Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Reconstrução tridimensional de folhas de feijão a partir de múltiplas imagens

Artur Welerson Sott Meyer

JUIZ DE FORA
MARÇO, 2025

Reconstrução tridimensional de folhas de feijão a partir de múltiplas imagens

Artur Welerson Sott Meyer

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Marcelo Bernardes Vieira

JUIZ DE FORA
MARÇO, 2025

RECONSTRUÇÃO TRIDIMENSIONAL DE FOLHAS DE FEIJÃO A PARTIR DE MÚLTIPLAS IMAGENS

Artur Welerson Sott Meyer

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Marcelo Bernardes Vieira
D.Sc em Ciência da Computação

Luiz Maurílio da Silva Maciel
D.Sc em Engenharia de Sistemas de Computação

Saulo Moraes Villela
D.Sc em Engenharia de Sistemas e Computação

JUIZ DE FORA
14 DE MARÇO, 2025

Aos meus amigos e irmãos.

Aos pais, pelo apoio e sustento.

Resumo

A integração da tecnologia no âmbito da agricultura e da biologia tem-se ampliado com o passar dos anos e, com isso, trouce grandes benefícios aos profissionais desses segmentos. Pesquisas sobre novos métodos para auxiliar essas áreas são de grande importância econômica. Nesse contexto, esta monografia aplica métodos da área de visão computacional para reconstruir folhas de feijão. Mais especificamente, são usadas as abordagens SfM (*Structure from Motion*) e estereoscopia monocular para promover uma análise não destrutiva dessas plantas, ou seja, estudá-las sem a necessidade de romoção das folhas. Busca-se complementar a base de dados de folhas de feijão utilizada neste trabalho com a adição da geometria dessas folhas. Essa base de dados é utilizada para treinar redes neurais profundas para estimar a área das folhas. Portanto, espera-se que o resultado deste trabalho reduza os erros de medição dessas redes ao considerar a própria geometria das folhas no treinamento. Resultados experimentais indicam que uma adequada reconstrução 3D de folhas de feijão a partir de múltiplas imagens pode ser alcançada, embora haja alguns desafios na configuração dos hiperparâmetros do método proposto.

Palavras-chave: reconstrução 3D de folhas de feijão, SfM, estereoscopia, estimativa de área foliar, visão computacional.

Abstract

The integration of technology in the field of agriculture and biology has expanded over the years, bringing great benefits to professionals in these segments. Research into new methods to help these areas is of great economic importance. In this context, this monograph applies computer vision methods to reconstruct bean leaves. More specifically, the SfM (Structure from Motion) and monocular stereoscopy approaches are used to promote a non-destructive analysis of these plants, i.e. to study them without the need to remove the leaves. The aim is to complement the database of bean leaves used in this work by adding the geometry of these leaves. This database is used to train deep neural networks to estimate leaf area. Therefore, the results of this work are expected to reduce the measurement errors of these networks by considering the geometry of the leaves themselves in the training. Experimental results indicate that an adequate 3D reconstruction of bean leaves from multiple images can be achieved, although there are some challenges in configuring the hyperparameters of the proposed method.

Keywords: 3D reconstruction of bean leaves, structure from motion, stereoscopy, leaf area estimation, 3D reconstruction, computer vision.

Agradecimentos

A todos os meus parentes, pelo encorajamento e apoio. Ao professor Marcelo pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria. Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

CNNs	Convolutional Neural Network
DoG	*Difference of Gaussians*
EPI	Imagem de Plano Epipolar
IA	Inteligência Artificial
MVS	*Multi-View Stereo*
PnP	*Perspective-n-Point*
SGM	*Semi-Global Matching*
SfM	*Structure from Motion*

1 Introdução

1.1 Contextualização do Problema

A agricultura é fundamental para a economia brasileira e para o abastecimento global. Portanto, monitorar o crescimento das plantas é crucial para atender à demanda mundial por alimentos e produtos agrícolas. Reconhecendo isso, pode-se considerar que as características fenotípicas, ou seja, aspectos visíveis como estrutura e tamanho das folhas, são essenciais para o estudo sobre a saúde e o desenvolvimento das lavouras, já que, estão diretamente relacionadas a fenômenos como fotossíntese e transpiração desses seres. Assim, métodos eficientes de monitoramento desses aspectos são cada vez mais necessários. Primeiramente, serão discutidos a motivação por trás do desenvolvimento de novos métodos para avaliação de plantações, será descrito os principais desafios associados ao problema, destacada a relevância do trabalho e em seguida é feita a descrição dos objetivos.

Métodos tradicionais de coleta de dados em plantações frequentemente requerem procedimentos invasivos e manuais, como a retirada de amostras do solo ou da vegetação, que podem ser demorados e dispendiosos. Diante disso, surge um interesse crescente em métodos mais eficientes e econômicos para a coleta de informações agrícolas. Esse cenário favorece a adoção da visão computacional, que propõe transformar a agricultura com métodos de monitoramento não invasivos e automatizados.

Este trabalho propõe uma abordagem não destrutiva, ou seja, fazer a análise das folhas sem a necessidade de removê-las, utilizando uma base de dados de imagens de folhas de feijão, a qual, possui imagens de diferentes poses para uma mesma folha. Desse modo, torna-se viável aplicar o método SfM (*Structure from Motion*), uma técnica fotogramétrica, que permite combinar imagens de uma mesma folha para criar um modelo 3D e obter informações morfológicas das folhas.

1.2 Motivação

A busca incessante pela eficiência na agricultura transcende a questão de produtividade, configurando-se como um imperativo de sustentabilidade e responsabilidade global. Em um cenário de crescimento populacional exponencial, garantir a segurança alimentar torna-se um desafio cada vez mais complexo e urgente. O cultivo do feijão, um dos pilares da dieta brasileira e de inúmeras outras culturas globais, exemplifica a relevância da agricultura para a sociedade. Contudo, as práticas tradicionais de monitoramento das plantações frequentemente se revelam ineficazes, podendo até prejudicar as culturas que pretendem proteger. A introdução de métodos não invasivos e baseados em dados para a análise fenotípica das folhas de feijão pode representar um avanço na gestão e melhoria da produção agrícola.

Este estudo investiga o papel da computação como uma aliada na resolução de desafios multidisciplinares. Através da reconstrução tridimensional das folhas de feijão a partir de múltiplas imagens, objetiva-se fornecer aos agricultores e cientistas uma ferramenta para o monitoramento preciso do crescimento das plantas. Dessa forma, espera-se contribuir para uma agricultura mais inteligente, que maximize a produção ao mesmo tempo, em que minimiza o impacto ambiental.

1.3 Objetivos

Nesta seção, serão apresentados os objetivos deste trabalho, que visam o desenvolvimento de um método baseado em visão computacional para a reconstrução tridimensional não destrutiva de folhas de feijão.

Desse modo busca-se como objetivo geral desenvolver um método baseado em visão computacional para a reconstrução tridimensional não destrutiva de folhas de feijão, visando melhorar posteriormente a precisão na estimativa da área foliar e contribuir para o monitoramento eficiente do crescimento dessas plantas em ambientes agrícolas.

E como objetivos específicos:

- Aplicar e avaliar a técnica de SfM e estereoscopia na reconstrução 3D de folhas de feijão, utilizando imagens digitais capturadas em diferentes ângulos;

- Ampliar a base de dados existente com informações geométricas tridimensionais das folhas, obtidas através das reconstruções 3D, para enriquecer as análises fenotípicas.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte forma, visando proporcionar uma leitura fluída e uma compreensão clara dos objetivos, métodos e resultados alcançados:

- **Capítulo 2 - Fundamentação Teórica:** Este capítulo descreve os conceitos teóricos aplicadas no método proposto;
- **Capítulo 3 - Método proposto:** Apresenta o conjunto de dados utilizados e detalha o método proposto para reconstrução das folhas de feijão;
- **Capítulo 4 - Resultados experimentais:** Detalha a configuração experimental, incluindo as métricas de avaliação, e discute os resultados obtidos;
- **Capítulo 5 - Conclusão e trabalhos futuros:** Conclui o trabalho, resumindo as principais contribuições e os resultados alcançados. Discute as limitações do estudo atual e propõe direções para pesquisas futuras.

A organização deste trabalho foi pensada para facilitar o entendimento da problemática abordada, das soluções propostas e dos resultados alcançados, contribuindo assim para o avanço do conhecimento na interseção da computação e da agronomia.

2 Fundamentação Teórica

Este capítulo apresenta os principais conceitos e teorias que fundamentam o processo de reconstrução tridimensional de objetos a partir de imagens bidimensionais, com foco na aplicação da técnica SfM para o estudo de folhas de feijão.

Inicialmente, abordam-se os conceitos fundamentais de visão computacional e fotogrametria, que constituem a base teórica para o SfM. Em seguida, exploram-se os princípios específicos do SfM, discutindo como essa técnica permite a geração de modelos tridimensionais a partir de um conjunto de imagens capturadas sob diferentes pontos de vista.

Este embasamento teórico é essencial para a compreensão da metodologia adotada, que será descrita em detalhes no capítulo 3, e para justificar a escolha dos métodos e ferramentas aplicados no desenvolvimento deste trabalho.

2.1 Extração de características invariantes

Na fotogrametria, é recomendável reduzir o conteúdo das imagens para diminuir o custo computacional, concentrando-se em áreas de interesse. Dessa forma, a extração de características permite transformar a imagem em um conjunto de pontos distintos acompanhados de suas descrições específicas. Esses pontos de interesse podem ser utilizados posteriormente para identificar correspondências entre diferentes imagens.

No presente trabalho, foi abordado o algoritmo SIFT (*Scale-Invariant Feature Transform*) Lowe (1999), um algoritmo de visão computacional utilizado para detectar, descrever e combinar características em imagens. Em Tareen e Saleem (2018) foram comparados diferentes algoritmos de detecção de características como AKAZE, BRISK, ORB, SIFT e, em geral, o SIFT foi considerado o mais preciso com relação a variações de escala, rotação e afins. Portanto, ele foi usado a fim de buscar obter resultados mais precisos para as reconstruções. Nos próximos parágrafos serão descritos os conceitos teóricos que fundamentam esse algoritmo.

Para descrever um ponto, pode-se considerar a vizinhança local à qual pertence, verificando os valores de intensidade e os gradientes presentes na imagem para criar um vetor descritor. Assim, obtém-se os pontos-chave que representam um ponto na imagem e o descritor, um vetor que descreve uma vizinhança local pertencente a um ponto-chave.

2.1.1 Pirâmide de escala-espaço

Para detectar esses pontos de importância e definir seus descritores em uma imagem, o SIFT aplica a pirâmide de escalas.

A pirâmide de escala-espaço é uma estrutura que facilita a identificação de características em múltiplas escalas, auxiliando na detecção de pontos de interesse que permanecem visíveis independentemente de transformações. Essa técnica se destaca nessa tarefa, explorando a propriedade de que pontos de interesse robustos devem ser localizáveis mesmo após transformações de escala e pequenas perturbações.

Uma forma de encontrar pontos de interesse é procurar áreas de alta frequência na imagem, que ainda são perceptíveis em situações de resolução reduzida ou desfoque. Uma maneira de fazer isso é usando o LoG (*Laplacian of Gaussian*), que consiste em aplicar um filtro gaussiano seguido de um filtro laplaciano. O filtro laplaciano é frequentemente utilizado para detectar partes da imagem com alta mudança de intensidade, pois em sua fórmula:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}, \tag{2.1}$$

calcula-se a segunda derivada parcial de uma imagem, destacando áreas com variações de intensidade, enquanto áreas com intensidades constantes tendem a ser zeradas. O filtro gaussiano é aplicado antes, pois o Laplaciano é sensível a ruídos, e essa suavização inicial ajuda a reduzir o impacto causado por eles. No entanto, o cálculo do LoG pode ser computacionalmente custoso.

Para contornar essa limitação, o SIFT utiliza a operação de diferenças gaussianas (DoG, *Difference of Gaussians*). O DoG aproxima o LoG ao subtrair duas imagens suavizadas com o filtro gaussiano, mas com desvios-padrão ligeiramente diferentes. Essa aproximação é vantajosa, por reduzir a complexidade computacional, enquanto ainda

destaca os pontos de interesse. A função gaussiana, dada por:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}, \tag{2.2}$$

é um filtro passa-baixas que suaviza as frequências da imagem, atenuando o ruído e destacando estruturas de maior escala. A aplicação repetida desse filtro com valores crescentes de desvio padrão (σ) em certa escala, que varia de acordo com sua posição na pirâmide, gera uma sequência de imagens suavizadas, formando uma pirâmide de escala-espaço. A suavização em diferentes escalas garante que os pontos selecionados sejam menos sensíveis ao ruído, tornando o SIFT funcional a diferentes tamanhos de imagem.

A base do SIFT reside na detecção de extremos locais proporcionados pela pirâmide de imagens construída a partir de convoluções com filtros gaussianos de diferentes escalas:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{2.3}$$

onde $L(x, y, \sigma)$ é a imagem desfocada, $G(x, y, \sigma)$ o filtro gaussiano e $I(x, y)$ a imagem original. A diferença entre duas imagens consecutivas nessa pirâmide é resulta no DoG. Nessa etapa, o objetivo é enfatizar as áreas de maior variação na imagem, o que é realizado subtraindo as imagens de maior desvio padrão pelas de menor desvio padrão

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma), \tag{2.4}$$

como exemplificado na imagem 2.1.

Após a descrição desses processos, a próxima etapa é utilizá-los para criar uma pirâmide de escala-espaço em si. A pirâmide simula o desfoque percebido ao se aproximar ou se distanciar de um objeto, permitindo identificar quais características de uma imagem permanecem visíveis para descrevê-la, mesmo em condições de grande desfoque ou em visibilidade reduzida.

A pirâmide é dividida em oitavas: a cada nova oitava, o tamanho da imagem é reduzido pela metade, permitindo que características maiores possam ser representadas de maneira mais compacta. Dentro de cada oitava, aplica-se uma variação dos valores de

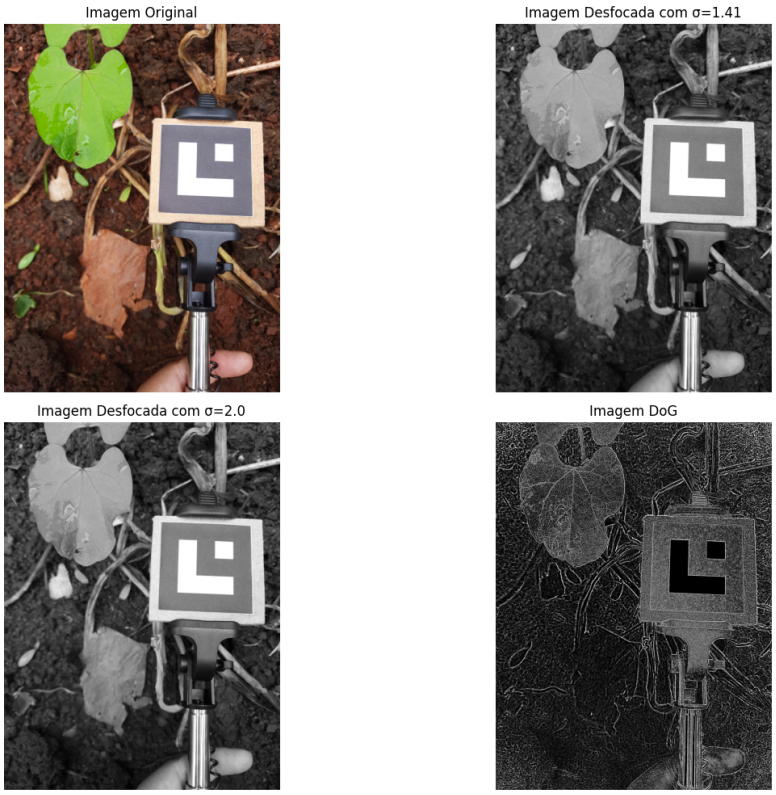


Figura 2.1: Na imagem, as áreas sem muita variação aparecem em preto, enquanto as partes com mudanças rápidas ou detalhes são destacadas em branco. Essa técnica é útil para evidenciar bordas e outros detalhes importantes da imagem.

σ.

Essa estrutura de múltiplos níveis permite capturar características em escalas decrescentes e é fundamental para garantir a eficiência do SIFT a diferentes tamanhos e resoluções, possibilitando a detecção de pontos de interesse independentemente da escala da imagem original.

Uma vez que o DoG é utilizado, parte-se para a procura de extremos locais em escala e espaço. Então, tendo um píxel em vista, compara-o com os oito píxeis vizinhos na imagem, e com os nove píxeis nas escalas acima e abaixo, se o ponto for um extremo comparado com todos os outros, ele será selecionado como um possível ponto-chave. Isso significa que esse é o ponto que possivelmente melhor representa a região na escala.

Após a detecção inicial dos pontos-chave, é necessário refiná-los para aumentar a precisão espacial. Essa etapa visa eliminar dois tipos de candidatos inadequados: pontos com baixo contraste (pouco distintivos) e aqueles mal posicionados ao longo de bordas (sensíveis a ruídos).

Para identificar os extremos locais com maior exatidão, o algoritmo SIFT aplica

uma expansão da série de Taylor à função de intensidade da imagem, aproximando seu comportamento em torno de cada ponto candidato por meio de um polinômio diferenciável. Essa modelagem matemática permite analisar a variação de intensidade nas vizinhanças do ponto e determinar sua posição: quando a derivada do polinômio se anula (valor zero), configura-se um extremo local.

Adicionalmente, um limiar de intensidade é estabelecido para validar os candidatos. Pontos cuja magnitude de intensidade no extremo seja inferior a esse limiar pré-definido são descartados. Essa filtragem não apenas aumenta a robustez dos descritores, mas também reduz significativamente a quantidade de pontos irrelevantes ou instáveis, garantindo maior confiabilidade nas características selecionadas.

Após a aplicação da série de Taylor, utilizam-se as derivadas parciais de segunda ordem para construir a matriz Hessiana. Considerando uma imagem $I(x, y)$, a matriz Hessiana $H(x, y)$ é definida como:

$$H(x,y) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix}, \tag{2.5}$$

esta matriz indica a curvatura da função em torno de um ponto crítico. Os autovalores α e β da matriz Hessiana correspondem à intensidade das curvaturas em direções principais ao redor desse ponto, enquanto os autovetores determinam as direções dessas curvaturas. Se ambos os autovalores são significativos e aproximadamente iguais, o ponto pode ser considerado um canto, uma característica de interesse. No entanto, se apenas um autovalor é grande, o ponto provavelmente representa uma borda e pode ser descartado. Para decidir se o ponto é uma borda, é comum utilizar a razão entre os autovalores:

$$r = \frac{\alpha}{\beta}, \tag{2.6}$$

se r for maior que um limite (tipicamente $r > 10$), o ponto é classificado como uma borda e, portanto, não é ideal para ser um ponto de interesse. Além disso, pode-se verificar essa

condição por meio da relação entre o traço e o determinante da matriz Hessiana:

$$\frac{\mathrm{Tr}(H)^2}{\det(H)} = \frac{(r+1)^2}{r}, \tag{2.7}$$

onde o traço $\mathrm{Tr}(H)$ e o determinante $\det(H)$ são dados por:

$$\mathrm{Tr}(H) = D_{xx} + D_{yy} = \alpha + \beta, \tag{2.8}$$

$$\det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta, \tag{2.9}$$

a expressão $\frac{\mathrm{Tr}(H)^2}{\det(H)}$ é minimizada quando $r = 1$, ou seja, quando os autovalores são iguais, sugerindo uma característica circular. Para pontos de interesse precisos, essa condição é ideal, ao indicar que o ponto possui uma forma mais estável e é menos sensível a mudanças de escala e rotação. Com esses pontos-chave, já se sabe a escala na qual cada um foi detectado, uma vez que essa escala corresponde ao nível de desfocagem da imagem onde o ponto-chave se destacou. Essa característica procura trazer a propriedade de invariância de escala, o que significa que o ponto poderá ser detectado independentemente de mudanças de escala na imagem original.

A próxima etapa é atribuir uma orientação a cada ponto-chave, possibilitando a propriedade de invariância de rotação. Isso permite que o ponto-chave possa ser identificado independentemente da orientação da imagem. Para fazer isso, toma-se uma vizinhança ao redor do ponto-chave com um tamanho proporcional à escala na qual o ponto foi detectado.

Dentro dessa vizinhança, calcula-se a magnitude e a direção do gradiente em cada píxel. A magnitude do gradiente indica a intensidade da mudança de luminosidade no ponto, enquanto a direção do gradiente representa o ângulo dessa mudança em relação a um eixo fixo. Para calcular a magnitude e a direção do gradiente em cada ponto (x, y) de uma imagem, utilizam-se as derivadas parciais da imagem nas direções x e y. A magnitude do gradiente $G(x, y)$ em um ponto (x, y) é dada pela combinação das derivadas parciais

da imagem $I(x, y)$ nas direções x e y:

$$||\nabla I(x,y)|| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}, \tag{2.10}$$

$\frac{\partial I}{\partial x}$ é a derivada parcial da imagem $I(x, y)$ em relação à direção x, que mede a variação da intensidade da imagem na direção horizontal. $\frac{\partial I}{\partial y}$ é a derivada parcial da imagem $I(x, y)$ em relação à direção y, que mede a variação da intensidade na direção vertical. A direção do gradiente $\theta(x, y)$ em um ponto (x, y) é o ângulo da direção da maior variação da intensidade da imagem.

Para organizar essas direções, constrói-se um histograma de orientação dividido em 36 compartimentos, cada um cobrindo um intervalo de 10 graus em um total de 360 graus. A quantidade adicionada a esse compartimento é proporcional à magnitude do gradiente, ou seja, a força com que a direção em questão está presente na vizinhança.

Repetindo esse processo para todos os píxeis ao redor do ponto-chave. O histograma se forma com picos em certas orientações normalizadas para ficarem menos sensíveis a variações de iluminação. O pico mais alto representa a orientação dominante da região ao redor do ponto-chave como demonstrado na figura 2.2, o que significa que o ponto-chave será rotacionado nessa direção ao ser registrado, possibilitando que ele seja identificado corretamente, mesmo se a imagem for girada.

Obtendo invariância, à localização, escala, rotação e ruído, parte-se para a descrição desses pontos com o auxílio de um descritor. Cada descritor vai ser relativo a uma área nos arredores de um ponto-chave. Inicialmente, em volta da área do ponto-chave, será definida uma região de 16×16 píxeis, posteriormente dividida em 16 quadrantes de tamanho 4×4. Para cada quadrante vão ser aplicados histogramas como acima, mas com apenas 8 compartimentos. Ou seja, tem-se 8 colunas no histograma e a orientação será inferida pelos ângulos que possuem magnitude mais frequente. No final, os descritores vão ser um vetor numérico de dimensão 128 (área 4x4 e 8 bins) Com os descritores criados, é necessário tratar o problema de invariância à rotação. Uma forma de fazer isso é deixar as informações de orientação dos descritores dependentes da orientação do ponto-chave. Logo, tem-se que cada orientação do descritor vai ser subtraída pela orientação

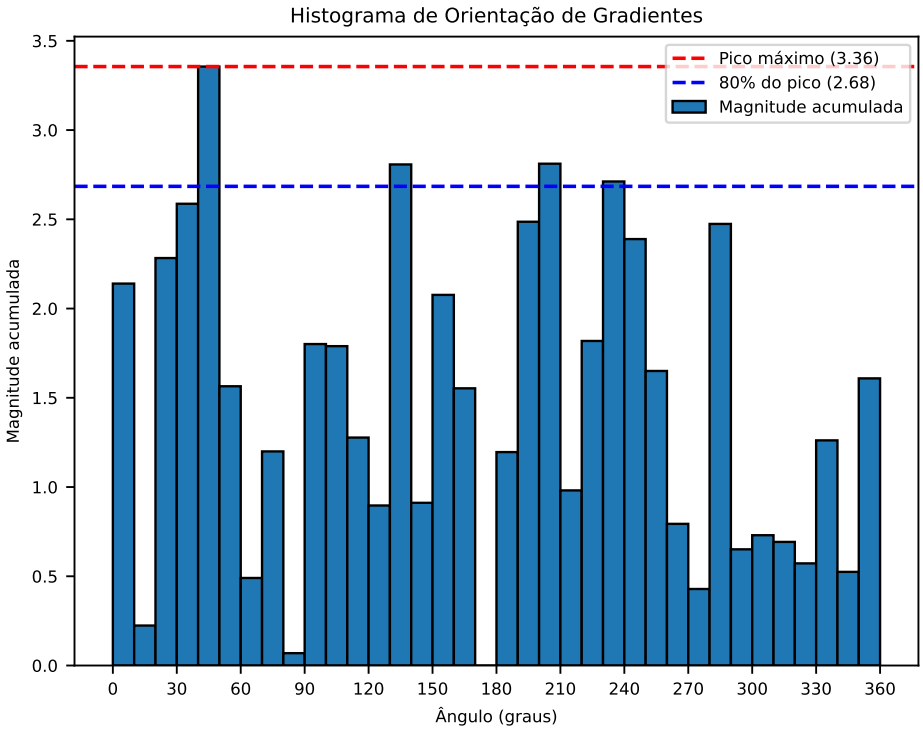


Figura 2.2: Um exemplo fictício mostra um gráfico onde o pico representa a orientação dominante da região ao redor de um ponto-chave. Nesse gráfico, existem linhas de referência marcando 100% e 80% da magnitude máxima, ajudando a visualizar a intensidade relativa das orientações.

do ponto-chave. Isso possibilita que, mesmo que a imagem for rotacionada, os descritores continuem sendo os mesmos.

Por exemplo, seja uma imagem I e um conjunto de ponto-chave $\{k_1, k_2, \ldots, k_n\}$, cada um com uma orientação θ_i associada. Para cada ponto-chave k_i, seja D_i o descritor correspondente, cuja orientação inicial é definida por θ_i. Para garantir a invariância a rotações, modifica-se D_i aplicando uma transformação que normaliza a orientação relativa ao ponto-chave. Formalmente, seja D_i' o descritor rotacionado, então tem-se:

$$D_i' = D_i - \theta_i, \tag{2.11}$$

essa operação garante que D_i' é independente da orientação absoluta de k_i na imagem. Assim, se a imagem sofrer uma rotação global de um ângulo α, os ponto-chave terão suas

orientações alteradas para $\theta_i + \alpha$, mas os descritores normalizados serão:

$$D'_i = (D_i + \alpha) - (\theta_i + \alpha) = D_i - \theta_i, \tag{2.12}$$

ou seja, os descritores D'_i permanecem inalterados. Agora, com a invariância à rotação, uma importante etapa é estabelecer uma invariância à iluminação. Isso pode ser feito truncando os valores relativos aos 128 números do descritor. Então, um valor t é usado para definir esse limite. Então, para atingir a independência de iluminação, limitam-se altos valores no vetor de características. Seja um vetor de características $\mathbf{v} = [v_1, v_2, \ldots, v_{128}]$. Para cada componente v_i de $\mathbf{v}$, aplica-se:

$$v_i = \min(v_i, t), \tag{2.13}$$

isso significa que qualquer valor $v_i > t$ é truncado para t. Em seguida, o vetor resultante é normalizado novamente, usando a norma L_2.

Assim, criam-se os ponto-chave, pontos de interesse pertencentes a uma imagem e seus descritores que possuem características da área ao redor do ponto de interesse. Essa etapa, além de reduzir o número de dados de uma imagem bidimensional de alta resolução para vetores numéricos, tornando os processos sequentes bem menos custosos computacionalmente, também cria informações cruciais para a correspondência entre imagens, o próximo passo para realizar a reconstrução.

2.2 Correspondência entre imagens

Após a identificação dos pontos característicos, descritos para possuir invariância de escala, rotação, iluminação e ruído, o próximo passo é encontrar correspondências entre eles em diferentes imagens. Esse processo consiste em identificar aparições do mesmo ponto de interesse em múltiplas imagens, de modo a relacionar estruturas comuns observadas sob diferentes perspectivas ou condições de captura.

Isso será feito com a ajuda de Árvores de Vocabulário. Essa estrutura é construída com base na técnica *k-means clustering* em múltiplos níveis. Na qual, k define o número

de filhos em cada nó e a árvore passa inicialmente por um processo de aprendizado não supervisionado, os descritores são divididos por suas similaridades numéricas intrínsecas, visto que descritores similares pertencem a regiões similares.

Primeiramente, encontram-se k médias definindo k centros de *clusters*, depois dividem-se os descritores em k grupos relativos a sua proximidade com o centro. Esse processo é realizado recursivamente, dividindo cada *cluster* em k novos *clusters* e isso ocorre até um nível L definido. Na fase *on-line*, propaga-se o descritor pela árvore a fim de encontrar o *cluster* que mais se aproxima de seu valor. Isso ocorre aplicando k produtos escalares nos L níveis da árvore, obtendo kL produtos escalares. Por fim, codifica-se o caminho da árvore por um número inteiro utilizado posteriormente como uma pontuação.

A árvore possibilita não só um bom vocabulário visual, mas também um procedimento de busca eficiente. O custo computacional na árvore hierárquica é relativo à $O(\log n)$ o que o torna mais eficiente que uma abordagem não hierárquica. E a memória usada é linear em relação ao tamanho do descritor, aproximadamente, a k^L, ou seja, k elevado ao número limite de níveis L.

A fórmula para o número total de nós PL_i na árvore é dada por:

$$PL_i = k \cdot \frac{L + 1 - k}{k - 1} \approx k^L. \tag{2.14}$$

O uso de memória é determinado pela quantidade total de nós na árvore e pela dimensionalidade dos descritores. A memória necessária pode ser estimada como:

$$\text{Memória} \approx D \cdot k^L \text{ bytes}. \tag{2.15}$$

Por exemplo, com o tamanho dos descritores $D = 128$, $L = 6$ e $k = 10$ ter-se-ia aproximadamente um milhão de folhas e usar-se-ia 148MB de memória.

O que garante grande eficiência desse algoritmo é a quantização hierárquica dos vetores. Ao transformá-los em grupos discretos, isso reduz significativamente o custo computacional ao manuseá-los. Após a quantização hierárquica, determina-se a relevância entre uma imagem de consulta e as imagens do banco de dados com base na similaridade dos caminhos percorridos pelos descritores na árvore de vocabulário. Para isso, atribuem-

se pesos w_i aos nós i e calculam-se os vetores de consulta q_i e banco de dados d_i como

$$q_i = n_i w_i \quad \text{e} \quad d_i = m_i w_i, \tag{2.16}$$

onde n_i e m_i são o número de descritores que passam pelo nó i na consulta e no banco, respectivamente. A pontuação de relevância $s(q, d)$ é dada pela diferença normalizada.

$$s(q, d) = \left\| \frac{q}{\|q\|} - \frac{d}{\|d\|} \right\|, \tag{2.17}$$

sendo a normalização L_1 mais eficaz que L_2 em experimentos do artigo original em Nister e Stewenius (2006). Os pesos w_i podem ser constantes, mas o desempenho pode melhorar ao usar

$$w_i = \ln \frac{N}{N_i}, \tag{2.18}$$

onde N é o total de imagens no banco, e N_i representa as imagens com descritores passando por i. Esse método favorece nós mais informativos.

A eficiência do algoritmo aumenta com o uso de um vocabulário extenso, geralmente com milhões de folhas. Nós finais são mais relevantes que intermediários para discriminação, enquanto listas de exclusão podem zerar pesos de símbolos muito frequentes ou raros, reduzindo custos computacionais sem prejudicar a recuperação. Por fim, a relevância é melhor calculada considerando entropia relativa à raiz da árvore, aproveitando a hierarquia para equilibrar distintividade e repetibilidade dos descritores.

Concluindo, a abordagem baseada em Árvores de Vocabulário se destaca como uma solução poderosa e eficiente para tarefas de reconhecimento e correspondência de imagens em grandes bases de dados. Conforme demonstrado, o uso de uma hierarquia para quantizar descritores de imagens permite não somente a construção de vocabulários extensos, mas também pode melhorar significativamente a eficiência do processo de busca. Resultados experimentais disponíveis no artigo original mostram que a desempenho do algoritmo aumenta proporcionalmente ao número de folhas na árvore, enquanto ajustes no fator de ramificação têm impacto mais moderado (NISTER; STEWENIUS, 2006).

Em síntese, a combinação de quantização hierárquica, otimizações baseadas em

entropia e normalização L_1 proporciona uma solução escalável, adequada para aplicações que exigem alta eficiência e qualidade, como buscas visuais em larga escala, reconhecimento de objetos e indexação.

Introduzido o conceito de árvores de vocabulário e definidos os pares de imagens candidatas, deve-se fazer a correspondência entre os descritores de imagens. Para isso, utiliza-se as técnicas de geometria epipolar e RANSAC que serão descritas na próxima seção.

2.3 Correspondência de Características

Uma vez que se obteve os descritores de cada imagem e os pares de imagens candidatas, necessita-se corresponder as características entre as imagens. Para isso, a cada característica da imagem I_A tenta-se encontrar uma característica equivalente na imagem I_B.

Os descritores extraídos de imagens residem em um espaço não linear, tornando a distância Euclidiana uma métrica pouco confiável para determinar proximidade. Para superar essa limitação, utiliza-se um critério relativo baseado nos dois descritores mais próximos.

Para cada descritor d_A na imagem A, encontram-se os dois descritores mais próximos, d_1 e d_2, na imagem B. A qualidade da correspondência é avaliada pela razão entre as distâncias:

$$r = \frac{\delta_1}{\delta_2}, \tag{2.19}$$

onde δ_1 é a distância ao descritor mais próximo e δ_2 ao segundo mais próximo. Se r for menor que um limite pré-definido ($r <$ limiar), considera-se a correspondência válida. Este critério elimina características ambíguas ou repetitivas, garantindo identificações de correspondências mais confiáveis, conforme proposto por Lowe (2004).

Após encontrar as posições das características que se assemelham, utiliza-se da geometria epipolar para realizar uma filtragem geométrica.

2.3.1 Geometria Epipolar e a Matriz Fundamental

A geometria epipolar, conforme definida por Hartley e Zisserman (2004), descreve a relação projetiva intrínseca entre duas visualizações. Essa relação é independente da estrutura tridimensional da cena, dependendo apenas dos parâmetros internos das câmeras e da pose relativa entre elas. Por meio dessa geometria, é possível verificar se as correspondências de pontos entre as imagens respeitam as restrições impostas pelas projeções.

Considere o problema em que $\mathbf{X} \in \mathbb{R}^3$ é um ponto no espaço tridimensional, projetado em duas visualizações $\mathbb{R}^2$. Na primeira imagem, o ponto projetado é $\mathbf{x}$, enquanto na segunda é projetada em $\mathbf{x}'$, como ilustrado na Figura 2.3.

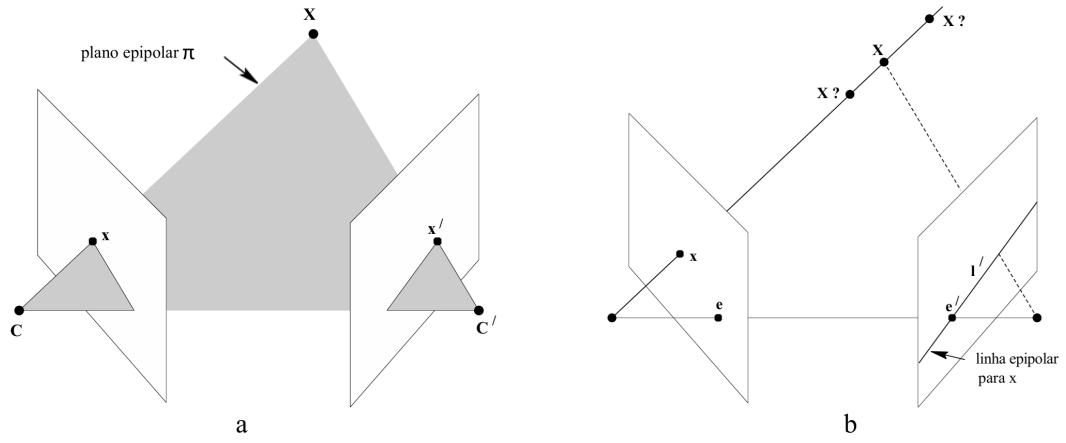


Figura 2.3: Representação do plano epipolar, contendo o ponto tridimensional $\mathbf{X}$, suas projeções $\mathbf{x}$ e $\mathbf{x}'$, além dos centros das câmeras, retirada de Hartley e Zisserman (2004)

A partir da Figura 2.3, observa-se que os pontos $\mathbf{X}$, $\mathbf{x}$, $\mathbf{x}'$ e os centros das câmeras são coplanares, formando o chamado plano epipolar π. A coplanaridade pode ser verificada matematicamente ao calcular o determinante de um sistema formado pelos vetores correspondentes às retas definidas por esses pontos. Caso o determinante seja zero, confirma-se que os pontos estão no mesmo plano. Se $\mathbf{x}$ for conhecido, é possível determinar a reta epipolar $\mathbf{l}'$, que representa a projeção do plano epipolar na segunda imagem. O ponto $\mathbf{x}'$ deve estar contido nessa reta para satisfazer a condição de coplanaridade com os demais elementos. Isso restringe a busca da imagem bidimensional inteira para uma simples reta, chamada linha epipolar.

Além desses pontos, tem-se o $\mathbf{e}$ e o $\mathbf{e}'$ que são os epipolos, os pontos formados na interseção da visualização com a reta que liga o centro das duas imagens. Obtém-se com

isso a ideia de que, dado um ponto $\mathbf{x}$, a linha epipolar é a projeção do raio de $\mathbf{x}$ até $\mathbf{C}$ na segunda visualização. Com isso, pode-se entender que existe um mapeamento $\mathbf{x} \mapsto \mathbf{l}'$ e esse é um mapeamento projetivo representado algebricamente pela matriz fundamental $\boldsymbol{F}$ quando não há calibração prévia e pela matriz essencial $\boldsymbol{E}$ quando há as matrizes de calibração $\boldsymbol{K}$ e $\boldsymbol{K'}$.

Sabendo que $\mathbf{x}$ e $\mathbf{x}'$ são projeções do mesmo ponto 3D $\mathbf{X}$ **que pertence a um plano** π, existe uma homografia 2D $\boldsymbol{H}_\pi$ que mapeia cada $\mathbf{x}$ para cada $\mathbf{x}'$. Assim, $\mathbf{x}'$ pode ser descrito como:

$$\mathbf{x}' = \boldsymbol{H}_\pi \mathbf{x}. \tag{2.20}$$

A linha epipolar $\mathbf{l}'$ no segundo sistema, associada ao epipolo $\mathbf{e}'$, é obtida pelo produto vetorial entre $\mathbf{e}'$ e $\mathbf{x}'$:

$$\mathbf{l}' = \mathbf{e}' \times \mathbf{x}' = [\mathbf{e}']_\times \mathbf{x}'. \tag{2.21}$$

$[\mathbf{e}']_\times$ indica a matriz antissimétrica de $\mathbf{e}'$ onde $\boldsymbol{A^T} = -\boldsymbol{A}$ e tem formato:

$$[\mathbf{e}']_\times = \begin{pmatrix} ,0 & -\mathbf{e}'_3 & \mathbf{e}'_2 \\ \mathbf{e}'_3 & 0 & -\mathbf{e}'_1 \\ -\mathbf{e}'_2 & \mathbf{e}'_1 & 0 \end{pmatrix}, \tag{2.22}$$

com ela torna-se o produto vetorial como uma transformação linear. Uma vez que essas equações foram definidas, pode-se substituir a equação 2.20 na 2.21 obtendo:

$$\mathbf{l}' = [\mathbf{e}']_x \boldsymbol{H}_\pi \mathbf{x}, \tag{2.23}$$

ou seja, a matriz que mapeia o ponto $\mathbf{x}$ para a linha epipolar correspondente, chamada de matriz fundamental, é dada por:

$$\boldsymbol{F} = [\mathbf{e}']_x \boldsymbol{H}_\pi \mathbf{x}, \tag{2.24}$$

tomando como base a geometria, $\boldsymbol{F}$ significa o mapeamento de um plano projetivo $\mathbb{P}^2$ para o conjunto de linhas epipolares da segunda imagem provindas do epipolo $\mathbf{e}'$. Exitem outros modos de encontrar a matriz fundamental descritos por Hartley e Zisserman (2004),

mas não serão descritos nesse trabalho. Até aqui considera-se $\mathbf{x} \mapsto \mathbf{l}'$ é definido por $\boldsymbol{F}$. Agora, é necessário definir a condição de correspondência, se $\mathbf{x}$ e $\mathbf{x}'$ são correspondentes, $\mathbf{x}'$ está na linha epipolar $\mathbf{l}' = \mathbf{F}\mathbf{x}$, como $\mathbf{x}'$ pertence a $\mathbf{l}'$ tem-se $\mathbf{l}'\mathbf{x}'^T = 0$. Substituindo, tem-se:

$$\mathbf{x}'^T \boldsymbol{F} \mathbf{x} = 0, \tag{2.25}$$

essa é a condição de satisfação. Isso é válido para todo par de pontos correspondentes entre as duas imagens. Com essa condição e obtendo pelo menos 7 pares de pontos correspondentes $\mathbf{x_i} \longleftrightarrow \mathbf{x'_i}$, pode-se encontrar a matriz $\boldsymbol{F}$. A partir disso, pode-se criar uma equação linear, utilizando as coordenadas homogêneas $(x, y, 1)^T$ e $(x', y', 1)^T$ e representando a matriz fundamental como f_{ij} define-se os vetores e a matriz da seguinte forma:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \mathbf{x}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}, \tag{2.26}$$

a equação fundamental 2.25 se expande como:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0, \tag{2.27}$$

primeiro, calcula-se $\boldsymbol{F}\mathbf{x}$:

$$\boldsymbol{F}\mathbf{x} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_{11}x + f_{12}y + f_{13} \\ f_{21}x + f_{22}y + f_{23} \\ f_{31}x + f_{32}y + f_{33} \end{bmatrix}, \tag{2.28}$$

em seguida, calcula-se $\mathbf{x}'^{\top}(\boldsymbol{F}\mathbf{x})$:

$$\mathbf{x}'^{\top}(\boldsymbol{F}\mathbf{x}) = \begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} f_{11}x + f_{12}y + f_{13} \\ f_{21}x + f_{22}y + f_{23} \\ f_{31}x + f_{32}y + f_{33} \end{bmatrix}, \tag{2.29}$$

expandindo os termos, tem-se:

$$x'(f_{11}x + f_{12}y + f_{13}) + y'(f_{21}x + f_{22}y + f_{23}) + (f_{31}x + f_{32}y + f_{33}) = 0. \tag{2.30}$$

Para n pontos correspondentes, o sistema de equações será da forma $\boldsymbol{A}\mathbf{f} = 0$, onde a matriz $\boldsymbol{A}$ tem a forma:

$$\mathbf{A}\mathbf{f} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ x'_2x_2 & x'_2y_2 & x'_2 & y'_2x_2 & y'_2y_2 & y'_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = 0, \tag{2.31}$$

onde cada linha de $\boldsymbol{A}$ contém os coeficientes dos elementos da matriz $\boldsymbol{F}$ correspondentes a cada par de pontos $(\mathbf{x}_i, \mathbf{x}'_i)$.

O sistema de equações homogêneas que descreve a relação epipolar entre duas imagens possui um espaço de soluções de dimensão maior que zero, o que significa que existem infinitas soluções que são múltiplas escalares de uma solução particular. Essa indeterminação é intrínseca à natureza projetiva das correspondências entre imagens. A matriz fundamental, que representa essa relação projetiva, é definida somente a menos de um fator de escala, pois multiplicar a matriz fundamental por um escalar não nulo não altera a relação epipolar.

Para obter uma boa estimativa da matriz fundamental, é necessário lidar com a presença de *outliers* nas correspondências. O algoritmo RANSAC (*Random Sample Consensus*) proposto em Fischler e Bolles (1981) é amplamente utilizado para esse fim. Ele funciona escolhendo aleatoriamente um subconjunto mínimo de correspondências, calculando a matriz fundamental a partir desse subconjunto e contando o número de correspondências que satisfazem a relação epipolar definida pela matriz fundamental. Esse processo é repetido várias vezes, e a matriz fundamental com o maior número de correspondências *inliers* é escolhida.

Após a estimação, a matriz fundamental é frequentemente normalizada, geralmente ajustando sua norma de Frobenius para 1. Essa normalização é útil para remover

a ambiguidade de escala e garantir que a matriz fundamental seja única, a menos do sinal. A norma de Frobenius é uma medida da magnitude de uma matriz e a normalização para 1 possibilita que todas as matrizes fundamentais tenham a mesma magnitude, facilitando a comparação entre elas, tornando-a adequada para aplicações posteriores em visão computacional.

2.3.2 Matriz Essencial

A matriz essencial é uma forma especial da matriz fundamental aplicada a câmeras calibradas, onde as coordenadas das imagens são normalizadas para eliminar os efeitos dos parâmetros intrínsecos. Seja uma câmera representada por:

$$\boldsymbol{P} = \boldsymbol{K}\,[\boldsymbol{R} \mid \mathbf{t}], \tag{2.32}$$

com $\boldsymbol{K}$ a matriz de calibração, $\boldsymbol{R}$ a matriz de rotação e $\mathbf{t}$ o vetor de translação. Ao projetar um ponto $\mathbf{X}$ no espaço 3D, obtém-se $\mathbf{x} = \boldsymbol{P}X$ e, ao aplicar o inverso de $\boldsymbol{K}$, chegam-se às coordenadas normalizadas $\hat{\mathbf{x}} = \boldsymbol{K}^{-1}\mathbf{x}$. Dessa forma, a câmera pode ser descrita por $[\boldsymbol{R} \mid t]$ e, para pontos correspondentes, a restrição epipolar assume a forma:

$$\hat{\mathbf{x}}^{\mathbf{T}}\boldsymbol{E}\,\hat{\mathbf{x}}' = 0, \tag{2.33}$$

sendo $\mathbf{E}$ a matriz essencial. Adicionalmente, se a matriz fundamental $\boldsymbol{F}$ satisfaz $\mathbf{x}^{\mathbf{T}}\boldsymbol{F}\mathbf{x}' = 0$ nas coordenadas originais, a relação entre elas é dada por:

$$\boldsymbol{E} = \boldsymbol{K}^T\boldsymbol{F}\boldsymbol{K}. \tag{2.34}$$

Como $\mathbf{E}$ depende apenas da rotação $\mathbf{R}$ e da translação $\mathbf{t}$ relativas entre as câmeras, embora cada uma tenha três graus de liberdade, a ambiguidade de escala reduz o total de parâmetros independentes para cinco. Um aspecto crucial dessa matriz é revelado em sua decomposição em valores singulares, que pode ser expressa como:

$$\boldsymbol{E} = \boldsymbol{U}\,\mathrm{diag}(\sigma,\ \sigma,\ 0)\,\boldsymbol{V}^T, \tag{2.35}$$

indicando que dois dos seus valores singulares são iguais e o terceiro é nulo, condição necessária e suficiente para que uma matriz 3×3 seja classificada como essencial. A decomposição de $\boldsymbol{E}$ em seus componentes pode ser escrita na forma:

$$\boldsymbol{E} = [\mathbf{t}]_{\times}\boldsymbol{R}, \tag{2.36}$$

em que $[\mathbf{t}]_{\times}$ é a matriz antissimétrica associada a $\mathbf{t}$, representando o produto vetorial. Utilizando a SVD Singular Value Decomposition $\boldsymbol{E} = \boldsymbol{U}\,\mathrm{diag}(1,1,0)\,\boldsymbol{V^T}$ e introduzindo matrizes auxiliares, como:

$$\boldsymbol{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{e} \quad \boldsymbol{Z} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.37}$$

pode-se representar $[\mathbf{t}]_{\times}$ por $\boldsymbol{U}\,\boldsymbol{Z}\,\boldsymbol{U}^T$ e identificar duas soluções para a rotação, isto é, $\boldsymbol{R} = \boldsymbol{U}\,\boldsymbol{W}\,\boldsymbol{V}^T$ ou $\boldsymbol{R} = \boldsymbol{U}\,\boldsymbol{W}^T\,\boldsymbol{V}^T$. Devido à natureza homogênea de E, essa decomposição não é única, havendo uma família contínua de soluções. No processo de reconstrução tridimensional, a primeira câmera é frequentemente fixada como:

$$\boldsymbol{P} = [\boldsymbol{I} \mid 0], \tag{2.38}$$

enquanto a segunda câmera, $\boldsymbol{P'}$, é determinada a partir da fatoração de $\boldsymbol{E}$. O vetor de translação $\mathbf{t}$ é geralmente extraído como a terceira coluna de $\boldsymbol{U}$ (isto é, $\mathbf{t} = \pm\mathbf{u_3}$), e, combinando-se com as duas alternativas para $\boldsymbol{R}$, obtém-se quatro configurações possíveis para $\boldsymbol{P'}$:

$$\boldsymbol{P'} = \left[\boldsymbol{U}\,\boldsymbol{W}\,\boldsymbol{V}^T \mid \pm\mathbf{u_3}\right] \quad \text{ou} \quad \boldsymbol{P'} = \left[\boldsymbol{U}\,\boldsymbol{W}^T\,\boldsymbol{V}^T \mid \pm\mathbf{u_3}\right]. \tag{2.39}$$

Essa ambiguidade quádrupla é resolvida pela condição de quiralidade, que garante que os pontos reconstruídos estejam posicionados à frente de ambas as câmeras, assegurando uma reconstrução consistente.

No caso de câmeras calibradas, utilizam-se as coordenadas de imagem normalizadas para calcular a matriz essencial em vez da fundamental. De maneira análoga à

obtenção de $\mathbf{F}$, $\mathbf{E}$ pode ser estimada por métodos lineares a partir de oito ou mais pontos, uma vez que os pontos correspondentes satisfazem a equação 2.33.

A principal diferença está na imposição das restrições: enquanto a matriz fundamental impõe a condição $\det \boldsymbol{F} = 0$, a matriz essencial requer, adicionalmente, que os dois maiores valores singulares sejam iguais. Esse requisito pode ser tratado pelo seguinte resultado: seja $\boldsymbol{E}$ uma matriz 3×3 com decomposição SVD $\boldsymbol{E} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, onde $\boldsymbol{D} = \text{diag}(a, b, c)$ com $a \geq b \geq c$; então, a matriz essencial mais próxima de E em norma de Frobenius é dada por $\hat{\boldsymbol{E}} = \boldsymbol{U}\hat{\boldsymbol{D}}\boldsymbol{V}^T$, onde $\hat{\boldsymbol{D}} = \text{diag}((a+b)/2, (a+b)/2, 0)$. Se o objetivo é determinar as matrizes de câmera normalizadas $\boldsymbol{P}$ e $\boldsymbol{P'}$ durante o processo de reconstrução, não é necessário calcular explicitamente $\hat{\boldsymbol{E}}$ via $\boldsymbol{U}\hat{\boldsymbol{D}}\mathbf{V}^T$, pois $\boldsymbol{P}$ pode ser obtida diretamente a partir da SVD. A escolha entre as quatro soluções possíveis para $\mathbf{P'}$ é feita com base na condição de que os pontos visíveis devem estar à frente de ambas as câmeras.

2.4 Inferência da Estrutura Rígida e da Pose da Cena 3D

A reconstrução tridimensional a partir de múltiplas imagens 2D permite a inferência da estrutura rígida de uma cena a partir de observações visuais. O objetivo principal dessa técnica é entender a relação geométrica entre as imagens fornecidas e, a partir dessa relação, estimar a posição, orientação das câmeras e os pontos 3D da cena. Este processo é frequentemente abordado por meio de *pipeline* incremental, onde a reconstrução começa com um conjunto inicial de imagens e é progressivamente expandida à medida que novas imagens são incorporadas. Nesse trabalho será usada a técnica SfM mais especificamente a *pipeline* descrita por Schönberger e Frahm (2016), que será descrita detalhadamente a seguir. A reconstrução incremental é uma abordagem eficaz, ao permitir a adição gradual de novas visualizações, ajustando a reconstrução à medida que mais dados estão disponíveis. A escolha de um par de imagens inicial adequado, a fusão de correspondências e a triangulação precisa de pontos 3D são etapas críticas para garantir a precisão da reconstrução final.

A seleção de pares de imagens para maximizar correspondências e extrair informações geométricas confiáveis é um aspecto fundamental em reconstrução 3D. Esses pares de imagens devem equilibrar dois fatores principais: a maximização do número de correspondências de pontos e a obtenção de um ângulo de base adequado entre as câmeras.

Maximizar o número de correspondências requer que ambas as imagens compartilhem uma quantidade significativa de elementos visíveis em comum. Isso implica que os pares de imagens devem ter uma sobreposição suficiente, com elementos distintos e bem definidos, para que algoritmos de correspondência possam identificar e rastrear os mesmos pontos em ambas as imagens. A distribuição dos pontos de interesse em cada imagem também é crucial, pois uma dispersão uniforme melhora a qualidade dos cálculos posteriores. Um par de imagens com correspondências bem distribuídas evita que os resultados fiquem enviesados por regiões de alta densidade de pontos e reduz a sensibilidade a erros locais.

Por outro lado, o ângulo entre as câmeras é igualmente importante. Esse ângulo, conhecido como ângulo de base ou base estereoscópica, afeta diretamente a qualidade da informação geométrica extraída. Um ângulo pequeno entre as câmeras pode produzir um número alto de correspondências, mas as informações de profundidade obtidas serão pouco precisas devido à baixa triangulação geométrica. Já um ângulo muito grande pode levar à diminuição do número de correspondências, pois os pontos visíveis em uma imagem podem não estar presentes na outra, especialmente em cenários com objetos dinâmicos ou diferenças significativas de perspectiva.

Idealmente, o ângulo de base deve ser suficientemente grande para fornecer informações geométricas precisas, sem sacrificar o número de correspondências. A escolha do ângulo depende do contexto da aplicação. Em reconstrução 3D, por exemplo, ângulos moderados são preferidos para alcançar um equilíbrio entre densidade de correspondências e precisão de profundidade.

Portanto, ao selecionar pares de imagens, é essencial adotar uma abordagem equilibrada. Deve-se buscar uma sobreposição suficiente e uma distribuição uniforme de correspondências nas imagens, ao mesmo tempo que se escolhe um ângulo de base que maximize a precisão geométrica sem comprometer a densidade de correspondências. Antes

de selecionar o par de imagens iniciais, as correspondências são fundidas em trilhas, cada trilha representa um ponto visualizado por várias câmeras. Nesse momento, os pontos que não possuem trilhas significativas ou incoerentes são descartados como *outliers*. A partir das trilhas, tem-se uma base maior para selecionar pares de imagens significativos.

2.4.1 Triangulação das correspondências

Após escolher as imagens e calcular a matriz fundamental entre elas, a primeira imagem é considerada a origem do sistema de coordenadas. Aqui considera-se que as matrizes de calibração das câmeras $\boldsymbol{K}$ e $\boldsymbol{K'}$ são conhecidas e são matrizes na forma:

$$\boldsymbol{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.40}$$

sendo f_x e f_y a distância focal da câmera e c_x e c_y o centro óptico. Definido isso, como a primeira visualização é a origem do sistema, sua matriz de projeção será dada como:

$$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{I}|0] = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.41}$$

Já a matriz de projeção da segunda visualização será obtida a partir da matriz essencial $\boldsymbol{E}$. A decomposição da matriz essencial fornece duas informações importantes para a matriz de projeção da segunda visualização, a rotação $\boldsymbol{R}$ e a translação $\mathbf{t}$ entre as duas câmeras, e aplicando isso tem-se:

$$\boldsymbol{P'} = \boldsymbol{K'} \begin{bmatrix} \boldsymbol{R} & \mathbf{t} \end{bmatrix} \boldsymbol{K'} = \begin{bmatrix} f'_x & 0 & c'_x \\ 0 & f'_y & c'_y \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \tag{2.42}$$

Após obter as informações de $\boldsymbol{P}$ e $\boldsymbol{P'}$ e tendo as correspondências $\mathbf{x}_i$ e $\mathbf{x'}_i$, objetiva-se encontrar os pontos homogêneos em $\mathbb{P}^3$ $\mathbf{X} = (X, Y, Z, 1)^T$. Para encontrar esse ponto, aplica-

se a técnica da triangulação, onde $\mathbf{X}$ deve satisfazer as equações $\mathbf{x}' = \boldsymbol{P}'\mathbf{X}$ e $\mathbf{x} = \boldsymbol{P}\mathbf{X}$. O método de triangulação utilizado para calcular pontos 3D a partir de correspondências de pontos 2D deve ser projetado para ser invariável sob transformações adequadas, como transformações afins ou projetivas. Quando as matrizes de câmera são conhecidas somente até essas transformações, é importante usar um método de triangulação que seja invariante em relação a transformações afins ou projetivas para obter pontos 3D consistentes. Denota-se o método de triangulação por τ, que computa um ponto 3D $\mathbf{X}$ a partir de uma correspondência de pontos $\mathbf{x} \leftrightarrow \mathbf{x}'$ e das matrizes de projeção das câmeras $\boldsymbol{P}$ e $\boldsymbol{P}'$. A equação que descreve esse processo é

$$\mathbf{X} = \tau(\mathbf{x}, \mathbf{x}', \boldsymbol{P}, \boldsymbol{P}'). \tag{2.43}$$

Diz-se que a triangulação é invariante sob uma transformação H se, ao aplicar essa transformação às câmeras, a posição do ponto 3D resultante também se transforma da mesma forma. Formalmente, isso é expresso como:

$$\tau(\mathbf{x}, \mathbf{x}', \boldsymbol{P}, \boldsymbol{P}') = \boldsymbol{H}^{-1}\tau(\mathbf{x}, \mathbf{x}', \boldsymbol{P}\boldsymbol{H}^{-1}, \boldsymbol{P}'\mathbf{H}^{-1}). \tag{2.44}$$

No problema de triangulação, as coordenadas de $\mathbf{X}$ são desconhecidas, enquanto as coordenadas dos píxeis podem ser obtidas, desde que a matriz de projeção tenha sido determinada a partir da calibração da câmera. O objetivo, portanto, é calcular as coordenadas de $\mathbf{X}$. Considerando que os vetores $\mathbf{x}$ e $\boldsymbol{P}\mathbf{X}$ são paralelos, o produto vetorial entre eles deve ser nulo, resultando na seguinte equação:

$$\mathbf{x} \times \boldsymbol{P}X = \mathbf{0} \begin{bmatrix} x \\ v_1 \\ 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{p_1}\mathbf{X} \\ \mathbf{p_2}\mathbf{X} \\ \mathbf{p_3}\mathbf{X} \end{bmatrix} = \begin{bmatrix} v_1\mathbf{p_3} - \mathbf{p_2} \\ \mathbf{p_1}\mathbf{p_3} - x\mathbf{p_3} \\ x\mathbf{p_2} - v_1\mathbf{p_1} \end{bmatrix} X = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{2.45}$$

os vetores de linha da matriz $\boldsymbol{P}$ são vetores de quatro dimensões que representam projeções de pontos 3D em um plano de imagem. Ao montar o sistema linear $\boldsymbol{A}\boldsymbol{X} = 0$, observa-se que a terceira linha é combinação linear das duas primeiras, ou seja, não acrescenta nova informação. Assim, tem-se somente duas equações independentes para determinar as três

incógnitas de $\mathbf{X}$. Essa indeterminação é inerente à utilização de uma única câmera, pois múltiplos pontos 3D podem projetar-se no mesmo ponto da imagem. A cada nova câmera adicionada, novas linhas são incluídas na matriz $\boldsymbol{A}$, aumentando o número de equações e, consequentemente, a precisão na determinação das coordenadas 3D. Obtém-se uma matriz assim:

$$\boldsymbol{A}\mathbf{X} = \begin{bmatrix} v_1\mathbf{p}_3 - \mathbf{p}_2 \\ \mathbf{p}_1 - u_1\mathbf{p}_3 \\ v_2\mathbf{p}_3 - \mathbf{p}_2 \\ \mathbf{p}_1 - u_2\mathbf{p}_3 \\ \vdots \end{bmatrix} \mathbf{X} = 0. \tag{2.46}$$

No processo de triangulação, a matriz $\boldsymbol{A}$ é fornecida, e o objetivo é resolver para $\mathbf{X}$. Para isso, utiliza-se o método DLT (*Direct Linear Transformation*). A solução desse sistema homogêneo de equações é obtida por meio da decomposição SVD (*Singular Value Decomposition*). O sistema $\boldsymbol{A}\mathbf{X} = \mathbf{0}$ é escrito, e a decomposição SVD é aplicada à matriz $\boldsymbol{A}$, resultando na expressão:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top,$$

onde $\boldsymbol{U}$ e $\boldsymbol{V}$ são matrizes ortogonais, e $\boldsymbol{S}$ é uma matriz diagonal contendo os valores singulares de $\boldsymbol{A}$, ordenados de forma decrescente ($\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$). O objetivo é minimizar a norma quadrática $\|\boldsymbol{A}\mathbf{X}\|^2$, o que é feito por meio da expressão:

$$\|\boldsymbol{A}\mathbf{X}\|^2 = \mathbf{X}^\top \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^\top\mathbf{X}.$$

A minimização ocorre ao escolher a última coluna de $\boldsymbol{V}$, que corresponde ao menor valor singular ($\sigma_{\min}$), pois essa coluna minimiza a soma $\sum_{i=1}^{n} \sigma_i^2 y_i^2$. Assim, a solução $\mathbf{X}$ é dada pela última coluna de $\boldsymbol{V}$, associada ao menor valor singular.

Este processo descrito envolve uma sequência de etapas cruciais para a reconstrução 3D de uma cena a partir de um conjunto de imagens. As etapas principais incluem a seleção das imagens mais úteis, a estimação das poses das câmeras associadas a essas imagens, e a atualização das características 3D, refinando continuamente a reconstrução. Será detalhado o funcionamento de cada uma dessas etapas.

2.4.2 *Pipeline* incremental e Otimização com *Bundle Adjustment*

O processo de reconstrução 3D otimizada começa após a fusão das correspondências em trilhas, sendo cada trilha a representação de um ponto candidato no espaço, visível por várias câmeras. A ideia central é selecionar as próximas imagens que forneçam novas informações à reconstrução, melhorando sua precisão e completude. Primeiramente, as imagens que possuem maior número de associações válidas com os pontos 3D já reconstruídos são selecionadas. Essa análise é feita verificando as correspondências entre pontos 2D nas imagens e suas projeções 3D no espaço. A seleção segue uma abordagem de próximas melhores vistas (*next best views*), priorizando imagens que tragam novas perspectivas, ângulos ou detalhes da cena. Isso maximiza a quantidade de informações úteis para o modelo.

Após a seleção das imagens, realiza-se a estimação da matriz essencial, considerando a primeira imagem como origem do sistema. As correspondências são trianguladas para encontrar os pontos 3D relacionados a cada correspondência. Uma vez que esse processo é encerrado, inicia-se a busca pela próxima melhor vista. Com base nas associações 2D-3D faz-se novas triangulações e utiliza-se uma ressecção para cada uma das novas câmeras, sendo ela um algoritmo Perspective-n-Point (PnP) com o método RANSAC que tem o objetivo de encontrar as características da câmera que melhor validam a maioria das correspondências.

Com as novas informações podem ser adicionados pontos observados por duas ou mais câmeras que podem ser trianguladas novamente. Depois desse processo, efetua-se o *Bundle Adjustment* para refinar os parâmetros intrínsecos e extrínsecos de todas as câmeras e os pontos 3D. A equação usada para *Bundle Adjustment* é dada por:

$$\boldsymbol{E} = \sum_{i} \sum_{j} \left\| \mathbf{x_{ij}} - \pi(\boldsymbol{R_j}\mathbf{X_i} + \mathbf{t_j}, \boldsymbol{K_j}) \right\|^2, \tag{2.47}$$

onde x_{ij} é a observação 2D do ponto X_i na câmera j, π representa a projeção de pontos 3D para o espaço 2D da imagem, R_j e t_j são os parâmetros extrínsecos da câmera j, e K_j são os parâmetros intrínsecos da câmera j. Esse ajuste utiliza métodos iterativos, como

Gauss-Newton, para encontrar a solução ótima.

Após o *Bundle Adjustment*, realiza-se uma filtragem dos resultados para eliminar observações inconsistentes, como aquelas com grandes erros de reprojeção ou pontos 3D mal condicionados devido a ângulos de triangulação pequenos. Além disso, uma nova triangulação pode ser feita para corrigir ou complementar a reconstrução. Essas etapas — seleção de imagens, estimação de poses, triangulação e refinamento — são repetidas iterativamente até que não sejam mais identificadas novas informações relevantes. A qualidade da reconstrução 3D refinada durante essas etapas é fundamental para a precisão dos algoritmos que seguem, como o algoritmo estéreo, o qual será discutido a seguir.

2.5 Estimativa de mapas de profundidade e superfície geométrica densa

A disparidade estéreo é um conceito fundamental em visão computacional, particularmente em tarefas de reconstrução 3D. A disparidade refere-se à diferença de posição de um ponto correspondente nas duas imagens. Essa diferença de posição ocorre devido ao ângulo de visão diferente de cada câmera.

A disparidade d é definida como a diferença entre as coordenadas horizontais de um ponto correspondente nas duas imagens capturadas pelas câmeras estéreo. Em uma cena 3D, se um ponto em um objeto for projetado nas duas imagens em diferentes posições, a disparidade d é inversamente proporcional à distância (ou profundidade) do ponto em relação ao sistema de câmeras. Ou seja, quanto maior a disparidade, mais próximo o objeto está das câmeras. A relação entre disparidade e profundidade é dada pela equação:

$$Z = \frac{f \cdot B}{d}, \tag{2.48}$$

onde, Z é a profundidade (a distância do ponto 3D à câmera), f é a distância focal da câmera, B é a distância entre as duas câmeras (conhecida como base estéreo), d é a disparidade, ou a diferença nas coordenadas do ponto nas imagens.

O algoritmo SGM (*Semi-Global Matching*) por Hirschmuller (2008) e explicado em Kak (2024) é projetado para estimar mapas de disparidade para imagens estéreo.

O Volume de Disparidade é um espaço 3D que codifica os custos de atribuir valores de disparidade a cada píxel da imagem de referência. Se o intervalo de disparidade esperado for $[d_{\min}, d_{\max}]$, o Volume de Disparidade terá dimensões $W \times H \times (d_{\max} - d_{\min})$, onde W e H são a largura e a altura da imagem de referência. Para cada píxel p, o algoritmo estima o custo total para cada disparidade possível no intervalo $[d_{\min}, d_{\max}]$.

A função de custo a ser minimizada para obter o melhor mapa de disparidade é expressa como:

$$C(d) = \sum_{p} C_{\text{data}}(p, d_p) + \sum_{q \in N_p} P_1 \cdot T\left(|d_p - d_q| = 1\right) + \sum_{q \in N_p} P_2 \cdot T\left(|d_p - d_q| > 1\right), \quad (2.49)$$

onde d é o mapa de disparidade, com d_p sendo a disparidade no píxel p, $C_{\text{data}}(p, d_p)$ é o custo de dados para o píxel p na disparidade d_p, N_p é a vizinhança do píxel p, P_1 e P_2 são os pesos definidos pelo usuário para penalizar descontinuidade de disparidade, e $T(\text{arg}) = 1$ se arg for verdadeiro, e $T(\text{arg}) = 0$ caso contrário. A fórmula distingue entre dois casos para descontinuidade de disparidade: quando $|d_p - d_q| = 1$, o custo de descontinuidade é penalizado por P_1; e quando $|d_p - d_q| > 1$, o custo de descontinuidade é penalizado por P_2. A ideia principal é que os valores de disparidade devem mudar gradualmente, com grandes mudanças de disparidade sendo desencorajadas pelo maior peso P_2.

Minimizar diretamente o custo para todos os píxeis e níveis de disparidade é um problema *NP-Hard*, então uma abordagem de programação dinâmica é usada. O cálculo é realizado para cada linha de píxeis de cada vez. Isso torna o problema computacionalmente viável, reduzindo a complexidade do problema para tempo polinomial. Para qualquer direção r, o custo $C_r(p, d)$ no píxel p para disparidade d é calculado recursivamente como:

$$C_r(p, d) = C_{\text{data}}(p, d) + \min\left(C_r(p - r, d - 1) + P_1, C_r(p - r, d + 1) + P_1, \min_i C_r(p - r, i) + P_2\right). \quad (2.50)$$

Essa fórmula calcula o custo de descontinuidade para a disparidade d no píxel p com base nos custos previamente calculados para píxeis adjacentes. Os três termos representam: o custo de mover para o nível de disparidade anterior $(d - 1)$, o custo de mover para o próximo nível de disparidade $(d + 1)$, e o custo mínimo de todos os níveis de disparidade no píxel anterior.

Após o cálculo dos custos para diferentes direções, os custos são agregados para obter o custo total em cada píxel:

$$S(p,d) = \sum_i C_{r_i}(p,d), \tag{2.51}$$

onde $S(p,d)$ é o custo agregado para o píxel p na disparidade d. Para cada píxel p, a disparidade d_p que minimiza o custo total agregado é escolhido:

$$d_p = \arg\min_d S(p,d). \tag{2.52}$$

O algoritmo usa várias direções para recursão: as implementações mais comuns do SGM usam 8 ou 16 direções. Essas direções são alinhadas com os eixos da imagem ou diagonais e ajudam a atualizar o custo de cada píxel com base nos píxeis vizinhos. O cálculo de custo de dados envolve comparar as vizinhanças de píxeis entre a imagem de referência e a outra imagem na disparidade d. Cada direção de recursão depende dos valores previamente calculados para píxeis adjacentes, e as direções definem a ordem de processamento durante a varredura.

O SGM é um algoritmo eficaz para correspondência estéreo, especialmente para imagens grandes, devido à sua viabilidade computacional. Ao usar programação dinâmica e agregação de custos ao longo de várias direções, ele consegue estimar eficientemente mapas de disparidade com mínima descontinuidade. A próxima etapa pretende estimar uma representação geométrica densa da superfície da cena. A ideia é combinar as informações de profundidade obtidas a partir de múltiplas imagens estéreo, gerando uma representação da geometria 3D. Para isso, utiliza-se a estrutura de dados octree.

Primeiro, calcula-se o mapa de profundidade pelo mapa de disparidade usando a relação na equação 2.48 funde-se todos os mapas de profundidade em uma octree global, sendo uma árvore hierárquica usada para representar a cena 3D eficientemente. A octree divide o espaço tridimensional em células cúbicas, permitindo representar grandes volumes de dados 3D de maneira compacta, com alta eficiência. Em uma octree, o espaço tridimensional é recursivamente dividido em oito subespaços menores, criando uma hierarquia de células com diferentes resoluções. Isso facilita o armazenamento e a

manipulação de dados espaciais em diferentes escalas de precisão.

Durante a fusão, os valores de profundidade compatíveis de diferentes mapas são mesclados nas células da octree. Para cada célula, são armazenados os valores de profundidade mais confiáveis, determinados pela consistência entre as múltiplas projeções do ponto nas imagens, provenientes dos mapas de disparidade estimados. Esse processo garante que a superfície geométrica seja representada de forma mais densa e precisa, ao integrar informações de profundidade de várias perspectivas. Após a fusão dos mapas de profundidade na octree, obtém-se uma representação densa da superfície geométrica da cena. A próxima fase do processo é dedicada à construção e refinamento da malha 3D a partir da representação geométrica densa obtida pela fusão dos mapas de profundidade. Após a integração dos dados de profundidade na octree, o objetivo agora é gerar uma malha tetraédrica precisa que capture as características da superfície da cena com maior detalhamento e fidelidade.

Inicialmente, realiza-se uma tetraedralização de Delaunay 3D (LABATUT; PONS; KERIVEN, 2009), uma técnica que permite dividir o espaço tridimensional em tetraedros. Essa divisão é fundamental para construir a malha e garantir que a geometria da cena seja representada de maneira eficiente, sem distorções. A tetraedralização é particularmente útil, pois a estrutura obtida pode ser utilizada para representar volumes complexos, além de facilitar a análise da topologia e da geometria da cena.

Uma vez formada a malha tetraédrica, é necessário aplicar uma votação nas células e facetas dessa malha para atribuir pesos de importância a cada elemento. Esse processo é descrito em detalhes nas publicações de Jancosek e Pajdla (2011) e Jancosek e Pajdla (2014) e visa identificar as partes da malha que são mais relevantes para a reconstrução da superfície 3D. As células e facetas que representam melhor a geometria da cena recebem maior peso, facilitando a construção de uma malha fiel e com boa precisão.

Seguindo a votação, aplica-se o algoritmo *Graph Cut Max-Flow* (BOYKOV; KOLMOGOROV, 2004), baseado na técnica de otimização de Boykov e Kolmogorov (2004), usado para segmentar a malha em regiões distintas, separando a superfície do fundo. Esse algoritmo realiza uma partição ótima do volume, isolando a superfície extraída da malha e criando uma representação clara das fronteiras do objeto ou cena em questão. A seg-

mentação gerada pelo algoritmo ajuda a determinar a camada externa da cena, crucial para reconstruções detalhadas ou análises geométricas subsequentes.

Essas etapas, que incluem a tetraedralização, votação, particionamento com *Graph Cut*, filtragem e simplificação, trabalham em conjunto para gerar uma malha 3D otimizada e densa.

3 Método proposto

Nesta seção, é apresentada a metodologia utilizada para o processamento e análise das imagens do conjunto de dados. Para fins explicativos e ilustrativos, é utilizada a folha 134 com 12 vistas como exemplo ao longo de toda a descrição dos procedimentos metodológicos. Este exemplo serve para ilustrar as etapas do processo de forma clara e didática. E a figura 3.1 mostra um resumo dos passos que são realizados ao decorrer do método. No entanto, vale ressaltar que, na análise dos dados, é realizada com os resultados obtidos de 5 folhas diferentes.

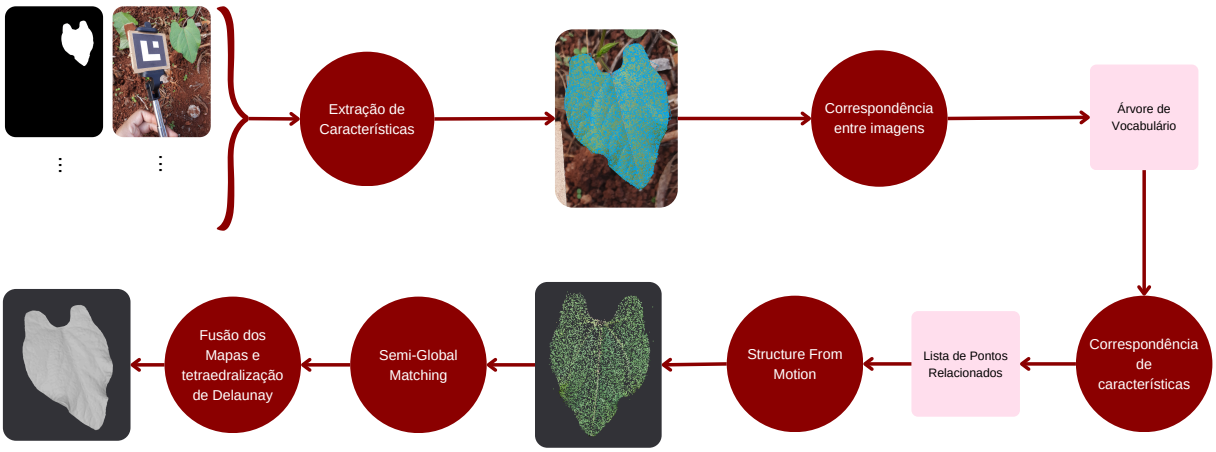


Figura 3.1: Diagrama do método

3.1 Base de Dados de Folhas de Feijão

O conjunto de dados utilizado neste estudo, originalmente publicado por Silva et al. (2025), é composto por 6981 imagens de 612 folhas de feijão, como mostrado na imagem 3.2, capturadas com um celular Samsung Galaxy M31 (modelo SM-M315F). Para garantir precisão métrica, o sistema óptico do dispositivo foi calibrado *a priori*, e todas as imagens foram registradas com resolução de 4624 × 3468 píxeis, utilizando parâmetros técnicos padronizados: abertura f/1.8, ISO 50 e tempo de exposição de 1/587 segundos. A desativação do *flash* e do balanço de branco automático garantiu uniformidade fotométrica, enquanto uma rotação pós-captura de 90° no sentido horário padronizou a orientação

espacial das amostras. Além das imagens em alta resolução, o conjunto de dados inclui:

- Arquivos XML com anotações de *bounding boxes* das folhas e marcadores ArUco;
- Máscaras de segmentação semântica geradas com suporte de uma rede neural e posteriormente avalidadas manualmente como mostrado na imagem 3.2.

3.2 Calibração de Câmeras e Metadados

Para assegurar correspondência entre pontos de vista no *pipeline*, cada imagem foi associada a um ID. A calibração prévia da câmera permitiu extrair:

- Distância focal: $f_{\text{mm}} = 5.23\,\text{mm}$;
- Dimensões do sensor: $7.42\,\text{mm} \times 5.565\,\text{mm}$;

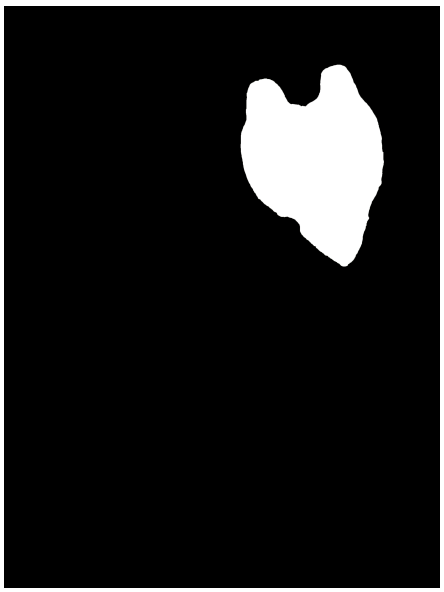
A conversão para píxeis é dada por:

$$f_{\text{píxeis}} = \left(\frac{f_{\text{mm}}}{\text{largura do sensor (mm)}} \right) \times \text{largura da imagem (píxeis)}. \tag{3.1}$$

A matriz de calibração intrínseca K, comum a todas as imagens, é definida como:

$$K = \begin{bmatrix} f_{\text{píxeis}} & 0 & c_x \\ 0 & f_{\text{píxeis}} & c_y \\ 0 & 0 & 1 \end{bmatrix}.$$

(a) Imagem com marcadores ArUco e folha.



(b) Máscara binária gerada por rede neural.

Figura 3.2: Exemplo de *input* (a) e *output* (b) do estágio de segmentação semântica. A máscara permite isolar a região de interesse, ou seja, a folha de feijão.

3.3 Extração de características

O algoritmo SIFT foi utilizado para detectar e descrever os ponto-chave. A pirâmide de escala-espaço (Seção 2.1.1) foi construída com 4 oitavas e 6 níveis de escala por oitava, permitindo a identificação de características invariantes a variações de escala. A aproximação DoG, foi aplicada para detectar extremos locais, que correspondem a pontos máximos ou mínimos numa escala, conferindo invariância a transformações de escala (Seção 2.1).

A filtragem de pontos de baixo contraste e bordas foi realizada por meio da matriz Hessiana (Equação 2.5), o que melhorou a precisão da extração, ao eliminar pontos de baixo contraste, que podem ser menos confiáveis. A orientação dominante dos descritores (Equação 2.12) possibilitou a invariância a rotações, permitindo que os pontos-chave fossem corretamente correspondidos entre imagens com diferentes orientações.

Antes de serem finalizados, os vetores de características passam por truncamento (Seção 2.13) com limiar definido empiricamente como 0.02, para suprimir magnitudes de gradiente excessivas, seguido de uma normalização L_2. Essa normalização assegura que todos os descritores tenham comprimento unitário, reduzindo a sensibilidade a variações de iluminação e destacando *padrões* de gradiente em vez de valores absolutos de inten-

sidade. Como resultado, a correspondência entre descritores torna-se mais robusta, uma vez que as comparações dependem da similaridade direcional, e não da magnitude. A figura 3.3 mostra a imagem original a máscara e como no resultado final os descritores só são criados na área de interesse. Os descritores normalizados gerados nesta etapa serão utilizados na correspondência entre as imagens, como discutido na seção seguinte.

Figura 3.3: Imagem original mais a esquerda, máscara da folha no meio e pontos-chave encontrados na imagem mais a direita.

3.4 Correspondência entre imagens

A correspondência entre as imagens foi realizada utilizando Árvores de Vocabulário. Inicialmente, os descritores invariantes a escala, rotação, iluminação e ruído foram quantizados hierarquicamente por meio de um processo multinível de *k-means clustering* utilizando k igual a 80 e 3 níveis (Seção 2.2).

Na fase *off-line*, os descritores foram agrupados recursivamente em aproximadamente k^L nós folha, seguindo a divisão por proximidade numérica. A memória necessária para essa estrutura foi estimada (Equação 2.15) resultando em um consumo de cerca de 76 MB para descritores com dimensão 128 (Seção 2.2).

Durante a fase *on-line*, cada descritor é propagado pela árvore, com um custo computacional de $O(\log n)$, e seu caminho é codificado em um único inteiro, (Seção 2.2). A similaridade entre as imagens é então determinada utilizando os vetores q_i e d_i (Equação 2.16), cujos componentes são ponderados por w_i, com base em entropia (Equação 2.18).

A pontuação final, calculada por meio de uma normalização (Equação 2.17), otimiza a discriminação dos descritores mais informativos.

Na imagem 3.4, são destacadas em amarelo as áreas semelhantes entre as duas imagens, que possuem características que poderiam gerar descritores semelhantes. Assim, os descritores correspondentes a essas áreas em ambas as imagens poderiam ser agrupados no mesmo conjunto.

Por fim, as correspondências brutas foram refinadas utilizando técnicas de geometria epipolar e RANSAC, que serão detalhadas na próxima seção.

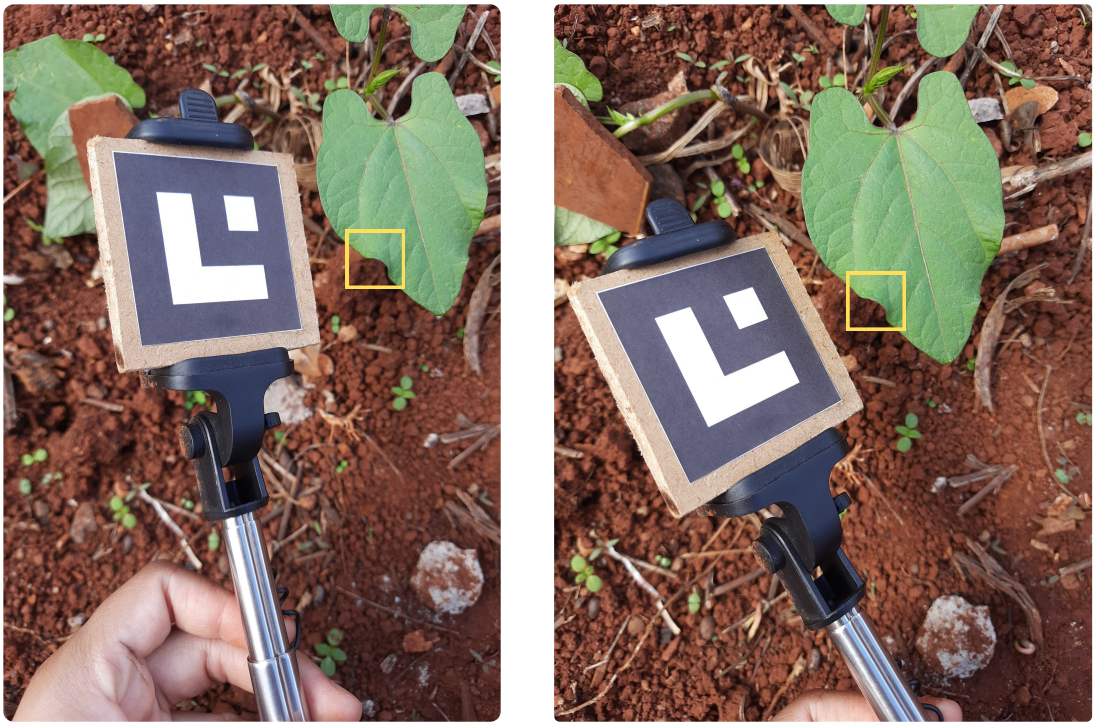


Figura 3.4: Imagens da folha 134 que mostram áreas semelhantes que poderiam estar relacionadas à descritores semelhantes.

3.5 Correspondência de Características

No final dos dois processos anteriores, espera-se que cada descritor extraído de uma imagem possua características específicas que permitem identificar pontos de interesse de maneira única. A partir dessa etapa, busca-se estabelecer correspondências entre os descritores extraídos de duas imagens I_A e I_B. A figura 3.5 ilustra que as correspondências são pontos presentes em ambas as imagens e representam o mesmo ponto na folha.

A correspondência entre os descritores é feita com base em uma métrica relativa,

considerando a proximidade entre os pontos. Para cada descritor d_A encontrado na imagem A, a tarefa é encontrar os dois descritores mais próximos na imagem B, denominados d_1 e d_2. A qualidade dessa correspondência é avaliada pela razão entre as distâncias dos dois descritores.

Se essa razão r for menor que um limite previamente determinado como 0.8 ($r < limiar$), considera-se que a correspondência é válida. Este critério ajuda a eliminar correspondências inconsistentes e características ambíguas, aumentando a precisão na identificação dos pontos correspondentes. Assim, é possível filtrar as correspondências mais confiáveis, evitando pares de descritores que possam ter sido erroneamente associados devido a pequenas variações ou ruídos nas imagens (Seção 2.3).

Após a correspondência inicial ser estabelecida, segue-se para a etapa de filtragem geométrica. A geometria epipolar é empregada para garantir que as correspondências entre os pontos observados em duas imagens diferentes respeitem as restrições geométricas impostas pelas projeções das câmeras. Como a câmera já estava calibrada, a estimativa das correspondências foi feita utilizando a matriz essencial E, que descreve a transformação geométrica entre as imagens com base nos parâmetros internos da câmera e na pose relativa entre as câmeras. A matriz essencial contém informações sobre a rotação e a translação entre as duas imagens, e sua estimativa permite refinar ainda mais as correspondências, garantindo que os pontos correspondam realmente a observações do mesmo ponto 3D no espaço (Seção 2.3.2).

A matriz essencial foi estimada a partir dos pares de pontos correspondentes identificados nas imagens, utilizando o método de estimação baseado no algoritmo de oito pontos. Esse algoritmo é projetado para calcular a matriz essencial a partir de um número mínimo de correspondências de pontos. Para lidar com a presença de *outliers*, foi utilizado o método RANSAC, eficaz na detecção e eliminação de correspondências incorretas. O procedimento do RANSAC envolve a seleção aleatória de um subconjunto mínimo de pontos correspondentes, a estimativa da matriz essencial a partir desse subconjunto e a verificação da consistência das demais correspondências com a matriz estimada, utilizando a distância de Sampson. O processo é repetido várias vezes para garantir que a solução final seja robusta, e a matriz essencial resultante no maior número de *inliers* é escolhida

como a solução final (Seção 2.3).

Uma vez que a matriz essencial foi estimada, ela é decomposta para obter a matriz de rotação $\boldsymbol{R}$ e o vetor de translação $\mathbf{t}$, os quais representam o movimento relativo entre as câmeras (Seção 2.3.2). Com essas informações, é possível melhorar a correspondência de pontos, uma vez que as projeções de um ponto 3D no espaço nas duas imagens devem satisfazer a condição de coplanaridade imposta pela geometria epipolar. Portanto, a matriz essencial, com a de rotação e translação, fornece uma maneira de validar as correspondências, garantindo que cada par de pontos projetados nas imagens esteja conforme as restrições geométricas e de movimento entre as câmeras.

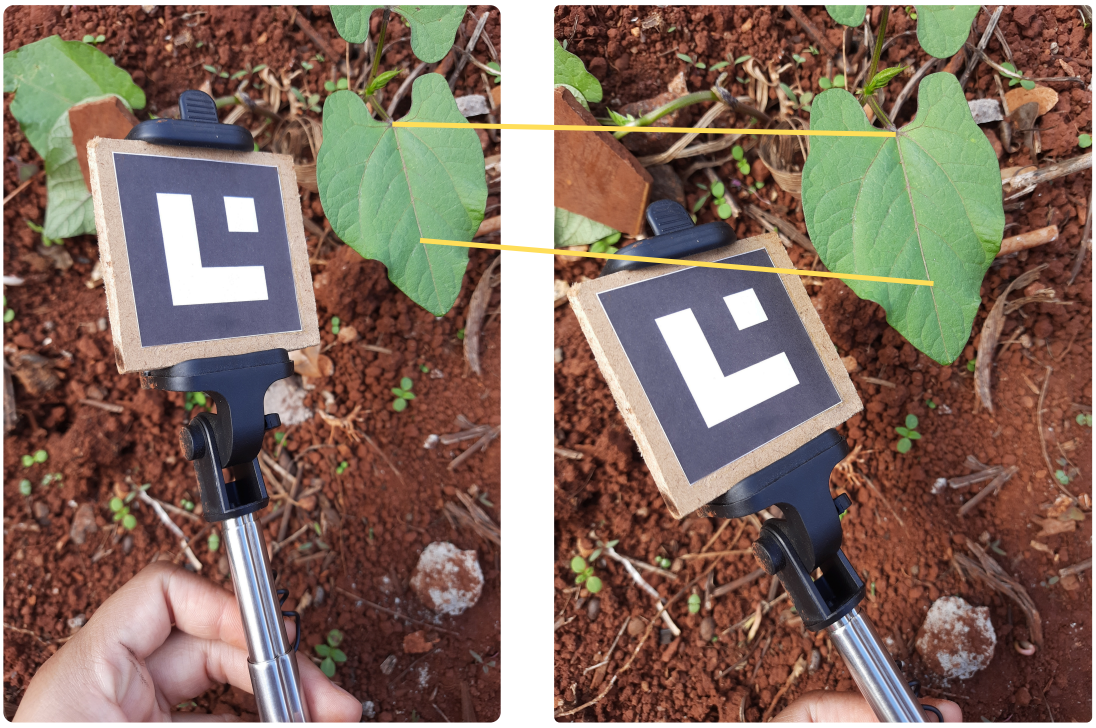


Figura 3.5: Imagens da folha 134, as linhas amarelas mostram possíveis correspondências entre as duas imagens

3.6 Inferência da Estrutura Rígida e da Pose da Cena 3D

Após fazer a correspondência de características inicial e verificar a quantidade e qualidade das correlações obtidas, o objetivo é agrupar todas as imagens em um contexto global.

O processo inicia-se com a seleção de imagens, etapa fundamental que influencia

diretamente a qualidade e a completude do modelo. O critério adotado baseia-se no número de associações válidas entre pontos 2D e pontos 3D já reconstruídos. Dessa forma, verifica-se a correspondência entre os pontos das novas imagens e suas projeções no modelo 3D. Para maximizar a aquisição de novas informações e evitar redundâncias, a abordagem segue o princípio da próxima melhor vista (*next best view*), priorizando imagens que fornecem novas perspectivas, ângulos ou detalhes da cena (Seção 2.4).

Uma vez selecionadas as imagens, procede-se à estimação das poses das câmeras, definindo sua posição e orientação em relação à cena. Utiliza-se o algoritmo *Perspective-n-Point* (PnP), que calcula a pose da câmera a partir das correspondências entre pontos 2D e 3D. Emprega-se o método RANSAC, que filtra possíveis *outliers* (Seção 2.4).

Com as poses refinadas, novos pontos 3D são triangulados a partir das correspondências entre pontos 2D-3D das imagens recém-adicionadas, enriquecendo o modelo 3D e permitindo uma representação mais detalhada da cena. A triangulação é realizada com base em critérios geométricos que evitam pontos mal condicionados. Pontos com ângulos de triangulação muito pequenos, que podem comprometer a precisão da reconstrução, são descartados ou ajustados (Seção 2.4.1). Periodicamente, aplica-se o *Bundle Adjustment*, uma técnica de otimização global que refina simultaneamente os parâmetros das câmeras (pose e intrínsecos) e as posições dos pontos 3D. O objetivo dessa etapa é minimizar o erro de reprojeção, garantindo que as projeções dos pontos 3D nas imagens correspondam o mais fielmente possível às observações reais. A função de erro a ser minimizada é dada pela equação 2.47.

Após o *Bundle Adjustment* , realiza-se uma filtragem para eliminar observações inconsistentes, como pontos com altos erros de reprojeção ou mal condicionados devido a ângulos de triangulação inadequados. Além disso, novas triangulações podem ser realizadas para corrigir ou complementar a reconstrução. Esse ciclo de seleção de imagens, estimação de poses, triangulação e refinamento é repetido iterativamente até que não sejam identificadas novas informações relevantes (Seção 2.4.2).

A imagem 3.6 mostra a saída final para a folha 134, sendo ela a pose relativa a cada imagem da reconstrução e a nuvem de pontos triangulados.

O uso do *Bundle Adjustment* é uma estratégia para que os erros de reprojeção

sejam minimizados, refinando continuamente a qualidade do modelo. O próximo passo é usar essas informações refinadas para obter o mapa de profundidade e, por fim, a superfície 3D da folha.

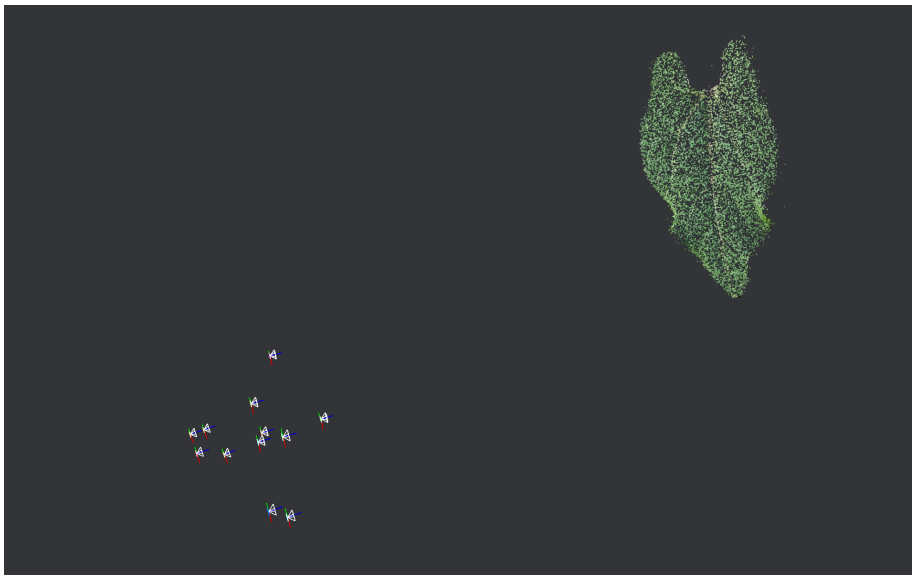


Figura 3.6: Nuvem de pontos e poses das câmeras obtidas para a folha 134.

3.7 Estimativa de mapas de profundidade e superfície geométrica densa

Após a aplicação desse método de minimização, segue-se visando obter uma representação geométrica densa da cena a partir das imagens estéreo.

O primeiro passo consiste na estimação da disparidade entre pares de imagens estéreo utilizando o algoritmo *Semi-Global Matching* (SGM). O volume de disparidade é construído e o custo de cada disparidade é calculado (Equação 2.49).

A disparidade final para cada píxel é escolhida minimizando o custo agregado (Equação 2.51). Com isso, obtêm-se os mapas de profundidade, a partir da relação de profundidade com disparidade (Equação 2.48) Os mapas de profundidade gerados são então integrados em uma estrutura octree, permitindo a fusão eficiente das informações de múltiplas visualizações. A fusão ocorre armazenando os valores de profundidade mais confiáveis em cada célula da octree, garantindo uma representação densa da superfície (Seção 2.5).

A partir da octree, constrói-se a malha 3D utilizando a tetraedralização de Delaunay, que divide o espaço tridimensional em tetraedros. Para refinar essa estrutura, aplica-se uma votação ponderada nas células e facetas, atribuindo maior peso às regiões que melhor representam a superfície da cena (JANCOSEK; PAJDLA, 2014).

A segmentação da malha é realizada por meio do algoritmo *Graph Cut Max-Flow* (BOYKOV; KOLMOGOROV, 2004), que separa a superfície do fundo com base na otimização global da estrutura. Em seguida, células malformadas são removidas, e um filtro Laplaciano é aplicado para suavizar a malha e eliminar imperfeições locais.

Em conclusão, o processo descrito gera uma representação tridimensional densa de uma folha numa cena a partir de múltiplas imagens estéreo. A aplicação do algoritmo SGM permite calcular disparidades e gerar mapas de profundidade, sendo então organizados em uma estrutura octree para fusão eficiente das informações. A tetraedralização de Delaunay e a votação ponderada refinam a malha 3D, enquanto a segmentação e os processos de filtragem, como o *Graph Cut Max-Flow* e o filtro Laplaciano, garantem uma superfície bem definida. Assim, o método resulta em uma reconstrução tridimensional refinada como mostrado na imagem 3.7. A seguir, são discutidos os resultados experimentais desse estudo.

Figura 3.7: Reconstrução final para a folha 134

4 Resultados Experimentais

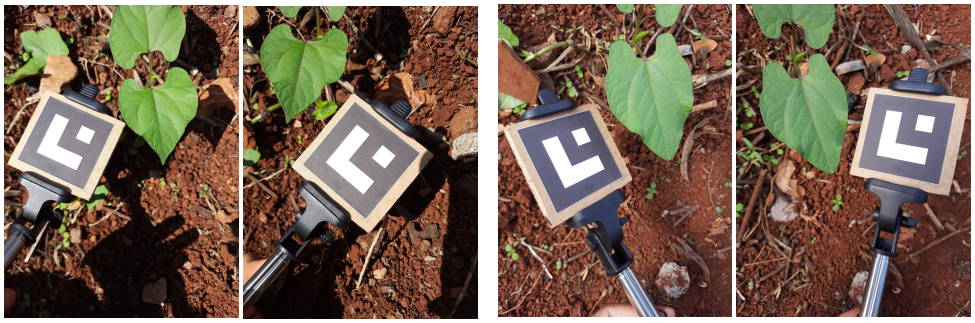
Este capítulo avalia o desempenho do método proposto para reconstrução 3D, aplicando-o em cinco folhas de feijão selecionadas da base de dados. A escolha de 5 folhas permitiu uma análise detalhada e controlada, viabilizando a validação inicial do método proposto. Estudos futuros devem expandir a amostra para garantir generalização. A discussão inicia-se abordando-se a estimativa inicial da matriz essencial, responsável por relacionar geometricamente as correspondências entre as vistas, e seus parâmetros de validação. A imagem 4.1 mostra as imagens usadas na estimativa inicial da matriz essencial.

Posteriormente, são detalhados os resultados das ressecções para cada nova imagem que refinam a posição 3D dos pontos e os parâmetros das câmeras, minimizando erros de reprojeção. Por fim, apresentam-se os resultados da reconstrução SfM, incluindo a precisão alcançada na modelagem tridimensional e a distribuição dos resíduos reprojetivos por meio de histogramas.

Folha	I_a **Imagem**	I_b **Imagem**	**limiar**	**# Pontos Usados**	**# Pontos Validados**	**% Pontos Validados**
52	20220506_103510.jpg	20220506_103747.jpg	12,1881	229	217	94,7598
134	20220509_104719.jpg	20220509_104716.jpg	6,86752	1810	1799	99,3923
192	20220509_145407.jpg	20220509_145348.jpg	6,01712	1043	1037	99,4247
221	20220510_090701.jpg	20220510_090634.jpg	9,02713	434	400	92,1659
353	20220511_105151.jpg	20220511_105122.jpg	9,08681	364	339	93,1319

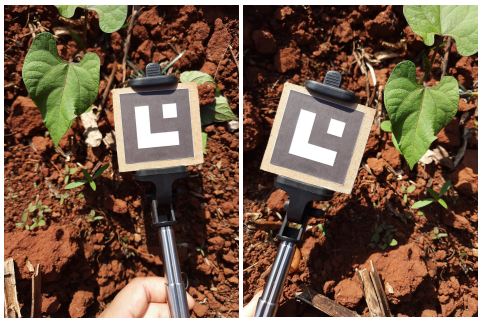
Tabela 4.1: Tabela com os resultados da estimativa da matriz essencial inicial para diferentes folhas.

A Tabela 4.1 resume os resultados da etapa inicial de estimativa da matriz essencial para pares de imagens de cinco folhas de feijão. As folhas 134 e 192 destacam-se pelo alto número de pontos usados (1810 e 1043, respectivamente) e pela alta porcentagem de validação ($\approx 99,4\%$). As folhas 52, 221 e 353, por outro lado, apresentam menos pontos usados (229, 434 e 364, respectivamente) e validação moderada ($\approx 92 - 93\%$). As folhas com menor número de pontos usados aparentam ser aquelas que possuem curvas mais acentuadas e maior complexidade geométrica, enquanto folhas mais planas parecem conter menos correspondências. Isso pode se relacionar também ao fato de que mesmo selecionando ambas as vistas com maior número de correspondências elas ainda não possuem uma variação muito significativa.

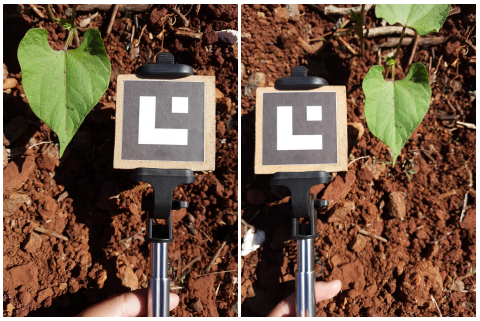


a) Par de imagens da folha 52 que possui um total de 12 imagens

c) Par de imagens da folha 134 que possui um total de 12 imagens



b) Par de imagens da folha 192 que possui um toltal de 12 imagens



d) Par de imagens da folha 221 que possui um total de 11 imagens

e)Par de imagens da folha 353 que possui um total de 16 imagens

Figura 4.1: A figura mostra as duas imagens usadas inicialmente no processo de reconstrução para cada folha, assim como indica a quantidade de imagens usadas na reconstrução para cada folha.

O limiar (limiar de erro reprojetivo para validação de pontos) varia significativamente. As folhas 134 e 192 operam com *limiares* baixos (6,87 e 6,02), mas mantêm validação quase perfeita. No processo inicial é importante garantir que haja pontos suficientes para a continuação da reconstrução, mas a alta validação pode introduzir *outliers* no modelo.

Tabela 4.2: Resultados de Ressecção por folha

Folha	Imagem	Thresh.	#Usados	#Valid.	%
52	20220506_103532.jpg	4,42	154	146	94,8
	20220506_103133.jpg	6,07	336	317	94,3
	20220506_103523.jpg	4,98	617	586	95,0
	20220506_103129.jpg	4,83	542	514	94,8
	20220506_103802.jpg	4,89	664	633	95,3
	20220506_103741.jpg	4,89	632	599	94,8
	20220506_103119.jpg	6,29	469	437	93,2
	20220506_103122.jpg	4,81	641	611	95,3
	20220506_103059.jpg	5,03	738	711	96,3
	20220506_103506.jpg	7,53	552	537	97,3
134	20220509_104706.jpg	5,65	1084	1060	97,8
	20220509_104655.jpg	3,90	1497	1447	96,7
	20220509_104703.jpg	3,60	2159	2095	97,0
	20220509_104659.jpg	2,50	2681	2555	95,3
	20220509_104650.jpg	2,98	3115	3010	96,6
	20220509_104647.jpg	3,14	2391	2314	96,8
	20220509_104653.jpg	3,28	2937	2822	96,1
	20220509_104645.jpg	3,82	2294	2221	96,8
	20220509_104642.jpg	3,25	2673	2589	96,9
	20220509_104640.jpg	3,87	2816	2730	96,9
192	20220509_145442.jpg	6,78	777	737	94,9
	20220509_145438.jpg	3,99	1857	1704	91,8
	20220509_145428.jpg	3,67	2618	2473	94,5
	20220509_145432.jpg	6,14	1647	1549	94,0
	20220509_145344.jpg	4,08	2187	2037	93,1
	20220509_145352.jpg	3,77	2452	2310	94,2
	20220509_145411.jpg	3,40	3020	2817	93,3
	20220509_145403.jpg	3,97	3341	3179	95,2
	20220509_145354.jpg	2,89	3219	3023	93,9
	20220509_145404.jpg	2,87	3699	3482	94,1
	20220510_090655.jpg	7,02	283	270	95,4
	20220510_090650.jpg	4,76	969	923	95,3

Continua na próxima página

Tabela 4.2 – continuação da página anterior

Folha	Imagem	Thresh.	#Usados	#Valid.	%
	20220510_090646.jpg	4,15	1507	1439	95,5
	20220510_090720.jpg	3,67	1826	1748	95,7
	20220510_090726.jpg	4,00	1970	1895	96,2
	20220510_090712.jpg	2,91	2292	2177	95,0
	20220510_090707.jpg	2,92	2254	2142	95,0
	20220510_090643.jpg	6,28	801	753	94,0
	20220510_090638.jpg	5,18	653	607	93,0
353	20220511_105134.jpg	5,30	227	222	97,8
	20220511_105205.jpg	7,16	473	456	96,4
	20220511_105120.jpg	4,82	711	658	92,5
	20220511_105145.jpg	4,48	838	788	94,0
	20220511_105129.jpg	7,59	538	492	91,4
	20220511_105148.jpg	4,70	932	875	93,9
	20220511_105157.jpg	6,05	830	776	93,5
	20220511_105127.jpg	5,09	792	725	91,5
	20220511_105153.jpg	4,73	1029	962	93,5
	20220511_105124.jpg	4,24	1008	944	93,7
	20220511_105113.jpg	4,28	1122	1050	93,6
	20220511_105139.jpg	5,21	1083	1040	96,0
	20220511_105117.jpg	4,37	1161	1098	94,6
	20220511_105141.jpg	4,62	1018	949	93,2

A tabela 4.2 indica uma validação superior a 90% em todas as folhas analisadas, mas essa métrica por si só não garante a precisão da reconstrução. O erro médio quadrático residual (RMSE) varia entre 0,698 e 1,024, mostrando que, mesmo com alta validação, a qualidade final pode ser impactada por outros fatores. Um dos aspectos relevantes é a quantidade de pontos reconstruídos, que parece influenciar a precisão, mas não linearmente. A Folha 134, com 11698 pontos reconstruídos, apresentou o menor erro (0,698), enquanto a Folha 353, com 5894 pontos reconstruídos, teve o maior erro (1,024). No entanto, a Folha 192, que contou com um número significativo de pontos reconstruídos (10506), apresentou um erro maior que a Folha 134, sugerindo que somente aumentar a quantidade de pontos não é suficiente para garantir uma melhor reconstrução. A distri-

buição e qualidade das correspondências também desempenham um papel fundamental.

Além disso, o tempo de processamento variou consideravelmente entre as folhas, indo de 2,16 a 12,09 segundos. Folhas com maior densidade de pontos tendem a demandar mais tempo, mas nem sempre essa relação se traduz em um menor erro. A Folha 353, por exemplo, teve um tempo intermediário de 6,23 segundos, mas apresentou o maior RMSE, sugerindo que o tempo de processamento por si só não é um bom indicador de precisão. Isso levanta questões sobre a eficiência do algoritmo e a necessidade de um balanceamento entre tempo de execução e qualidade final da reconstrução.

Outro fator que requer atenção é o impacto dos limiares na seleção dos *inliers*. Observa-se que limiares mais elevados podem levar à inclusão de *outliers*, como é o caso da Folha 353, que, com um limiar de 7,59, apresentou o maior RMSE. Por outro lado, a Folha 134, que utilizou limiares moderados entre 2,5 e 3,8, obteve a melhor precisão, o que sugere que valores mais controlados podem ser ideais para evitar contaminação por pontos errôneos. Esse aspecto deveria ser investigado mais profundamente, testando diferentes configurações de limiares para identificar padrões e determinar a melhor abordagem conforme a complexidade da cena.

Diante desses pontos, a análise dos resultados deveria ir além da simples observação das validações elevadas e considerar com mais profundidade a relação entre os diferentes fatores que influenciam a precisão da reconstrução. A seleção de *inliers*, a distribuição dos pontos reconstruídos, o tempo de processamento e a configuração dos limiares precisam ser estudados de maneira conjunta para entender melhor os impactos na qualidade final. Embora os resultados sejam promissores, há inconsistências que precisam ser melhor investigadas.

Folha	**# Nº Imagens**	**# Landmarks**	**Tempo Decorrido (s)**	**RMSE Residual**
52	12	4459	2,164	0,946523
134	12	11698	7,015	0,698394
192	12	10506	12,093	0,88206
221	11	5242	3,324	0,863202
353	16	5894	6,23	1,02486

Tabela 4.3: Resultados do processo SfM

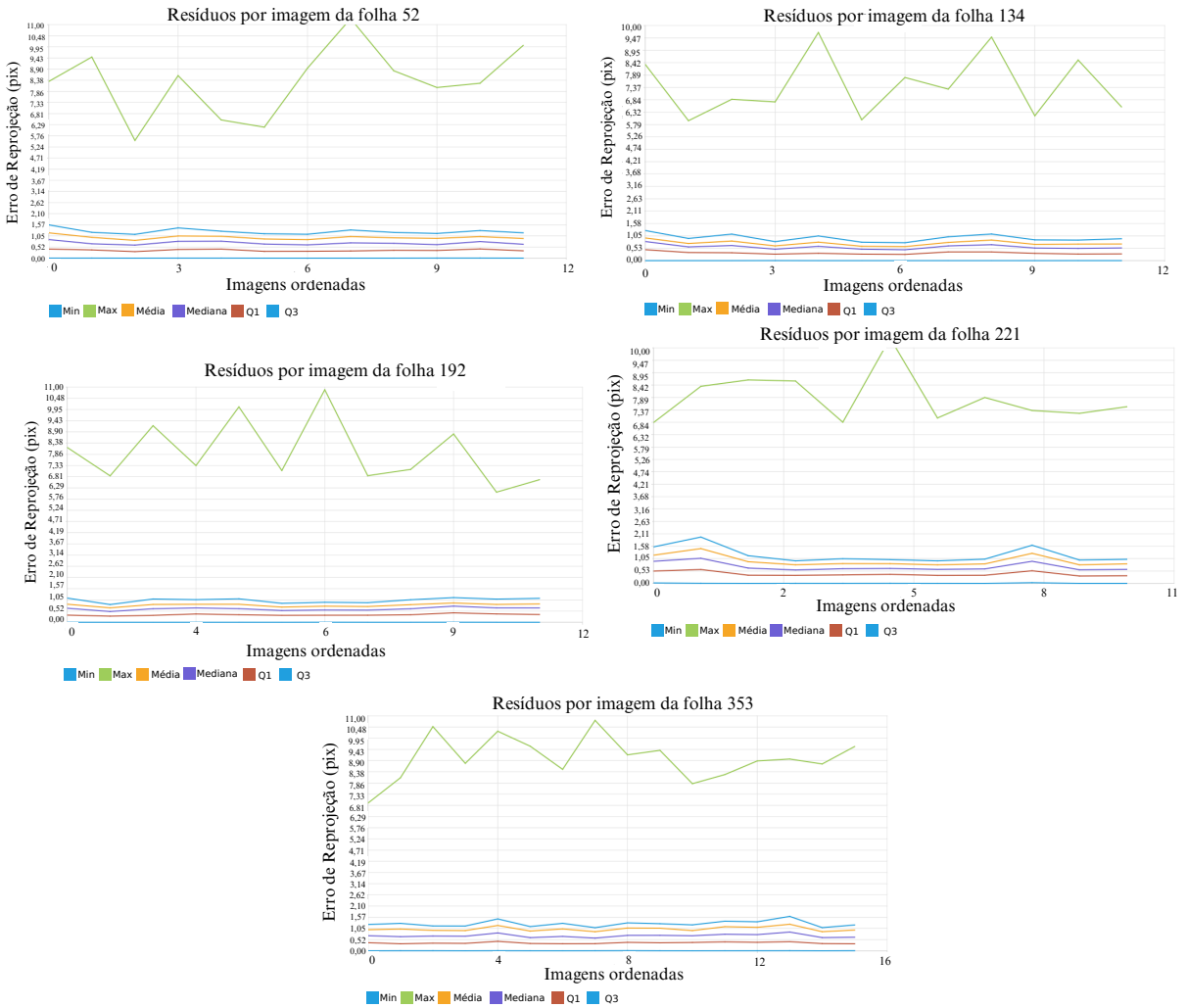


Figura 4.2: Gráficos de cada folha que mostram os resíduos por imagem

A análise dos dados revela que o processo de SfM apresenta variações significativas em desempenho e precisão entre os diferentes conjuntos de imagens folhas. Em geral, observa-se uma relação entre o número de pontos reconstruídos e a precisão do modelo reconstruído, medida pelo RMSE residual. Por exemplo, a Folha 134, com 11698 pontos reconstruídos, obteve o menor RMSE (0,698) e um tempo de processamento relativamente baixo (7,015 s), indicando eficiência tanto na reconstrução quanto na otimização. Contudo, essa correlação não é absoluta: a Folha 192, com 10506 pontos reconstruídos, registrou o RMSE de 0,88206 equivalente à folhas com menos pontos como a 221, possivelmente devido a problemas de baixa sobreposição entre imagens ou presença de *outliers*.

O tempo de processamento variou amplamente, de 2,164 s (Folha 52) a 12,093 s (Folha 192), sem uma relação direta com a quantidade de pontos reconstruídos. A Folha 192, por exemplo, demandou quase o dobro do tempo da Folha 134, apesar de ter

menos pontos reconstruídos, o que sugere complexidade adicional na otimização do *Bundle Adjustment* ou desafios na convergência do algoritmo. Já a Folha 221, com 11 imagens e 5.242 pontos reconstruídos, apresentou um RMSE intermediário (0,863), reforçando que a qualidade das correspondências entre características influencia mais a precisão do que a mera quantidade de dados.

Casos críticos destacam desafios específicos. A Folha 353, com o maior número de imagens (16), teve o pior desempenho em RMSE, possivelmente por obter uma baixa densidade de pontos reconstruídos. Por outro lado, a Folha 134 demonstrou ser possível alcançar boa precisão mesmo com grandes volumes de dados, se houver pontos reconstruídos bem distribuídos. Recomenda-se, portanto, priorizar a aquisição de imagens com sobreposição adequada e texturas ricas para maximizar pontos reconstruídos válidos e ajustes no *Bundle Adjustment* para reduzir tempos de processamento em casos complexos, como o da Folha 192. Conclui-se que a eficácia do SfM depende não somente da quantidade de dados, mas da integração entre qualidade das imagens, robustez das correspondências e eficiência computacional.

De maneira geral, os valores de RMSE indicam um bom ajuste do modelo, com a maioria das execuções apresentando erros residuais relativamente baixos conforme mostrado nos gráficos 4.2 no qual a média de resíduo por imagem se manteve estável e entre 1 e 0. A concentração dos resíduos em torno de 0 indica que a maioria dos pontos estão bem ajustados ao modelo, com desvios pequenos e aceitáveis. E isso se confirma também na análise dos histogramas em 4.3 que mostram a distribuição dos resíduos, com as maiores frequências próximas aos valores 0 e 1. Isso condiz com o RMSE residual obtido para o ajuste, que reflete a alta precisão da reconstrução 3D. A precisão não é necessariamente determinada pelo número de pontos reconstruídos ou imagens, mas sim pela qualidade do alinhamento, como exemplificado na execução da folha 353, que, apesar de ter mais dados, obteve um RMSE mais alto. A execução da folha 134, com o RMSE mais baixo, foi a que apresentou a melhor precisão, refletindo a qualidade da reconstrução 3D. Assim, o RMSE proporciona uma boa visão da qualidade do modelo 3D, sendo que valores menores indicam um ajuste mais preciso e fiel às imagens de entrada.

As reconstruções finais, em geral, condizem bem com as folhas, apresentando uma

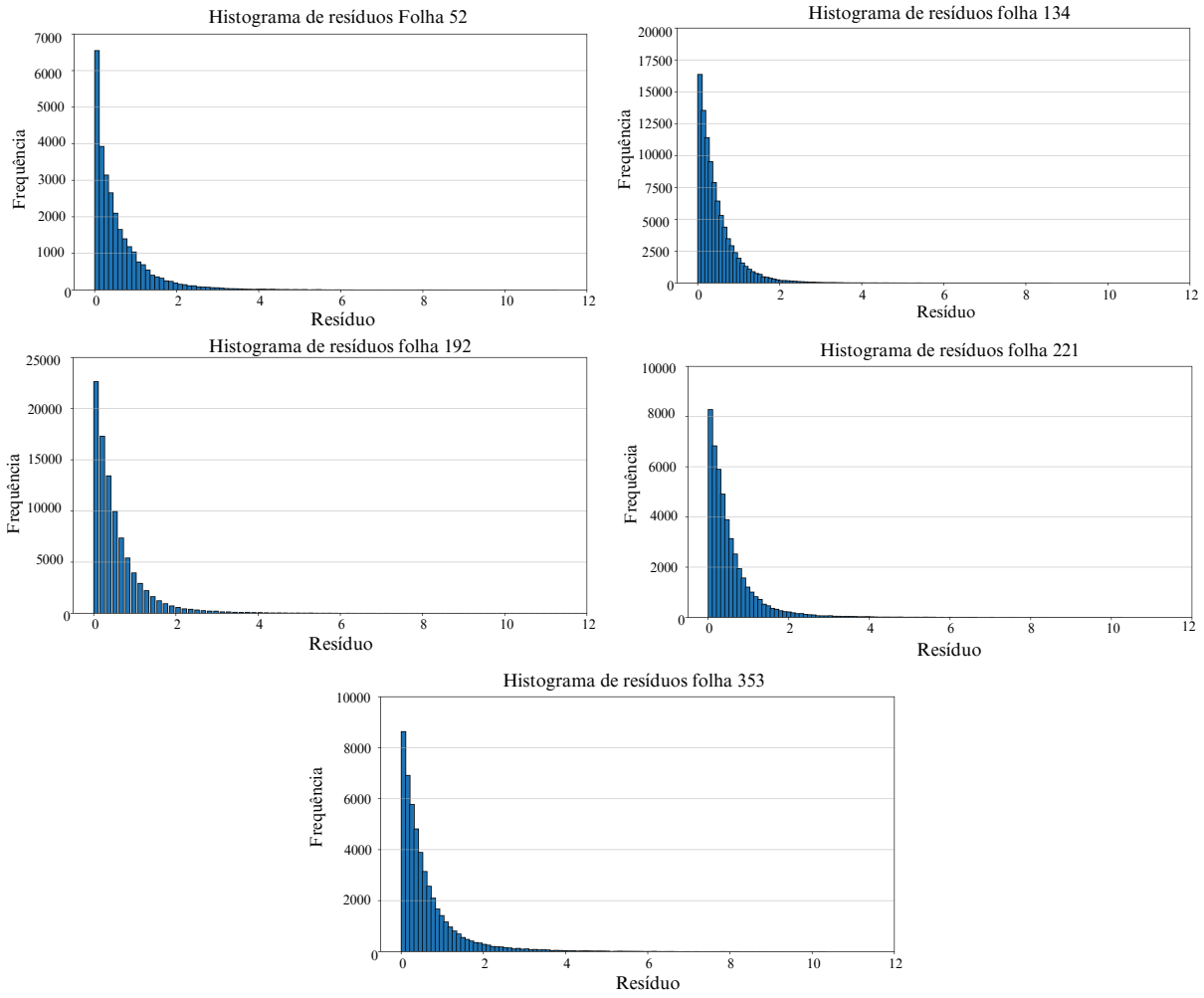


Figura 4.3: Histogramas dos resíduos do processo SfM.

boa correspondência em termos de forma como mostrado nas figuras 4.4. As estruturas modeladas, de modo geral, replicam de forma satisfatória os contornos principais das folhas, indicando uma precisão razoável no processo de reconstrução. No entanto, alguns detalhes finos foram perdidos durante a reconstrução. Isso se traduz, principalmente, em uma suavização excessiva de certas protuberâncias menores nas folhas e na dificuldade de preencher buracos pequenos que são mais desafiadores de capturar. Embora o formato global das folhas tenha sido bem preservado, essas omissões menores afetam a fidelidade dos detalhes mais sutis da estrutura da folha.

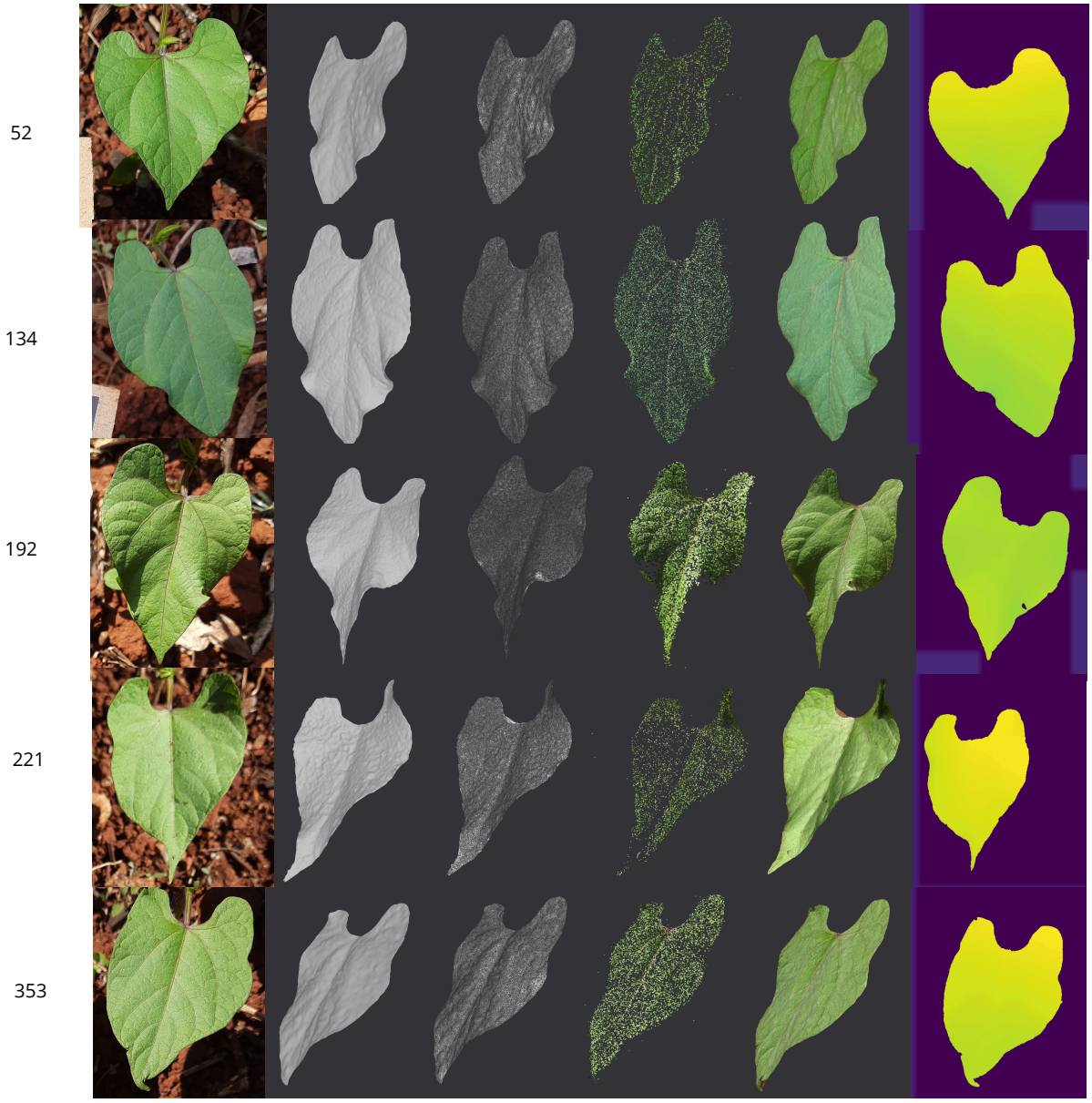


Figura 4.4: Etapas da reconstrução para cada uma das 5 folhas. Da esquerda para a direita tem-se, malha 3D, malha 3D com visualização das faces, nuvem de pontos, malha texturizada e, por fim, mapa de profundidade de uma das vistas

5 Conclusão e trabalhos futuros

Por fim, a proposta deste trabalho alinha-se ao objetivo geral que é desenvolver um método baseado em visão computacional para a reconstrução tridimensional não destrutiva de folhas de feijão. E propõe-se aplicar e avaliar técnicas de SfM (Structure from Motion) e estereoscopia na reconstrução 3D de folhas, além de ampliar a base de dados com informações geométricas tridimensionais para enriquecer as análises fenotípicas.

Os resultados experimentais evidenciam um bom indício que o algoritmo SfM consegue refinar relativamente bem tanto os parâmetros intrínsecos quanto os pontos, ajustando-os de maneira a se adequarem ao contexto global de todas as imagens analisadas. Essa capacidade de adaptação é fundamental para a obtenção de uma malha 3D precisa e consistente, conforme indicado pelos valores finais de RMSE, que se mantêm relativamente baixos, demonstrando ser um método promissor para a reconstrução da superfície de folhas de feijão.

Contudo, os experimentos também revelam que as etapas de reconstrução podem enfrentar desafios significativos quando submetidas a condições adversas. Em cenários com variações intensas de iluminação, mudanças abruptas de textura, presença de ruídos e movimentos acentuados, a integridade dos dados pode ser comprometida, afetando negativamente a qualidade da reconstrução final das folhas. Tais fatores impõem complexidades adicionais ao processo, exigindo uma abordagem mais robusta para garantir resultados satisfatórios.

Diante desses desafios, torna-se evidente a necessidade de aprimoramento contínuo. Assim, trabalhos futuros buscarão integrar métodos que permitam uma menor sensibilidade a situações adversas. A aplicação de redes neurais convolucionais, por exemplo, surge como uma estratégia promissora, ao poder contribuir para a invariância do processo relacionada a variações ambientais, melhorando potencialmente a reconstrução 3D.

Abordagens híbridas, que combinam técnicas tradicionais de SfM com métodos de *deep learning*, têm o potencial de ampliar a versatilidade do sistema, permitindo que ele lide de forma mais eficaz com diferentes tipos de interferência. Com isso, não só se

aprimora a qualidade da malha gerada, como também se abre caminho para aplicações mais avançadas em análise de imagens e modelagem tridimensional.

Outra meta futura importante é aplicar o método no conjunto total de folhas de feijão da base de dados a fim de prover informação de profundidade em relação à câmera para todas as suas imagens e conter mais informações para garantir a generalização da análise de dados.

Bibliografia

BOYKOV, Y.; KOLMOGOROV, V. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 9, p. 1124–1137, 2004.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 24, n. 6, p. 381–395, jun. 1981. ISSN 0001-0782. Disponível em: ⟨https://doi.org/10.1145/358669.358692⟩.

HARTLEY, R.; ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. 2. ed. [S.l.]: Cambridge University Press, 2004.

HIRSCHMULLER, H. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 30, n. 2, p. 328–341, 2008.

JANCOSEK, M.; PAJDLA, T. Multi-view reconstruction preserving weakly-supported surfaces. In: *CVPR 2011*. [S.l.: s.n.], 2011. p. 3121–3128.

JANCOSEK, M.; PAJDLA, T. Exploiting visibility information in surface reconstruction to preserve weakly supported surfaces. *International Scholarly Research Notices*, v. 2014, p. 798595, 2014. Published 2014 Aug 11. Disponível em: ⟨https://doi.org/10.1155/2014/798595⟩.

KAK, P. U. A. *An Introduction to the SGM Algorithm for Dense Matching in Binocular Stereo*. 2024. Disponível em: ⟨https://engineering.purdue.edu/kak/Tutorials/SemiGlobalMatching.pdf⟩.

LABATUT, P.; PONS, J.-P.; KERIVEN, R. Robust and Efficient Surface Reconstruction From Range Data. *Computer Graphics Forum*, The Eurographics Association and Blackwell Publishing Ltd, 2009. ISSN 1467-8659.

LOWE, D. Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. [S.l.: s.n.], 1999. v. 2, p. 1150–1157 vol.2.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, Kluwer Academic Publishers, USA, v. 60, n. 2, p. 91–110, nov. 2004. ISSN 0920-5691. Disponível em: ⟨https://doi.org/10.1023/B:VISI.0000029664.99615.94⟩.

NISTER, D.; STEWENIUS, H. Scalable recognition with a vocabulary tree. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. [S.l.: s.n.], 2006. v. 2, p. 2161–2168.

SCHöNBERGER, J. L.; FRAHM, J.-M. Structure-from-motion revisited. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 4104–4113.

SILVA, K. G. F. da et al. Bean leaf image dataset annotated with leaf dimensions, segmentation masks, and camera calibration. *Data in Brief*, v. 59, p. 111328, 2025. ISSN 2352-3409. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S2352340925000605⟩.

TAREEN, S. A. K.; SALEEM, Z. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In: *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. [S.l.: s.n.], 2018. p. 1–10.