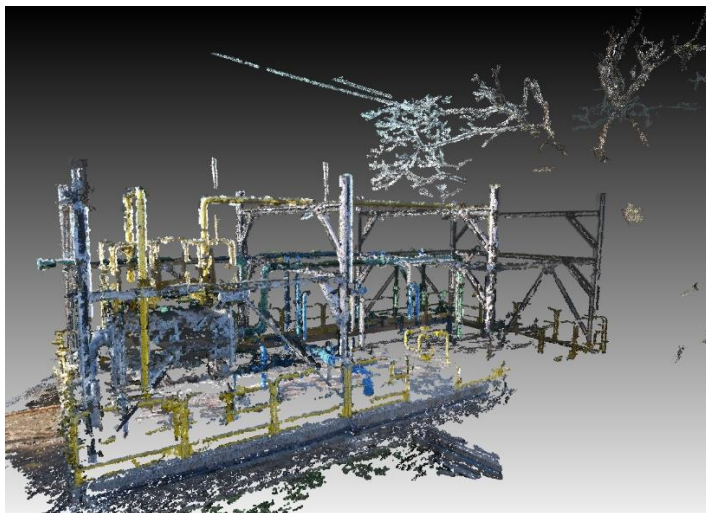


Fonte: autoria própria

Foi realizado também um teste utilizando a nuvem gerada como entrada em uma biblioteca de densificação, a OpenMVS (*Open Multiple View Stereo Vision*). O OpenMVS é um software de código aberto que implementa algoritmos de visão computacional para realizar a reconstrução densa de modelos 3D. Ele utiliza imagens e informações sobre as poses da câmera para gerar uma nuvem de pontos densa a partir de uma nuvem esparsa previamente gerada, melhorando a qualidade da reconstrução 3D ao estimar a posição de novos pontos e preencher lacunas na nuvem esparsa.

Com o uso do OpenMVS, além da nuvem de pontos esparsa, foram fornecidos as poses da câmera e as imagens capturadas, permitindo que o sistema realizasse a densificação e gerasse uma nuvem de pontos mais detalhada (Figura 72).

Figura 72 – Densificação da nuvem gerada por SfM utilizando a biblioteca OpenMVS.



Fonte: autoria própria

5.2 Incerteza de medição

Para se avaliar a incerteza do sistema, um objeto que pudesse servir como ponto de referência. Esse objeto foi avaliado em uma máquina com uma precisão de medição conhecida de (Anexo C), a Zeiss Prismo Ultra. Nos subcapítulos 5.2.1, foi detalhado os procedimentos realizados, enquanto no subcapítulo 5.2.2, foi descrito os passos executados para medir, comparar e extrair a precisão de medição do sistema.

5.2.1 Medições da peça de referência com a Zeiss Prismo Ultra

Durante o processo de obtenção de uma peça de referência, utilizaram-se medições precisas realizadas com a máquina de medição por coordenadas ZEISS Prismo Ultra. Para facilitar a compreensão é apresentado na Figura 74 o modelo CAD da peça usinada, que serviu como base para as medições subsequentes. Na Figura 75, detalhamos as definições dos planos a serem medidos, fornecendo uma visão clara dos detalhes selecionados. Além disso, na Figura 73, destacamos visualmente todos os planos definidos, permitindo uma análise minuciosa desses detalhes.

Figura 74 - Modelo de referência.

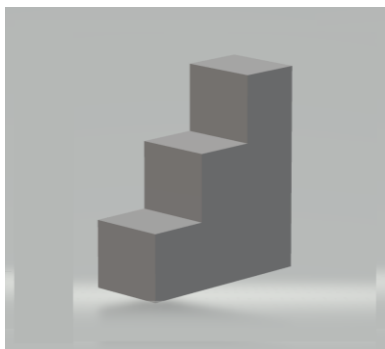


Figura 75 - Marcação das medições.

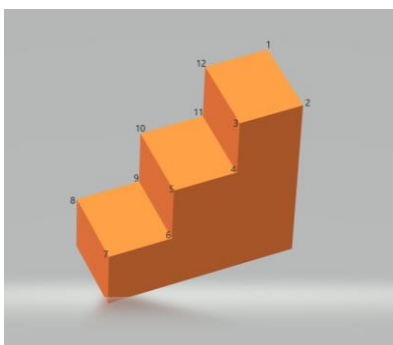
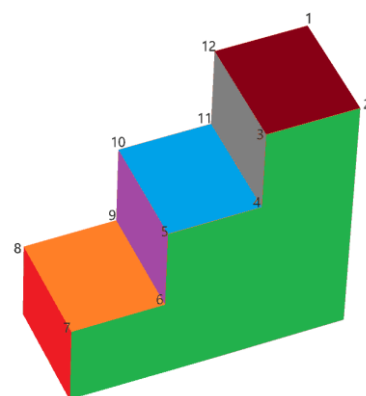


Figura 73 - Planos utilizados.

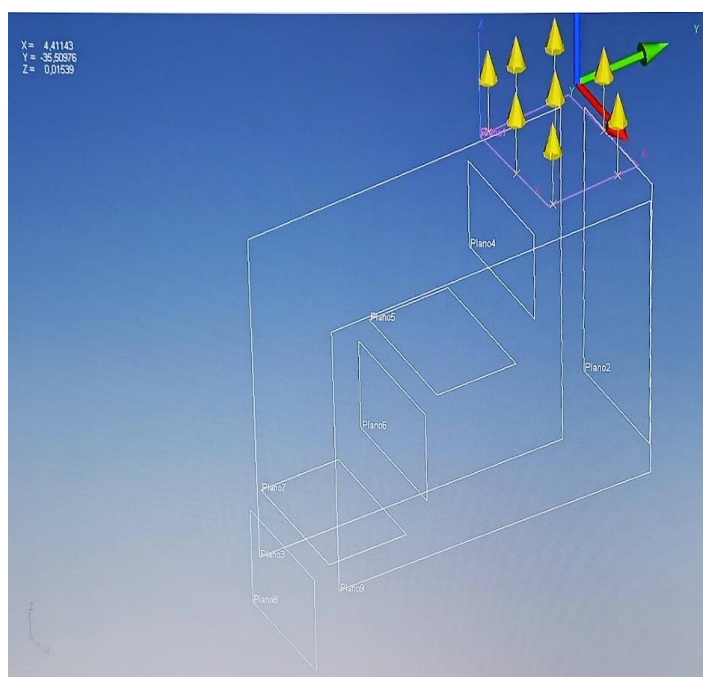


Fonte: autoria própria

Os planos no software ZEISS Prismo Ultra desempenham um papel importante na medição precisa de objetos em nuvens de pontos. Esses planos são superfícies virtuais que são criadas com base em pontos selecionados na nuvem de pontos. Eles permitem que os profissionais realizem medições específicas sobre os detalhes dimensionais dos objetos.

Na Figura 76, é possível visualizar os planos no software ZEISS Prismo Ultra. Nessa tela, é possível selecionar a região de interesse na nuvem de pontos e criar planos virtuais com base nesses pontos selecionados. Esses planos podem representar superfícies planas ou outras formas geométricas, dependendo das necessidades de medição.

Figura 76 - Planos definidos no software da ZEISS Prismo ultra.



Fonte: autoria própria

Uma vez que os planos foram determinados, o software ZEISS Prismo Ultra envia as coordenadas correspondentes para a máquina realizar as medições. Isso garante que as medições sejam executadas com base nos pontos de interesse determinados anteriormente (Figura 73), garantindo uma precisão e consistência confiáveis.

Na Figura 78, é observado as medições sendo realizadas na peça. O software fornece ferramentas avançadas de medição, como distância, ângulo, raio e área, que podem ser aplicadas diretamente nas superfícies dos planos. Essas medições fornecem informações das dimensões do objeto.

Além disso, na Figura 77, pode-se ver em destaque o apalpador utilizado para medir os planos. Esse apalpador, um dispositivo físico conectado à máquina, é responsável por coletar os dados necessários para realizar as medições com precisão. Ele desliza ao longo das superfícies definidas pelos planos, capturando os pontos necessários para realizar as medições.

Figura 78 - Medições sendo realizadas pelo ZEISS Prismo ultra.

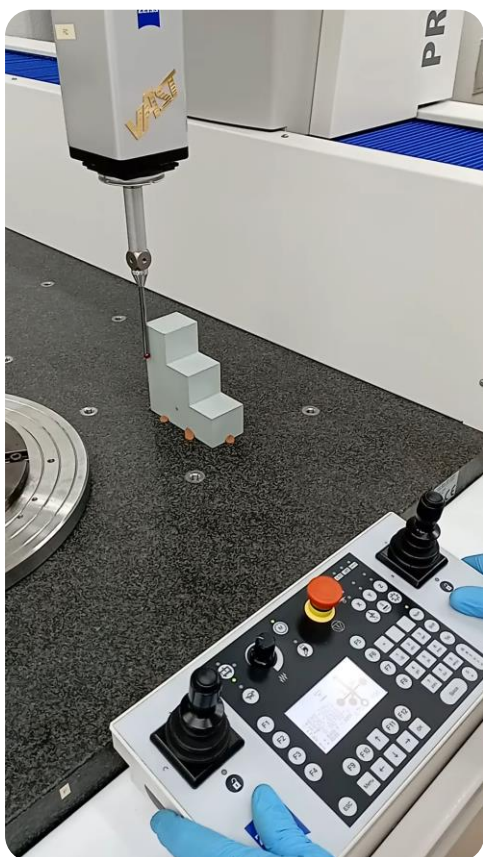
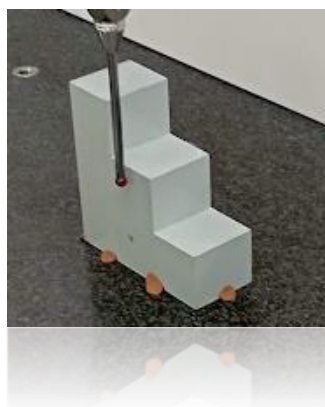


Figura 77 - Destaque apalpador.



Na Tabela 8, são apresentados os resultados das medições realizadas, totalizando 7 repetições de cada medida. Também são apresentadas as médias dos valores medidos, seus desvios, a variância e o desvio padrão. Essas medidas são fundamentais para avaliar a geometria e as dimensões do objeto em questão. O software gera um relatório com todas as informações de medição (Anexo C).

Tabela 8 - Resultado objeto padrão (software ZEISS Prismo).
(Obs.: medidas estão milímetro)

Medidas	Valores	Valores	Valores	Valores	Valores	Valores	Valores	Média	Desvio	Desvio	Desvio	Desvio	Desvio	Desvio	Desvio	Desvio	Variância	Desvio padrão
Distance_1-2_X	50,30170	50,30940	50,30082	50,30090	50,30059	50,30066	50,30069	50,30211	0,00041	0,00729	0,00129	0,00121	0,00152	0,00145	0,00142	0,00000125	0,00324	
Distance_2-3_Y	39,95271	39,95171	39,95164	39,95173	39,95190	39,95167	39,95157	39,95185	0,00086	0,00014	0,00021	0,00012	0,00005	0,00018	0,00028	0,00000002	0,00039	
Distance_3-4_Z	39,94989	39,95033	39,95039	39,95016	39,95004	39,94991	39,95011	39,95012	0,00023	0,00021	0,00027	0,00004	0,00008	0,00021	0,00001	0,00000001	0,00019	
Distance_4-5_Y	39,99207	39,99184	39,99212	39,99215	39,99212	39,99219	39,99218	39,99210	0,00003	0,00026	0,00002	0,00005	0,00002	0,00009	0,00008	0,00000000	0,00012	
Distance_5-6_Z	40,00780	40,00784	40,00772	40,00760	40,00769	40,00758	40,00766	40,00770	0,00010	0,00014	0,00002	0,00010	0,00001	0,00012	0,00004	0,00000000	0,00010	
Distance_6-7_Y	40,25156	40,25160	40,25147	40,25129	40,25126	40,25119	40,25118	40,25136	0,00020	0,00024	0,00011	0,00007	0,00010	0,00017	0,00018	0,00000000	0,00018	
Distance_7-8_X	50,21269	50,21301	50,21271	50,21259	50,21237	50,21239	50,21236	50,21259	0,00010	0,00042	0,00012	0,00000	0,00022	0,00020	0,00023	0,00000001	0,00024	
Distance_8-9_Y	40,32398	40,32401	40,32394	40,32378	40,32360	40,32360	40,32362	40,32379	0,00019	0,00022	0,00015	0,00001	0,00019	0,00019	0,00017	0,00000001	0,00019	
Distance_9-10_Z	39,96864	39,96878	39,96846	39,96838	39,96873	39,96834	39,96843	39,96854	0,00010	0,00024	0,00008	0,00016	0,00019	0,00020	0,00011	0,00000000	0,00018	
Distance_10-11_Y	39,99094	39,99074	39,99084	39,99091	39,99099	39,99094	39,99091	39,99090	0,00004	0,00016	0,00006	0,00001	0,00009	0,00004	0,00001	0,00000000	0,00008	
Distance_11-12_Z	39,82840	39,82905	39,82880	39,82857	39,82845	39,82827	39,82841	39,82856	0,00016	0,00049	0,00024	0,00001	0,00011	0,00029	0,00015	0,00000001	0,00027	

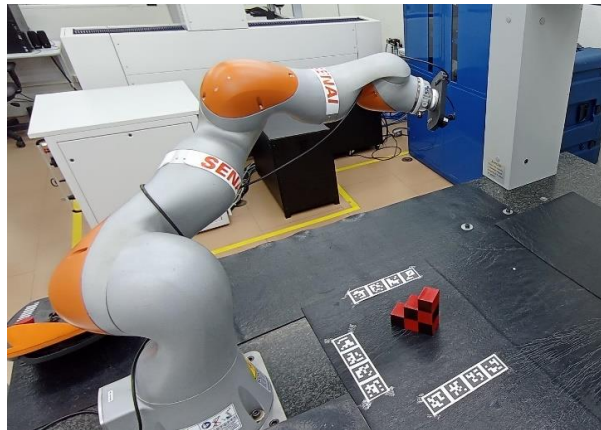
Fonte: autoria própria

5.2.2 Medições da peça de referência com SfM

Para realizar as medições no sistema, empregamos o mesmo conjunto de configurações e trajetória do braço robótico Kuka, conforme explicado detalhadamente no subcapítulo 5.1. Isso garantiu uma consistência na abordagem e permitiu que as medições fossem feitas de forma comparável.

Além disso, foi utilizado esse conjunto de configurações para capturar imagens do objeto padrão, conforme ilustrado na Figura 79.

Figura 79 - Braço robótico KUKA.

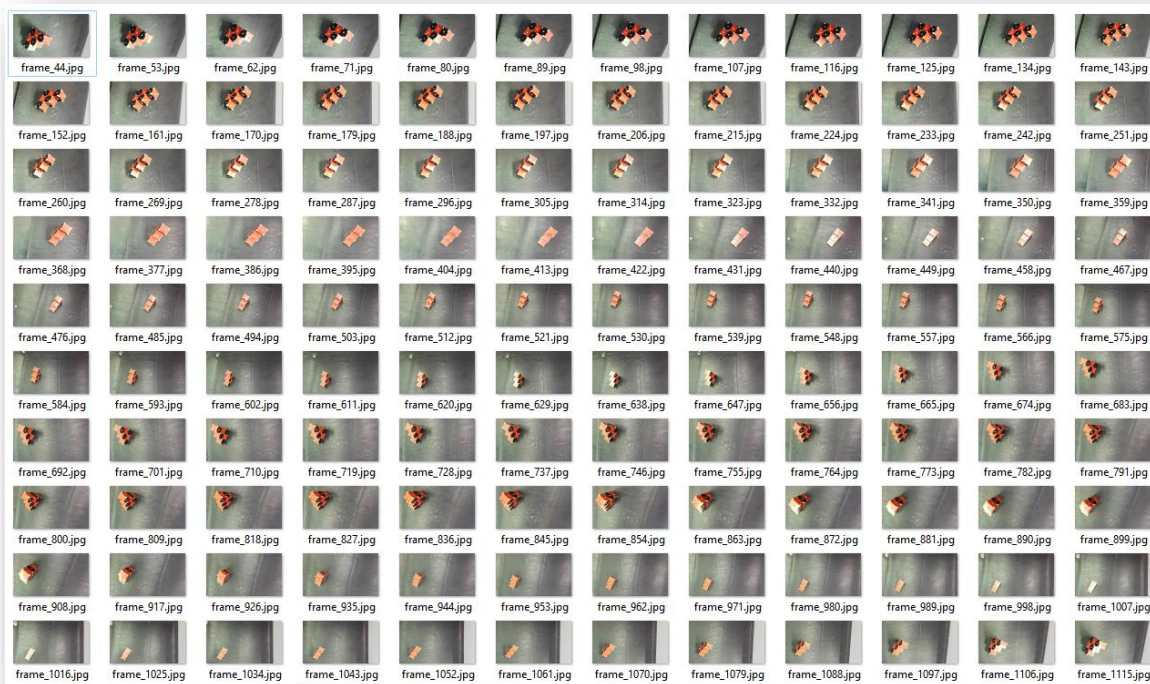


Fonte: autoria própria

Essas imagens foram posteriormente processadas para extrair os frames necessários para aplicar a metodologia de SfM. É importante destacar que a escolha deliberada dessas configurações e trajetória do braço robótico foi feita com o propósito de garantir a precisão e a consistência dos dados coletados.

Na Figura 80, identificamos os frames específicos que foram utilizados para realizar a reconstrução 3D do objeto.

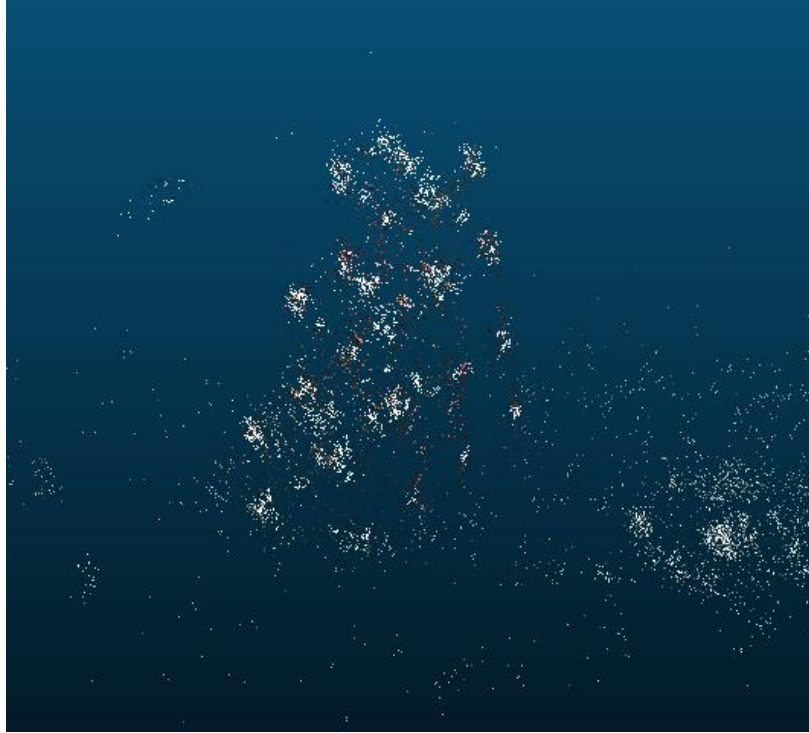
Figura 80 - Frames do objeto padrão.



Fonte: autoria própria

Na Figura 81 é ilustrado o resultado na nuvem de pontos do objeto de referência.

Figura 81 - Nuvem de pontos gerada.



Fonte: autoria própria

Com a nuvem de pontos finalizada, os planos do objeto padrão foram determinados utilizando o método dos mínimos quadrados.

A análise de mínimos quadrados é uma técnica estatística que busca encontrar a melhor representação linear que se ajusta aos dados de um conjunto de pontos. Seu objetivo é minimizar a soma dos quadrados das diferenças entre os valores observados e os valores previstos pela linha de ajuste. Essa linha é comumente conhecida como "linha de regressão" ou "linha de melhor ajuste".

A equação geral de uma linha de regressão linear (equação 31).

$$y = mx + b \quad (31)$$

onde:

- y é a variável dependente (a que você está tentando prever).
- x é a variável independente (a que você está usando para prever y).
- m é a inclinação da linha (coeficiente angular), que representa como y muda em relação a x .
- b é a intercepção da linha com o eixo y (coeficiente linear), que representa o valor de y quando x é zero.

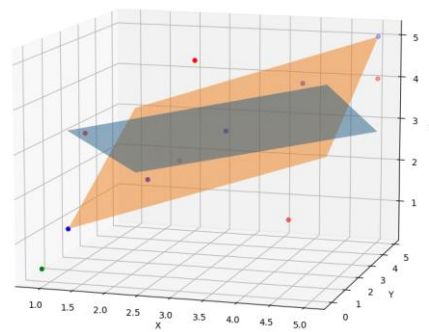
A técnica dos mínimos quadrados visa encontrar os valores de m e b que minimizam a soma dos quadrados dos resíduos (diferenças entre os valores observados de y e os valores previstos pela linha de regressão). Isso pode ser representado matematicamente (equação 48).

$$\min = \sum_i^n (y_i - \hat{y}_i)^2 \quad (48)$$

onde n é o número de pontos de dados, $((x_i, y_i))$ são os pontos individuais e $(mx_i + b)$ é o valor previsto pela linha de regressão para o ponto (x_i) .

Os coeficientes (m) e (b) que minimizam essa soma dos quadrados dos resíduos são calculados usando métodos matemáticos, como a solução de equações normais ou técnicas de otimização. A linha de regressão resultante é a que melhor se ajusta aos dados em termos de minimizar as diferenças entre os valores observados e os valores previstos (Figura 82).

Figura 82 - Interseção entre planos.



Fonte: autoria própria

Na Figura 84 mostra a definição de 4 planos e suas interseções é possível terminar os pontos necessários para realizar medições, seguindo o mesmo planejamento feito na Zeiss Prismo Ultra. Desta pode se realizar as 11 medições (Figura 83):

- Distância_1-2_Y
- Distância_1-2_X
- Distância_2-3_Y
- Distância_3-4_Z
- Distância_4-5_Y

- Distância_5-6_Z
- Distância_6-7_Y
- Distância_7-8_X
- Distância_8-9_Y
- Distância_9-10_Z
- Distância_10-11_Y
- Distância_11-12_Z.

Repetida 7 vezes para determinar o desvio padrão das medições.

Figura 84 - Definição da interseção entre planos.

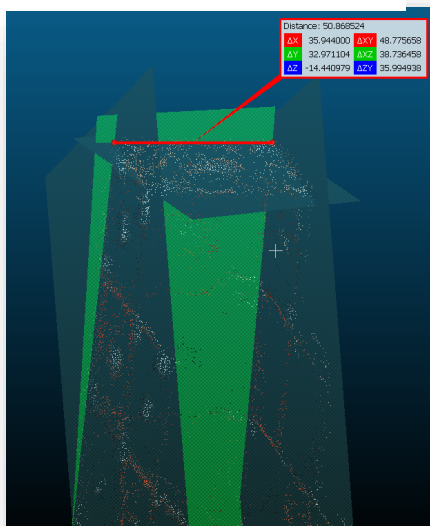
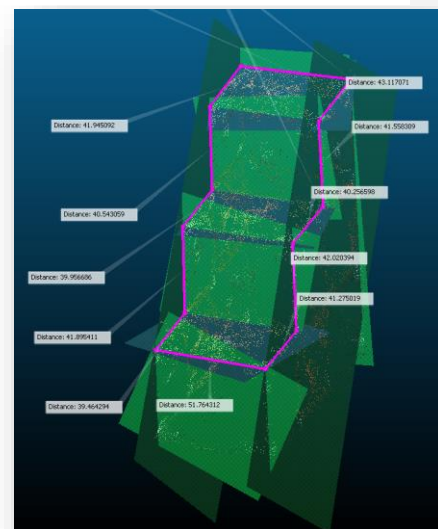


Figura 83 - Todo os planos utilizados nas medições.



Fonte: autoria própria

A Tabela 9, mostra os resultados das 7 medições, médias das medições, desvio, variância e o desvio padrão encontrado.

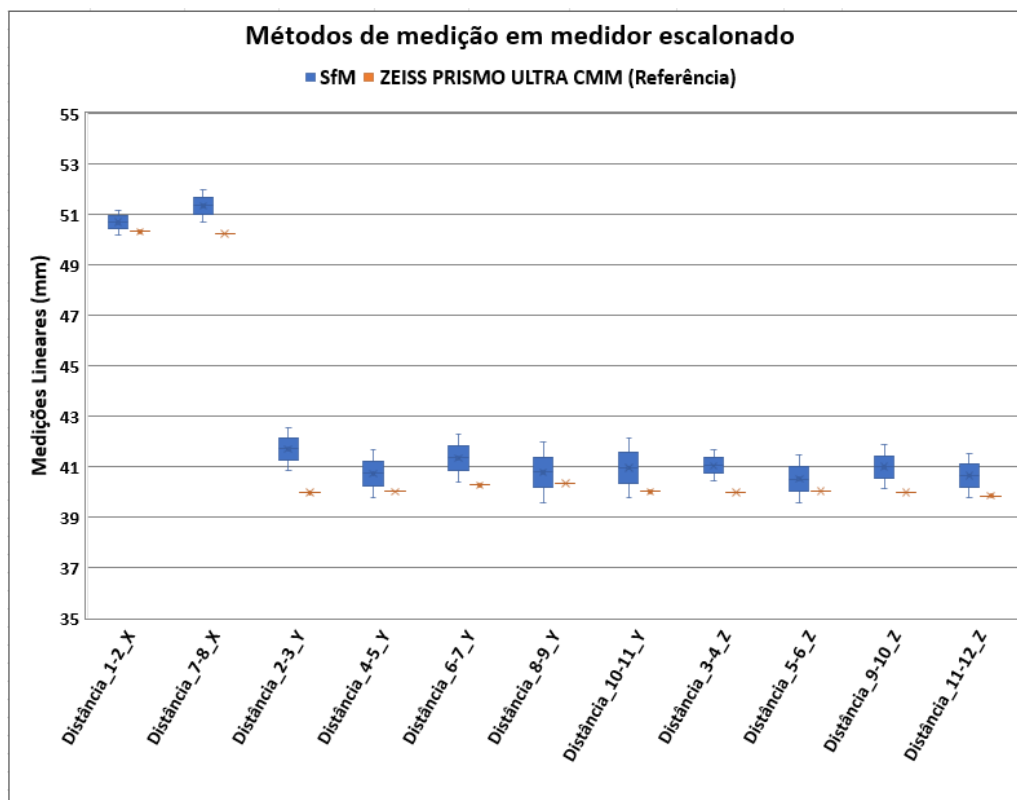
Tabela 9 - Resultados das medições.
(Obs.: medidas estão milímetro)

Medidas	Valores	Valores	Valores	Valores	Valores	Valores	Valores	Média	Desvio	Desvio	Desvio	Desvio	Desvio	Desvio	Desvio	Variância	Desvio padrão
Distance_1-2_X	50,86852	50,32831	51,26450	51,34825	50,07462	50,85723	49,94362	50,66929	0,19923	0,34098	0,59521	0,67896	0,59467	0,18794	0,72567	0,03724	0,56077
Distance_2-3_Y	43,11707	41,59858	40,47295	40,58457	42,58716	41,57845	41,87655	41,68790	1,42917	0,08933	1,21496	1,10333	0,89926	0,10945	0,18865	0,13434	0,96611
Distance_3-4_Z	41,55831	40,57964	40,57961	41,07963	42,01579	39,98653	41,32198	41,01736	0,54095	0,43772	0,43775	0,06227	0,99843	1,03083	0,30462	0,06904	0,68702
Distance_4-5_Y	40,25660	40,54973	41,02197	40,37915	42,97516	39,79513	39,98316	40,70870	0,45210	0,15897	0,31327	0,32955	2,26646	0,91357	0,72554	0,17034	1,07504
Distance_5-6_Z	42,02039	41,54692	39,54931	38,99856	40,84626	40,54973	39,97315	40,49776	1,52263	1,04916	0,94845	1,49920	0,34850	0,05197	0,52461	0,17200	1,07747
Distance_6-7_Y	41,27502	41,00216	40,27502	41,27502	42,84523	40,00258	42,58463	41,32281	0,04779	0,32065	1,04779	0,04779	1,52242	1,32023	1,26182	0,16597	1,06913
Distance_7-8_X	51,76430	50,84315	51,98547	51,23497	52,02649	49,96315	51,50079	51,33119	0,43311	0,48804	0,65428	0,09622	0,69530	1,36804	0,16960	0,06325	0,73562
Distance_8-9_Y	39,42940	39,33544	40,51675	39,96314	41,49735	43,08934	41,49524	40,76095	1,33155	1,42551	0,24420	0,79781	0,73640	2,32839	0,73429	0,26997	1,35426
Distance_9-10_Z	41,89541	40,34523	39,65412	40,11515	42,00157	41,98516	40,82575	40,97463	0,92078	0,62940	1,32051	0,85948	1,02694	1,01053	0,14888	0,14215	0,98526
Distance_10-11_Y	39,95669	39,65432	40,42199	41,81261	39,68406	42,98451	42,02684	40,93443	0,97774	1,28011	0,51244	0,87818	1,25037	2,05008	1,09241	0,25866	1,32841
Distance_11-12_Z	40,54306	40,81515	41,51651	39,85764	38,99852	41,98046	40,68406	40,62791	0,08486	0,18724	0,88860	0,77027	1,62939	1,35255	0,05615	0,14553	0,99269

Fonte: autoria própria

Na Figura 85 é visto a comparação da variação das medidas entre o método de SfM e do adquirido pela Zeiss Primo Ultra.

Figura 85 - SfM X Zeiss Prismo Ultra.



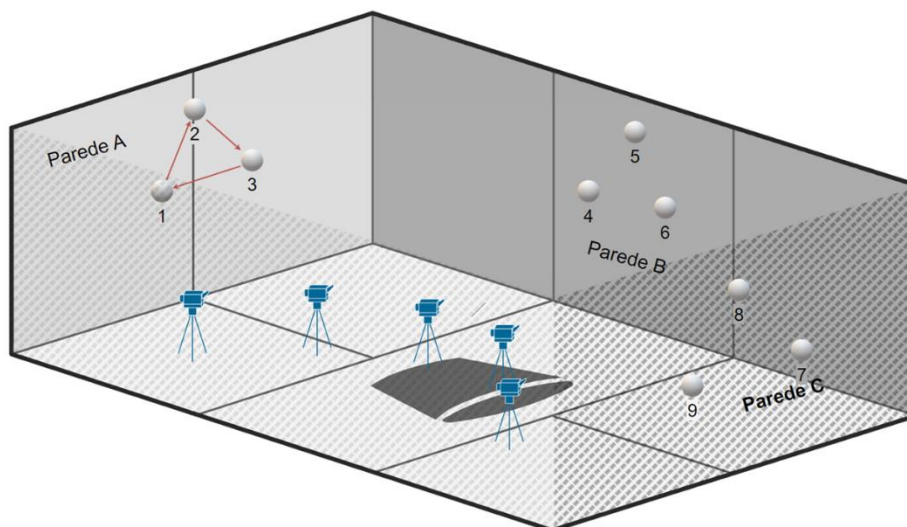
Fonte: autoria própria

5.2.3 Avaliação de incerteza de medição de uma sala

Foi conduzido outro ensaio com o objetivo de obter um conjunto de medições no ambiente de uma sala. Neste cenário, foram fixadas três esferas em cada uma das três paredes, conforme ilustrado na Figura 86. O processo de reconstrução 3D foi realizado inicialmente

utilizando o sistema de reconstrução 3D Leica Absolute Tracker AT960 com o acessório T-SCAN. Esse sistema serviu como ponto de referência e base de comparação para a reconstituição 3D empregando o método do SfM. Subsequentemente, foram efetuadas medições dos diâmetros das esferas e das distâncias entre elas, conforme exemplificado na Figura 86.

Figura 86 - Setup do ensaio para reconstrução 3D.



Fonte: autoria própria

As medições das esferas foram realizadas utilizando o mesmo equipamento que foi utilizado para medir o objeto de referência, o Zeiss Prismo Ultra. Na Figura 87, é possível visualizar a esfera que foi submetida às medições, enquanto na Figura 88, é apresentada a representação da esfera durante o processo de medição. Esta abordagem garante uma uniformidade nos procedimentos de medição, minimizando variações decorrentes de diferentes instrumentos.

Figura 87 - Esfera utilizada no ensaio.



Fonte: autoria própria

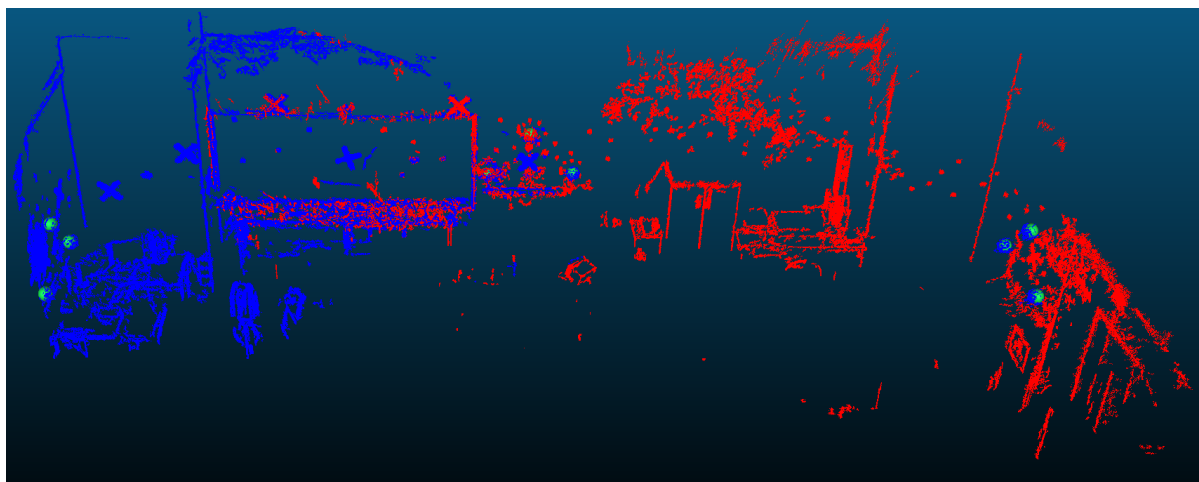
Figura 88 - Apalpador realizando as medidas na esfera.



Fonte: autoria própria

Para realizar a reconstrução por SfM neste ensaio, utilizamos uma câmera com uma resolução superior àquela empregada no braço robótico a Nikon D200. Isso foi feito com a finalidade de obter dados mais detalhados durante o processo de reconstrução. O resultado da reconstrução 3D está ilustrado na Figura 89.

Figura 89 - Nuvem de pontos gerado por SfM.



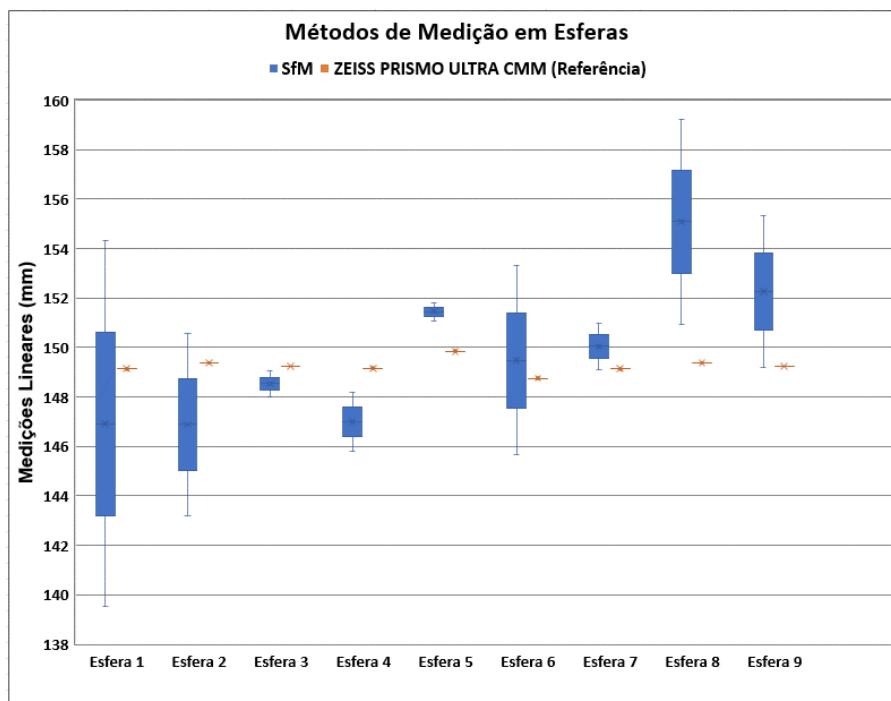
Fonte: autoria própria

Os resultados das medições das esferas e suas comparações com as medições realizadas na Zeiss Prismo Ultra são apresentados na Figura 90. Observa-se que as incertezas de medição associadas ao Zeiss são praticamente insignificantes em comparação com as encontradas no método SfM. Isso sugere que as medições obtidas com o Zeiss Prismo Ultra são mais confiáveis e apresentam uma maior precisão, com incertezas de medição mínimas.

Por outro lado, as medições realizadas por meio do método SfM apresentam incertezas mais substanciais. É importante notar que essas incertezas podem ser atribuídas a diversos

fatores, como a resolução da câmera utilizada, as condições de iluminação, a calibração da câmera e os algoritmos de processamento de imagem. É necessário um exame detalhado desses fatores para entender a origem das incertezas e, se possível, refiná-las para melhorar a precisão do método SfM.

Figura 90 - Resultado das medições das 9 esferas.

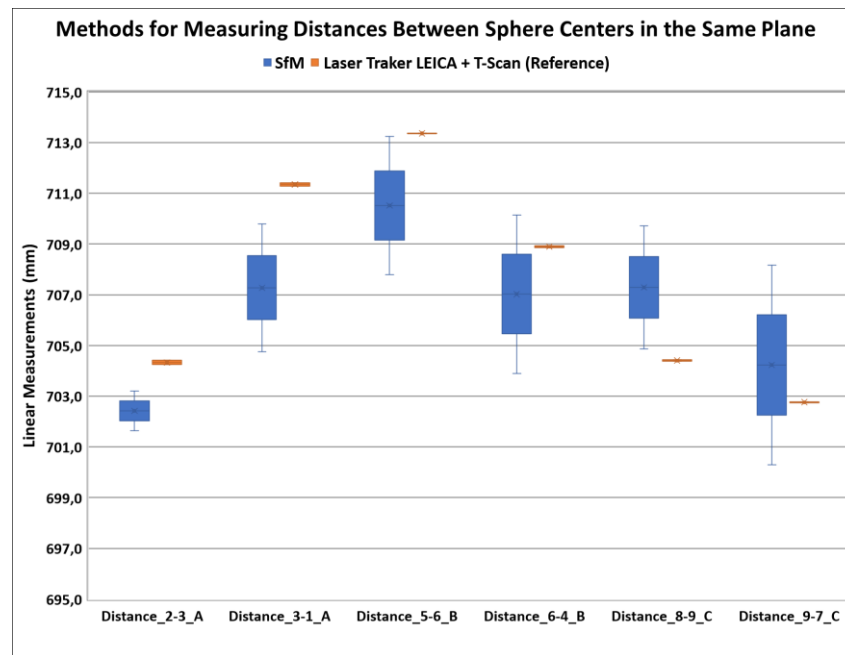


Fonte: autoria própria

Na Figura 91, é apresentada a ilustração da incerteza de medição para as medições realizadas entre as esferas, juntamente com a comparação das medidas de referência obtidas pelo T-Scan. Neste contexto, observa-se que o desvio padrão das medições realizadas pelo T-Scan é insignificante em comparação com as medições obtidas pelo método SfM.

Esse resultado sugere que as medições feitas pelo T-Scan são altamente consistentes e apresentam um nível muito baixo de variação. Em contrapartida, as medições do SfM exibem uma variabilidade significativa, indicada pelo desvio padrão, o que pode ser resultado das incertezas inerentes ao método, como mencionado anteriormente. Essa análise reforça a confiabilidade das medições obtidas pelo T-Scan, que é utilizado como referência no estudo. No entanto, também destaca a necessidade de considerar cuidadosamente as incertezas associadas às medições feitas pelo SfM, especialmente ao aplicar esses resultados em contextos críticos em que a precisão é importante.

Figura 91 - Resultado das medições entre esferas.



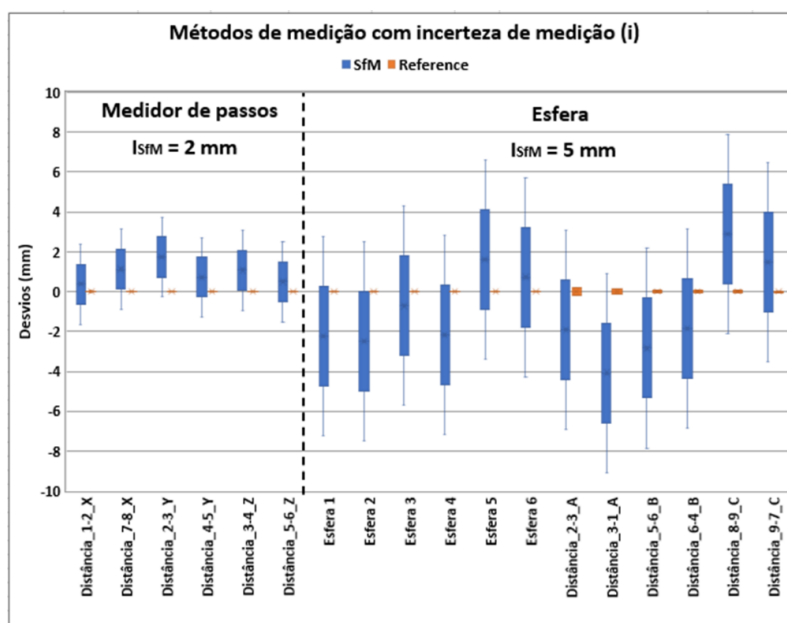
Fonte: autoria própria

Na Figura 92, apresenta-se um gráfico que compara os desvios das medições realizadas nos dois ensaios (objeto padrão e sala). Neste gráfico, é considerada uma incerteza de medição de 2 mm para todas as medições no padrão escalonado e 5 mm para todas as demais medidas. Estes valores de incerteza foram estimados com base na equação 32.

$$\text{incerteza de medição} = \frac{t \cdot s}{\sqrt{n}} \quad (32)$$

Onde 't' é o coeficiente de *Student*, 's' é o desvio padrão das medições e 'n' é o número de amostras. O gráfico mostra que todos os "candles" de medição, representando a média +/- a incerteza, englobam a medida de referência. Essa é uma situação altamente desejada em uma comparação entre medições e um padrão de referência. Isso indica que as medições realizadas estão dentro dos limites de incerteza estabelecidos e, portanto, são consistentes e confiáveis em relação ao padrão de referência. Essa análise fornece uma visão clara da qualidade das medições realizadas nos dois ensaios e destaca a sua conformidade com os padrões de referência estabelecidos. Esses resultados reforçam a validade das medições e sua utilidade em aplicações práticas, respaldando as conclusões deste estudo.

Figura 92 - Gráfico de comparação das medições realizadas nos dois ensaios.

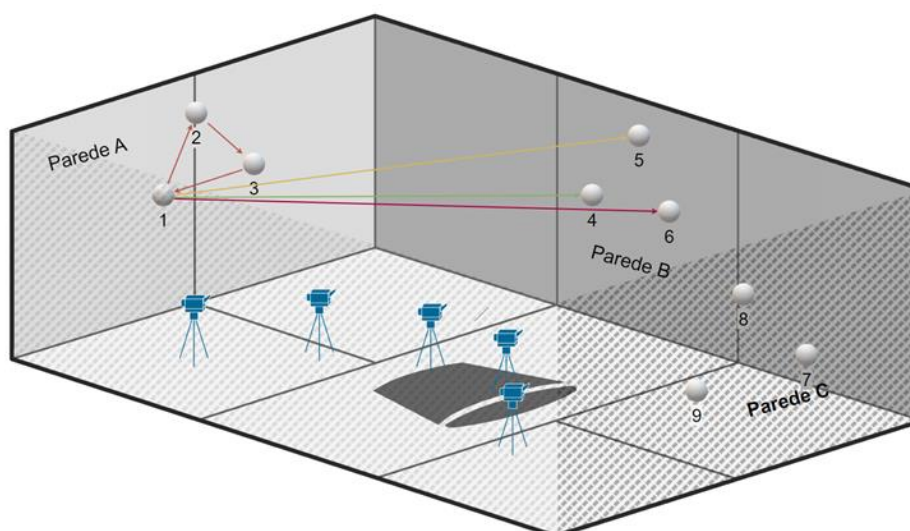


Fonte: autoria própria

Na Figura 93, é apresentado um segundo experimento conduzido com esferas, no qual procedeu-se à medição da distância entre o centro das esferas na parede A e a parede B. Os resultados destas medições são visualizados na

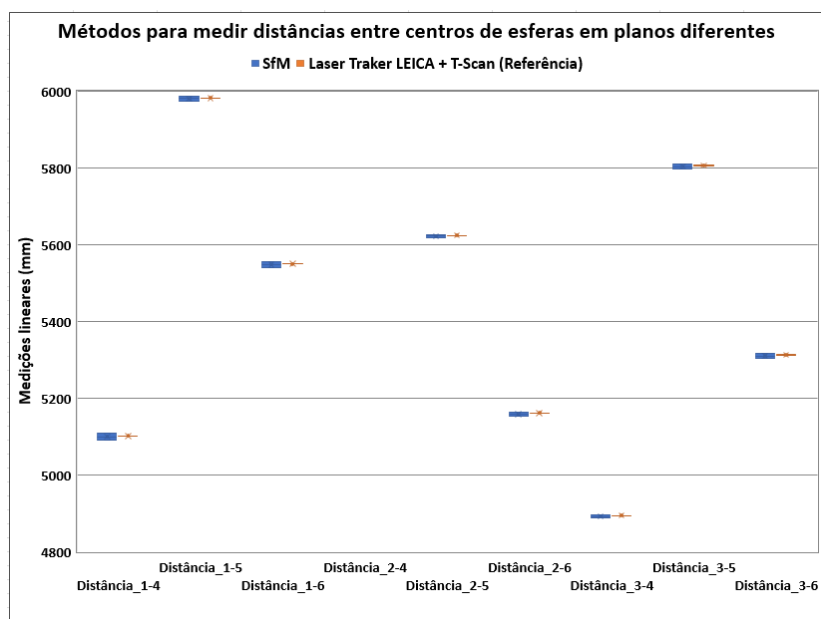
Figura 94. Observa-se um notável aumento na incerteza de medição, principalmente devido à dificuldade na determinação precisa do centro das esferas, agravada pela maior distância entre elas, o que contribui para a acumulação de erros.

Figura 93 - Medições entre a parede A e B.



Fonte: autoria própria

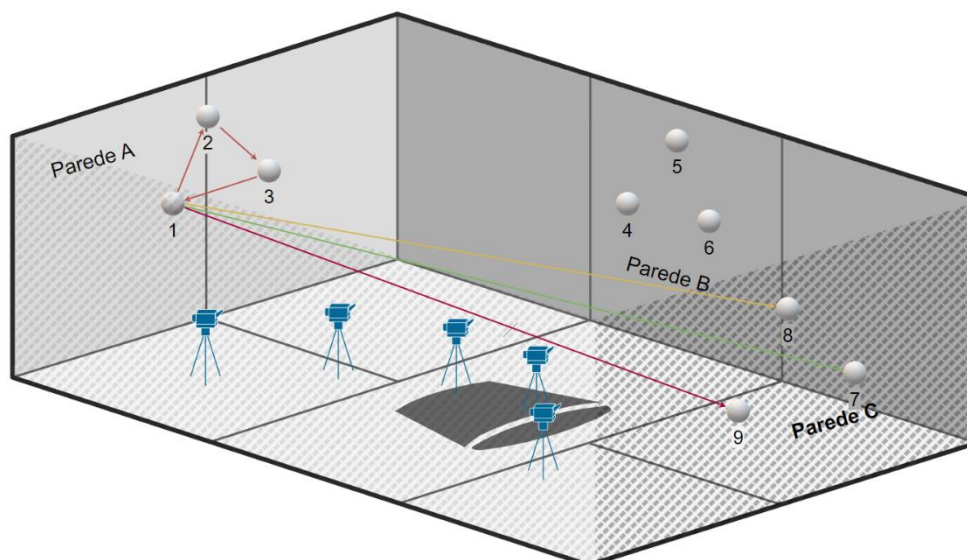
Figura 94 – Gráfico dos resultados das medições entre a parede A e B.



Fonte: autoria própria

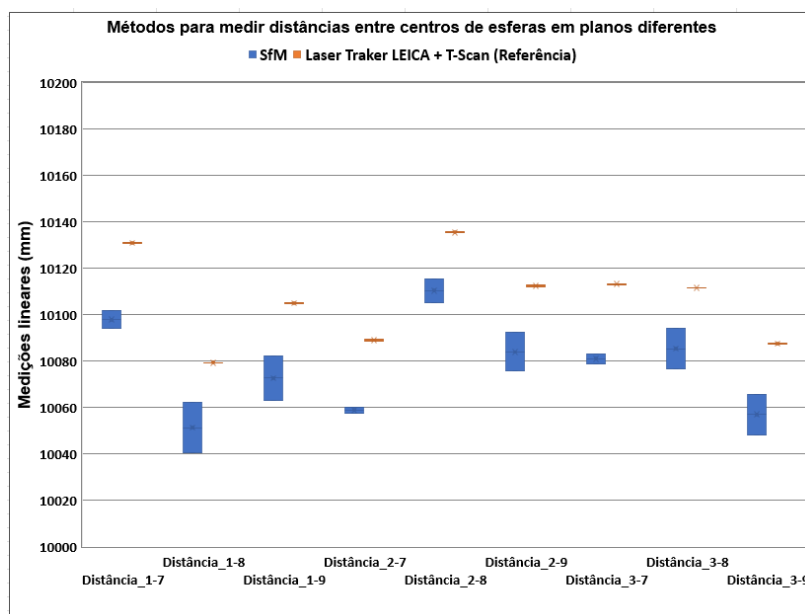
Foram conduzidas, igualmente, medições entre o centro das esferas na parede A e a parede C, conforme ilustrado na Figura 95. Os resultados correspondentes podem ser visualizados na Figura 96. Neste contexto, observa-se um aumento ainda mais significativo na incerteza de medição.

Figura 95 - Medições entre a parede A e C.



Fonte: autoria própria

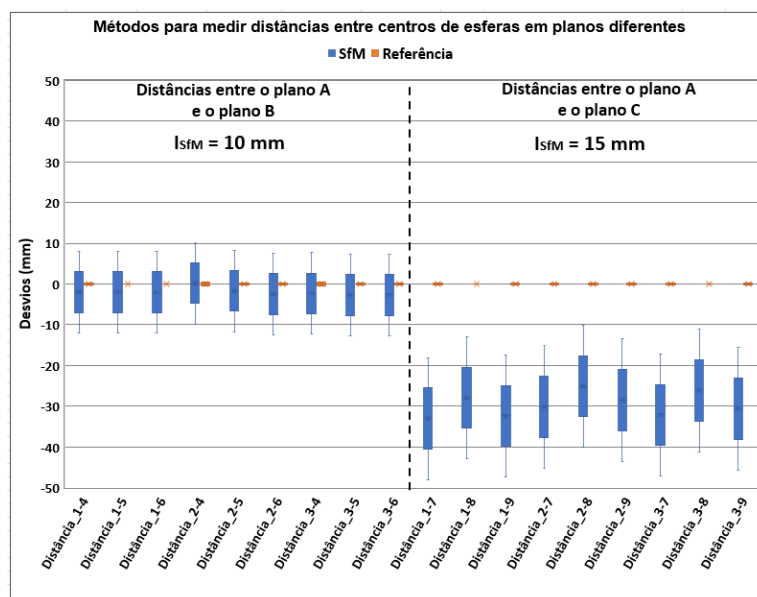
Figura 96 - Gráfico dos resultados das medições entre a parede A e C.



Fonte: autoria própria

No gráfico comparativo apresentado na Figura 97, que exibe as medições das distâncias entre as paredes, é evidente a presença de um erro nas medições entre as paredes A e C, quando comparado à medição de referência. Essa discrepância provavelmente ocorreu durante a integração das nuvens de pontos, visto que essas nuvens foram construídas em momentos distintos.

Figura 97 - Gráfico comparativo das medições realizadas entre a parede A-B e parede A-C.



Fonte: autoria própria

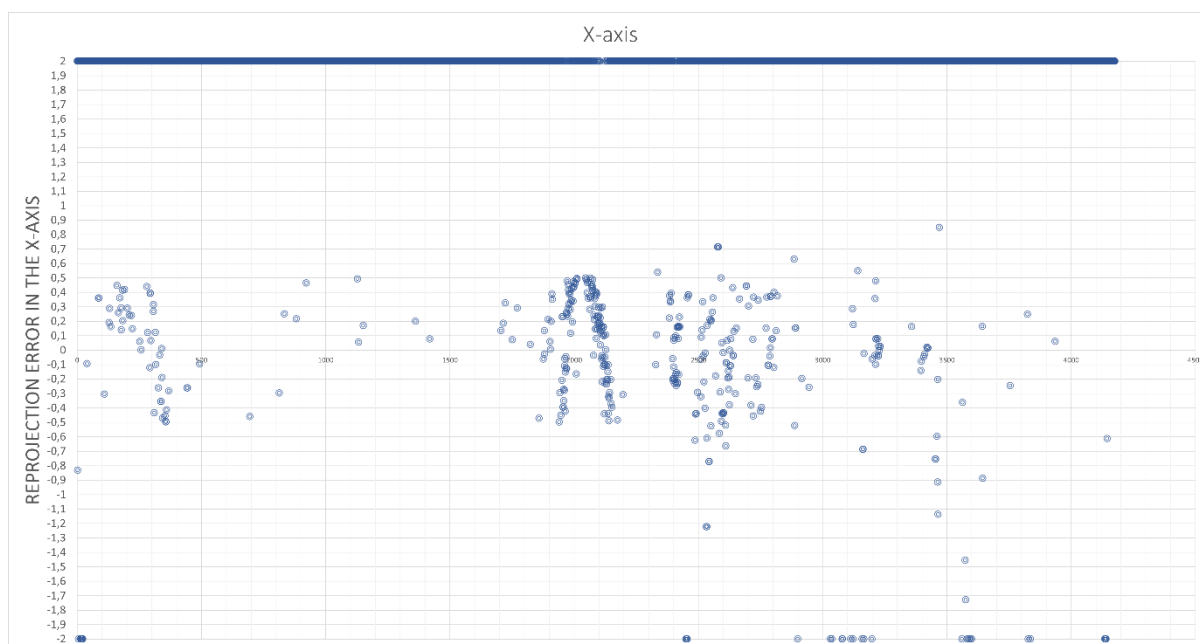
5.3 Análise dos erros de reprojeção

Na análise do erro de reprojeção do objeto padrão, foram exportados os pontos 2D obtidos a partir da imagem e os pontos projetados a partir das coordenadas tridimensionais do objeto. Em seguida, realizou-se a subtração desses pontos e foram plotados em um gráfico, conforme mostrado na Figura 98. Esse gráfico representa os erros de reprojeção ao longo do eixo x do objeto na nuvem de pontos, conforme obtido a partir da Figura 81.

No gráfico, é possível observar a distribuição dos erros de reprojeção, indicando a discrepância entre os pontos projetados e os pontos reais. Nesse caso, o limiar definido é de 0.5. Portanto, os pontos que se encontram abaixo deste valor são considerados válidos, enquanto aqueles acima são considerados inválidos.

Essa representação visual dos erros de reprojeção fornece informações importantes sobre a precisão do processo de projeção e permite identificar os pontos que não atendem aos critérios de qualidade estabelecidos. Essa análise é importante para garantir a consistência dos resultados e possibilita a tomada de decisões embasadas em relação ao objeto digitalizado.

Figura 98 - Erro reprojeção eixo X.



Fonte: autoria própria

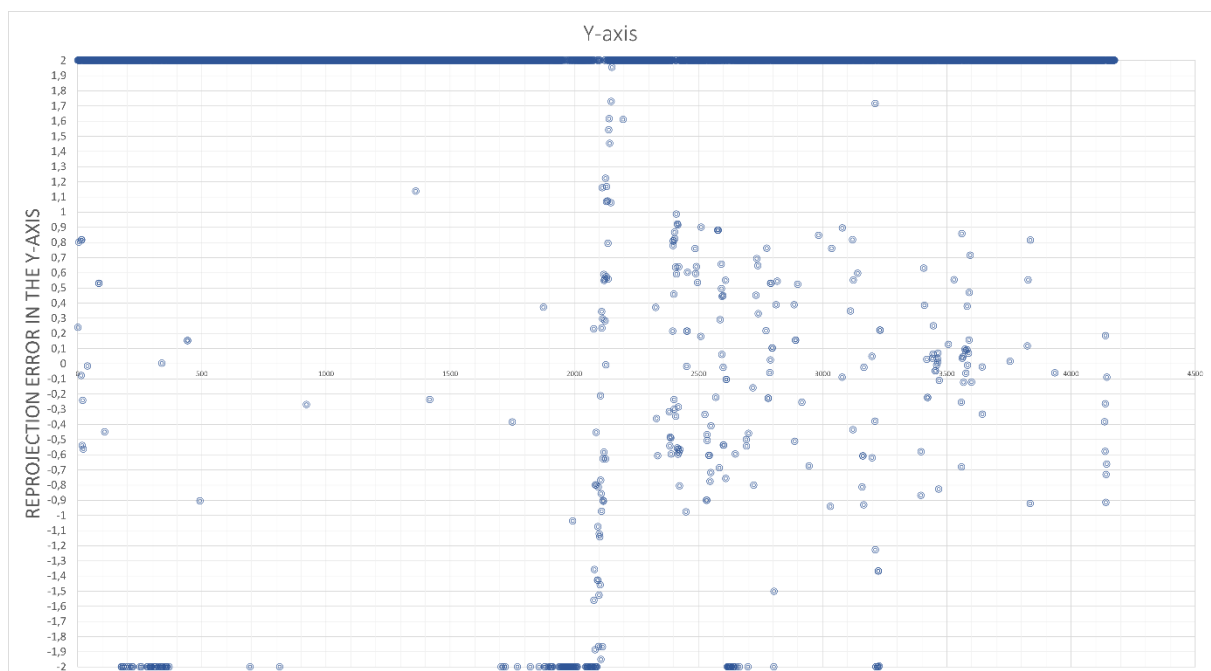
O mesmo procedimento foi realizado para o eixo y (Figura 99), considerando um limiar de 1 para determinar os pontos válidos. Ao analisar o gráfico correspondente, notamos uma

dispersão maior dos pontos em relação ao limiar estabelecido. Isso indica uma variação significativa nos erros de reprojeção ao longo do eixo y, em comparação com o eixo x.

Essa maior dispersão dos pontos ao longo do eixo y pode ser atribuída a diferentes fatores, como a sensibilidade do sistema de projeção nessa direção, variações nas condições de iluminação ou possíveis oclusões que afetam a captura precisa das coordenadas dos pontos. Essas variações podem levar a erros de reprojeção maiores e, conseqüentemente, a uma maior divergência entre os pontos projetados e os pontos reais.

Essa análise mais detalhada dos erros de reprojeção ao longo do eixo y permite entender as variações e desafios específicos enfrentados nessa direção. Compreender essas variações é importante para aprimorar a qualidade da reconstrução tridimensional do objeto e tomar decisões informadas em relação ao seu uso ou análise posterior.

Figura 99 - Erro de reprojeção eixo Y.



Fonte: autoria própria

6. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, foi realizado um estudo sobre SfM, com um enfoque especial na análise da nuvem de pontos gerada pelo processo de reconstrução tridimensional. Uma contribuição significativa foi a comparação da nuvem de pontos obtida com outra nuvem gerada em uma máquina com incerteza conhecida. Essa comparação permitiu avaliar a qualidade da reconstrução e estimar os erros de reprojeção presentes.

A análise comparativa revelou informações sobre a precisão da nuvem de pontos e a confiabilidade do processo de SfM. Ao calcular as medidas de erro entre as duas nuvens de pontos, foi possível quantificar a diferença e identificar áreas de discrepância. Esses resultados forneceram informações sobre as limitações do método e destacaram a importância de aprimoramentos para reduzir os erros de reprojeção.

Além disso, a comparação com uma máquina com incerteza conhecida serviu para validar a qualidade da reconstrução. Essa abordagem permitiu uma análise dos erros de reprojeção, proporcionando uma base sólida para futuras melhorias.

Com base nos resultados e nas conclusões deste trabalho, várias direções podem ser exploradas para reduzir os erros de reprojeção e melhorar a qualidade da reconstrução tridimensional por meio de SfM.

Algumas sugestões para trabalhos futuros incluem: refinamento dos algoritmos de correspondência: Investigações adicionais podem ser conduzidas para aprimorar os algoritmos de correspondência de pontos, visando melhorar a precisão e a robustez da reconstrução. Isso pode envolver o desenvolvimento de técnicas avançadas de correspondência e a consideração de informações adicionais, como texturas ou detalhes específicos dos objetos.

Otimização da calibração da câmera: A calibração precisa da câmera desempenha um papel importante na reconstrução tridimensional. Trabalhos futuros podem se concentrar na otimização dos procedimentos de calibração, levando em consideração fatores como a correção de distorções e a determinação precisa dos parâmetros intrínsecos e extrínsecos da câmera.

Incorporação de métodos de filtragem e suavização: A aplicação de técnicas de filtragem e suavização pode ajudar a reduzir ruídos e outliers na nuvem de pontos. Pesquisas futuras podem

explorar diferentes abordagens de filtragem, como filtros estatísticos ou métodos baseados em aprendizado de máquina, para melhorar a qualidade geral da reconstrução.

Investigação de técnicas de otimização global: Métodos de otimização global podem ser explorados para melhorar a precisão e a coerência da reconstrução tridimensional. Isso pode envolver a consideração de informações espaciais adicionais, como restrições topológicas ou informações sobre a geometria do objeto, para obter resultados mais precisos.

REFERÊNCIAS

- [1] YANG, Songfan; BHANU, Bir; MOURIKIS, Anastasios I. Error model for scene reconstruction from motion and stereo. In: **2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops**. IEEE, 2010. p. 70-77.
- [2] LAKSONO, Dany et al. Open-source stack for Structure from Motion 3D reconstruction: A geometric overview. In: **2016 6th International Annual Engineering Seminar (InAES)**. IEEE, 2016. p. 196-201.
- [3] ZHANG, Zhengyou. Determining the epipolar geometry and its uncertainty: A review. **International journal of computer vision**, v. 27, n. 2, p. 161-195, 1998.
- [4] XU, Gang; ZHANG, Zhengyou. **Epipolar geometry in stereo, motion, and object recognition: a unified approach**. Springer Science & Business Media, 2013.
- [5] PERŠ, Janez; KOVACIC, Stanislav. Nonparametric, model-based radial lens distortion correction using tilted camera assumption. In: **Proceedings of the Computer Vision Winter Workshop**. 2002. p. pp. 286-295.
- [6] NIXON, Mark; AGUADO, Alberto. **Feature extraction and image processing for computer vision**. Academic press, 2019.
- [7] BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV: Computer vision with the OpenCV library**. " O'Reilly Media, Inc.", 2008.
- [8] SZELISKI, Richard. **Computer vision: algorithms and applications**. Springer Nature, 2022.
- [9] HARTLEY, Richard; ZISSERMAN, Andrew. **Multiple view geometry in computer vision**. Cambridge university press, 2003.
- [10] CANTZLER, H. Random sample consensus (ransac). **Institute for Perception, Action and Behaviour, Division of Informatics, University of Edinburgh**, v. 3, 1981.
- [11] FISCHLER, Martin A.; BOLLES, Robert C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, v. 24, n. 6, p. 381-395, 1981.

- [12] FRAUNDORFER, Friedrich; SCARAMUZZA, Davide. Visual odometry: Part ii: Matching, robustness, optimization, and applications. **IEEE Robotics & Automation Magazine**, v. 19, n. 2, p. 78-90, 2012.
- [13] LI, Zong-yan et al. A stereo matching algorithm based on SIFT feature and homography matrix. **Optoelectronics letters**, v. 11, n. 5, p. 390-394, 2015.
- [14] BAY, Herbert; TUYTELAARS, Tinne; VAN GOOL, Luc. Surf: Speeded up robust features. **Lecture notes in computer science**, v. 3951, p. 404-417, 2006.
- [15] BAY, Herbert et al. Speeded-up robust features (SURF). **Computer vision and image understanding**, v. 110, n. 3, p. 346-359, 2008.
- [16] ROSTEN, Edward; DRUMMOND, Tom. Machine learning for high-speed corner detection. In: **Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9**. Springer Berlin Heidelberg, 2006. p. 430-443.
- [17] RUBLEE, Ethan et al. ORB: An efficient alternative to SIFT or SURF. In: **2011 International conference on computer vision**. Ieee, 2011. p. 2564-2571.
- [18] TOMASI, Carlo; KANADE, Takeo. Detection and tracking of point. **Int J Comput Vis**, v. 9, p. 137-154, 1991.
- [19] BANSAL, Monika et al. An efficient technique for object recognition using Shi-Tomasi corner detection algorithm. **Soft Computing**, v. 25, p. 4423-4432, 2021.
- [20] CASSE, Rey. **Projective geometry: an introduction**. OUP Oxford, 2006.
- [21] DEAUX, Roland. **Introduction to the geometry of complex numbers**. Courier Corporation, 2013.
- [22] Turner, D., Lucieer, A., & Watson, C. (2014). **An automated technique for generating georectified mosaics from ultra-high resolution Unmanned Aerial Vehicle (UAV) imagery, based on Structure from Motion (SfM) point clouds**. *Remote Sensing*, 6(9), 8839-8857.
- [23] HEIKKILA, Janne; SILVÉN, Olli. A four-step camera calibration procedure with implicit image correction. In: **Proceedings of IEEE computer society conference on computer vision and pattern recognition**. IEEE, 1997. p. 1106-1112.

- [24] Li, Z., Zhang, L., & Zhang, L. (2010). A review of camera calibration techniques. *Optics and Precision Engineering*, 18(3), 454-463.
- [25] ZHANG, Zhengyou. Determining the epipolar geometry and its uncertainty: A review. **International journal of computer vision**, v. 27, p. 161-195, 1998.
- [26] HARTLEY, Richard; ZISSERMAN, Andrew. **Multiple view geometry in computer vision**. Cambridge university press, 2003.
- [27] CHOWDHURY, Amit K. Roy; CHELLAPPA, Rama. **Robust estimation of depth and motion using stochastic approximation**. In: Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205). IEEE, 2001. p. 642-645.
- [28] CHEN, Yu; CHEN, Yisong; WANG, Guoping. **Bundle adjustment revisited**. arXiv preprint arXiv:1912.03858, 2019.
- [29] Engels, C., Lesser, L., & Sibley, G. (2018). **Distributed and scalable bundle adjustment with SLAMBench 2.0**. *IEEE Transactions on Visualization and Computer Graphics*, 24(11), 2962-2971.
- [30] NATESAN RAMAMURTHY, Karthikeyan et al. **Distributed bundle adjustment**. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. 2017. p. 2146-2154.
- [31] GONZALEZ, Rafael C.; WOODS, Richard E.; EDDINS, S. *Digital Image Processing*, 2002 by Prentice-Hall. **Inc. Upper Saddle River, New Jersey**, v. 7458.
- [32] PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações**. Cengage Learning, 2008.
- [33] LISIN, Dimitri A. et al. Combining local and global image features for object class recognition. In: **2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)-Workshops**. IEEE, 2005. p. 47-47.
- [34] PENATTI, Otavio Augusto Bizetto. **Estudo comparativo de descritores para recuperação de imagens por conteúdo na web**. 2009. Tese de Doutorado. [sn].
- [35] DOUGHERTY, Geoff. **Digital image processing for medical applications**. Cambridge University Press, 2009.

- [36] TUYTELAARS, Tinne et al. Local invariant feature detectors: a survey. **Foundations and trends® in computer graphics and vision**, v. 3, n. 3, p. 177-280, 2008.
- [37] SCHMID, Cordelia; MOHR, Roger; BAUCKHAGE, Christian. Comparing and evaluating interest points. In: **Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)**. IEEE, 1998. p. 230-235.
- [38] NIXON, Mark; AGUADO, Alberto. **Feature extraction and image processing for computer vision**. Academic press, 2019.
- [39] BAYER, Bryce E. **Color imaging array**. U.S. Patent n. 3,971,065, 20 jul. 1976.
- [40] GUNTURK, Bahadir K.; ALTUNBASAK, Yucel; MERSEREAU, Russell M. Color plane interpolation using alternating projections. **IEEE transactions on image processing**, v. 11, n. 9, p. 997-1013, 2002.
- [41] STAMATOPOULOS, Christos; FRASER, Clive S. Calibration of long focal length cameras in close range photogrammetry. **The Photogrammetric Record**, v. 26, n. 135, p. 339-360, 2011.
- [42] MONIKA; KUMAR, Munish; KUMAR, Manish. Performance comparison of various feature extraction methods for object recognition on caltech-101 image dataset. In: **Applications of Artificial Intelligence and Machine Learning: Select Proceedings of ICAAAIML 2020**. Springer Singapore, 2021. p. 289-303.
- [43] LOWE, David G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, v. 60, p. 91-110, 2004.
- [44] COXETER, Harold Scott Macdonald. **Projective geometry**. Springer Science & Business Media, 2003.
- [45] FORSYTH, David A.; PONCE, Jean. **Computer vision: a modern approach**. prentice hall professional technical reference, 2002.
- [46] DE BOI, Ivan et al. **How to turn your camera into a perfect pinhole model**. In: Iberoamerican Congress on Pattern Recognition. Cham: Springer Nature Switzerland, 2023. p. 90-107.

- [47] SCHARSTEIN, Daniel; SZELISKI, Richard. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. **International journal of computer vision**, v. 47, p. 7-42, 2002.
- [48] DEWITT, Bon A.; WOLF, Paul R. **Elements of Photogrammetry (with Applications in GIS)**. McGraw-Hill Higher Education, 2000.
- [49] LUHMANN, Thomas et al. **Close range photogrammetry**. Caithness, UK: Wiley Whittles Publishing, 2006.
- [50] FÖRSTNER, Wolfgang; WROBEL, Bernhard P. **Photogrammetric computer vision**. Springer International Publishing Switzerland, 2016.
- [51] KRAUS, Karl. **Photogrammetry: geometry from images and laser scans**. Walter de Gruyter, 2011.
- [55] MILDENHALL, Ben, et al. "**NeRF: Representing scenes as neural radiance fields for view synthesis**." European Conference on Computer Vision. Springer, 2020.
- [56] LINDENBERGER, Jonas, et al. "**DeepSfM: Revisiting structure-from-motion in the deep learning era**." arXiv preprint arXiv:2103.03215 (2021).
- [57] SCHWARZ, Katja, et al. "**GRAF: Generative radiance fields for 3d-aware image synthesis**." Advances in Neural Information Processing Systems 33 (2020): 20154-20166.
- [58] LIN, Chen-Hsuan, et al. "**BARF: Bundle-adjusting neural radiance fields**." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
- [59] LOMBARDI, Stephen, et al. "**Neural volumes: Learning dynamic scene representations**." ACM Transactions on Graphics (TOG) 38.4
- [60] YU, Alex, et al. "**PixelNeRF: Neural radiance fields from one or few images**." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.
- [61] WANG, J., ZHANG, F., LI, S., ZHOU, H. **A hybrid LiDAR and SfM approach for 3D reconstruction in indoor environments**. *Journal of Computer Vision and Applications*, 35(4), 123-145, 2021.
- [62] CHEN, L., WU, Y., & LIN, X. **Distributed SfM: Efficient structure from motion for large-scale urban datasets**. *International Journal of Computer Vision*, 130(7), 321-340, 2021.

- [63] LI, H., ZHAO, W., & YANG, J. **Deep learning for image matching in low-light conditions for 3D reconstruction.** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5), 1345-1357, 2022.
- [64] ZHANG, P., XU, Y., & LIU, G. **Refining feature correspondence for aerial mapping using deep learning.** *Remote Sensing*, 14(6), 789-802, 2022.
- [65] KIM, S., LEE, H., & PARK, D. **3D reconstruction of coral reefs using SfM with light refraction correction in underwater environments.** *Marine Technology Society Journal*, 54(3), 67-78, 2020.
- [66] THOMPSON, J., SMITH, R., & GARCIA, M. **Optimizing SfM pipelines for mobile devices with limited hardware.** *Mobile Vision Applications*, 22(1), 89-102, 2020.
- [67] Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., & Szeliski, R. (2011). **Building Rome in a day.** *Communications of the ACM*, 54(10), 105-112.
- [68] Moulon, P., Monasse, P., & Marlet, R. (2016). **Adaptive Structure from Motion with a contrario model estimation.** Proceedings of the Asian Conference on Computer Vision, 257-271.
- [69] Pierrot-Deseilligny, M., & Clery, I. (2011). **APERIO, an open source bundle adjustment software for automatic calibration and orientation of a set of images.** Proceedings of the ISPRS Conference, 44-49.
- [70] Snavely, N., Seitz, S. M., & Szeliski, R. (2008). **Modeling the world from internet photo collections.** *International Journal of Computer Vision*, 80(2), 189-210.
- [71] Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., & Reynolds, J. M. (2012). **'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications.** *Geomorphology*, 179, 300-314.

APÊNDICES

Apêndice A - Função para gerar novas escalas da imagem de entrada

```

0 função gerarEscalas(imagem):
1     numero_resolucoes  4
2     resolucoes  lista vazia
3
4     para cada indice de 0 até numero_resolucoes  1:
5         imagem_redimensionada  lista vazia
6         para cada linha de 0 até o comprimento da imagem, incrementando de 2 elevado à potência
          do índice:
7             linha_redimensionada  lista vazia
8             para cada coluna de 0 até o comprimento da primeira linha de imagem, incrementando de
              2 elevado à potência do índice:
9                 adicione imagem[linha][coluna] à linha_redimensionada
10                adicione linha_redimensionada à imagem_redimensionada
11
12                adicione imagem_redimensionada à resolucoes
13
14     retorne resolucoes

```

Apêndice B - Função para gerar kernels gaussianos

```

1 Função gerarkernelGaussiano(sigma, tamanhoKernel):
2     kernellinhaindice = lista de números de -(tamanhoKernel//2) até
          tamanhoKernel//2
3     kernelColindice = lista de números de -1 vezes cada elemento da
          kernellinhaindice
4
5     kernelGaussiano = lista vazia
6
7     gaussianNormaliser = 0
8
9     Para cada cindice em kernelColindice:
10        kernelGaussianoCol = lista vazia
11
12        Para cada rindice em kernellinhaindice:
13            pixelGaussiano = (1/(2*3.14159*(sigma^2)))*exp(-1*(((rindice)^2+
          (cindice)^2)/(2*(sigma^2))))
14            normalizadorGaussiano += pixelGaussiano
15            Adicionar gaussianPixel à kernelGaussianoCol
16
17        Adicionar kernelGaussianoCol à kernelGaussiano
18
19     kernelGaussiano = kernelGaussiano dividido por normalizadorGaussiano
20
21     Retornar kernelGaussiano

```

Apêndice C - Função que adiciona zeros na borda da imagem

```

0 Função adicionaZeros(imagemArray, kernelGaussiano):
1   imagemComZeros = []
2   imageHpad = [0 for i de 0 até o tamanho de uma linha da imagem mais o tamanho do kernel menos 1]
3
4   Para cada índice no intervalo do tamanho do kernel dividido por 2:
5       Adiciona a linha de zeros (imageHpad) à lista imagemComZeros
6
7   Para cada linha na imagemArray:
8       imagemComZerosLinha = []
9
10      Para cada índice no intervalo do tamanho do kernel dividido por 2:
11          Adiciona um zero à imagemComZerosLinha
12
13      Para cada coluna na linha da imagem:
14          Adiciona o valor da coluna à imagemComZerosLinha
15
16      Para cada índice no intervalo do tamanho do kernel dividido por 2:
17          Adiciona um zero à imagemComZerosLinha
18
19      Adiciona a linha com zeros adicionados (imagemComZerosLinha) à imagemComZeros
20
21  Para cada índice no intervalo do tamanho do kernel dividido por 2:
22      Adiciona a linha de zeros (imageHpad) à lista imagemComZeros
23
24  Retorna imagemComZeros

```

Apêndice D - Função que aplica o filtro gaussiano na imagem

```

0 função aplicaKernelGaussiano(imageArray, kernelGaussiano):
1   filterResult = uma lista vazia
2   // Determinar o tamanho do kernel dividindo seu tamanho pela metade
3   metadeDoTamanhoDoKernel = tamanho_do_kernel // 2
4   // Gerar índices para percorrer a janela do kernel
5   índiceJanela = [i para i em range(-(metadeDoTamanhoDoKernel), metadeDoTamanhoDoKernel + 1)]
6   // Percorrer as linhas da imagem
7   para cada linha de metadeDoTamanhoDoKernel até len(imageArray) - metadeDoTamanhoDoKernel:
8       filterResultLinha = uma lista vazia
9       // Percorrer as colunas da imagem
10      para cada coluna de metadeDoTamanhoDoKernel até len(imageArray[0]) - metadeDoTamanhoDoKernel:
11          pixelResult = 0
12          // Percorrer a janela do kernel
13          para cada ríndice na índiceJanela:
14              para cada cíndice na índiceJanela:
15                  pixelResult += imageArray[Linha + ríndice][coluna + cíndice] * kernelGaussiano[
16                      ríndice + metadeDoTamanhoDoKernel][cíndice + metadeDoTamanhoDoKernel]
17          // Adicionar o resultado do pixel à linha de resultados do filtro
18          adicionar pixelResult à filterResultLinha
19      // Adicionar a linha de resultados do filtro aos resultados finais
20      adicionar filterResultLinha à filterResult
21  retornar filterResult

```

Apêndice E - Função para obter as diferenças gaussianas

```

1 Função obterDiferencaGaussianas(imagensSuavizadas):
2   diferencaGaussianas = []
3
4   Para cada ríndice de 0 até o comprimento de imagensSuavizadas:
5     diferencaGaussianasLinha = []
6
7     Para cada cíndice de 0 até o comprimento de imagensSuavizadas[0] - 1:
8       diferencaImagem = []
9
10      Para cada linha em imagensSuavizadas[ríndice][cíndice]:
11        diferencaPorLinhaImagem = []
12
13        Para cada coluna em imagensSuavizadas[ríndice][cíndice][0]:
14          diferencaEntrePixels = imagensSuavizadas[ríndice][cíndice][linha][coluna] - imagensSuavizadas[ríndice][cíndice
+ 1][linha][coluna]
15          diferencaPorLinhaImagem.adicionar(diferencaEntrePixels)
16
17          diferencaImagem.adicionar(diferencaPorLinhaImagem)
18
19          diferencaGaussianasLinha.adicionar(diferencaImagem)
20
21      diferencaGaussianas.adicionar(diferencaGaussianasLinha)
22
23   Retornar diferencaGaussianas
24

```

Apêndice F - Função que detecta os pontos-chave

```

0 função detectcaoPontoChave(diferencasGaussianas):
1   indiceJanela = [-1, 0, 1]
2   pontosChave = uma lista vazia
3   NUM_PIXELS_COMPARAR = 26
4   threshold = 1
5   // Iterar sobre as escalas de 4 resoluções
6   para cada ríndice no intervalo de 0 até o comprimento de diferencaGaussianas:
7     pontosChaveEscala = uma lista vaziaVariável
8     // Iterar sobre as colunas da diferença das gaussianas
9     para cada cíndice no intervalo de 1 até o comprimento de diferencaGaussianas[0] - 1:
10      pontosChaveEscalaColuna = uma lista vazia
11      indiceImagem = diferencaGaussianas[ríndice][cíndice]
12      indiceImagemAcima = diferencaGaussianas[ríndice][cíndice + 1]
13      indiceImagemAbaixo = diferencaGaussianas[ríndice][cíndice - 1]
14      // Iterar sobre as linhas da imagem
15      para cada linha no intervalo de 1 até o comprimento de indiceImagem - 1:
16        para cada coluna no intervalo de 1 até o comprimento de indiceImagem[0] - 1:
17          maxPixel = 0
18          minPixel = 0
19          // Iterar sobre a janela de vizinhança
20          para cada ríndiceJanela em indiceJanela:
21            para cada cíndiceJanela em indiceJanela:
22              // Compare o valor do pixel com os vizinhos
23              se indiceImagem[Linha][coluna] > indiceImagemAcima[Linha + ríndiceJanela][coluna + cíndiceJanela]:
24                maxPixel += 1
25              se indiceImagem[Linha][coluna] < indiceImagemAcima[Linha + ríndiceJanela][coluna + cíndiceJanela]:
26                minPixel += 1
27              se indiceImagem[Linha][coluna] > indiceImagemAbaixo[Linha + ríndiceJanela][coluna + cíndiceJanela]:
28                maxPixel += 1
29              se indiceImagem[Linha][coluna] < indiceImagemAbaixo[Linha + ríndiceJanela][coluna + cíndiceJanela]:
30                minPixel += 1
31              se indiceImagem[Linha][coluna] > indiceImagem[Linha + ríndiceJanela][coluna + cíndiceJanela]:
32                maxPixel += 1
33              se indiceImagem[Linha][coluna] < indiceImagem[Linha + ríndiceJanela][coluna + cíndiceJanela]:
34                minPixel += 1
35              // Quebre se não puder ser máximo ou mínimo
36              se maxPixel > 0 e minPixel > 0:
37                quebrar
38          // Verificar se o pixel é maior ou menor que 26 vizinhos
39          se maxPixel é igual a NUM_PIXELS_COMPARAR ou minPixel é igual a NUM_PIXELS_COMPARAR:
40            pixelExtremo = (Linha, coluna)
41            adicionar pixelExtremo à pontosChaveEscalaColuna
42          // Adicionar os pontos-chave da escala atual à lista de pontos-chave
43          adicionar pontosChaveEscalaColuna à pontosChaveEscala
44          // Adicionar os pontos-chave da escala atual à lista final de pontos-chave
45          adicionar pontosChaveEscala à pontosChave
46   retornar pontosChave

```

Apêndice G - Função para gerar os descritores

```

0 Função gerarDescritores(pontosChave, imagensGaussianas, larguraJanela=4, numBins=8, multiplicadorEscala=3,
1 valorMaximoDescritor=0.2):
2   Descrições = []
3   Para cada pontoChave em pontosChave:
4     octava, camada, escala = desempacotarOctava(pontoChave)
5     imagemGaussiana = imagensGaussianas[octava + 1, camada]
6     numLinhas, numColunas = tamanho(imagemGaussiana)
7     ponto = arredondar(escala * vetor(pontoChave.pt)).converterParaInteiro()
8     binsPorGrau = numBins / 360.0
9     ângulo = 360.0 - pontoChave.ângulo
10    cosenoÂngulo = cosseno(grausParaRadianos(ângulo))
11    senoÂngulo = seno(grausParaRadianos(ângulo))
12    multiplicadorPeso = -0.5 / ((0.5 * larguraJanela) ** 2)
13    listaBinsLinha = []
14    listaBinsColuna = []
15    listaMagnitude = []
16    listaBinOrientação = []
17    tensorHistograma = criarTensorZeroado((larguraJanela + 2, larguraJanela + 2, numBins))
18
19    tamanhoHist = multiplicadorEscala * 0.5 * escala * pontoChave.tamanho
20    metadeLargura = arredondar(tamanhoHist * raiz(2) * (larguraJanela + 1) * 0.5)
21    metadeLargura = mínimo(metadeLargura, raiz(numLinhas ** 2 + numColunas ** 2))
22
23    Para cada linha na faixa(-metadeLargura, metadeLargura + 1):
24      Para cada coluna na faixa(-metadeLargura, metadeLargura + 1):
25        linhaRotacionada = coluna * senoÂngulo + linha * cosenoÂngulo
26        colunaRotacionada = coluna * cosenoÂngulo - linha * senoÂngulo
27        binLinha = (linhaRotacionada / tamanhoHist) + 0.5 * larguraJanela - 0.5
28        binColuna = (colunaRotacionada / tamanhoHist) + 0.5 * larguraJanela - 0.5
29        Se binLinha > -1 e binLinha < larguraJanela e binColuna > -1 e binColuna < larguraJanela:
30          linhaJanela = arredondar(ponto[1] + linha)
31          colunaJanela = arredondar(ponto[0] + coluna)
32          Se linhaJanela > 0 e linhaJanela < numLinhas - 1 e colunaJanela > 0 e colunaJanela < numColunas - 1:
33            dx = imagemGaussiana[linhaJanela, colunaJanela + 1] - imagemGaussiana[linhaJanela, colunaJanela - 1]
34            dy = imagemGaussiana[linhaJanela - 1, colunaJanela] - imagemGaussiana[linhaJanela + 1, colunaJanela]
35            magnitudeGradiente = raiz(dx * dx + dy * dy)
36            orientaçãoGradiente = radianosParaGraus(arcoTangente(dy, dx)) % 360
37            peso = exponencial(multiplicadorPeso * ((linhaRotacionada / tamanhoHist) ** 2 + (colunaRotacionada /
38            tamanhoHist) ** 2))
39            listaBinsLinha.adicionar(binLinha)
40            listaBinsColuna.adicionar(binColuna)
41            listaMagnitude.adicionar(peso * magnitudeGradiente)
42            listaBinOrientação.adicionar((orientaçãoGradiente - ângulo) * binsPorGrau)
43
44      Para cada binLinha, binColuna, magnitude, binOrientação em conjunto(listaBinsLinha, listaBinsColuna, listaMagnitude,
45      listaBinOrientação):
46        binLinhaPiso, binColunaPiso, binOrientaçãoPiso = arredondarParaBaixo([binLinha, binColuna,
47        binOrientação]).converterParaInteiro()
48        fraçãoLinha, fraçãoColuna, fraçãoOrientação = binLinha - binLinhaPiso, binColuna - binColunaPiso, binOrientação -
49        binOrientaçãoPiso
50        Se binOrientaçãoPiso < 0:
51          binOrientaçãoPiso += numBins
52        Se binOrientaçãoPiso ≥ numBins:
53          binOrientaçãoPiso -= numBins
54
55        c1 = magnitude * fraçãoLinha
56        c0 = magnitude * (1 - fraçãoLinha)
57        c11 = c1 * fraçãoColuna
58        c10 = c1 * (1 - fraçãoColuna)
59        c01 = c0 * fraçãoColuna
60        c00 = c0 * (1 - fraçãoColuna)
61        c111 = c11 * fraçãoOrientação
62        c110 = c11 * (1 - fraçãoOrientação)
63        c101 = c10 * fraçãoOrientação
64        c100 = c10 * (1 - fraçãoOrientação)
65        c011 = c01 * fraçãoOrientação
66        c010 = c01 * (1 - fraçãoOrientação)
67        c001 = c00 * fraçãoOrientação
68        c000 = c00 * (1 - fraçãoOrientação)
69
70        tensorHistograma[binLinhaPiso + 1, binColunaPiso + 1, binOrientaçãoPiso] += c000
71        tensorHistograma[binLinhaPiso + 1, binColunaPiso + 1, (binOrientaçãoPiso + 1) % numBins] += c001
72        tensorHistograma[binLinhaPiso + 1, binColunaPiso + 2, binOrientaçãoPiso] += c010
73        tensorHistograma[binLinhaPiso + 1, binColunaPiso + 2, (binOrientaçãoPiso + 1) % numBins] += c011
74        tensorHistograma[binLinhaPiso + 2, binColunaPiso + 1, binOrientaçãoPiso] += c100
75        tensorHistograma[binLinhaPiso + 2, binColunaPiso + 1, (binOrientaçãoPiso + 1) % numBins] += c101
76        tensorHistograma[binLinhaPiso + 2, binColunaPiso + 2, binOrientaçãoPiso] += c110
77        tensorHistograma[binLinhaPiso + 2, binColunaPiso + 2, (binOrientaçãoPiso + 1) % numBins] += c111
78
79        vetorDescritor = tensorHistograma[1:-1, 1:-1, :].aplanar() # Remove as bordas do histograma
80        Limite = norma(vetorDescritor) * valorMaximoDescritor
81        vetorDescritor[vetorDescritor > Limite] = Limite
82        vetorDescritor ≠ máximo(norma(vetorDescritor), tolerânciaFloat)
83        vetorDescritor = arredondar(512 * vetorDescritor)
84        vetorDescritor[vetorDescritor < 0] = 0
85        vetorDescritor[vetorDescritor > 255] = 255
86        Descrições.adicionar(vetorDescritor)
87
88   Retornar conjunto(Descrições)

```

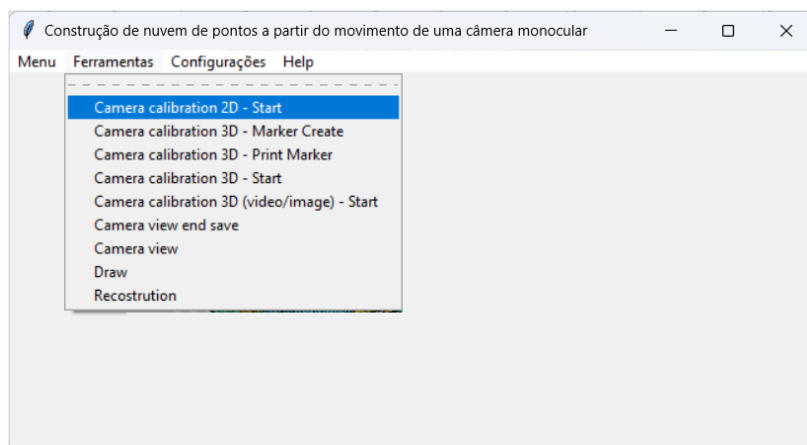
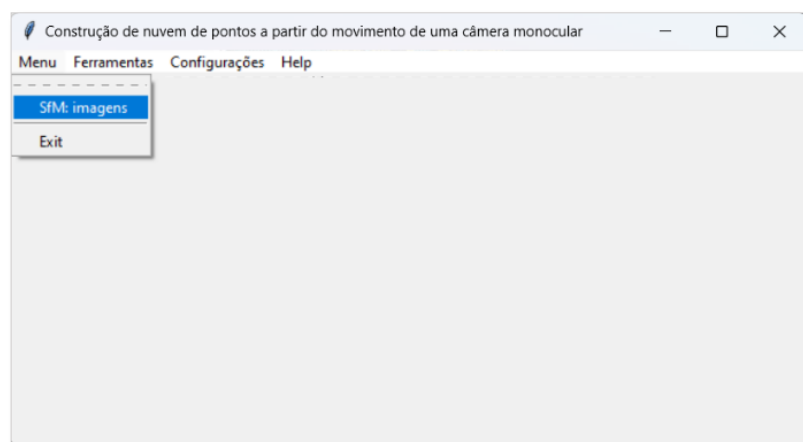
Apêndice H - Função que executa a correspondências entre descritores

```

0 função correspondencias(descritor1, descritor2)
1   bf = criarBFMatcher() // Cria um objeto BFMatcher
2   matches = bf.knnMatch(descritor1, descritor2, k=2) // Realiza a correspondência de vizinhos mais próximos
3   bons_matches = [] // Lista para armazenar correspondências boas
4   contagem_boas_correspondencias = 0 // Contador para o número de correspondências boas
5   para cada m, n em correspondencias:
6     fator = 0.5 // Fator de filtragem para determinar bons matches
7     se m.distância < fator * n.distância: // Verifica se a distância está abaixo do limite
8       adicionar m a boas_correspondencias // Adiciona a correspondência à lista de boas correspondencias
9       contagem_boas_correspondencias += 1 // Incrementa o contador de boas correspondencias
10  retornar boas_correspondencias // Retorna a lista de boas correspondencias
11

```

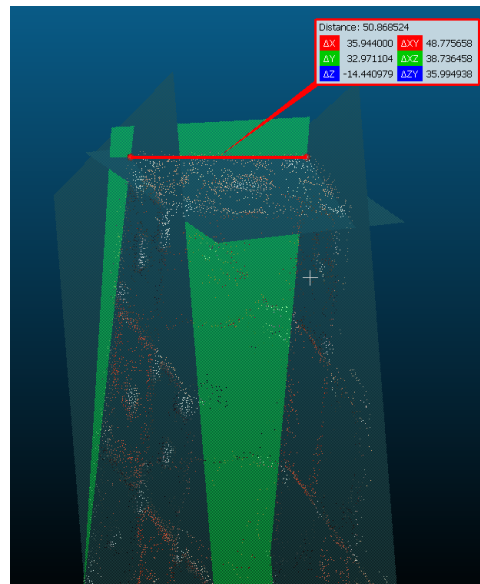
Apêndice I - Interface desenvolvida para o sistema



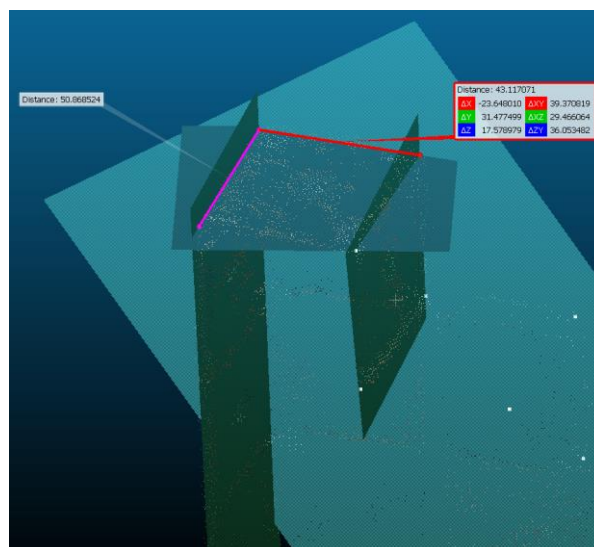
Apêndice J - Imagem capturadas pelo braço robótico kuka para calibração da câmera.



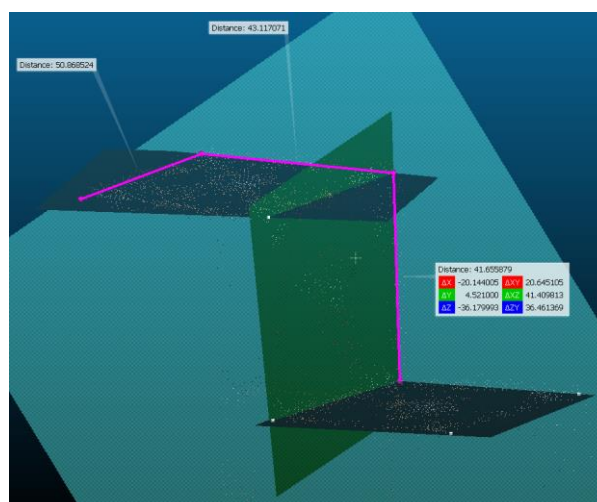
Apêndice K - Planos de definidos para medir a distância entre pontos 1 e 2 X.



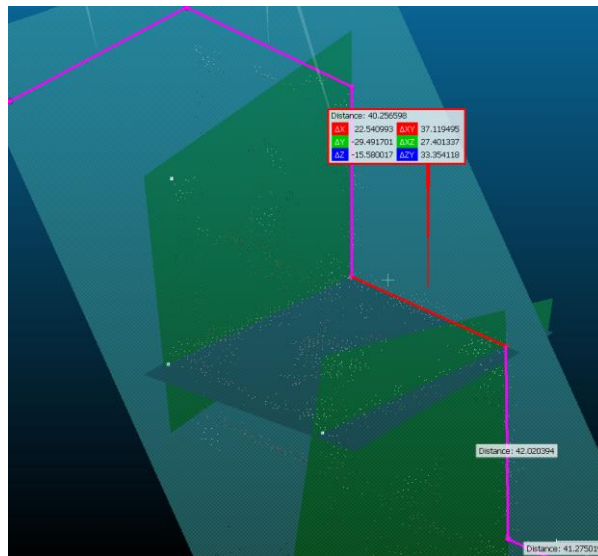
Apêndice L - Planos de definidos para medir a distância entre pontos 1 e 2 X.



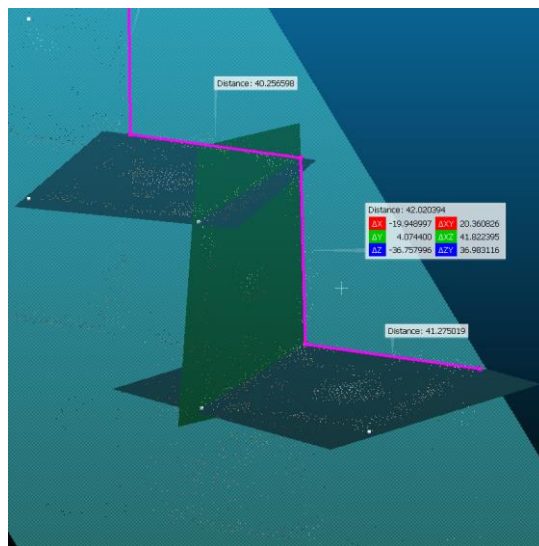
Apêndice M - Planos de definidos para medir a distância entre pontos 3 e 4 Z.



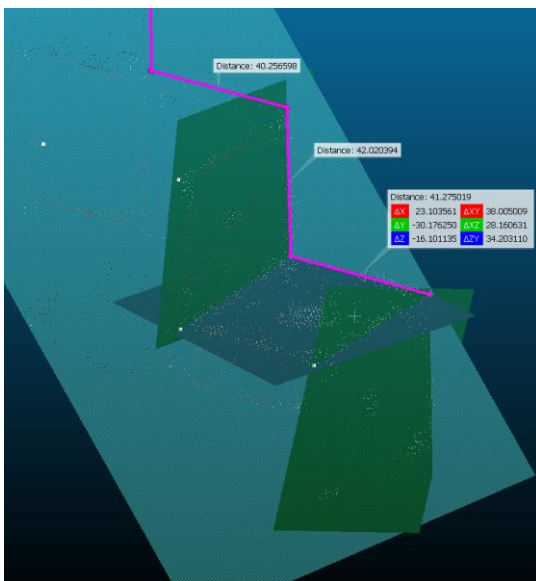
Apêndice N - Planos de definidos para medir a distância entre pontos 4 e 5 Y.



Apêndice O - Planos de definidos para medir a distância entre pontos 5 e 6 Z.



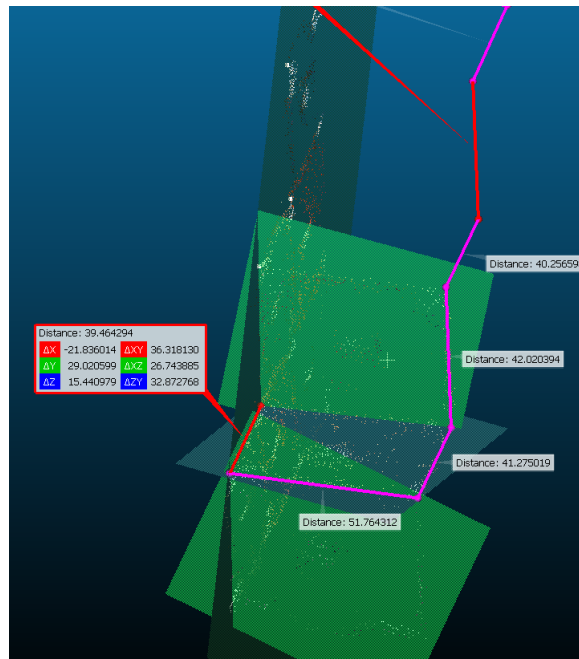
Apêndice P - Planos de definidos para medir a distância entre pontos 6 e 7 Y.



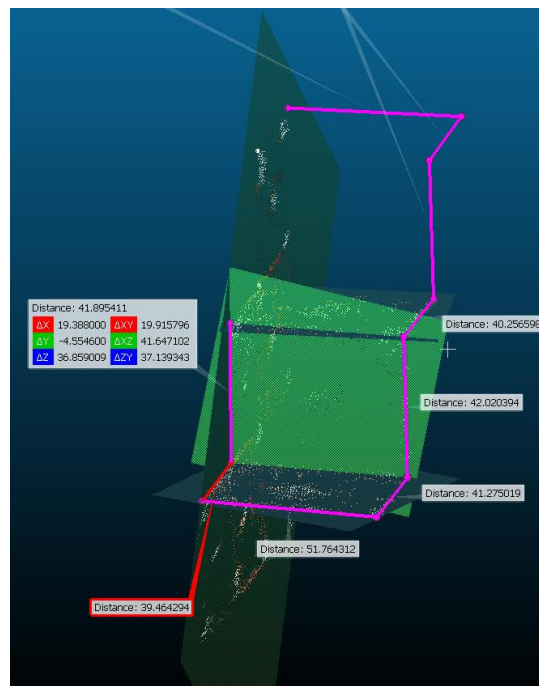
Apêndice Q - Planos de definidos para medir a distância entre pontos 7 e 8 X.



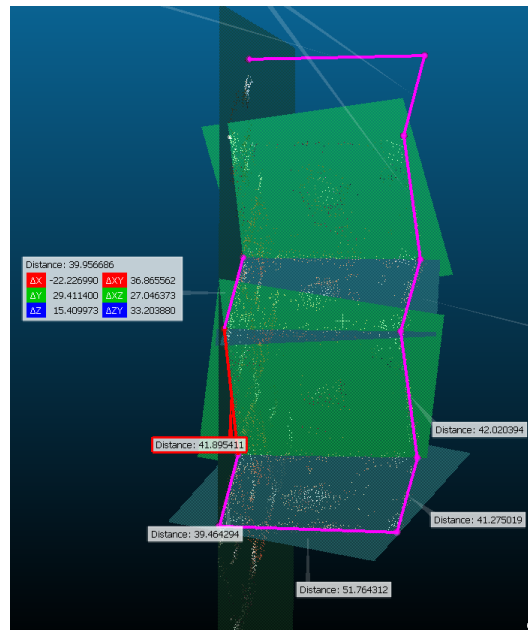
Apêndice R - Planos de definidos para medir a distância entre pontos 8 e 9 Y.



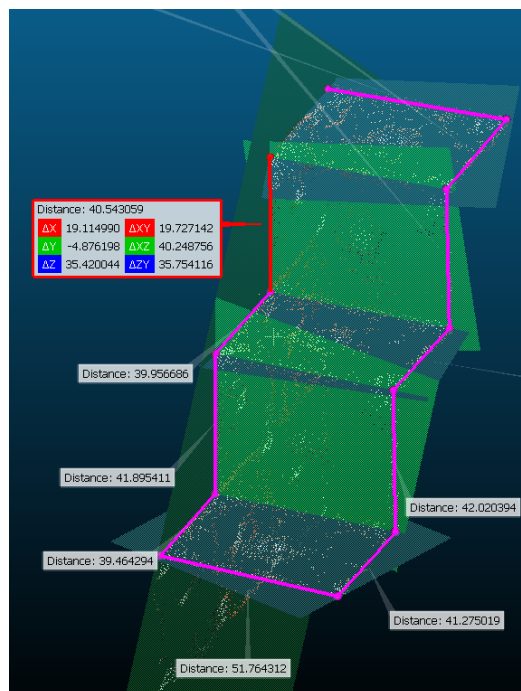
Apêndice S - Planos de definidos para medir a distância entre pontos 9 e 10 Z



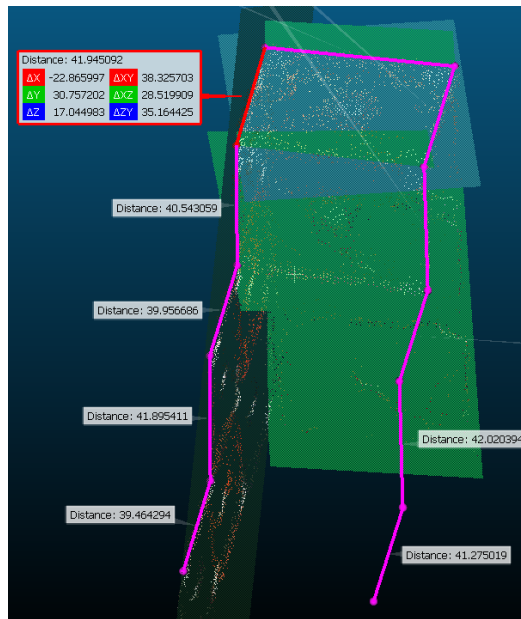
Apêndice T - Planos de definidos para medir a distância entre pontos 10 e 11 Y.



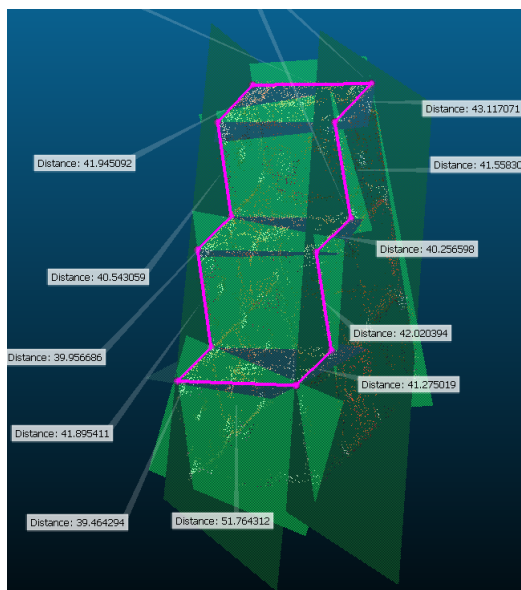
Apêndice U - Planos de definidos para medir a distância entre pontos 11 e 12



Apêndice V - Planos de definidos para medir a distância entre pontos 12 e 1 Y.

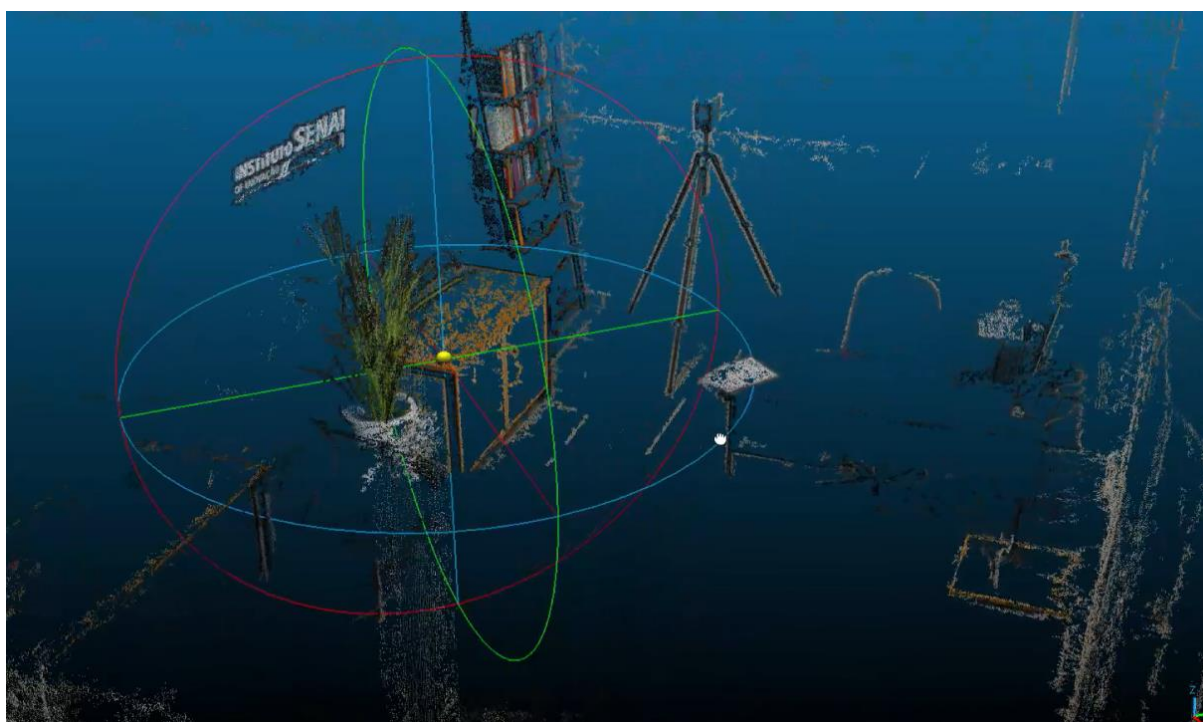
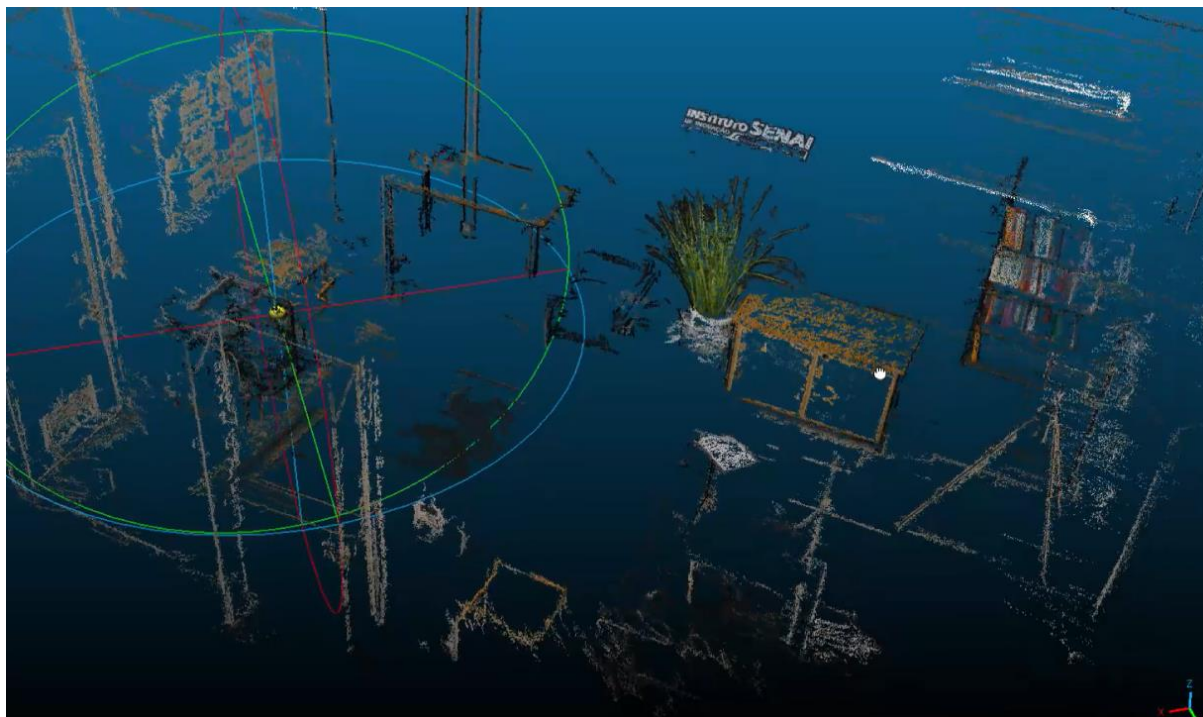


Apêndice W - Todos os planos definidos

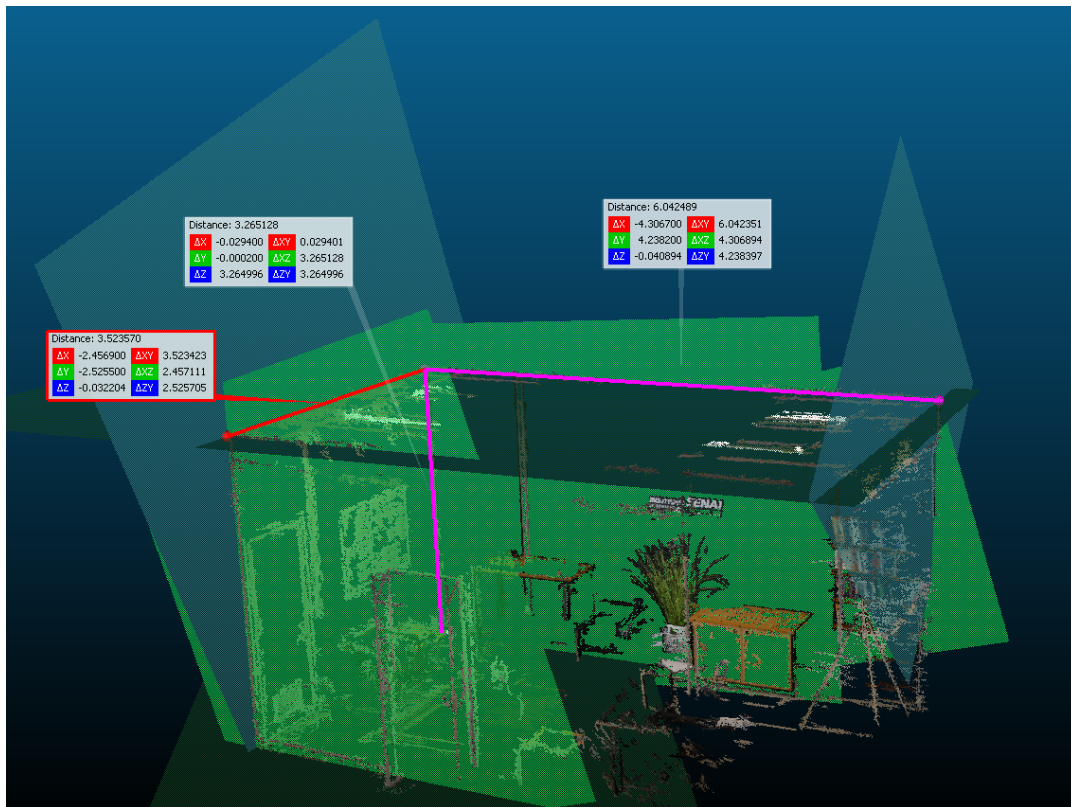


Apêndice X - Outras medições realizadas – escritório

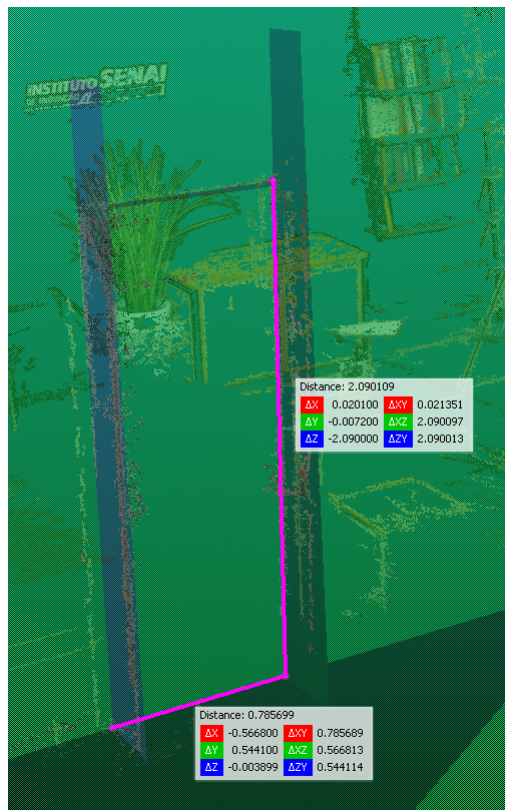
- **Nuvem de pontos gerada**



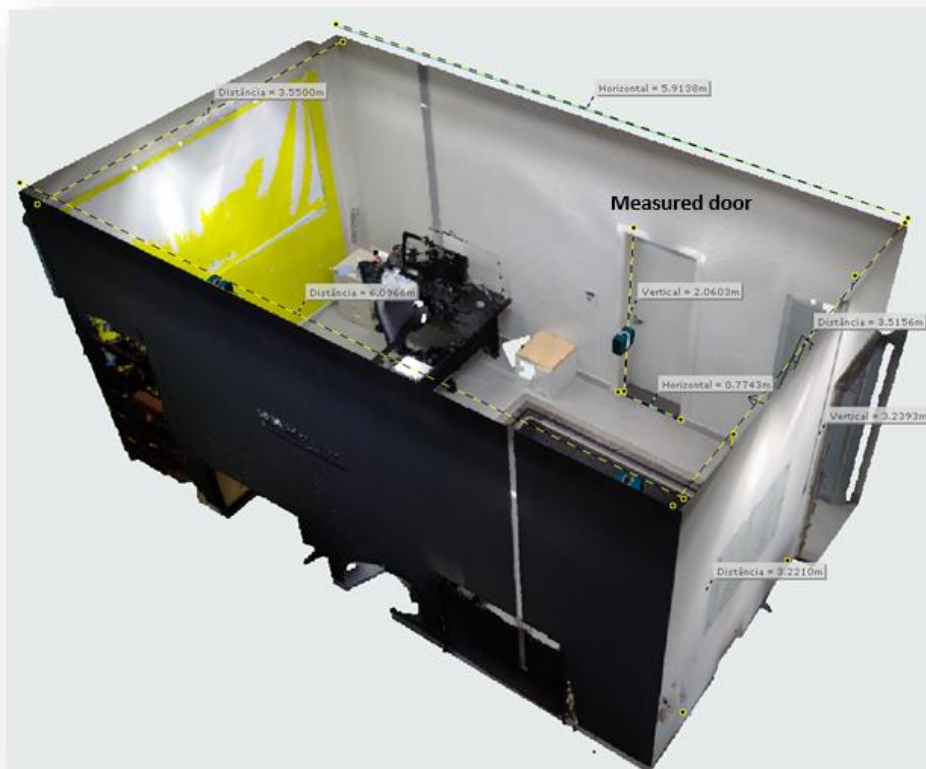
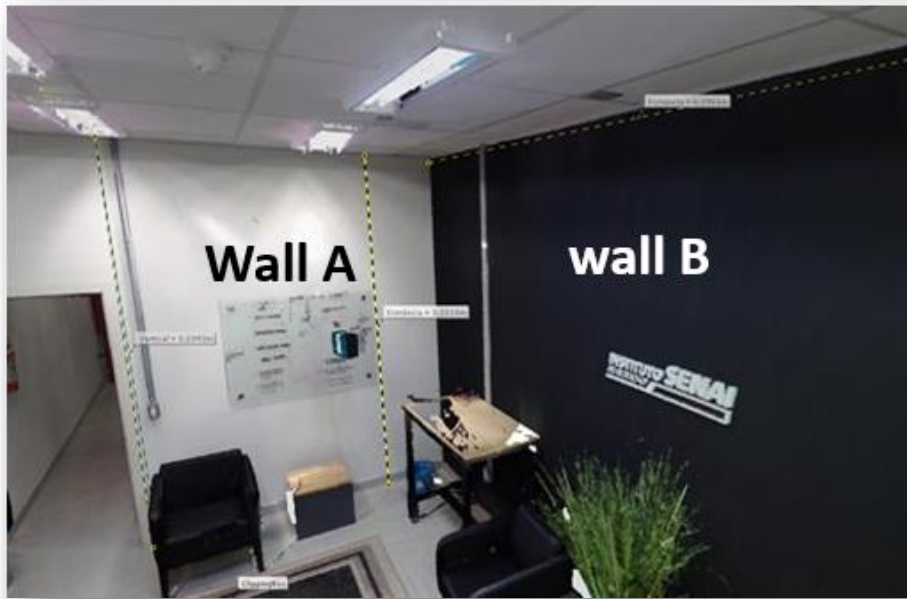
- **Medições do escritório: altura, largura e comprimento.**



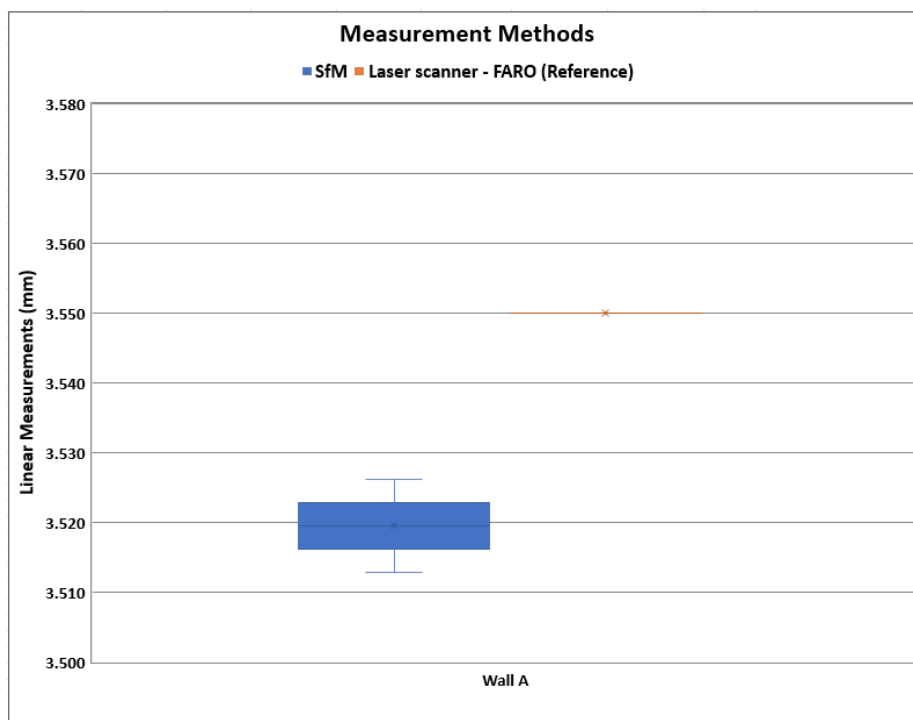
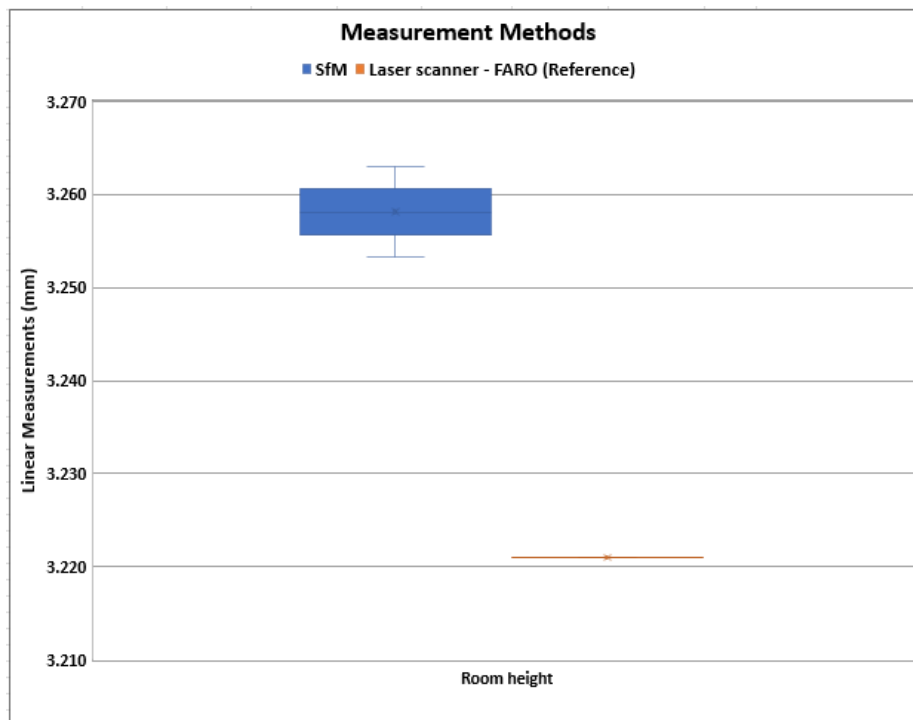
- **Medições da porta**

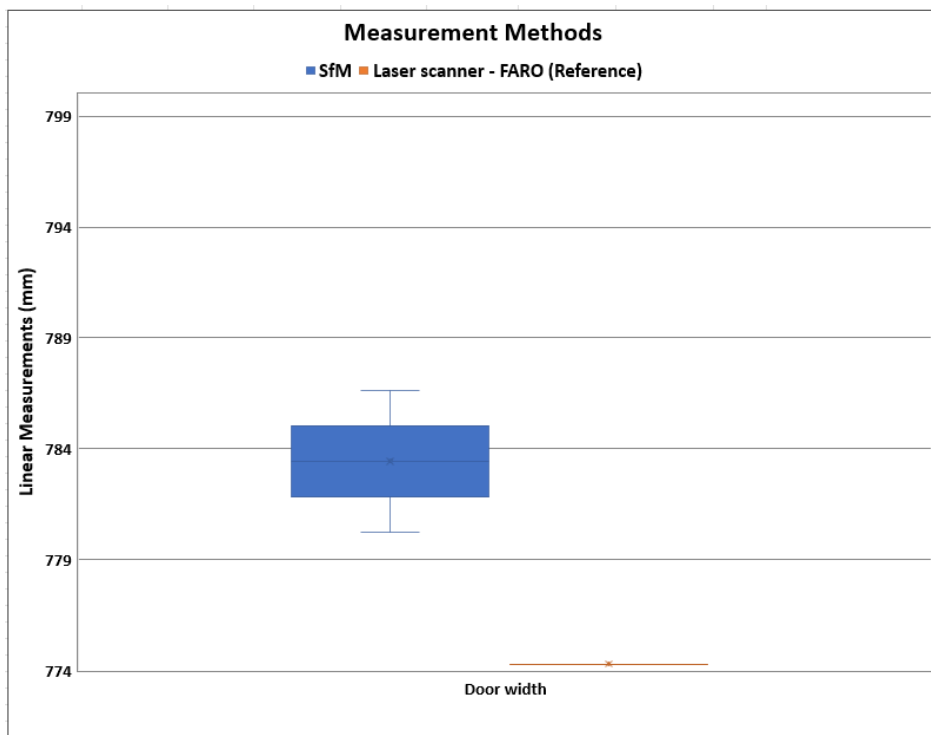
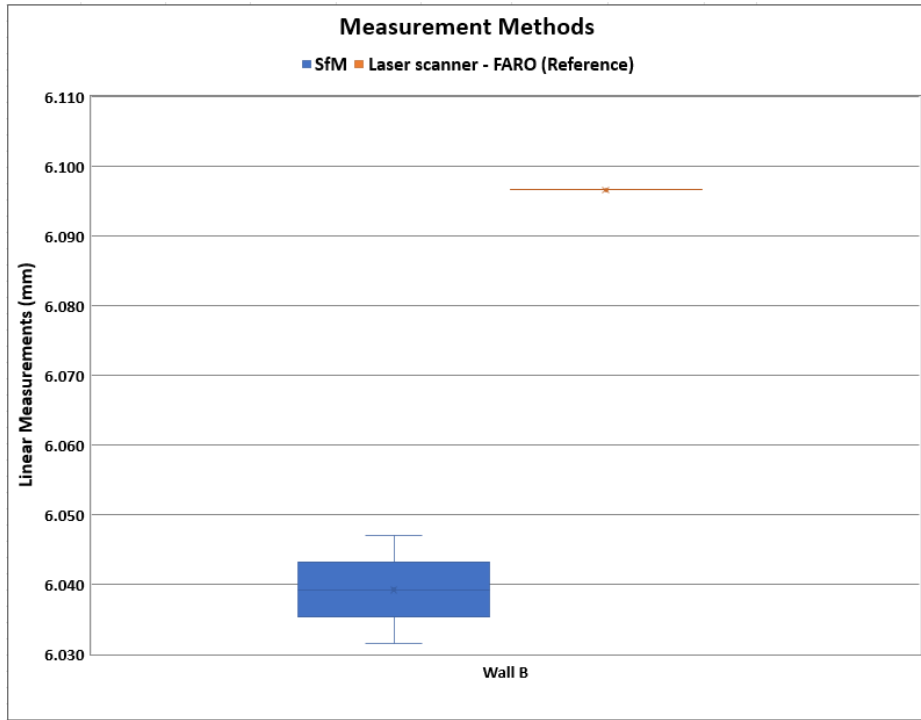


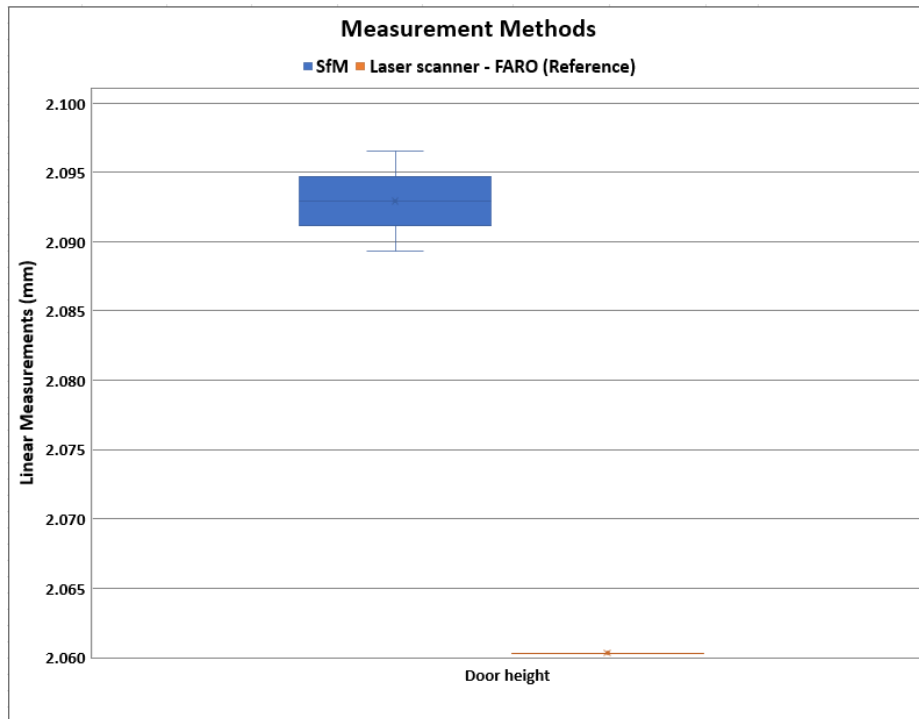
- Medições com laser scanner (Anexo A)



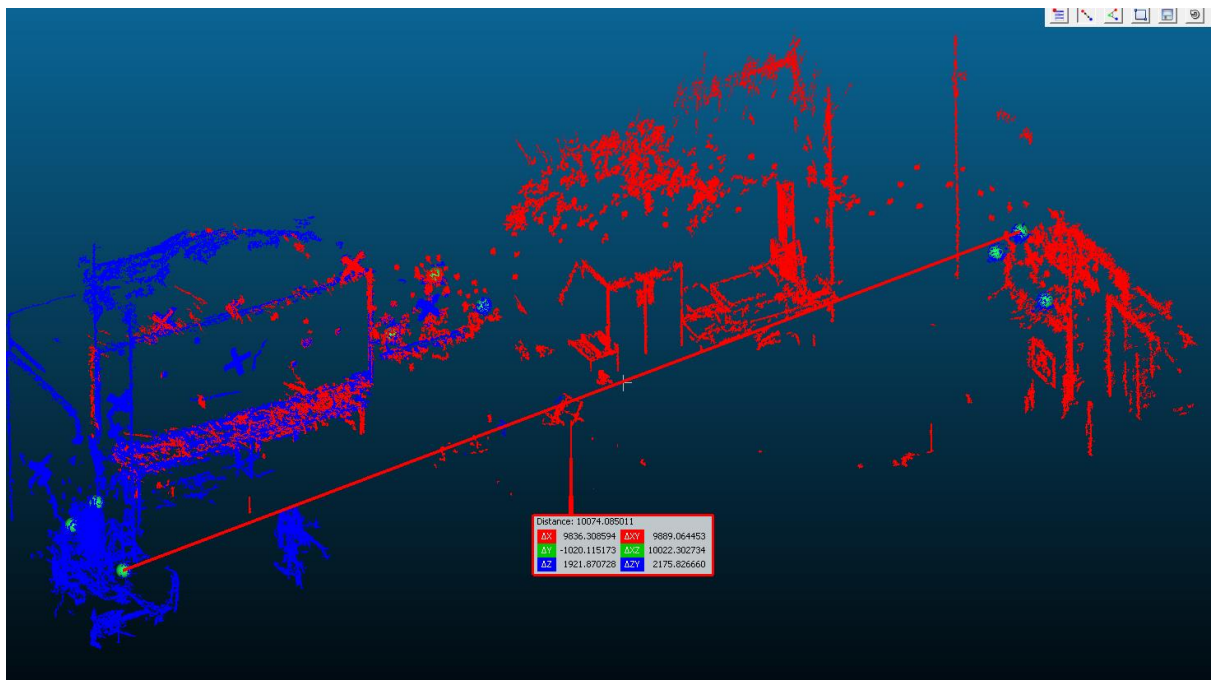
Apêndice Y - Outras medições – Resultados das medições do escritório



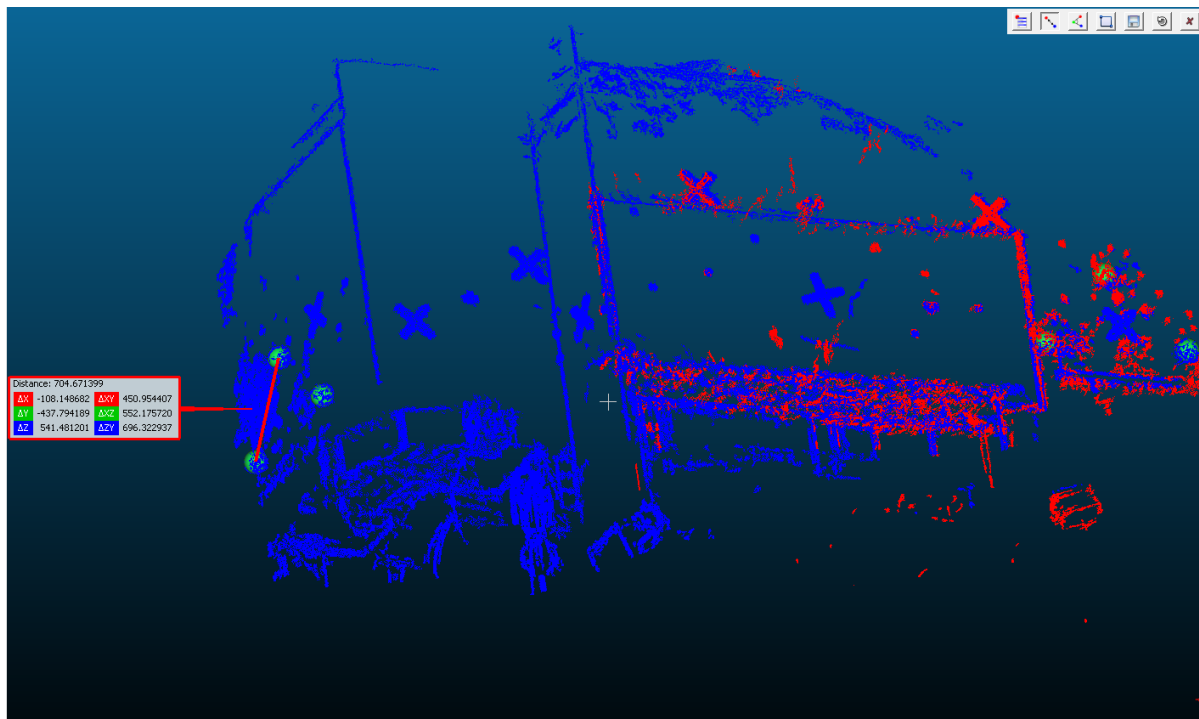




Apêndice Z - Medições das paredes do laboratório utilizando esferas de referência



Apêndice AA - Medições das paredes do laboratório utilizando esferas de referência

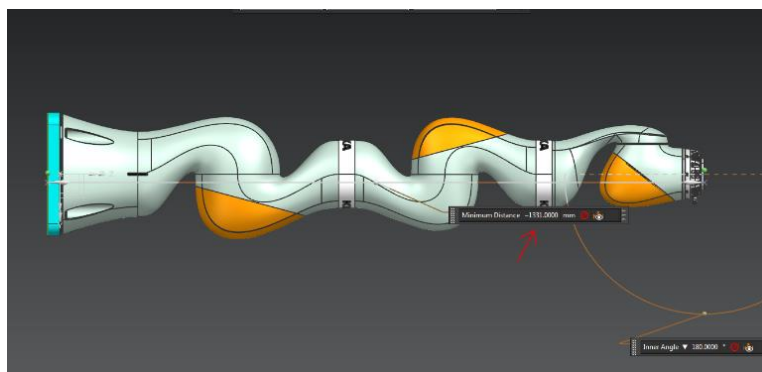
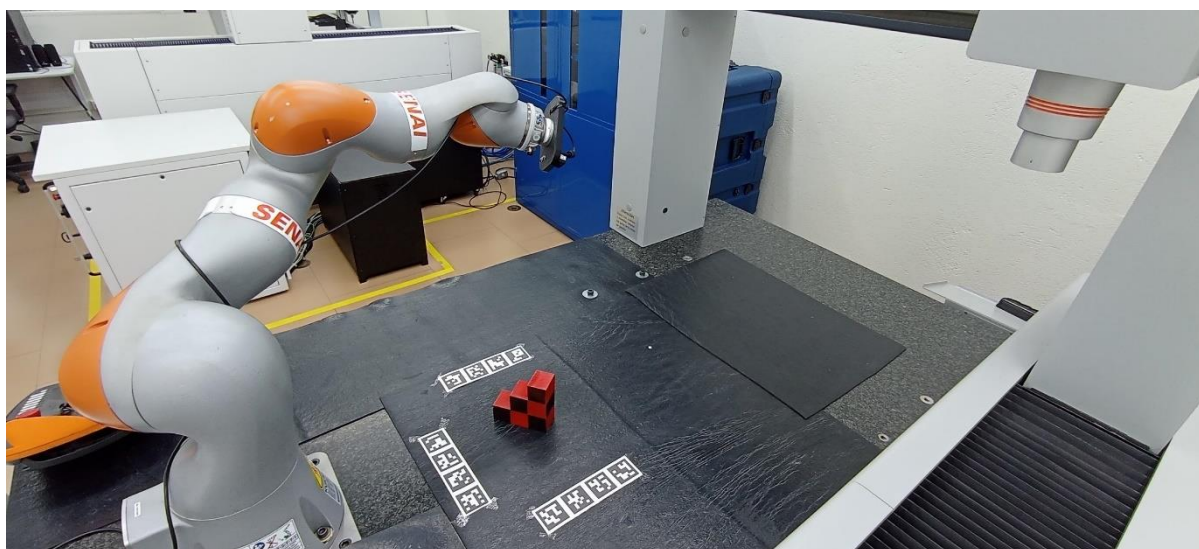


ANEXOS











Anexo A - Laser scanner



























Anexo B - Braço robótico kuka













Anexo C - Medições realizadas pela ZEISS (Peça escada)
Primeira rodada de medições

ZEISS Calypro					
Plano Medição peca_ref1		Data 22 de Setembro de 2022			
No. Desenho * drawingno *	Hora 3:24:27 pm	Ordem OS		No. Sequencial Peça 1	
Operador Master	MMC C32Bit				
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
					
 Resultados Globais Todas Características: 45 ...Tolerância: 43 ...Fora da Tolerância: 2 ...Acima Limite Alerta: 0 ...Não Calculado: 0 Total de Sistemas de coord.: 1 ...Não Calculado: 0 Total de elementos de texto: 0					
 Distância_1-2_X 50.30107	50.20000	0.15000	-0.15000	--	
 Distância_1-2_Y 0.09302	40.00000	0.15000	-0.15000	-39.75698 -39.90698	
 Distância_1-2_Z 0.13850	0.17509	0.05000	-0.05000	-- -0.03659	
 Distância_2-3_X 0.02392	0.00000	0.05000	-0.05000	--	
 Distância_2-3_Y 39.95271	40.04485	0.15000	-0.15000	-- -0.09214	
 Distância_2-3_Z 0.00007	39.97509	0.15000	-0.15000	-39.82502 -39.97502	
 Distância_3-4_X 0.00819	0.00000	0.05000	-0.05000		
 Distância_3-4_Y 0.00968	0.00000	0.05000	-0.05000		










Nome Plano peca_ref1	Operador Master	Hora 3:24:27 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_3-4_Z 39.94989	39.97509	0.15000	-0.15000	- -0.02520	
 Distância_4-5_X 0.02394	0.00000	0.05000	-0.05000	--	
 Distância_4-5_Y 39.99207	40.04485	0.15000	-0.15000	- -0.05278	
 Distância_4-5_Z 0.01815	0.00000	0.05000	-0.05000	--	
 Distância_4-11_X 50.26895	50.20000	0.15000	-0.15000	--	
 Distância_4-11_Y 0.01458	0.00000	0.05000	-0.05000	--	
 Distância_4-11_Z 0.01695	0.00000	0.05000	-0.05000	--	
 Distância_5-6_X 0.00823	0.00000	0.05000	-0.05000		
 Distância_5-6_Y 0.05117	0.06981	0.05000	-0.05000	- -0.01864	
 Distância_5-6_Z 40.00780	40.00000	0.15000	-0.15000		
 Distância_5-10_X 50.24501	50.20000	0.15000	-0.15000	--	
 Distância_5-10_Y 0.01571	0.00000	0.05000	-0.05000	--	













Nome Plano peca_ref1	Operador Master	Hora 3:24:27 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_5-10_Z 0.01694	0.00000	0.05000	-0.05000	--	
 Distância_6-7_X 0.02410	0.00000	0.05000	-0.05000	--	
 Distância_6-7_Y 40.25156	40.16995	0.15000	-0.15000	--	
 Distância_6-7_Z 0.00417	0.00000	0.05000	-0.05000		
 Distância_6-9_X 50.23678	50.20000	0.15000	-0.15000		
 Distância_6-9_Y 0.01575	0.00000	0.05000	-0.05000	--	
 Distância_6-9_Z 0.02222	0.00000	0.05000	-0.05000	--	
 Distância_7-8_X 50.21269	50.20000	0.15000	-0.15000		
 Distância_7-8_Y 0.05666	0.08762	0.05000	-0.05000	- -0.03095	
 Distância_7-8_Z 0.02221	0.00000	0.05000	-0.05000	--	
 Distância_8-9_X 0.00000	0.00000	0.05000	-0.05000		
 Distância_8-9_Y 40.32398	40.25757	0.15000	-0.15000	--	







Nome Plano peca_ref1	Operador Master	Hora 3:24:27 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_8-9_Z 0.00417	0.00000	0.05000	-0.05000		
 Distância_9-10_X 0.00000	0.00000	0.05000	-0.05000		
 Distância_9-10_Y 0.05113	0.06981	0.05000	-0.05000	- -0.01869	
 Distância_9-10_Z 39.96864	40.00000	0.15000	-0.15000	- -0.03136	
 Distância_10-11_X 0.00000	0.00000	0.05000	-0.05000		
 Distância_10-11_Y 39.99094	40.04485	0.15000	-0.15000	- -0.05390	
 Distância_10-11_Z 0.01815	0.00000	0.05000	-0.05000	--	
 Distância_11-12_X 0.00000	0.00000	0.05000	-0.05000		
 Distância_11-12_Y 0.00966	0.00000	0.05000	-0.05000	--	
 Distância_11-12_Z 39.82840	39.80000	0.15000	-0.15000		
 Distância_1-12_X 0.00000	0.00000	0.05000	-0.05000		
 Distância_1-12_Y 40.03117	40.00000	0.15000	-0.15000	--	











Nome Plano peca_ref1	Operador Master	Hora 3:24:27 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_1-12_Z 0.00000	0.00000	0.05000	-0.05000		


Segunda rodada de medições

ZEISS Calypso					
Plano Medição peca_ref1		Data 22 de Setembro de 2022		ZEISS	
No. Desenho * drawingno *	Ordem OS	Hora 3:32:08 pm		No. Sequencial Peça 2	
Operador Master	MMC C32Bit				
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Resultados Globais Todas Características: 45 ...Tolerância: 2 ...Fora da Tolerância: 0 ...Acima Limite Alerta: 0 ...Não Calculado: 1 Total de Sistemas de coord.: 1 ...Não Calculado: 0 Total de elementos de texto: 0					
 Distância_1-2_X	50.30094	50.20000	0.15000	-0.15000	--- 0.10094
 Distância_1-2_Y	0.09308	40.00000	0.15000	-0.15000	-39.75692 -39.90692
 Distância_1-2_Z	0.13827	0.17509	0.05000	-0.05000	--- -0.03682
 Distância_2-3_X	0.02384	0.00000	0.05000	-0.05000	--- 0.02384
 Distância_2-3_Y	39.95171	40.04485	0.15000	-0.15000	--- -0.09314
 Distância_2-3_Z	0.00007	39.97509	0.15000	-0.15000	-39.82503 -39.97503
 Distância_3-4_X	0.00808	0.00000	0.05000	-0.05000	--- 0.00808
 Distância_3-4_Y	0.00933	0.00000	0.05000	-0.05000	--- 0.00933










Nome Plano peca_ref1	Operador Master	Hora 3:32:08 pm	Data 22 de Setembro de 2022	ZEISS	
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_3-4_Z	39.95033	39.97509	0.15000	-0.15000	- 0.02476
 Distância_4-5_X	0.02386	0.00000	0.05000	-0.05000	--- 0.02386
 Distância_4-5_Y	39.99184	40.04485	0.15000	-0.15000	--- -0.05301
 Distância_4-5_Z	0.01761	0.00000	0.05000	-0.05000	--- 0.01761
 Distância_4-11_X	50.26901	50.20000	0.15000	-0.15000	--- 0.06901
 Distância_4-11_Y	0.01443	0.00000	0.05000	-0.05000	--- 0.01443
 Distância_4-11_Z	0.01692	0.00000	0.05000	-0.05000	--- 0.01692
 Distância_5-6_X	0.00812	0.00000	0.05000	-0.05000	--- 0.00812
 Distância_5-6_Y	0.05095	0.06981	0.05000	-0.05000	--- -0.01886
 Distância_5-6_Z	40.00784	40.00000	0.15000	-0.15000	--- 0.00784
 Distância_5-10_X	50.24515	50.20000	0.15000	-0.15000	--- 0.04515
 Distância_5-10_Y	0.01553	0.00000	0.05000	-0.05000	--- 0.01553













Nome Plano peca_ref1	Operador Master	Hora 3:32:08 pm	Data 22 de Setembro de 2022	ZEISS	
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_5-10_Z	0.01691	0.00000	0.05000	-0.05000	--- 0.01691
 Distância_6-7_X	0.02402	0.00000	0.05000	-0.05000	--- 0.02402
 Distância_6-7_Y	40.25160	40.16995	0.15000	-0.15000	--- 0.08165
 Distância_6-7_Z	0.00348	0.00000	0.05000	-0.05000	--- 0.00348
 Distância_6-9_X	50.23703	50.20000	0.15000	-0.15000	--- 0.03703
 Distância_6-9_Y	0.01558	0.00000	0.05000	-0.05000	--- 0.01558
 Distância_6-9_Z	0.02214	0.00000	0.05000	-0.05000	--- 0.02214
 Distância_7-8_X	50.21301	50.20000	0.15000	-0.15000	--- 0.01301
 Distância_7-8_Y	0.05683	0.08762	0.05000	-0.05000	--- -0.03079
 Distância_7-8_Z	0.02214	0.00000	0.05000	-0.05000	--- 0.02214
 Distância_8-9_X	0.00000	0.00000	0.05000	-0.05000	0.00000
 Distância_8-9_Y	40.32401	40.25757	0.15000	-0.15000	--- 0.06644













Nome Plano peca_ref1	Operador Master	Hora 3:32:08 pm	Data 22 de Setembro de 2022	ZEISS	
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_8-9_Z	0.00347	0.00000	0.05000	-0.05000	--- 0.00347
 Distância_9-10_X	0.00000	0.00000	0.05000	-0.05000	0.00000
 Distância_9-10_Y	0.05090	0.06981	0.05000	-0.05000	--- -0.01891
 Distância_9-10_Z	39.96878	40.00000	0.15000	-0.15000	--- -0.03122
 Distância_10-11_X	0.00000	0.00000	0.05000	-0.05000	0.00000
 Distância_10-11_Y	39.99074	40.04485	0.15000	-0.15000	--- -0.05411
 Distância_10-11_Z	0.01762	0.00000	0.05000	-0.05000	--- 0.01762
 Distância_11-12_X	0.00000	0.00000	0.05000	-0.05000	0.00000
 Distância_11-12_Y	0.00931	0.00000	0.05000	-0.05000	--- 0.00931
 Distância_11-12_Z	39.82905	39.80000	0.15000	-0.15000	--- 0.02905
 Distância_1-12_X	0.00000	0.00000	0.05000	-0.05000	0.00000
 Distância_1-12_Y	40.03039	40.00000	0.15000	-0.15000	--- 0.03039













Nome Plano peca_ref1	Operador Master	Hora 3:32:08 pm	Data 22 de Setembro de 2022	ZEISS	
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_1-12_Z	0.00000	0.00000	0.05000	-0.05000	0.00000


Terceira rodada de medições

ZEISS Calypso					
Plano Medição peca_ref1		Data 22 de Setembro de 2022		ZEISS	
No. Desenho *drawingno*	Hora 3:39:31 pm	Ordem OS			
Operador Master	MMC C32Bit	No. Sequencial Peça 3			
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Resultados Globais Todas Características: 45 ...Tolerância: 2 ...Fora da Tolerância: 0 ...Acima Limite Alerta: 0 ...Não Calculado: 0 Total de Sistemas de coord.: 1 ...Não Calculado: 0 Total de elementos de texto: 0					
 Distância_1-2_X 50.30082	50.20000	0.15000	-0.15000	--	
 Distância_1-2_Y 0.09300	40.00000	0.15000	-0.15000	-39.75700 -39.90700	
 Distância_1-2_Z 0.13849	0.17509	0.05000	-0.05000	--	
 Distância_2-3_X 0.02392	0.00000	0.05000	-0.05000	--	
 Distância_2-3_Y 39.95164	40.04485	0.15000	-0.15000	--	
 Distância_2-3_Z 0.00007	39.97509	0.15000	-0.15000	-39.82502 -39.97502	
 Distância_3-4_X 0.00806	0.00000	0.05000	-0.05000	--	
 Distância_3-4_Y 0.00938	0.00000	0.05000	-0.05000	--	









ZEISS					
Nome Plano peca_ref1		Operador Master	Hora 3:39:31 pm	Data 22 de Setembro de 2022	ZEISS
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_5-10_Z 0.01683	0.00000	0.05000	-0.05000	--	
 Distância_6-7_X 0.02410	0.00000	0.05000	-0.05000	--	
 Distância_6-7_Y 40.25147	40.16995	0.15000	-0.15000	--	
 Distância_6-7_Z 0.00359	0.00000	0.05000	-0.05000	--	
 Distância_6-9_X 50.23681	50.20000	0.15000	-0.15000	--	
 Distância_6-9_Y 0.01580	0.00000	0.05000	-0.05000	--	
 Distância_6-9_Z 0.02243	0.00000	0.05000	-0.05000	--	
 Distância_7-8_X 50.21271	50.20000	0.15000	-0.15000	--	
 Distância_7-8_Y 0.05667	0.08762	0.05000	-0.05000	--	
 Distância_7-8_Z 0.02243	0.00000	0.05000	-0.05000	--	
 Distância_8-9_X 0.00000	0.00000	0.05000	-0.05000	--	
 Distância_8-9_Y 40.32394	40.25757	0.15000	-0.15000	--	













ZEISS					
Nome Plano peca_ref1		Operador Master	Hora 3:39:31 pm	Data 22 de Setembro de 2022	ZEISS
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_3-4_Z 39.95039	39.97509	0.15000	-0.15000	--	
 Distância_4-5_X 0.02394	0.00000	0.05000	-0.05000	--	
 Distância_4-5_Y 39.99212	40.04485	0.15000	-0.15000	--	
 Distância_4-5_Z 0.01762	0.00000	0.05000	-0.05000	--	
 Distância_4-11_X 50.26885	50.20000	0.15000	-0.15000	--	
 Distância_4-11_Y 0.01448	0.00000	0.05000	-0.05000	--	
 Distância_4-11_Z 0.01684	0.00000	0.05000	-0.05000	--	
 Distância_5-6_X 0.00809	0.00000	0.05000	-0.05000	--	
 Distância_5-6_Y 0.05112	0.06981	0.05000	-0.05000	--	
 Distância_5-6_Z 40.00772	40.00000	0.15000	-0.15000	--	
 Distância_5-10_X 50.24491	50.20000	0.15000	-0.15000	--	
 Distância_5-10_Y 0.01576	0.00000	0.05000	-0.05000	--	









ZEISS					
Nome Plano peca_ref1		Operador Master	Hora 3:39:31 pm	Data 22 de Setembro de 2022	ZEISS
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_8-9_Z 0.00358	0.00000	0.05000	-0.05000	--	
 Distância_9-10_X 0.00000	0.00000	0.05000	-0.05000	--	
 Distância_9-10_Y 0.05107	0.06981	0.05000	-0.05000	--	
 Distância_9-10_Z 39.96846	40.00000	0.15000	-0.15000	--	
 Distância_10-11_X 0.00000	0.00000	0.05000	-0.05000	--	
 Distância_10-11_Y 39.99084	40.04485	0.15000	-0.15000	--	
 Distância_10-11_Z 0.01763	0.00000	0.05000	-0.05000	--	
 Distância_11-12_X 0.00000	0.00000	0.05000	-0.05000	--	
 Distância_11-12_Y 0.00936	0.00000	0.05000	-0.05000	--	
 Distância_11-12_Z 39.82880	39.80000	0.15000	-0.15000	--	
 Distância_1-12_X 0.00000	0.00000	0.05000	-0.05000	--	
 Distância_1-12_Y 40.03018	40.00000	0.15000	-0.15000	--	












ZEISS					
Nome Plano peca_ref1		Operador Master	Hora 3:39:31 pm	Data 22 de Setembro de 2022	ZEISS
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_1-12_Z 0.00000	0.00000	0.05000	-0.05000	--	


Quarta rodada de medições

ZEISS Calypso					
Plano Medição peca_ref1	Data 22 de Setembro de 2022				
No. Desenho * drawingno *	Hora 3:46:31 pm	Ordem OS			
Operador Master	MMC C32Bit	No. Sequencial Peça 4			
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Resultados Globais					
Todas Características: 45					
...Tolerância: 43					
...Fora da Tolerância: 2					
...Acima Limite Alerta: 0					
...Não Calculado: 0					
Total de Sistemas de coord.: 1					
...Não Calculado: 0					
Total de elementos de texto: 0					
 Distância_1-2_X 50.30090	50.20000	0.15000	-0.15000	--	
 Distância_1-2_Y 0.09302	40.00000	0.15000	-0.15000	-39.75698 -39.90698	
 Distância_1-2_Z 0.13846	0.17509	0.05000	-0.05000	--	
 Distância_2-3_X 0.02398	0.00000	0.05000	-0.05000	--	
 Distância_2-3_Y 39.95173	40.04485	0.15000	-0.15000	--	
 Distância_2-3_Z 0.00007	39.97509	0.15000	-0.15000	-39.82502 -39.97502	
 Distância_3-4_X 0.00806	0.00000	0.05000	-0.05000	--	
 Distância_3-4_Y 0.00961	0.00000	0.05000	-0.05000	--	

ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 3:46:31 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_3-4_Z 39.95016	39.97509	0.15000	-0.15000	--	
 Distância_4-5_X 0.02400	0.00000	0.05000	-0.05000	--	
 Distância_4-5_Y 39.99215	40.04485	0.15000	-0.15000	--	
 Distância_4-5_Z 0.01784	0.00000	0.05000	-0.05000	--	
 Distância_4-11_X 50.26885	50.20000	0.15000	-0.15000	--	
 Distância_4-11_Y 0.01460	0.00000	0.05000	-0.05000	--	
 Distância_4-11_Z 0.01680	0.00000	0.05000	-0.05000	--	
 Distância_5-6_X 0.00810	0.00000	0.05000	-0.05000	--	
 Distância_5-6_Y 0.05146	0.06981	0.05000	-0.05000	--	
 Distância_5-6_Z 40.00760	40.00000	0.15000	-0.15000	--	
 Distância_5-10_X 50.24485	50.20000	0.15000	-0.15000	--	
 Distância_5-10_Y 0.01584	0.00000	0.05000	-0.05000	--	

ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 3:46:31 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_5-10_Z 0.01679	0.00000	0.05000	-0.05000	--	
 Distância_6-7_X 0.02416	0.00000	0.05000	-0.05000	--	
 Distância_6-7_Y 40.25129	40.16995	0.15000	-0.15000	--	
 Distância_6-7_Z 0.00385	0.00000	0.05000	-0.05000	--	
 Distância_6-9_X 50.23675	50.20000	0.15000	-0.15000	--	
 Distância_6-9_Y 0.01589	0.00000	0.05000	-0.05000	--	
 Distância_6-9_Z 0.02243	0.00000	0.05000	-0.05000	--	
 Distância_7-8_X 50.21259	50.20000	0.15000	-0.15000	--	
 Distância_7-8_Y 0.05659	0.08762	0.05000	-0.05000	--	
 Distância_7-8_Z 0.02243	0.00000	0.05000	-0.05000	--	
 Distância_8-9_X 0.00000	0.00000	0.05000	-0.05000	--	
 Distância_8-9_Y 40.32378	40.25757	0.15000	-0.15000	--	

ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 3:46:31 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_8-9_Z 0.00384	0.00000	0.05000	-0.05000	--	
 Distância_9-10_X 0.00000	0.00000	0.05000	-0.05000	--	
 Distância_9-10_Y 0.05142	0.06981	0.05000	-0.05000	--	
 Distância_9-10_Z 39.96838	40.00000	0.15000	-0.15000	--	
 Distância_10-11_X 0.00000	0.00000	0.05000	-0.05000	--	
 Distância_10-11_Y 39.99091	40.04485	0.15000	-0.15000	--	
 Distância_10-11_Z 0.01785	0.00000	0.05000	-0.05000	--	
 Distância_11-12_X 0.00000	0.00000	0.05000	-0.05000	--	
 Distância_11-12_Y 0.00958	0.00000	0.05000	-0.05000	--	
 Distância_11-12_Z 39.82857	39.80000	0.15000	-0.15000	--	
 Distância_1-12_X 0.00000	0.00000	0.05000	-0.05000	--	
 Distância_1-12_Y 40.03017	40.00000	0.15000	-0.15000	--	

ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 3:46:31 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Distância_1-12_Z 0.00000	0.00000	0.05000	-0.05000	--	

Quinta rodada de medições

ZEISS Calypso					
Plano Medição peca_ref1		Data 22 de Setembro de 2022			
No. Desenho * drawingno *		Hora 3:53:40 pm	Ordem OS		
Operador Master		MMC C32Bit	No. Sequencial Peça 5		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Resultados Globais Todas Características: 45 ...Tolerância: 43 ...Fora da Tolerância: 2 ...Acima Limite Alerta: 0 ...Não Calculado: 0 Total de Sistemas de coord.: 1 ...Não Calculado: 0 Total de elementos de texto: 0					
Distância_1-2_X	50.30059	50.20000	0.15000	-0.15000	-- 0.10059
Distância_1-2_Y	0.09304	40.00000	0.15000	-0.15000	-39.75696 -39.90696
Distância_1-2_Z	0.13838	0.17509	0.05000	-0.05000	-- -0.03671
Distância_2-3_X	0.02400	0.00000	0.05000	-0.05000	-- 0.02400
Distância_2-3_Y	39.95190	40.04485	0.15000	-0.15000	-- -0.09294
Distância_2-3_Z	0.00007	39.97509	0.15000	-0.15000	-39.82502 -39.97502
Distância_3-4_X	0.00800	0.00000	0.05000	-0.05000	-- 0.00800
Distância_3-4_Y	0.00965	0.00000	0.05000	-0.05000	-- 0.00965










ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 3:53:40 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_3-4_Z	39.95004	39.97509	0.15000	-0.15000	- -0.02505
Distância_4-5_X	0.02402	0.00000	0.05000	-0.05000	-- 0.02402
Distância_4-5_Y	39.99212	40.04485	0.15000	-0.15000	-- -0.05273
Distância_4-5_Z	0.01811	0.00000	0.05000	-0.05000	-- 0.01811
Distância_4-11_X	50.26859	50.20000	0.15000	-0.15000	-- 0.06859
Distância_4-11_Y	0.01459	0.00000	0.05000	-0.05000	-- 0.01459
Distância_4-11_Z	0.01673	0.00000	0.05000	-0.05000	-- 0.01673
Distância_5-6_X	0.00804	0.00000	0.05000	-0.05000	-- 0.00804
Distância_5-6_Y	0.05156	0.06981	0.05000	-0.05000	-- -0.01825
Distância_5-6_Z	40.00769	40.00000	0.15000	-0.15000	-- 0.00769
Distância_5-10_X	50.24458	50.20000	0.15000	-0.15000	-- 0.04458
Distância_5-10_Y	0.01572	0.00000	0.05000	-0.05000	-- 0.01572













ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 3:53:40 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_5-10_Z	0.01672	0.00000	0.05000	-0.05000	-- 0.01672
Distância_6-7_X	0.02418	0.00000	0.05000	-0.05000	-- 0.02418
Distância_6-7_Y	40.25126	40.16995	0.15000	-0.15000	-- 0.08131
Distância_6-7_Z	0.00417	0.00000	0.05000	-0.05000	-- 0.00417
Distância_6-9_X	50.23654	50.20000	0.15000	-0.15000	-- 0.03654
Distância_6-9_Y	0.01576	0.00000	0.05000	-0.05000	-- 0.01576
Distância_6-9_Z	0.02224	0.00000	0.05000	-0.05000	-- 0.02224
Distância_7-8_X	50.21237	50.20000	0.15000	-0.15000	-- 0.01237
Distância_7-8_Y	0.05657	0.08762	0.05000	-0.05000	-- -0.03104
Distância_7-8_Z	0.02224	0.00000	0.05000	-0.05000	-- 0.02224
Distância_8-9_X	0.00000	0.00000	0.05000	-0.05000	0.00000
Distância_8-9_Y	40.32360	40.25757	0.15000	-0.15000	-- 0.06603











ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 3:53:40 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_8-9_Z	0.00417	0.00000	0.05000	-0.05000	-- 0.00417
Distância_9-10_X	0.00000	0.00000	0.05000	-0.05000	0.00000
Distância_9-10_Y	0.05151	0.06981	0.05000	-0.05000	-- -0.01830
Distância_9-10_Z	39.96873	40.00000	0.15000	-0.15000	-- -0.03127
Distância_10-11_X	0.00000	0.00000	0.05000	-0.05000	0.00000
Distância_10-11_Y	39.99099	40.04485	0.15000	-0.15000	-- -0.05386
Distância_10-11_Z	0.01811	0.00000	0.05000	-0.05000	-- 0.01811
Distância_11-12_X	0.00000	0.00000	0.05000	-0.05000	0.00000
Distância_11-12_Y	0.00963	0.00000	0.05000	-0.05000	-- 0.00963
Distância_11-12_Z	39.82845	39.80000	0.15000	-0.15000	-- 0.02845
Distância_1-12_X	0.00000	0.00000	0.05000	-0.05000	0.00000
Distância_1-12_Y	40.03038	40.00000	0.15000	-0.15000	-- 0.03038













ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 3:53:40 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_1-12_Z	0.00000	0.00000	0.05000	-0.05000	0.00000


Sexta rodada de medições

ZEISS Calypso					
Plano Medição peca_ref1		Data 22 de Setembro de 2022			
No. Desenho * drawingno *		Hora 4:06:22 pm	Ordem OS		
Operador Master		MMC C32Bit	No. Sequencial Peça 6		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
 Resultados Globais Todas Características: ...Tolerância: 43 ...Fora da Tolerância: 2 ...Acima Limite Alerta: 0 ...Não Calculado: 0 Total de Sistemas de coord.: 1 ...Não Calculado: 0 Total de elementos de texto: 0					
	Distância_1-2_X 50.30066	50.20000	0.15000	-0.15000	-
	Distância_1-2_Y 0.09299	40.00000	0.15000	-0.15000	-39.75701 -39.90701
	Distância_1-2_Z 0.13851	0.17509	0.05000	-0.05000	-
	Distância_2-3_X 0.02398	0.00000	0.05000	-0.05000	-
	Distância_2-3_Y 39.95167	40.04485	0.15000	-0.15000	-
	Distância_2-3_Z 0.00007	39.97509	0.15000	-0.15000	-39.82502 -39.97502
	Distância_3-4_X 0.00804	0.00000	0.05000	-0.05000	-
	Distância_3-4_Y 0.00983	0.00000	0.05000	-0.05000	-

Nome Plano peca_ref1	Operador Master	Hora 4:06:22 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
	Distância_3-4_Z 39.94991	39.97509	0.15000	-0.15000	-
	Distância_4-5_X 0.02401	0.00000	0.05000	-0.05000	-
	Distância_4-5_Y 39.99219	40.04485	0.15000	-0.15000	-
	Distância_4-5_Z 0.01806	0.00000	0.05000	-0.05000	-
	Distância_4-11_X 50.26863	50.20000	0.15000	-0.15000	-
	Distância_4-11_Y 0.01455	0.00000	0.05000	-0.05000	-
	Distância_4-11_Z 0.01681	0.00000	0.05000	-0.05000	-
	Distância_5-6_X 0.00808	0.00000	0.05000	-0.05000	-
	Distância_5-6_Y 0.05179	0.06981	0.05000	-0.05000	-
	Distância_5-6_Z 40.00758	40.00000	0.15000	-0.15000	-
	Distância_5-10_X 50.24463	50.20000	0.15000	-0.15000	-
	Distância_5-10_Y 0.01580	0.00000	0.05000	-0.05000	-

Nome Plano peca_ref1	Operador Master	Hora 4:06:22 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
	Distância_5-10_Z 0.01680	0.00000	0.05000	-0.05000	-
	Distância_6-7_X 0.02416	0.00000	0.05000	-0.05000	-
	Distância_6-7_Y 40.25119	40.16995	0.15000	-0.15000	-
	Distância_6-7_Z 0.00428	0.00000	0.05000	-0.05000	-
	Distância_6-9_X 50.23655	50.20000	0.15000	-0.15000	-
	Distância_6-9_Y 0.01585	0.00000	0.05000	-0.05000	-
	Distância_6-9_Z 0.02244	0.00000	0.05000	-0.05000	-
	Distância_7-8_X 50.21239	50.20000	0.15000	-0.15000	-
	Distância_7-8_Y 0.05656	0.08762	0.05000	-0.05000	-
	Distância_7-8_Z 0.02244	0.00000	0.05000	-0.05000	-
	Distância_8-9_X 0.00000	0.00000	0.05000	-0.05000	-
	Distância_8-9_Y 40.32360	40.25757	0.15000	-0.15000	-

Nome Plano peca_ref1	Operador Master	Hora 4:06:22 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
	Distância_8-9_Z 0.00428	0.00000	0.05000	-0.05000	-
	Distância_9-10_X 0.00000	0.00000	0.05000	-0.05000	-
	Distância_9-10_Y 0.05174	0.06981	0.05000	-0.05000	-
	Distância_9-10_Z 39.96834	40.00000	0.15000	-0.15000	-
	Distância_10-11_X 0.00000	0.00000	0.05000	-0.05000	-
	Distância_10-11_Y 39.99094	40.04485	0.15000	-0.15000	-
	Distância_10-11_Z 0.01807	0.00000	0.05000	-0.05000	-
	Distância_11-12_X 0.00000	0.00000	0.05000	-0.05000	-
	Distância_11-12_Y 0.00980	0.00000	0.05000	-0.05000	-
	Distância_11-12_Z 39.82827	39.80000	0.15000	-0.15000	-
	Distância_1-12_X 0.00000	0.00000	0.05000	-0.05000	-
	Distância_1-12_Y 40.03014	40.00000	0.15000	-0.15000	-

Nome Plano peca_ref1	Operador Master	Hora 4:06:22 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
	Distância_1-12_Z 0.00000	0.00000	0.05000	-0.05000	-

Sétima rodada de medições

ZEISS Calypso					
Plano Medição peca_ref1		Data 22 de Setembro de 2022			
No. Desenho	Hora	Ordem			
*drawingno *	4:12:45 pm	OS			
Operador	MMC	No. Sequencial Peça			
Master	C32Bit	7			
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Resultados Globais Todas Características: 45 Tolerância: 43 Fora da Tolerância: 2 Acima Limite Alerta: 0 Não Calculado: 0 Total de Sistemas de coord.: 1 Não Calculado: 0 Total de elementos de texto: 0					
Distância_1-2_X	50.30069	50.20000	0.15000	-0.15000	---
Distância_1-2_Y	0.09300	40.00000	0.15000	-0.15000	-39.75700 -39.90700
Distância_1-2_Z	0.13853	0.17509	0.05000	-0.05000	---
Distância_2-3_X	0.02393	0.00000	0.05000	-0.05000	---
Distância_2-3_Y	39.95157	40.04485	0.15000	-0.15000	---
Distância_2-3_Z	0.00007	39.97509	0.15000	-0.15000	-39.82502 -39.97502
Distância_3-4_X	0.00816	0.00000	0.05000	-0.05000	---
Distância_3-4_Y	0.00977	0.00000	0.05000	-0.05000	---

Nome Plano peca_ref1	Operador Master	Hora 4:12:45 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_3-4_Z	39.95011	39.97509	0.15000	-0.15000	- -
Distância_4-5_X	0.02395	0.00000	0.05000	-0.05000	---
Distância_4-5_Y	39.99218	40.04485	0.15000	-0.15000	- -
Distância_4-5_Z	0.01814	0.00000	0.05000	-0.05000	---
Distância_4-11_X	50.26861	50.20000	0.15000	-0.15000	---
Distância_4-11_Y	0.01447	0.00000	0.05000	-0.05000	---
Distância_4-11_Z	0.01677	0.00000	0.05000	-0.05000	---
Distância_5-6_X	0.00820	0.00000	0.05000	-0.05000	---
Distância_5-6_Y	0.05181	0.06981	0.05000	-0.05000	- -
Distância_5-6_Z	40.00766	40.00000	0.15000	-0.15000	---
Distância_5-10_X	50.24466	50.20000	0.15000	-0.15000	---
Distância_5-10_Y	0.01575	0.00000	0.05000	-0.05000	---

Nome Plano peca_ref1	Operador Master	Hora 4:12:45 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_5-10_Z	0.01676	0.00000	0.05000	-0.05000	---
Distância_6-7_X	0.02410	0.00000	0.05000	-0.05000	---
Distância_6-7_Y	40.25118	40.16995	0.15000	-0.15000	---
Distância_6-7_Z	0.00432	0.00000	0.05000	-0.05000	---
Distância_6-9_X	50.23647	50.20000	0.15000	-0.15000	---
Distância_6-9_Y	0.01579	0.00000	0.05000	-0.05000	---
Distância_6-9_Z	0.02247	0.00000	0.05000	-0.05000	---
Distância_7-8_X	50.21236	50.20000	0.15000	-0.15000	---
Distância_7-8_Y	0.05665	0.08782	0.05000	-0.05000	---
Distância_7-8_Z	0.02246	0.00000	0.05000	-0.05000	---
Distância_8-9_X	0.00000	0.00000	0.05000	-0.05000	---
Distância_8-9_Y	40.32362	40.25757	0.15000	-0.15000	---

Nome Plano peca_ref1	Operador Master	Hora 4:12:45 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_8-9_Z	0.00432	0.00000	0.05000	-0.05000	---
Distância_9-10_X	0.00000	0.00000	0.05000	-0.05000	---
Distância_9-10_Y	0.05176	0.06981	0.05000	-0.05000	- -
Distância_9-10_Z	39.96843	40.00000	0.15000	-0.15000	- -
Distância_10-11_X	0.00000	0.00000	0.05000	-0.05000	---
Distância_10-11_Y	39.99091	40.04485	0.15000	-0.15000	- -
Distância_10-11_Z	0.01815	0.00000	0.05000	-0.05000	---
Distância_11-12_X	0.00000	0.00000	0.05000	-0.05000	---
Distância_11-12_Y	0.00974	0.00000	0.05000	-0.05000	---
Distância_11-12_Z	39.82841	39.80000	0.15000	-0.15000	---
Distância_1-12_X	0.00000	0.00000	0.05000	-0.05000	---
Distância_1-12_Y	40.03013	40.00000	0.15000	-0.15000	---

Nome Plano peca_ref1	Operador Master	Hora 4:12:45 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_1-12_Z	0.00000	0.00000	0.05000	-0.05000	---

Oitava rodada de medições

ZEISS Calypso					
Plano Medição peca_ref1		Data 22 de Setembro de 2022			
No. Desenho * drawingno *		Hora 4:37:26 pm	Ordem OS		
Operador Master		MMC C32Bit	No. Sequencial Peça 8		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Resultados Globais Todas Características: 45 ...Tolerância: 43 ...Fora da Tolerância: 2 ...Acima Limite Alerta: 0 ...Não Calculado: 0 Total de Sistemas de coord.: 1 ...Não Calculado: 0 Total de elementos de texto: 0					
Distância_1-2_X	50.30095	50.20000	0.15000	-0.15000	0.10095
Distância_1-2_Y	0.09300	40.00000	0.15000	-0.15000	-39.75700
Distância_1-2_Z	0.13850	0.17509	0.05000	-0.05000	-0.03660
Distância_2-3_X	0.02402	0.00000	0.05000	-0.05000	0.02402
Distância_2-3_Y	39.95169	40.04485	0.15000	-0.15000	-0.09316
Distância_2-3_Z	0.00007	39.97509	0.15000	-0.15000	-39.82502
Distância_3-4_X	0.00818	0.00000	0.05000	-0.05000	0.00818
Distância_3-4_Y	0.00981	0.00000	0.05000	-0.05000	0.00981


ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 4:37:26 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_5-10_Z	0.01675	0.00000	0.05000	-0.05000	0.01675
Distância_6-7_X	0.02420	0.00000	0.05000	-0.05000	0.02420
Distância_6-7_Y	40.25100	40.16995	0.15000	-0.15000	0.08105
Distância_6-7_Z	0.00422	0.00000	0.05000	-0.05000	0.00422
Distância_6-9_X	50.23649	50.20000	0.15000	-0.15000	0.03649
Distância_6-9_Y	0.01585	0.00000	0.05000	-0.05000	0.01585
Distância_6-9_Z	0.02266	0.00000	0.05000	-0.05000	0.02266
Distância_7-8_X	50.21229	50.20000	0.15000	-0.15000	0.01229
Distância_7-8_Y	0.05663	0.08762	0.05000	-0.05000	-0.03098
Distância_7-8_Z	0.02266	0.00000	0.05000	-0.05000	0.02266
Distância_8-9_X	0.00000	0.00000	0.05000	-0.05000	0.00000
Distância_8-9_Y	40.32349	40.25757	0.15000	-0.15000	0.06592

ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 4:37:26 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_3-4_Z	39.94998	39.97509	0.15000	-0.15000	-0.02512
Distância_4-5_X	0.02404	0.00000	0.05000	-0.05000	0.02404
Distância_4-5_Y	39.99218	40.04485	0.15000	-0.15000	-0.05267
Distância_4-5_Z	0.01807	0.00000	0.05000	-0.05000	0.01807
Distância_4-11_X	50.26875	50.20000	0.15000	-0.15000	0.06875
Distância_4-11_Y	0.01464	0.00000	0.05000	-0.05000	0.01464
Distância_4-11_Z	0.01675	0.00000	0.05000	-0.05000	0.01675
Distância_5-6_X	0.00822	0.00000	0.05000	-0.05000	0.00822
Distância_5-6_Y	0.05184	0.06981	0.05000	-0.05000	-0.01797
Distância_5-6_Z	40.00762	40.00000	0.15000	-0.15000	0.00762
Distância_5-10_X	50.24471	50.20000	0.15000	-0.15000	0.04471
Distância_5-10_Y	0.01580	0.00000	0.05000	-0.05000	0.01580

ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 4:37:26 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_8-9_Z	0.00422	0.00000	0.05000	-0.05000	0.00422
Distância_9-10_X	0.00000	0.00000	0.05000	-0.05000	0.00000
Distância_9-10_Y	0.05179	0.06981	0.05000	-0.05000	-0.01802
Distância_9-10_Z	39.96821	40.00000	0.15000	-0.15000	-0.03179
Distância_10-11_X	0.00000	0.00000	0.05000	-0.05000	0.00000
Distância_10-11_Y	39.99101	40.04485	0.15000	-0.15000	-0.05384
Distância_10-11_Z	0.01808	0.00000	0.05000	-0.05000	0.01808
Distância_11-12_X	0.00000	0.00000	0.05000	-0.05000	0.00000
Distância_11-12_Y	0.00978	0.00000	0.05000	-0.05000	0.00978
Distância_11-12_Z	39.82830	39.80000	0.15000	-0.15000	0.02830
Distância_1-12_X	0.00000	0.00000	0.05000	-0.05000	0.00000
Distância_1-12_Y	40.03008	40.00000	0.15000	-0.15000	0.03008

ZEISS					
Nome Plano peca_ref1	Operador Master	Hora 4:37:26 pm	Data 22 de Setembro de 2022		
Atual	Nominal	Tol. Superior	Tol. Inferior	Desvio	
Distância_1-12_Z	0.00000	0.00000	0.05000	-0.05000	0.00000

Anexo C - Relatório de calibração tabuleiro de xadrez



INSTITUTO SENAI
DE INOVAÇÃO METALMECÂNICA

Instituto SENAI de Inovação em Metalmeccânica

Rede Brasileira de Calibração

RELATÓRIO DE MEDIÇÃO Nº 02824/22

Laboratório da área dimensional Ordem de serviço: 943 Data da calibração: 22/11/2022
 Data de emissão: 24/11/2022

1 - CLIENTE Instituto SENAI de Inovação em Metalmeccânica
 Avenida Getúlio Vargas, 3239 - São Leopoldo - RS

2 - IDENTIFICAÇÃO DO INSTRUMENTO CALIBRADO/ENSAIADO
 Placa Chessboard 12 x 8. Código de identificação do proprietário: 631-MET-216, Desenho: - - - -, Fabricante: - - - -, Modelo: - - - -, Código de série: - - - -.

3 - RASTREABILIDADE DOS PADRÕES UTILIZADOS
 DataLogger 631-MET-020, certificado Nº 631-MET-020, emitido por Novus, válido até 08/2023.
 Máquina de medição por coordenadas com cabeçote ótico 631-MET-075, certificado Nº 210420173688, emitido por Carl Zeiss, válido até 04/2023.

4 - PROCEDIMENTO DE MEDIÇÃO
 Procedimento interno: PRI 631-99 rev. 03
 Condições ambientais: Temperatura de (20,0 ± 0,5) °C e umidade ≤ 65 %.
 Método de medição: A placa foi posicionada diretamente sobre o desempenho da máquina de medição por coordenadas sem a necessidade de fixação, sendo alinhada pelas extremidades da marcação xadrez com eixo Y da MMC. O alinhamento foi estabelecido pela lateral superior alinhado com eixo X da MMC, a origem do eixo Z foi definido conforme ajuste de foco do cabeçote ótico da MMC. A cada lateral foi medida uma linha com 5 pontos, com a interseção entre 2 linhas formando um ponto na aresta do quadrado medido e retirada posição entre 2 pontos (ponto média entre arestas) conforme figura na página 12. Foram realizados 3 ciclos de medição sendo apresentado a média, foi utilizado cabeçote ótico com ampliação de 1x.


5 - RESULTADOS
 Ver resultados na página 2 de 12.

6 - INCERTEZA DE MEDIÇÃO
 A incerteza expandida de medição relatada é declarada como a incerteza padrão de medição multiplicada pelo fator de abrangência [k], o qual para uma distribuição t com [v_{eff}] graus de liberdade efetivos corresponde a uma probabilidade de abrangência de aproximadamente 95%. A incerteza padrão da medição foi determinada de acordo com a publicação EA-4/02. Os valores de k e v_{eff} são apresentados na tabela de resultados.


7 - NOTAS
 - - - -


Douglas Bergamo
 Signatário Autorizado

Este documento dispensa a assinatura física por ser autorizado eletronicamente, cumprindo integralmente os requisitos da Norma ABNT NBR ISO/IEC 17025.
 Este relatório atende aos requisitos de acreditação da Cgcre que avalia a competência do laboratório e comprovou sua rastreabilidade ao Sistema Internacional de Unidades - SI. A reprodução deste relatório só poderá ser feita de forma integral. Os resultados deste relatório referem-se exclusivamente a pago diverso submetido à medição nas condições especificadas, não sendo extensivos a qualquer item.



Instituto SENAI de Inovação em Metalmeccânica
 Av. Getúlio Vargas, 3239 - São Leopoldo / RS
 CEP 93025-753 - Tel.: (51) 3904-2691
 e-mail: servicos.isisim@senai.rs.gov.br






INSTITUTO SENAI
DE INOVAÇÃO METALMECÂNICA

RELATÓRIO DE MEDIÇÃO Nº 02824/22


Laboratório de calibração acreditado pela Cgcre de acordo com a ABNT NBR ISO/IEC 17025, sob o número CAL 0013.


RESULTADOS

Item	Característica	Valor Encontrado (mm)	Incerteza	k	Veff
1	Valor X_Simetria1	-0.012	0,004	4,53	2
2	Valor Y_Simetria1	-9.645	0,006	4,53	2
3	Valor X_Simetria2	9.622	0,004	4,53	2
4	Valor Y_Simetria2	0.004	0,004	4,53	2
5	Valor X_Simetria3	19.231	0,005	4,53	2
6	Valor Y_Simetria3	-9.637	0,004	4,53	2
7	Valor X_Simetria4	9.597	0,005	4,53	2
8	Valor Y_Simetria4	-19.285	0,006	4,53	2
9	Valor X_Simetria5	38.359	0,008	4,53	2
10	Valor Y_Simetria5	-9.633	0,005	4,53	2
11	Valor X_Simetria6	47.977	0,008	4,53	2
12	Valor Y_Simetria6	0.013	0,004	4,53	2
13	Valor X_Simetria7	57.563	0,005	4,53	2
14	Valor Y_Simetria7	-9.638	0,004	4,53	2
15	Valor X_Simetria8	47.945	0,004	4,53	2
16	Valor Y_Simetria8	-19.284	0,005	4,53	2
17	Valor X_Simetria9	76.729	0,005	4,53	2
18	Valor Y_Simetria9	-9.641	0,003	4,53	2
19	Valor X_Simetria10	86.345	0,007	4,53	2
20	Valor Y_Simetria10	0.002	0,003	4,53	2
21	Valor X_Simetria11	95.936	0,006	4,53	2
22	Valor Y_Simetria11	-9.650	0,004	4,53	2
23	Valor X_Simetria12	86.321	0,004	4,53	2
24	Valor Y_Simetria12	-19.292	0,004	4,53	2
25	Valor X_Simetria13	115.064	0,005	4,53	2
26	Valor Y_Simetria13	-9.662	0,002	3,31	3
27	Valor X_Simetria14	124.712	0,007	4,53	2
28	Valor Y_Simetria14	-0.002	0,004	4,53	2
29	Valor X_Simetria15	134.323	0,006	4,53	2
30	Valor Y_Simetria15	-9.674	0,007	4,53	2
31	Valor X_Simetria16	124.674	0,006	4,53	2
32	Valor Y_Simetria16	-19.333	0,006	4,53	2
33	Valor X_Simetria17	-19.200	0,003	4,53	2
34	Valor Y_Simetria17	-28.877	0,008	4,53	2
35	Valor X_Simetria18	-9.580	0,006	4,53	2
36	Valor Y_Simetria18	-19.246	0,006	4,53	2
37	Valor X_Simetria19	0.011	0,008	4,53	2
38	Valor Y_Simetria19	-28.887	0,006	4,53	2
39	Valor X_Simetria20	-9.608	0,004	4,53	2



Instituto SENAI de Inovação em Metalmeccânica
 Av. Getúlio Vargas, 3239 - São Leopoldo / RS
 CEP 93025-753 - Tel.: (51) 3904-2691
 e-mail: servicos.isisim@senai.rs.gov.br






INSTITUTO SENAI
DE INOVAÇÃO METALMECÂNICA

RELATÓRIO DE MEDIÇÃO Nº 02824/22


Laboratório de calibração acreditado pela Cgcre de acordo com a ABNT NBR ISO/IEC 17025, sob o número CAL 0013.


RESULTADOS

Item	Característica	Valor Encontrado (mm)	Incerteza	k	Veff
40	Valor Y_Simetria20	-38.517	0,006	4,53	2
41	Valor X_Simetria21	19.173	0,005	4,53	2
42	Valor Y_Simetria21	-28.867	0,005	4,53	2
43	Valor X_Simetria22	28.792	0,005	4,53	2
44	Valor Y_Simetria22	-19.234	0,004	4,53	2
45	Valor X_Simetria23	38.376	0,004	4,53	2
46	Valor Y_Simetria23	-28.875	0,003	4,53	2
47	Valor X_Simetria24	28.757	0,004	4,53	2
48	Valor Y_Simetria24	-38.507	0,002	3,31	3
49	Valor X_Simetria25	57.494	0,006	4,53	2
50	Valor Y_Simetria25	-28.882	0,004	4,53	2
51	Valor X_Simetria26	67.129	0,006	4,53	2
52	Valor Y_Simetria26	-19.243	0,004	4,53	2
53	Valor X_Simetria27	76.746	0,004	4,53	2
54	Valor Y_Simetria27	-28.893	0,003	4,53	2
55	Valor X_Simetria28	67.111	0,005	4,53	2
56	Valor Y_Simetria28	-38.532	0,004	4,53	2
57	Valor X_Simetria29	95.859	0,005	4,53	2
58	Valor Y_Simetria29	-28.909	0,002	3,31	3
59	Valor X_Simetria30	105.483	0,006	4,53	2
60	Valor Y_Simetria30	-19.268	0,003	4,53	2
61	Valor X_Simetria31	115.063	0,006	4,53	2
62	Valor Y_Simetria31	-28.924	0,004	4,53	2
63	Valor X_Simetria32	105.439	0,005	4,53	2
64	Valor Y_Simetria32	-38.565	0,003	4,53	2
65	Valor X_Simetria33	-0.054	0,005	4,53	2
66	Valor Y_Simetria33	-48.103	0,005	4,53	2
67	Valor X_Simetria34	9.583	0,005	4,53	2
68	Valor Y_Simetria34	-38.473	0,003	4,53	2
69	Valor X_Simetria35	19.192	0,004	4,53	2
70	Valor Y_Simetria35	-48.114	0,003	4,53	2
71	Valor X_Simetria36	9.554	0,003	4,53	2
72	Valor Y_Simetria36	-57.745	0,004	4,53	2
73	Valor X_Simetria37	38.312	0,003	4,53	2
74	Valor Y_Simetria37	-48.126	0,004	4,53	2
75	Valor X_Simetria38	47.930	0,004	4,53	2
76	Valor Y_Simetria38	-38.481	0,003	4,53	2
77	Valor X_Simetria39	57.505	0,003	3,31	3



Instituto SENAI de Inovação em Metalmeccânica
 Av. Getúlio Vargas, 3239 - São Leopoldo / RS
 CEP 93025-753 - Tel.: (51) 3904-2691
 e-mail: servicos.isisim@senai.rs.gov.br






INSTITUTO SENAI
DE INOVAÇÃO METALMECÂNICA

RELATÓRIO DE MEDIÇÃO Nº 02824/22


Laboratório de calibração acreditado pela Cgcre de acordo com a ABNT NBR ISO/IEC 17025, sob o número CAL 0013.


RESULTADOS

Item	Característica	Valor Encontrado (mm)	Incerteza	k	Veff
78	Valor Y_Simetria39	-48.122	0,002	2,52	6
79	Valor X_Simetria40	47.887	0,002	2,65	5
80	Valor Y_Simetria40	-57.767	0,003	4,53	2
81	Valor X_Simetria41	76.678	0,010	4,53	2
82	Valor Y_Simetria41	-48.149	0,005	4,53	2
83	Valor X_Simetria42	86.290	0,012	4,53	2
84	Valor Y_Simetria42	-38.508	0,005	4,53	2
85	Valor X_Simetria43	95.867	0,003	3,31	3
86	Valor Y_Simetria43	-48.157	0,004	4,53	2
87	Valor X_Simetria44	86.255	0,005	4,53	2
88	Valor Y_Simetria44	-57.798	0,004	4,53	2
89	Valor X_Simetria45	115.000	0,003	3,31	3
90	Valor Y_Simetria45	-48.177	0,006	4,53	2
91	Valor X_Simetria46	124.648	0,004	4,53	2
92	Valor Y_Simetria46	-38.531	0,006	4,53	2
93	Valor X_Simetria47	134.247	0,003	3,31	3
94	Valor Y_Simetria47	-48.183	0,006	4,53	2
95	Valor X_Simetria48	124.599	0,005	4,53	2
96	Valor Y_Simetria48	-57.830	0,006	4,53	2
97	Valor X_Simetria49	-19.256	0,004	4,53	2
98	Valor Y_Simetria49	-67.353	0,005	4,53	2
99	Valor X_Simetria50	-9.637	0,003	4,53	2
100	Valor Y_Simetria50	-57.694	0,005	4,53	2
101	Valor X_Simetria51	-0.049	0,003	4,53	2
102	Valor Y_Simetria51	-67.372	0,006	4,53	2
103	Valor X_Simetria52	-9.668	0,003	4,53	2
104	Valor Y_Simetria52	-77.031	0,006	4,53	2
105	Valor X_Simetria53	19.113	0,005	4,53	2
106	Valor Y_Simetria53	-67.375	0,004	4,53	2
107	Valor X_Simetria54	28.729	0,004	4,53	2
108	Valor Y_Simetria54	-57.701	0,005	4,53	2
109	Valor X_Simetria55	38.316	0,004	4,53	2
110	Valor Y_Simetria55	-67.379	0,007	4,53	2
111	Valor X_Simetria56	28.700	0,005	4,53	2
112	Valor Y_Simetria56	-77.054	0,006	4,53	2
113	Valor X_Simetria57	57.437	0,002	2,26	10
114	Valor Y_Simetria57	-67.408	0,002	3,31	3
115	Valor X_Simetria58	67.078	0,003	3,31	3



Instituto SENAI de Inovação em Metalmeccânica
 Av. Getúlio Vargas, 3239 - São Leopoldo / RS
 CEP 93025-753 - Tel.: (51) 3904-2691
 e-mail: servicos.isisim@senai.rs.gov.br






RELATÓRIO DE MEDIÇÃO Nº 02824/22


Laboratório de calibração acreditado pela Cgcre de acordo com a ABNT NBR ISO/IEC 17025, sob o número CAL 0013.

RESULTADOS

Item	Característica	Valor Encontrado (mm)	Incerteza	k	Veff
116	Valor Y_Simetria58	-57,739	0,005	4,53	2
117	Valor X_Simetria59	76,682	0,006	4,53	2
118	Valor Y_Simetria59	-67,427	0,003	4,53	2
119	Valor X_Simetria60	67,041	0,007	4,53	2
120	Valor Y_Simetria60	-77,097	0,002	2,32	9
121	Valor X_Simetria61	95,796	0,003	2,87	4
122	Valor Y_Simetria61	-67,449	0,004	4,53	2
123	Valor X_Simetria62	105,411	0,003	3,31	3
124	Valor Y_Simetria62	-57,775	0,005	4,53	2
125	Valor X_Simetria63	115,005	0,003	3,31	3
126	Valor Y_Simetria63	-67,470	0,004	4,53	2
127	Valor X_Simetria64	105,390	0,003	3,31	3
128	Valor Y_Simetria64	-77,144	0,004	4,53	2
129	Valor X_Simetria65	-0,119	0,004	4,53	2
130	Valor Y_Simetria65	-86,641	0,005	4,53	2
131	Valor X_Simetria66	9,521	0,003	4,53	2
132	Valor Y_Simetria66	-77,000	0,005	4,53	2
133	Valor X_Simetria67	19,128	0,004	4,53	2
134	Valor Y_Simetria67	-86,656	0,005	4,53	2
135	Valor X_Simetria68	9,489	0,004	4,53	2
136	Valor Y_Simetria68	-96,297	0,004	4,53	2
137	Valor X_Simetria69	38,246	0,005	4,53	2
138	Valor Y_Simetria69	-86,669	0,007	4,53	2
139	Valor X_Simetria70	47,864	0,004	4,53	2
140	Valor Y_Simetria70	-77,028	0,006	4,53	2
141	Valor X_Simetria71	57,447	0,004	4,53	2
142	Valor Y_Simetria71	-86,694	0,004	4,53	2
143	Valor X_Simetria72	47,830	0,005	4,53	2
144	Valor Y_Simetria72	-96,334	0,004	4,53	2
145	Valor X_Simetria73	76,601	0,005	4,53	2
146	Valor Y_Simetria73	-86,715	0,005	4,53	2
147	Valor X_Simetria74	86,216	0,005	4,53	2
148	Valor Y_Simetria74	-77,073	0,002	2,43	7
149	Valor X_Simetria75	95,803	0,003	2,87	4
150	Valor Y_Simetria75	-86,743	0,004	4,53	2
151	Valor X_Simetria76	86,187	0,002	2,65	5
152	Valor Y_Simetria76	-96,384	0,005	4,53	2
153	Valor X_Simetria77	114,931	0,004	4,53	2

Instituto SENAI de Inovação em Metalmeccânica
 Av. Getúlio Vargas, 3239 - São Leopoldo / RS
 CEP 93025-753 - Tel.: (51) 3904-2691
 e-mail: servicos.isism@senairs.org.br





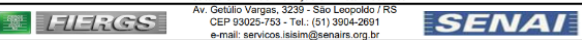
RELATÓRIO DE MEDIÇÃO Nº 02824/22


Laboratório de calibração acreditado pela Cgcre de acordo com a ABNT NBR ISO/IEC 17025, sob o número CAL 0013.

RESULTADOS

Item	Característica	Valor Encontrado (mm)	Incerteza	k	Veff
154	Valor Y_Simetria77	-86,763	0,006	4,53	2
155	Valor X_Simetria78	124,575	0,003	3,31	3
156	Valor Y_Simetria78	-77,112	0,006	4,53	2
157	Valor X_Simetria79	134,183	0,005	4,53	2
158	Valor Y_Simetria79	-86,770	0,006	4,53	2
159	Valor X_Simetria80	124,539	0,005	4,53	2
160	Valor Y_Simetria80	-96,422	0,005	4,53	2
161	Valor X_Simetria81	-19,330	0,005	4,53	2
162	Valor Y_Simetria81	-105,875	0,006	4,53	2
163	Valor X_Simetria82	-9,697	0,007	4,53	2
164	Valor Y_Simetria82	-96,235	0,006	4,53	2
165	Valor X_Simetria83	-0,119	0,003	4,53	2
166	Valor Y_Simetria83	-105,889	0,006	4,53	2
167	Valor X_Simetria84	-9,751	0,007	4,53	2
168	Valor Y_Simetria84	-115,530	0,006	4,53	2
169	Valor X_Simetria85	19,042	0,004	4,53	2
170	Valor Y_Simetria85	-105,891	0,005	4,53	2
171	Valor X_Simetria86	28,667	0,003	4,53	2
172	Valor Y_Simetria86	-96,264	0,006	4,53	2
173	Valor X_Simetria87	38,242	0,004	4,53	2
174	Valor Y_Simetria87	-105,923	0,003	3,31	3
175	Valor X_Simetria88	28,617	0,005	4,53	2
176	Valor Y_Simetria88	-115,550	0,002	2,00	infinito
177	Valor X_Simetria89	57,361	0,003	3,31	3
178	Valor Y_Simetria89	-105,950	0,004	4,53	2
179	Valor X_Simetria90	67,007	0,003	3,31	3
180	Valor Y_Simetria90	-96,322	0,004	4,53	2
181	Valor X_Simetria91	76,603	0,004	4,53	2
182	Valor Y_Simetria91	-105,972	0,003	4,53	2
183	Valor X_Simetria92	66,957	0,004	4,53	2
184	Valor Y_Simetria92	-115,600	0,002	2,87	4
185	Valor X_Simetria93	95,721	0,005	4,53	2
186	Valor Y_Simetria93	-105,993	0,004	4,53	2
187	Valor X_Simetria94	105,349	0,005	4,53	2
188	Valor Y_Simetria94	-96,360	0,005	4,53	2
189	Valor X_Simetria95	114,931	0,005	4,53	2
190	Valor Y_Simetria95	-106,011	0,006	4,53	2
191	Valor X_Simetria96	105,303	0,005	4,53	2

Instituto SENAI de Inovação em Metalmeccânica
 Av. Getúlio Vargas, 3239 - São Leopoldo / RS
 CEP 93025-753 - Tel.: (51) 3904-2691
 e-mail: servicos.isism@senairs.org.br






RELATÓRIO DE MEDIÇÃO Nº 02824/22


Laboratório de calibração acreditado pela Cgcre de acordo com a ABNT NBR ISO/IEC 17025, sob o número CAL 0013.

RESULTADOS

Item	Característica	Valor Encontrado (mm)	Incerteza	k	Veff
192	Valor Y_Simetria96	-115,644	0,006	4,53	2
193	Valor X_Simetria97	-0,202	0,005	4,53	2
194	Valor Y_Simetria97	-125,164	0,007	4,53	2
195	Valor X_Simetria98	9,435	0,003	4,53	2
196	Valor Y_Simetria98	-115,500	0,006	4,53	2
197	Valor X_Simetria99	19,039	0,005	4,53	2
198	Valor Y_Simetria99	-125,182	0,008	4,53	2
199	Valor X_Simetria100	9,402	0,006	4,53	2
200	Valor Y_Simetria100	-134,646	0,008	4,53	2
201	Valor X_Simetria101	38,164	0,004	4,53	2
202	Valor Y_Simetria101	-125,194	0,004	4,53	2
203	Valor X_Simetria102	47,782	0,003	4,53	2
204	Valor Y_Simetria102	-115,529	0,004	4,53	2
205	Valor X_Simetria103	57,352	0,002	2,37	8
206	Valor Y_Simetria103	-125,213	0,004	4,53	2
207	Valor X_Simetria104	47,734	0,006	4,53	2
208	Valor Y_Simetria104	-134,878	0,003	3,31	3
209	Valor X_Simetria105	76,500	0,005	4,53	2
210	Valor Y_Simetria105	-125,243	0,005	4,53	2
211	Valor X_Simetria106	86,142	0,006	4,53	2
212	Valor Y_Simetria106	-115,579	0,003	3,31	3
213	Valor X_Simetria107	95,715	0,006	4,53	2
214	Valor Y_Simetria107	-125,260	0,012	4,53	2
215	Valor X_Simetria108	86,073	0,005	4,53	2
216	Valor Y_Simetria108	-134,924	0,012	4,53	2
217	Valor X_Simetria109	114,838	0,004	4,53	2
218	Valor Y_Simetria109	-125,276	0,008	4,53	2
219	Valor X_Simetria110	124,484	0,003	3,31	3
220	Valor Y_Simetria110	-115,616	0,003	3,31	3
221	Valor X_Simetria111	134,088	0,006	4,53	2
222	Valor Y_Simetria111	-125,298	0,004	4,53	2
223	Valor X_Simetria112	124,442	0,007	4,53	2
224	Valor Y_Simetria112	-134,958	0,008	4,53	2
225	Valor X_Simetria113	-19,423	0,005	4,53	2
226	Valor Y_Simetria113	-144,435	0,005	4,53	2
227	Valor X_Simetria114	-9,792	0,007	4,53	2
228	Valor Y_Simetria114	-134,795	0,005	4,53	2
229	Valor X_Simetria115	-0,215	0,007	4,53	2

Instituto SENAI de Inovação em Metalmeccânica
 Av. Getúlio Vargas, 3239 - São Leopoldo / RS
 CEP 93025-753 - Tel.: (51) 3904-2691
 e-mail: servicos.isism@senairs.org.br






RELATÓRIO DE MEDIÇÃO Nº 02824/22

Laboratório de calibração acreditado pela Cgcre de acordo com a ABNT NBR ISO/IEC 17025, sob o número CAL 0013.

RESULTADOS

Item	Característica	Valor Encontrado (mm)	Incerteza	k	Veff
230	Valor Y_Simetria115	-144,454	0,006	4,53	2
231	Valor X_Simetria116	-9,846	0,005	4,53	2
232	Valor Y_Simetria116	-154,094	0,005	4,53	2
233	Valor X_Simetria117	18,949	0,006	4,53	2
234	Valor Y_Simetria117	-144,462	0,002	2,87	4
235	Valor X_Simetria118	28,574	0,006	4,53	2
236	Valor Y_Simetria118	-134,822	0,003	3,31	3
237	Valor X_Simetria119	38,149	0,006	4,53	2
238	Valor Y_Simetria119	-144,479	0,004	4,53	2
239	Valor X_Simetria120	28,524	0,006	4,53	2
240	Valor Y_Simetria120	-154,118	0,004	4,53	2
241	Valor X_Simetria121	57,299	0,006	4,53	2
242	Valor Y_Simetria121	-144,503	0,006	4,53	2
243	Valor X_Simetria122	66,917	0,005	4,53	2
244	Valor Y_Simetria122	-134,860	0,008	4,53	2
245	Valor X_Simetria123	76,504	0,006	4,53	2
246	Valor Y_Simetria123	-144,524	0,006	4,53	2
247	Valor X_Simetria124	66,857	0,006	4,53	2
248	Valor Y_Simetria124	-154,167	0,005	4,53	2
249	Valor X_Simetria125	95,623	0,005	4,53	2
250	Valor Y_Simetria125	-144,541	0,005	4,53	2
251	Valor X_Simetria126	105,250	0,007	4,53	2
252	Valor Y_Simetria126	-134,907	0,003	3,31	3
253	Valor X_Simetria127	114,829	0,007	4,53	2
254	Valor Y_Simetria127	-144,571	0,003	3,31	3
255	Valor X_Simetria128	105,202	0,006	4,53	2
256	Valor Y_Simetria128	-154,205	0,005	4,53	2
257	Valor X_Simetria129	-0,300	0,007	4,53	2
258	Valor Y_Simetria129	-163,696	0,006	4,53	2
259	Valor X_Simetria130	9,343	0,006	4,53	2
260	Valor Y_Simetria130	-154,071	0,005	4,53	2
261	Valor X_Simetria131	18,938	0,006	4,53	2
262	Valor Y_Simetria131	-163,709	0,003	3,31	3
263	Valor X_Simetria132	9,294	0,008	4,53	2
264	Valor Y_Simetria132	-173,334	0,004	4,53	2
265	Valor X_Simetria133	38,063	0,007	4,53	2
266	Valor Y_Simetria133	-163,727	0,004	4,53	2
267	Valor X_Simetria134	47,677	0,005	4,53	2

Instituto SENAI de Inovação em Metalmeccânica
 Av. Getúlio Vargas, 3239 - São Leopoldo / RS
 CEP 93025-753 - Tel.: (51) 3904-2691
 e-mail: servicos.isism@senairs.org.br



INSTITUTO SENAI
DE INOVAÇÃO METALMECÂNICA

RELATÓRIO DE MEDIÇÃO N° 02824/22

Laboratório de calibração acreditado pela Cgcre de acordo com a ABNT NBR ISO/IEC 17025, sob o número CAL 0013.

RESULTADOS

Item	Característica	Valor Encontrado (mm)	Incerteza	k	Veff
266	Valor Y_Simetria134	-154,101	0,003	2,87	4
269	Valor X_Simetria135	57,253	0,006	4,53	2
270	Valor Y_Simetria135	-163,747	0,004	4,53	2
271	Valor X_Simetria136	47,839	0,008	4,53	2
272	Valor Y_Simetria136	-173,373	0,005	4,53	2
273	Valor X_Simetria137	76,416	0,005	4,53	2
274	Valor Y_Simetria137	-163,778	0,005	4,53	2
275	Valor X_Simetria138	86,030	0,006	4,53	2
276	Valor Y_Simetria138	-154,150	0,005	4,53	2
277	Valor X_Simetria139	95,816	0,007	4,53	2
278	Valor Y_Simetria139	-163,788	0,004	4,53	2
279	Valor X_Simetria140	86,001	0,006	4,53	2
280	Valor Y_Simetria140	-173,417	0,003	3,31	3
281	Valor X_Simetria141	114,739	0,007	4,53	2
282	Valor Y_Simetria141	-163,812	0,005	4,53	2
283	Valor X_Simetria142	124,362	0,006	4,53	2
284	Valor Y_Simetria142	-154,192	0,008	4,53	2
285	Valor X_Simetria143	133,996	0,007	4,53	2
286	Valor Y_Simetria143	-163,824	0,012	4,53	2
287	Valor X_Simetria144	124,343	0,008	4,53	2
288	Valor Y_Simetria144	-173,444	0,009	4,53	2
289	Valor X_Simetria145	-19,522	0,004	4,53	2
290	Valor Y_Simetria145	-182,921	0,006	4,53	2
291	Valor X_Simetria146	-9,899	0,004	4,53	2
292	Valor Y_Simetria146	-173,284	0,004	4,53	2
293	Valor X_Simetria147	-0,314	0,007	4,53	2
294	Valor Y_Simetria147	-182,951	0,005	4,53	2
295	Valor X_Simetria148	-9,847	0,007	4,53	2
296	Valor Y_Simetria148	-192,608	0,006	4,53	2
297	Valor X_Simetria149	18,851	0,008	4,53	2
298	Valor Y_Simetria149	-182,965	0,007	4,53	2
299	Valor X_Simetria150	28,469	0,008	4,53	2
300	Valor Y_Simetria150	-173,304	0,007	4,53	2
301	Valor X_Simetria151	38,057	0,008	4,53	2
302	Valor Y_Simetria151	-182,985	0,006	4,53	2
303	Valor X_Simetria152	28,440	0,008	4,53	2
304	Valor Y_Simetria152	-192,645	0,005	4,53	2
305	Valor X_Simetria153	57,177	0,007	4,53	2

Instituto SENAI de Inovação em Metalmeccânica
Av. Getúlio Vargas, 3239 - São Leopoldo / RS
CEP 93025-753 - Tel.: (51) 3904-2691
e-mail: servicos.isisim@senairs.org.br

INSTITUTO SENAI
DE INOVAÇÃO METALMECÂNICA

RELATÓRIO DE MEDIÇÃO N° 02824/22

Laboratório de calibração acreditado pela Cgcre de acordo com a ABNT NBR ISO/IEC 17025, sob o número CAL 0013.

RESULTADOS

Item	Característica	Valor Encontrado (mm)	Incerteza	k	Veff
306	Valor Y_Simetria153	-163,009	0,006	4,53	2
307	Valor X_Simetria154	66,814	0,007	4,53	2
308	Valor Y_Simetria154	-173,346	0,007	4,53	2
309	Valor X_Simetria155	76,418	0,007	4,53	2
310	Valor Y_Simetria155	-183,028	0,008	4,53	2
311	Valor X_Simetria156	66,781	0,007	4,53	2
312	Valor Y_Simetria156	-192,691	0,007	4,53	2
313	Valor X_Simetria157	95,535	0,008	4,53	2
314	Valor Y_Simetria157	-183,080	0,008	4,53	2
315	Valor X_Simetria158	105,155	0,006	4,53	2
316	Valor Y_Simetria158	-173,396	0,006	4,53	2
317	Valor X_Simetria159	114,744	0,008	4,53	2
318	Valor Y_Simetria159	-183,075	0,005	4,53	2
319	Valor X_Simetria160	105,124	0,010	4,53	2
320	Valor Y_Simetria160	-192,738	0,007	4,53	2
321	Valor X_Simetria161	-0,399	0,009	4,53	2
322	Valor Y_Simetria161	-202,211	0,004	3,31	3
323	Valor X_Simetria162	9,253	0,008	4,53	2
324	Valor Y_Simetria162	-192,264	0,003	3,31	3
325	Valor X_Simetria163	18,840	0,009	4,53	2
326	Valor Y_Simetria163	-202,223	0,003	2,87	4
327	Valor X_Simetria164	9,188	0,010	4,53	2
328	Valor Y_Simetria164	-211,849	0,004	3,31	3
329	Valor X_Simetria165	37,974	0,009	4,53	2
330	Valor Y_Simetria165	-202,241	0,005	4,53	2
331	Valor X_Simetria166	47,598	0,010	4,53	2
332	Valor Y_Simetria166	-192,626	0,005	4,53	2
333	Valor X_Simetria167	57,170	0,009	4,53	2
334	Valor Y_Simetria167	-202,285	0,007	4,53	2
335	Valor X_Simetria168	47,546	0,008	4,53	2
336	Valor Y_Simetria168	-211,879	0,007	4,53	2
337	Valor X_Simetria169	76,328	0,009	4,53	2
338	Valor Y_Simetria169	-202,288	0,003	2,87	4
339	Valor X_Simetria170	85,953	0,008	4,53	2
340	Valor Y_Simetria170	-192,672	0,005	4,53	2
341	Valor X_Simetria171	95,531	0,008	4,53	2
342	Valor Y_Simetria171	-202,297	0,006	4,53	2
343	Valor X_Simetria172	85,905	0,010	4,53	2

Instituto SENAI de Inovação em Metalmeccânica
Av. Getúlio Vargas, 3239 - São Leopoldo / RS
CEP 93025-753 - Tel.: (51) 3904-2691
e-mail: servicos.isisim@senairs.org.br

INSTITUTO SENAI
DE INOVAÇÃO METALMECÂNICA

RELATÓRIO DE MEDIÇÃO N° 02824/22

Laboratório de calibração acreditado pela Cgcre de acordo com a ABNT NBR ISO/IEC 17025, sob o número CAL 0013.

RESULTADOS

Item	Característica	Valor Encontrado (mm)	Incerteza	k	Veff
344	Valor Y_Simetria172	-211,913	0,003	3,31	3
345	Valor X_Simetria173	114,680	0,009	4,53	2
346	Valor Y_Simetria173	-202,301	0,007	4,53	2
347	Valor X_Simetria174	124,311	0,008	4,53	2
348	Valor Y_Simetria174	-192,708	0,006	4,53	2
349	Valor X_Simetria175	133,909	0,008	4,53	2
350	Valor Y_Simetria175	-202,323	0,007	4,53	2
351	Valor X_Simetria176	124,258	0,009	4,53	2
352	Valor Y_Simetria176	-211,915	0,008	4,53	2
353	Valor X_Simetria177	-19,627	0,009	4,53	2
354	Valor Y_Simetria177	-221,360	0,003	2,52	6
355	Valor X_Simetria178	-9,993	0,009	4,53	2
356	Valor Y_Simetria178	-211,783	0,004	3,31	3
357	Valor X_Simetria179	-0,413	0,010	4,53	2
358	Valor Y_Simetria179	-221,383	0,008	4,53	2
359	Valor X_Simetria180	-10,047	0,010	4,53	2
360	Valor Y_Simetria180	-230,960	0,007	4,53	2
361	Valor X_Simetria181	18,750	0,011	4,53	2
362	Valor Y_Simetria181	-221,390	0,003	3,31	3
363	Valor X_Simetria182	28,382	0,011	4,53	2
364	Valor Y_Simetria182	-211,812	0,003	3,31	3
365	Valor X_Simetria183	37,980	0,011	4,53	2
366	Valor Y_Simetria183	-221,405	0,004	3,31	3
367	Valor X_Simetria184	28,328	0,011	4,53	2
368	Valor Y_Simetria184	-230,983	0,005	4,53	2
369	Valor X_Simetria185	57,083	0,009	4,53	2
370	Valor Y_Simetria185	-221,422	0,005	4,53	2
371	Valor X_Simetria186	66,725	0,009	4,53	2
372	Valor Y_Simetria186	-211,852	0,002	2,52	6
373	Valor X_Simetria187	76,332	0,011	4,53	2
374	Valor Y_Simetria187	-221,440	0,007	4,53	2
375	Valor X_Simetria188	66,689	0,011	4,53	2
376	Valor Y_Simetria188	-231,010	0,003	3,31	3
377	Valor X_Simetria189	95,450	0,011	4,53	2
378	Valor Y_Simetria189	-221,446	0,008	4,53	2
379	Valor X_Simetria190	105,075	0,010	4,53	2
380	Valor Y_Simetria190	-211,872	0,008	4,53	2
381	Valor X_Simetria191	114,659	0,010	4,53	2

Instituto SENAI de Inovação em Metalmeccânica
Av. Getúlio Vargas, 3239 - São Leopoldo / RS
CEP 93025-753 - Tel.: (51) 3904-2691
e-mail: servicos.isisim@senairs.org.br

INSTITUTO SENAI
DE INOVAÇÃO METALMECÂNICA

RELATÓRIO DE MEDIÇÃO N° 02824/22

Laboratório de calibração acreditado pela Cgcre de acordo com a ABNT NBR ISO/IEC 17025, sob o número CAL 0013.

RESULTADOS

Item	Característica	Valor Encontrado (mm)	Incerteza	k	Veff
382	Valor Y_Simetria191	-221,451	0,002	2,37	8
383	Valor X_Simetria192	105,035	0,011	4,53	2
384	Valor Y_Simetria192	-231,025	0,003	2,87	4

FIGURA

Instituto SENAI de Inovação em Metalmeccânica
Av. Getúlio Vargas, 3239 - São Leopoldo / RS
CEP 93025-753 - Tel.: (51) 3904-2691
e-mail: servicos.isisim@senairs.org.br