UNIVERSIDADE FEDERAL DE OURO PRETO
Instituto de Ciências Exatas e Aplicadas
Curso de Engenharia de Computação

UTILIZAÇÃO DO AWS REKOGNITION NA IDENTIFICAÇÃO DE OBJETOS PARA RECONSTRUÇÃO 3D

Vinicius Assis e Silva

João Monlevade – MG
2026

Vinicius Assis e Silva

UTILIZAÇÃO DO AWS REKOGNITION NA IDENTIFICAÇÃO DE OBJETOS PARA RECONSTRUÇÃO 3D

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Federal de Ouro Preto, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Darlan Nunes de Brito

João Monlevade – MG

2026

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTACAO E SISTEMAS

FOLHA DE APROVAÇÃO

Vinicius Assis e Silva

Utilização do AWS RECKOGNITION na identificação de objetos para reconstrução 3D

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de bacharel em Engenharia de Computação

Aprovada em 27 de julho de 2026

Membros da banca

Doutor - Darlan Nunes de Brito - Orientador - Universidade Federal de Ouro Preto
Doutora - Gilda Aparecida de Assis - Universidade Federal de Ouro Preto
Mestre - Maurício José Aureliano Júnior - Universidade Federal de Ouro Preto

Darlan Nunes de Brito, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 11/08/2026

Documento assinado eletronicamente por **Darlan Nunes de Brito**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 11/08/2026, às 11:22, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **1159035** e o código CRC **BB7CF119**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3808-0819 - www.ufop.br

Agradecimentos

Agradeço, primeiramente, a Deus, a quem atribuo todas as minhas conquistas.

Agradeço aos meus pais, aos meus avós e a todos que, de alguma forma, me apoiaram ao longo desta jornada. Sem a presença e o apoio de cada um, este momento não seria possível.

À Universidade Federal de Ouro Preto (UFOP), em especial ao Instituto de Ciências Exatas e Aplicadas (ICEA), agradeço pela oportunidade de acesso a um ensino público, gratuito e de qualidade.

Agradeço aos professores que contribuíram para minha formação. Em especial, ao Professor Darlan Nunes de Brito, pela orientação, pelo apoio, pela paciência, pelos conhecimentos compartilhados e pelas contribuições ao desenvolvimento deste trabalho.

Agradeço também à Professora Gilda Aparecida de Assis e ao Professor Maurício José Aureliano Júnior, por aceitarem compor a banca examinadora e pelas contribuições decorrentes da avaliação deste trabalho.

A todos, expresso minha mais sincera gratidão.

Resumo

Este trabalho investiga o uso do AWS Rekognition para a identificação e o processamento de objetos em imagens e vídeos, integrando essas funcionalidades a uma pipeline modular de reconstrução tridimensional (3D). A reconstrução 3D exige a identificação precisa de elementos visuais, sendo um processo tradicionalmente complexo e com alto custo computacional. Em cenários reais, conjuntos de imagens capturados por câmeras convencionais ou por drones frequentemente contêm redundâncias, imagens com baixa nitidez ou ângulos inadequados, o que eleva desnecessariamente a carga computacional nas etapas subsequentes e pode comprometer a qualidade do modelo final. Nesse contexto, o AWS Rekognition, um serviço de visão computacional baseado em técnicas de aprendizado profundo disponibilizado pela Amazon Web Services, é utilizado como etapa de pré-processamento para filtragem automática dessas imagens irrelevantes antes da reconstrução. O estudo contempla desde a aquisição em vídeo e a extração de frames até a visualização do modelo gerado no MeshLab. A pipeline proposta é estruturada de forma modular, dividida em etapas independentes e interconectadas: extração de frames do vídeo por amostragem uniforme, filtragem via AWS Rekognition, reconstrução esparsa e densa por meio de Structure from Motion (SfM) e Multi-View Stereo (MVS) com o COLMAP, e alinhamento de nuvens de pontos por *Global Registration* (RANSAC sobre descritores FPFH) seguido de refinamento por Iterative Closest Point (ICP) e reconstrução de superfície com a técnica de *Screened Poisson*. Essa arquitetura favorece maior flexibilidade, escalabilidade e facilidade de análise individual de cada fase do processo. As vantagens, limitações e possíveis aplicações da solução são discutidas, com foco no seu potencial para cenários de reconstrução 3D com hardware convencional e recursos limitados.

Palavras-chave: Reconstrução 3D. AWS Rekognition. Structure from Motion. Nuvem de pontos. Fotogrametria.

Abstract

This work investigates the use of AWS Rekognition for object identification and processing in images and videos, integrating these capabilities into a modular three-dimensional (3D) reconstruction pipeline. 3D reconstruction requires the accurate identification of visual elements, a process that is traditionally complex and computationally expensive. In real-world scenarios, image sets captured by conventional cameras or drones frequently contain redundancies, blurry frames, or inadequate viewing angles, which unnecessarily increase the computational load in subsequent stages and may degrade the quality of the final model. In this context, AWS Rekognition, a deep learning-based computer vision service provided by Amazon Web Services, is employed as a preprocessing step for the automatic filtering of irrelevant images prior to the reconstruction. The study covers the entire workflow, from video acquisition and frame extraction to the visualization of the generated model in MeshLab. The proposed pipeline is structured in a modular fashion, divided into independent yet interconnected stages: uniform frame extraction from video, filtering via AWS Rekognition, sparse and dense reconstruction using Structure from Motion (SfM) and Multi-View Stereo (MVS) with COLMAP, and point cloud registration via Global Registration (RANSAC over FPFH descriptors) followed by Iterative Closest Point (ICP) refinement, and surface reconstruction using the Screened Poisson technique. This architecture enhances flexibility, scalability, and the ability to individually analyze each processing step. Finally, the advantages, limitations, and potential applications of the proposed solution are discussed, highlighting its suitability for scenarios that require 3D reconstruction under limited computational resources.

Keywords: 3D Reconstruction. AWS Rekognition. Structure from Motion. Point cloud. Photogrammetry.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

2D	Bidimensional
3D	Tridimensional
3DGS	*3D Gaussian Splatting*
4PCS	*4-Points Congruent Sets*
API	*Application Programming Interface* (Interface de Programação de Aplicações)
AWS	*Amazon Web Services*
BA	*Bundle Adjustment* (ajuste de feixes)
CUDA	*Compute Unified Device Architecture*
DCT	*Discrete Cosine Transform* (Transformada Discreta de Cosseno)
FPFH	*Fast Point Feature Histograms*
GPU	*Graphics Processing Unit* (Unidade de Processamento Gráfico)
GR	*Global Registration* (registro global)
IAM	*Identity and Access Management*
ICP	*Iterative Closest Point*
JPEG	*Joint Photographic Experts Group*
MVS	*Multi-View Stereo*
NCC	*Normalized Cross-Correlation* (correlação cruzada normalizada)
NeRF	*Neural Radiance Fields*
PCA	*Principal Component Analysis* (Análise de Componentes Principais)
pHash	*Perceptual Hash*
PLY	*Polygon File Format*
RAM	*Random Access Memory* (Memória de Acesso Aleatório)
RANSAC	*Random Sample Consensus*
RGB	*Red, Green and Blue* (vermelho, verde e azul)
RMSE	*Root Mean Square Error* (Raiz do Erro Quadrático Médio)
S3	*Simple Storage Service*
SDK	*Software Development Kit*
SfM	*Structure from Motion*
SIFT	*Scale-Invariant Feature Transform*
SPSR	*Screened Poisson Surface Reconstruction*
SSD	*Solid-State Drive* (unidade de estado sólido)

Sumário

1 Introdução

A reconstrução 3D é uma das áreas mais ativas da visão computacional, com aplicações que vão da preservação de patrimônio histórico à inspeção industrial, passando por robótica, medicina e entretenimento [6]. O que antes dependia de equipamentos caros e equipes especializadas tornou-se gradualmente acessível com câmeras convencionais e software livre. Essa democratização do acesso trouxe uma consequência prática importante: hoje é comum capturar centenas de imagens de um objeto em campo, sendo que boa parte delas é redundante, desfocada ou simplesmente não contribui para a reconstrução.

A Figura 1 representa diferentes formas de aquisição de múltiplas vistas e destaca a captura orbital por smartphone adotada neste trabalho.

Figura 1: Representação das formas de aquisição de múltiplas vistas de um objeto para reconstrução 3D, com destaque para a captura orbital por smartphone adotada neste trabalho.

Fonte: Imagem gerada com inteligência artificial, sob orientação do autor (2026).

Técnicas como Structure from Motion (SfM) e Multi-View Stereo (MVS) são sensíveis à qualidade das imagens de entrada [5]. Frames desfocados, posições redundantes ou imagens em que o objeto está parcialmente fora do enquadramento introduzem ruído nas estimativas de câmera e degradam a nuvem de pontos resultante. Abordagens mais recentes baseadas em aprendizado profundo, como Neural Radiance Fields (NeRF) e 3D Gaussian Splatting (3DGS), alcançam qualidade superior, mas exigem GPUs de alto desempenho e longos tempos de processamento, o que as torna inviáveis em hardware convencional [6].

A triagem das imagens de entrada raramente é tratada de forma sistemática. Capturar um vídeo orbital de 30 segundos com smartphone produz mais de 1.000 frames,

sendo que boa parte é redundante ou está fora de foco. Selecionar os melhores manualmente é viável para amostras pequenas, mas rapidamente se torna inescalável. A filtragem automática antes do SfM pode, portanto, reduzir o custo computacional sem comprometer a qualidade do modelo final.

Os principais tipos de frames que motivam essa triagem são ilustrados na Figura 2: imagens úteis, imagens desfocadas, vistas redundantes e quadros nos quais o objeto não aparece no enquadramento.

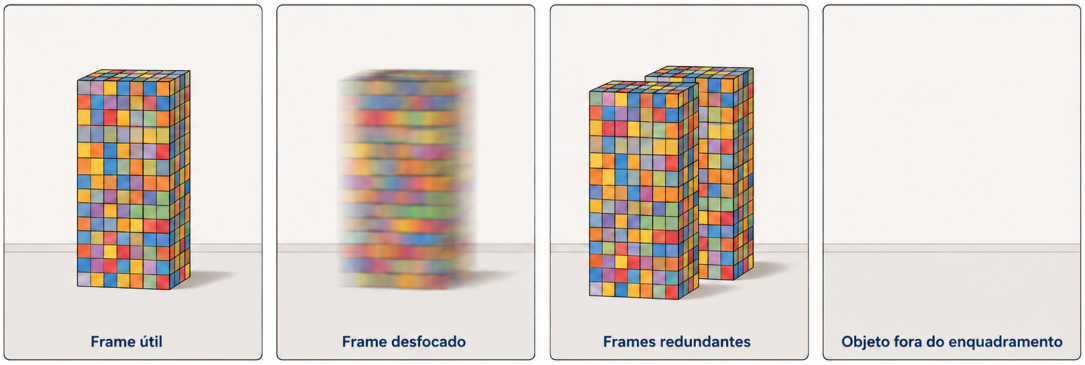


Figura 2: Representação dos principais tipos de frames encontrados durante a aquisição em vídeo: frame útil, frame desfocado, frames redundantes e ausência do objeto no enquadramento.

Fonte: Imagem gerada com inteligência artificial, sob orientação do autor (2026).

Este trabalho propõe uma pipeline modular de reconstrução 3D a partir de vídeos capturados com smartphone, com foco na etapa de pré-filtragem automática dos frames antes do SfM. O AWS Rekognition é usado como filtro semântico: apenas frames em que o objeto de interesse é identificado com confiança suficiente avançam para as etapas computacionalmente custosas. O restante da pipeline usa o COLMAP para SfM e MVS, o Open3D para alinhamento por ICP e reconstrução de superfície por *Screened Poisson*. Todas essas ferramentas são consolidadas e não dependem de aprendizado profundo no núcleo da reconstrução.

A Figura 3 apresenta uma visão conceitual desse fluxo, desde a captura orbital até a geração do modelo tridimensional.

O objetivo principal é avaliar se a filtragem prévia via AWS Rekognition reduz o custo computacional da pipeline sem comprometer a qualidade do modelo 3D gerado.

1.1 Objetivos Específicos

Para atingir o objetivo principal do trabalho, definem-se os seguintes objetivos específicos:

1. Desenvolver um módulo de filtragem automática de imagens capaz de avaliar nitidez,

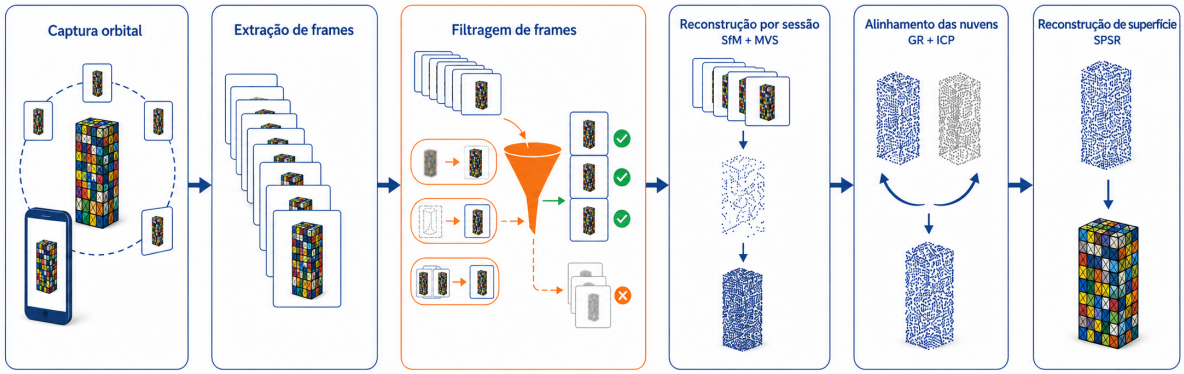


Figura 3: Visão geral conceitual da pipeline proposta, abrangendo captura orbital, extração e filtragem dos frames, reconstrução das nuvens de pontos, alinhamento entre sessões e geração do modelo tridimensional.

Fonte: Imagem gerada com inteligência artificial, sob orientação do autor (2026).

indícios semânticos do objeto de interesse no cenário controlado do experimento e redundância entre frames extraídos de vídeo.

2. Integrar um serviço de visão computacional em nuvem (AWS Rekognition) como etapa de verificação semântica, sem treinamento ou hospedagem de modelos locais.

3. Reconstruir modelos tridimensionais de um objeto real a partir de vídeos capturados por smartphone, utilizando técnicas consolidadas de reconstrução baseada em imagens.

4. Alinhar nuvens de pontos obtidas em sessões de captura distintas em um sistema de coordenadas comum.

5. Gerar malhas poligonais adequadas à visualização a partir das nuvens de pontos alinhadas.

6. Avaliar experimentalmente o impacto da filtragem prévia sobre o tempo de processamento e sobre a qualidade das reconstruções.

2 Fundamentação Teórica

Antes de descrever a metodologia e os resultados, vale revisar os fundamentos das principais técnicas envolvidas. Cada etapa da pipeline apoia-se em um conjunto específico de conceitos matemáticos e algorítmicos que influenciam diretamente as decisões de implementação tomadas ao longo do trabalho.

2.1 Detecção de Nitidez por Variância do Laplaciano

O operador Laplaciano mede a variação local do gradiente de uma função, somando suas derivadas parciais de segunda ordem nas direções x e y. Este trabalho utiliza a implementação `cv2.Laplacian` da biblioteca OpenCV, cuja documentação oficial [14] define o operador como:

$$\texttt{dst} = \Delta\,\texttt{src} = \frac{\partial^2\,\texttt{src}}{\partial x^2} + \frac{\partial^2\,\texttt{src}}{\partial y^2} \tag{1}$$

Como a Equação 1 é definida para funções contínuas, o OpenCV a aproxima numericamente por convolução com o seguinte kernel (`ksize=1`):

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{2}$$

Em imagens, esse operador produz respostas altas nas bordas dos objetos, onde a intensidade muda abruptamente, e respostas próximas de zero em regiões uniformes. Uma imagem nítida tem bordas bem definidas, o que gera alta variância sobre os pixels. Uma imagem desfocada tem transições suaves, e por consequência reduz a variância do operador sobre a imagem inteira.

A variância da resposta Laplaciana sobre a imagem é utilizada neste trabalho como métrica de nitidez (Seção 3.3.2).

2.2 Eliminação de duplicidade por *Perceptual Hash* (pHash)

O *perceptual hash* (pHash) é um resumo compacto do conteúdo visual de uma imagem. Diferentemente de uma comparação pixel a pixel, a imagem passa por uma sequência de operações antes da comparação: é convertida para escala de cinza, redimensionada para 32×32 pixels e submetida à Transformada Discreta de Cosseno (DCT-II) bidimensional [33]. Do resultado, retém-se apenas o bloco 8×8 de coeficientes de baixa frequência, que concentra a estrutura global da imagem e descarta detalhes finos e ruído, cada coeficiente é então comparado à mediana do bloco, gerando um *hash* binário de 64 bits. Este trabalho utiliza a implementação do pacote *imagehash* [16].

A comparação entre duas imagens é feita pela distância de Hamming entre seus *hashes*, o número de posições em que dois vetores binários de mesmo comprimento diferem [34]. Como o *hash* preserva apenas a estrutura visual de baixa frequência, duas imagens visualmente semelhantes produzem *hashes* com pequena distância de Hamming entre si. A similaridade percentual entre dois frames é dada por $s = (1 - d/64) \times 100\%$, em que d representa a distância de Hamming. Dois frames são considerados redundantes quando $s \geq \rho_{\max}$, sendo $\rho_{\max}$ o limiar de similaridade configurável (Seção 3.3.2).

2.3 Structure from Motion (SfM)

O Structure from Motion (SfM) reconstrói geometria 3D esparsa e estima as posições das câmeras simultaneamente, usando correspondências de pontos entre imagens com sobreposição [1]. O ponto de partida é detectar pontos de interesse que sejam estáveis a variações de escala e rotação.

2.3.1 Extração de Características: SIFT

O extrator SIFT (*Scale-Invariant Feature Transform*), originalmente proposto por Lowe [29], detecta pontos de interesse como extremos locais no espaço de escala gaussiano, definido por:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3}$$

onde $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$ é um filtro Gaussiano de desvio-padrão σ e $*$ denota convolução. A diferença de gaussianas $D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$ aproxima o Laplaciano de Gaussiana normalizado por escala e é usada para detectar eficientemente extremos no espaço de escala. Para cada ponto detectado, o SIFT calcula um descritor de 128 dimensões baseado em histogramas de gradientes orientados em uma vizinhança ao redor do ponto, normalizado para invariância a variações de iluminação [29].

2.3.2 Geometria Epipolar e Estimativa da Posição

A correspondência de características é feita comparando os descritores dos pontos de interesse por meio de uma métrica de distância no espaço de características. Como essa comparação considera apenas a semelhança entre descritores, e não a consistência geométrica da cena, nem todos os pares encontrados representam, de fato, o mesmo ponto físico. Em outras palavras, algumas correspondências podem estar incorretas, gerando os chamados *outliers*. Para lidar com esse problema, o COLMAP realiza uma verificação geométrica entre cada par de imagens, estimando uma transformação projetiva que seja compatível com as correspondências obtidas [1].

O tipo de transformação estimada depende da configuração entre as câmeras. Quando a câmera apenas gira em torno do próprio eixo ou a cena é aproximadamente plana, utiliza-se uma homografia H. Já no caso mais comum, em que a câmera também se desloca durante a captura, a relação entre as imagens é descrita pela geometria epipolar. Nessa situação, emprega-se a matriz essencial E, quando as câmeras são calibradas, ou a matriz fundamental F, quando não há calibração disponível [2].

Como as correspondências obtidas a partir da aparência visual normalmente incluem erros, a estimativa desses modelos é realizada de forma robusta utilizando o algoritmo RANSAC (*Random Sample Consensus*) [4]. O algoritmo seleciona repetidamente pequenos subconjuntos de correspondências, calcula um modelo geométrico candidato e

verifica quantos dos demais pontos são compatíveis com esse modelo. Esse conjunto de correspondências consistentes é chamado de conjunto de consenso. Ao final das iterações, o RANSAC mantém o modelo que apresenta o maior número de correspondências consistentes, reduzindo significativamente a influência dos *outliers* na reconstrução.

2.3.3 Refinamento do ajuste

O Refinamento do ajuste (Chamado em inglês de *Bundle Adjustment*, BA) é a etapa de otimização global do SfM. O COLMAP o define como o refinamento conjunto não linear dos parâmetros de câmera $\mathbf{P}_c$ e dos pontos 3D $\mathbf{X}_k$, minimizando o erro de reprojeção [1]:

$$E = \sum_j \rho_j\left(\|\pi(\mathbf{P}_c, \mathbf{X}_k) - \mathbf{x}_j\|_2^2\right) \tag{4}$$

onde $\pi(\cdot)$ é a função que projeta pontos da cena para o espaço da imagem, $\mathbf{x}_j$ é a observação 2D correspondente, e ρ_j é uma função de perda usada para reduzir o peso de outliers [1]. O COLMAP resolve o problema com a biblioteca Ceres Solver, usando como ρ_j a função de Cauchy:

$$\rho(s) = \log(1 + s) \tag{5}$$

onde $s = \|\pi(\mathbf{P}_c, \mathbf{X}_k) - \mathbf{x}_j\|_2^2$ é o quadrado residual [17]. O problema é resolvido pelo algoritmo de Levenberg–Marquardt [3, 2], que aproveita a esparsidade da hessiana (cada ponto é visível em apenas um subconjunto de câmeras) para manter a otimização viável mesmo com centenas de imagens. O COLMAP usa reconstrução incremental, começa com o par de imagens de maior sobreposição, registra novas câmeras uma a uma por resseção de perspectiva e executa ajustes finos locais ao longo do processo, com um ajuste global ao final [1].

2.4 Multi-View Stereo (MVS): PatchMatch Stereo

Com as posições da câmera estimadas pelo SfM, o Multi-View Stereo (MVS) aumenta a densidade da reconstrução, estimando profundidade e normal para cada pixel das imagens, e gera uma nuvem de pontos com fidelidade geométrica maior [30]. O COLMAP usa uma variante do PatchMatch Stereo: para cada pixel da imagem de referência, estima conjuntamente profundidade, normal e visibilidade (quais imagens vizinhas realmente enxergam aquele ponto sem oclusão), usando a correlação cruzada normalizada (NCC) entre as partes como medida de similaridade fotométrica [30]. Estimativas de alta qualidade se propagam espacialmente entre pixels vizinhos, e perturbações aleatórias de profundidade e normal diversificam o espaço de busca para evitar ótimos locais. Essa é

a mesma estratégia de propagação e amostragem aleatória que caracteriza o algoritmo PatchMatch [30].

O COLMAP acrescenta uma verificação de consistência geométrica: cada estimativa de profundidade é projetada nas imagens vizinhas e, em seguida, projetada de volta na imagem original. As estimativas cujo erro de reprojeção ultrapassa um limiar configurável são descartadas ou refinadas [30]. Esse passo reduz profundidades falsas. Os valores adotados neste trabalho são descritos na Seção 3.3.4. Os mapas de profundidade de todas as vistas são então fundidos em uma única nuvem de pontos, processo que elimina redundâncias e reduz ruído [30].

2.5 Alinhamento de Nuvens de Pontos: ICP

O algoritmo Iterative Closest Point (ICP), introduzido por Besl e McKay [31] na variante ponto-a-ponto, resolve o problema de registro de nuvens de pontos: encontrar a transformação rígida (R, t) que minimiza a distância entre pontos correspondentes de uma nuvem fonte e uma nuvem alvo. O objetivo dessa transformação é permitir que capturas realizadas de perspectivas ou sessões distintas, cada uma reconstruída em seu próprio sistema de coordenadas, sejam fundidas em uma única representação consistente do objeto. Este trabalho usa a variante *point-to-plane*, proposta por Chen e Medioni [32], que minimiza a distância de cada ponto fonte ao plano tangente à superfície alvo no ponto correspondente, em vez da distância ponto-a-ponto direta. A função objetivo dessa variante está documentada pelo Open3D [11], biblioteca que implementa o algoritmo neste trabalho. O ICP executa os seguintes passos de forma iterativa até a convergência:

1. **Correspondência**: para cada ponto da nuvem fonte, encontrar o ponto mais próximo na nuvem alvo dentro de uma distância máxima $d_{\max}$.
2. **Estimativa**: linearizar (R, t) pela aproximação de pequenos ângulos e resolver o sistema linear resultante, minimizando a soma das distâncias ponto-a-plano sobre o conjunto de correspondências [32].
3. **Atualização**: aplicar a transformação estimada à nuvem fonte.
4. **Verificação**: repetir até que a variação do RMSE entre iterações seja inferior a ϵ ou o número máximo de iterações seja atingido.

Tanto a distância máxima de correspondência $d_{\max}$ quanto os critérios de parada são parâmetros definidos pelo usuário do algoritmo. Os valores adotados neste trabalho são descritos na Seção 3.3.5. A limitação fundamental do ICP, em qualquer variante, é a dependência da posição inicial. Se as nuvens de pontos estiverem muito distantes no começo, o algoritmo converge para um mínimo local errado [20]. Por isso, costuma-se executar antes um alinhamento global, que fornece uma estimativa grosseira da transformação

para o ICP refinar. Neste trabalho, emprega-se o *Global Registration* por RANSAC sobre descritores FPFH (*Fast Point Feature Histograms*), implementado no Open3D. Outra alternativa descrita na literatura é o *4-Points Congruent Sets* (4PCS) [10].

2.6 Reconstrução de Superfície: *Screened Poisson*

A *Screened Poisson Surface Reconstruction* (SPSR) [7] recebe uma nuvem de pontos com normais orientadas e devolve uma malha fechada. A ideia por trás do método é tratar a superfície como o contorno de uma função indicadora χ, um campo escalar que separa o interior do exterior do objeto, recuperada resolvendo uma equação de Poisson a partir das normais dos pontos de entrada, que funcionam como um campo de gradientes a ser aproximado.

A formulação original do método busca a função escalar χ cujo gradiente melhor aproxima o campo vetorial $\vec{V}$ derivado das normais dos pontos de entrada, minimizando a energia

$$E(\chi) = \int \left\| \vec{V}(p) - \nabla\chi(p) \right\|^2 dp \tag{6}$$

A versão *screened*, usada neste trabalho, acrescenta à Equação 6 um termo de fidelidade que penaliza o desvio de χ em relação a zero nos próprios pontos da nuvem [7]:

$$E(\chi) = \int \left\| \vec{V}(p) - \nabla\chi(p) \right\|^2 dp \; + \; \alpha \cdot \frac{\text{Area}(P)}{\sum_{p\in P} w(p)} \sum_{p\in P} w(p)\, \chi^2(p) \tag{7}$$

onde P é o conjunto de pontos de entrada, $w(p)$ são pesos por amostra (unitários na implementação de referência), $\text{Area}(P)$ é a área estimada da superfície e α é o peso de *screening*, que controla o equilíbrio entre ajustar o campo de gradientes e interpolar os pontos [7]. Nessa formulação, a função indicadora χ é definida com valor $+\frac{1}{2}$ no interior e $-\frac{1}{2}$ no exterior do objeto, de modo que sua isosuperfície de nível zero passe próxima aos pontos de entrada [7]. É por esse motivo que o termo de fidelidade penaliza $\chi^2(p)$: ele puxa a superfície para perto de onde os pontos realmente estão. Na prática, isso corrige a tendência do Poisson original de suavizar demais os detalhes finos e reduz distorções em regiões com ruído [7].

Toda essa otimização acontece sobre uma octree adaptativa, cuja profundidade máxima é controlada pelo parâmetro `depth`, regiões mais densas da nuvem recebem uma árvore mais profunda e, portanto, maior resolução local. Ao final, a malha é extraída por *marching cubes*, percorrendo a isosuperfície de χ no isovalor calculado como a média de χ avaliada nos próprios pontos de amostra de entrada, e não no nível zero, uma vez que a solução da equação de Poisson é definida apenas a menos de uma constante aditiva [7]. O método costuma lidar bem com ruído moderado, produzindo malhas suaves mesmo quando os pontos de entrada têm imperfeições [7].

As normais usadas nessa etapa têm origem majoritariamente no próprio COLMAP: o `stereo_fusion` (Seção 2.4) já funde, junto com a posição de cada ponto, a normal estimada pelo PatchMatch Stereo a partir do plano local ajustado a cada pixel dos mapas de profundidade [30]. Apenas quando essa fusão falha e a pipeline recorre ao mecanismo de contingência de retroprojeção dos mapas de profundidade (Seção 3.3.4), a nuvem resultante chega sem normais. Nesse caso, o Open3D as calcula por análise de covariância dos k vizinhos mais próximos de cada ponto, sem uma etapa adicional de orientação consistente entre pontos vizinhos [11].

2.7 Visão Computacional em Nuvem: AWS Rekognition

O AWS Rekognition é um serviço de análise de imagens e vídeos da Amazon Web Services. Neste trabalho, ele é consumido como uma caixa-preta por meio da API *DetectLabels*. A AWS não documenta publicamente a arquitetura interna do modelo empregado [18].

A API *DetectLabels* recebe uma imagem e retorna rótulos referentes a objetos, cenas ou conceitos, acompanhados de valores de confiança entre 0 e 100. O parâmetro *MinConfidence* define o limiar mínimo para que um rótulo seja retornado pela API [18]. Esses valores são utilizados neste trabalho como critério operacional de filtragem, sem interpretá-los como probabilidades calibradas.

A integração é feita via SDK Python (*boto3*): a imagem é enviada em bytes e a resposta contém os rótulos detectados e seus respectivos valores de confiança. Embora a API possa disponibilizar informações adicionais para determinados rótulos, como instâncias e caixas delimitadoras, a pipeline utiliza apenas o nome do rótulo e a confiança para decidir se o frame segue para as etapas posteriores.

3 Desenvolvimento

3.1 Problema

Quando se captura um objeto em campo, o conjunto de frames gerado raramente é homogêneo. Frames desfocados, posições redundantes, imagens em que o objeto saiu do enquadramento e ângulos que nada acrescentam à reconstrução acabam no mesmo lote que os frames úteis. Isso acontece com câmeras convencionais e é ainda mais frequente em capturas com drones, onde o volume de dados é grande.

SfM e MVS são sensíveis à qualidade do conjunto de entrada. Imagens ruins dificultam a correspondência de características entre pares, introduzem erro nas posições estimadas e adicionam ruído à nuvem de pontos. Além disso, o tempo de processamento cresce com o número de imagens: mesmo os frames inúteis precisam passar por todas as etapas se não forem removidos antes.

Selecionar as imagens manualmente funciona para um número pequeno de fotos. Com centenas de frames extraídos de um vídeo, essa triagem não é prática.

Este trabalho propõe usar o AWS Rekognition como filtro automático nessa etapa. Cada frame é analisado pela API antes de entrar no SfM. Apenas os que contêm o objeto de interesse com confiança suficiente seguem adiante. O objetivo é reduzir o volume de entrada sem precisar de intervenção manual e sem treinar ou hospedar nenhum modelo local.

A hipótese central deste trabalho é que a filtragem prévia via AWS Rekognition contribui para a melhoria da eficiência computacional da pipeline de reconstrução 3D, sem comprometer significativamente a qualidade do modelo tridimensional gerado.

3.2 Trabalhos Relacionados

A reconstrução 3D a partir de imagens é um problema estudado há décadas, e a literatura disponível é ampla. Esta seção apresenta os trabalhos mais diretamente relacionados ao contexto desta proposta.

3.2.1 Reconstrução 3D Clássica: SfM e MVS

A pipeline clássica de reconstrução 3D a partir de imagens divide o problema em duas etapas: Structure from Motion (SfM), que estima a geometria esparsa e as posições da câmera a partir de correspondências de características, e Multi-View Stereo (MVS), que usa essas posições para calcular mapas de profundidade e gerar uma nuvem densa. Zhu [28], em tese de doutorado de 2015, foi um dos primeiros a tratar essa separação de forma sistemática, estruturando cada módulo de forma independente, organização que os sistemas posteriores adotaram como padrão.

O COLMAP [1], de Schönberger e Frahm, é hoje a implementação de referência para essa pipeline. Usa SIFT para extração de características, correspondência exaustiva entre pares de imagens e ajustes finos (*bundle adjustment*) para refinar globalmente as posições estimadas. O módulo MVS, baseado em PatchMatch Stereo, estima profundidade e normais por correspondência fotométrica entre imagens vizinhas.

3.2.2 Abordagens Baseadas em Aprendizado Profundo

Redes neurais profundas abriram caminhos diferentes para a reconstrução 3D. Samavati e Soryani [5] fazem uma revisão do estado da arte, cobrindo redes convolucionais, transformers e modelos generativos para estimativa de profundidade e reconstrução volumétrica. A conclusão é clara: a qualidade melhorou, mas o custo computacional continua sendo um obstáculo real para quem não tem acesso a GPUs de alto desempenho.

O DUSt3R [24] é um exemplo representativo desse trade-off: usa um transformer para estimar diretamente pontos 3D a partir de pares de imagens sem calibração prévia

de câmera, com resultados impressionantes em estimativa de posição e profundidade, mas com custo de processamento muito acima das abordagens clássicas. NeRF e 3DGS seguem na mesma direção: qualidade fotorrealista, mas com exigências de hardware que a maioria dos cenários práticos não tem [6].

Num contexto mais próximo ao deste trabalho, Wu, Shen e Ke [26] propõem uma pipeline eficiente para reconstrução com múltiplas imagens e nuvens de pontos, mostrando que a qualidade das imagens de entrada tem impacto direto na qualidade final do modelo. Isso reforça a motivação da etapa de pré-filtragem proposta neste trabalho.

3.2.3 Alinhamento de Nuvens de Pontos

Alinhar duas nuvens de pontos, como as reconstruções obtidas em duas sessões de captura do mesmo objeto, significa encontrar a rotação e translação que melhor as sobrepõe. O ICP [20] é o algoritmo mais usado para isso: associa cada ponto da nuvem fonte ao vizinho mais próximo na nuvem alvo, estima a transformação que minimiza a distância e repete até convergir. O problema é que ele precisa de uma inicialização razoavelmente próxima da solução correta. Nuvens com posições muito diferentes simplesmente não convergem para o alinhamento certo.

O 4PCS [10] e sua variante Super 4PCS contornam isso fazendo um alinhamento global sem depender de inicialização. A combinação 4PCS + ICP (grosseiro depois fino) tornou-se padrão exatamente por isso [21].

Shang e Shen [27] exploram a qualidade do alinhamento usando arranjos de câmeras de profundidade em pipelines de reconstrução em tempo real, mostrando que sensores complementares reduzem erros de alinhamento mesmo sem iterações extensas de ICP. Da Silva e Apaza-Agüero [25], em artigo publicado na Revista Brasileira de Computação Aplicada, propõem uma pipeline de reconstrução de baixo custo que combina câmeras RGB e de profundidade para melhorar o alinhamento sem hardware especializado. O objetivo é próximo ao deste trabalho: bons resultados dentro de restrições de custo, trocando hardware adicional por uma seleção mais cuidadosa das imagens de entrada.

3.2.4 Reconstrução de Superfície

A etapa de reconstrução de superfície converte uma nuvem de pontos densa em uma malha poligonal fechada. A técnica de *Screened Poisson Surface Reconstruction* [7], desenvolvida por Kazhdan e Hoppe, permanece como uma das abordagens mais eficientes e amplamente utilizadas para essa finalidade. O método formula o problema como a resolução de uma equação de Poisson sobre uma grade octree adaptativa, produzindo superfícies suaves e sem artefatos mesmo em presença de ruído moderado.

Desenvolvimentos recentes propõem extensões significativas ao método original. O Neural Poisson Surface Reconstruction (nPSR) [9] utiliza operadores de Fourier neurais

para resolver a equação de Poisson, o que permite extrair a malha em qualquer resolução sem reprocessar a nuvem de entrada. O p-Poisson [8], apresentado no NeurIPS 2023, elimina a necessidade de normais orientadas, como entrada, ampliando a aplicabilidade do método a nuvens de pontos brutas. Apesar desses avanços, este trabalho adota a implementação clássica do *Screened Poisson* disponível na biblioteca Open3D [11], por ser consolidada e amplamente validada. Conforme a documentação oficial da biblioteca, essa função implementa o método proposto por Kazhdan e Hoppe [7] e utiliza a implementação original de Kazhdan [13].

3.2.5 Fotogrametria com Dispositivos Móveis

Smartphones evoluíram ao ponto de se tornarem uma alternativa real a câmeras dedicadas para fotogrametria de pequena escala [22]. Sensores de 50 a 200 megapixels, estabilização óptica (OIS) e processamento computacional em tempo real colocam esses dispositivos num patamar que antes exigia equipamento profissional.

Comparado a drones, capturar com smartphone em movimento orbital tem vantagens práticas: menor custo, mais controle em ambientes fechados e sem burocracia de operação. A desvantagem é a cobertura angular menor por posição, o que exige mais posições de captura para garantir sobreposição suficiente. Mesmo assim, trabalhos recentes confirmam que SfM + MVS a partir de imagens de smartphone produz resultados comparáveis a câmeras dedicadas para objetos de pequena e média escala [23].

3.2.6 Visão Computacional em Nuvem

Serviços de visão computacional em nuvem como o AWS Rekognition são amplamente usados em aplicações industriais e comerciais, mas quase não aparecem no contexto acadêmico de reconstrução 3D. Os poucos trabalhos que os mencionam os usam em tarefas isoladas de detecção ou moderação de conteúdo. Usar o Rekognition especificamente como filtro semântico antes do SfM, verificando se o objeto de interesse está presente em cada frame, é o que diferencia este trabalho dos demais encontrados na literatura.

3.3 Metodologia

A pipeline modular implementada é composta pelas seguintes etapas:

1. **Aquisição de vídeo**: captura do objeto alvo em vídeo por câmera de smartphone, realizando movimento orbital ao redor do objeto.

2. **Extração de frames**: seleção de N frames uniformemente espaçados ao longo do vídeo e exportação como imagens JPEG.

3. **Filtragem com AWS Rekognition**: descarte automático de frames onde o objeto de interesse não é identificado com confiança suficiente.

4. **Structure from Motion (SfM)**: estimativa da nuvem de pontos esparsa e das posições da câmera a partir do conjunto de frames filtrado.

5. **Multi-View Stereo (MVS)**: geração da nuvem de pontos densa a partir dos frames e das posições estimadas.

6. **Alinhamento com *Global Registration* + ICP**: registro de múltiplas nuvens de pontos em um único sistema de coordenadas. O alinhamento grosseiro é feito por *Global Registration* (RANSAC sobre descritores FPFH) e o refinamento, por ICP.

7. **Reconstrução de superfície**: conversão da nuvem de pontos densa em malha poligonal para visualização e exportação.

A Figura 4 apresenta o diagrama da pipeline, com o fluxo de dados entre as etapas. As subseções a seguir detalham cada uma delas.

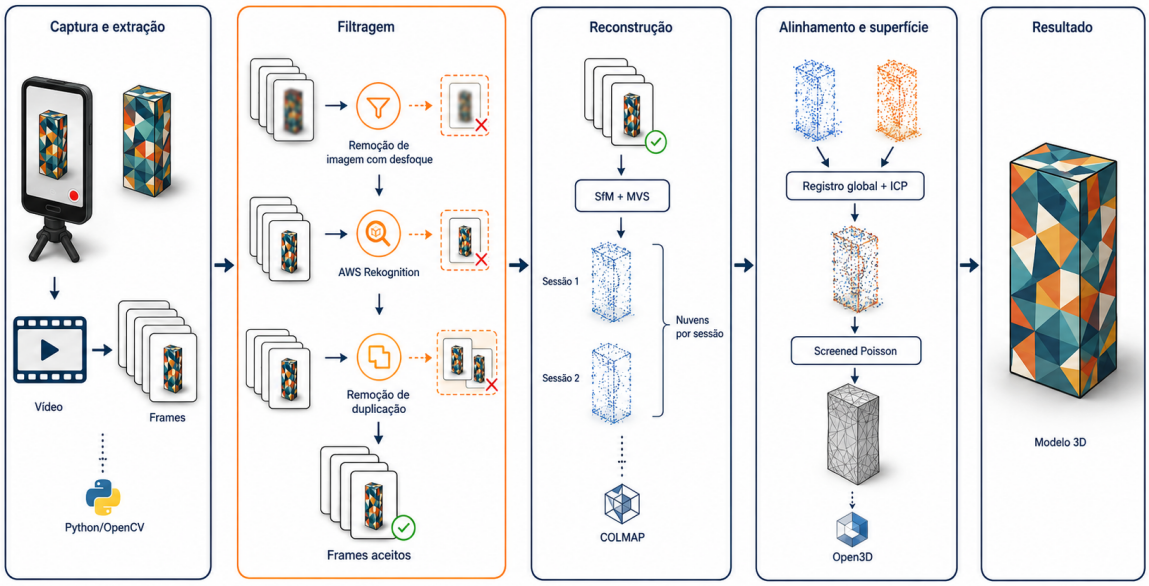


Figura 4: Detalhamento da pipeline modular implementada, desde a captura e extração dos frames até a filtragem, reconstrução das nuvens de pontos, alinhamento entre sessões e geração da malha tridimensional.

Fonte: Imagem gerada com inteligência artificial, sob orientação do autor (2026).

3.3.1 Aquisição de Vídeo e Extração de Frames

A aquisição de dados constitui a etapa inicial da pipeline e determina diretamente a qualidade máxima alcançável pelo modelo 3D resultante. Para os experimentos realizados neste trabalho, utilizou-se um smartphone Xiaomi Poco X6, equipado com câmera traseira

principal de 64 megapixels (sensor OmniVision OV64B, 1/2.0", abertura f/1,79, distância focal equivalente a 26 mm, estabilização óptica e eletrônica de imagem: OIS + EIS). A captura foi realizada pela câmera nativa do dispositivo, sem aplicativos de câmera de terceiros.

Optou-se pela gravação de vídeo e posterior divisão do vídeo em frames para obter as imagens utilizadas na reconstrução. Em todas as capturas, o objeto permaneceu imóvel no centro da trajetória, enquanto o smartphone foi segurado manualmente, em orientação vertical (retrato), e direcionado para a caixa. As distâncias foram medidas com trena, e marcações no chão foram utilizadas para orientar o raio do deslocamento. O operador caminhou lentamente ao redor do objeto, no sentido horário, procurando completar uma volta e manter a caixa enquadrada. A velocidade foi controlada apenas pelas passadas do operador, sem cronometragem ou equipamento externo de estabilização.

No dataset de calibração, capturado a aproximadamente 1 m, foram realizadas duas sessões orbitais com perspectivas distintas. Na primeira, o smartphone foi mantido aproximadamente na altura do centro da caixa, com enquadramento frontal e o aparelho paralelo ao objeto. Na segunda, o smartphone foi posicionado um pouco acima da caixa e inclinado entre aproximadamente 30° e 45° para baixo, permitindo registrar também sua região superior. Em ambas, a caixa foi enquadrada desde o início do vídeo e mantida visível durante o deslocamento circular. No dataset de validação, capturado a aproximadamente 40 cm, as duas sessões foram realizadas com distância, altura e inclinação aparentes semelhantes, mantendo-se o aparelho em orientação retrato e aproximadamente na altura da caixa. Nessas gravações, a captura começou com a caixa fora do enquadramento. Em seguida, o smartphone foi direcionado para o objeto, que se procurou manter enquadrado durante o restante da órbita.

Como os frames extraídos são amostras discretas do movimento orbital, a cobertura angular resultante é densa e regular, embora não contínua. Cada frame captura o objeto a partir de um ângulo ligeiramente distinto do anterior. O movimento físico buscou completar uma volta, enquanto a cobertura de aproximadamente 355° mencionada neste trabalho corresponde à trajetória posteriormente estimada pelo SfM para o dataset de calibração.

Foram gravadas duas sessões de captura do mesmo objeto a partir de perspectivas distintas. O vídeo da sessão 1 (`VIDEO1.mp4`) tem duração de 54,3 segundos e 1627 frames. O da sessão 2 (`VIDEO2.mp4`) tem 35,6 segundos e 1069 frames. Ambos foram gravados em resolução 4K retrato (2160 × 3840 pixels), a aproximadamente 30 fps, no formato H.264/MP4.

Esse conjunto de dados foi usado para calibrar a pipeline, capturado a aproximadamente 1 metro do objeto (Tabela 1), é sobre ele que se apoiam a varredura de parâmetros (*sweep*) de MVS e a identificação da distância de captura como principal fator associado à dominância do piso no alinhamento (Seção 4.6). Com a pipeline calibrada a partir

dessas descobertas, a validação final foi conduzida sobre um novo conjunto de capturas a aproximadamente 40 cm do objeto. A redução da distância motivou os demais ajustes do protocolo de captura, descritos na Seção 4.6 e avaliados na Seção 4.8.

A partir de cada vídeo, são extraídos $N = 130$ frames por amostragem uniforme: os índices de frame são espaçados linearmente do primeiro ao último frame, garantindo cobertura homogênea de todo o arco de captura. O valor de N foi definido empiricamente durante a calibração das etapas de SfM e MVS. Como os vídeos capturados são curtos, extrair mais frames elevava a proporção de quadros quase idênticos, e o valor adotado reduz essa duplicidade sem comprometer a cobertura da órbita. As trajetórias de câmera estimadas pelo SfM indicam varredura angular de aproximadamente 355 graus por sessão (Figura 11), o que corresponde a um espaçamento médio de cerca de 2,8 graus entre os 130 frames extraídos. Além disso, o custo da correspondência exaustiva do SfM cresce com o quadrado do número de imagens, o que desaconselha conjuntos de entrada desnecessariamente grandes. Os frames são exportados como imagens JPEG com qualidade 95%, na resolução original do vídeo (2160 × 3840 px). Esse nível de qualidade preserva os detalhes de textura necessários à extração de características, com artefatos de compressão visualmente imperceptíveis, e evita o custo de armazenamento de formatos sem perdas. Esse processo é realizado pelo módulo `frame_extractor` da pipeline, que utiliza a biblioteca OpenCV para leitura e exportação dos frames [15].

Tabela 1: Especificação do conjunto de vídeos utilizado na etapa de calibração da pipeline.

Parâmetro	**Valor**
Dispositivo	Xiaomi Poco X6
Câmera	Traseira principal, OmniVision OV64B, 64 MP
Sensor	1/2.0”, f/1,79, OIS + EIS
Modo de captura	Vídeo nativo, modo automático
Resolução do vídeo	2160 × 3840 px (4K retrato)
Taxa de quadros	≈ 30 fps
Codec	H.264 (MP4)
Padrão de captura	Orbital, cobertura angular densa
Sessão 1 (`VIDEO1.mp4`)	54,3 s – 1627 frames totais
Sessão 2 (`VIDEO2.mp4`)	35,6 s – 1069 frames totais
Frames extraídos por sessão	130 (amostragem uniforme)
Total de frames na pipeline	260 (130 por sessão)
Frames exportados	JPEG, qualidade 95, 2160 × 3840 px
Distância ao objeto	≈ 1 metro
Objeto fotografado	Caixa de papelão revestida com papéis coloridos

Fonte: Elaborado pelo autor com base nas especificações técnicas do fabricante e nos dados experimentais da pesquisa (2026).

O objeto de estudo é uma caixa de papelão originalmente lisa e de coloração uniforme, uma superfície de baixa textura, desfavorável às técnicas de reconstrução baseadas

em correspondência fotométrica, que dependem de regiões visualmente distintivas para estabelecer correspondências confiáveis entre imagens. Para viabilizar a reconstrução, a textura do objeto foi ampliada artificialmente: a caixa foi revestida com papéis coloridos em padrão xadrez multicolorido, criando bordas de alto contraste e regiões distintivas em todas as faces. A Figura 5 apresenta o objeto preparado, fotografado antes das capturas em vídeo.

Figura 5: Objeto fotografado (caixa de papelão revestida com papéis coloridos em padrão xadrez multicolorido), fotografias de referência tiradas antes da captura em vídeo.

Fonte: Fotografias produzidas pelo autor (2026).

3.3.2 Filtragem de Imagens

O módulo de filtragem desenvolvido neste trabalho é estruturado em três estágios independentes, executados sequencialmente sobre o conjunto de frames extraídos antes de qualquer etapa computacionalmente intensiva da pipeline.

Estágio 1 : Detecção de nitidez por variância do Laplaciano. Cada frame é convertido para escala de cinza e submetido ao operador Laplaciano, cuja variância constitui uma métrica de nitidez amplamente utilizada na literatura para detecção de desfoque [19]. Frames com variância abaixo do limiar configurável $\sigma_{\min}$ são descartados. Esta etapa opera localmente, sem custo de chamadas à API, sendo especialmente adequada para frames extraídos de vídeo: o codec H.264 e a estabilização eletrônica de imagem (EIS) presentes no dispositivo de captura suavizam a imagem de forma uniforme, reduzindo a variância global em relação a fotografias individuais. O limiar foi calibrado empiricamente a partir da distribuição da variância do Laplaciano nos 260 frames extraídos no dataset de calibração. Os valores observados variam de 14,8 a 73,3, com mediana de 40,8 e percentil de 5% em 28,6. O valor $\sigma_{\min} = 25{,}0$ situa-se abaixo desse percentil e descarta apenas os 8 frames da cauda inferior da distribuição, nos quais o borramento é visualmente perceptível, sem remover frames nítidos. Por depender da distribuição de variâncias de cada conjunto de imagens, o limiar é recalibrado quando o enquadramento muda, como ocorreu no dataset de validação, em que foi reduzido para 12,0 (Seção 4.8). A Figura 6 mostra um exemplo real de frame aceito e de frame descartado nesse estágio, extraído do dataset de validação e avaliado com o limiar recalibrado de 12,0.

Estágio 2 : Verificação semântica contextual via AWS Rekognition. Os frames aprovados no estágio anterior são enviados à API *DetectLabels* do AWS Rekognition [18] por meio da biblioteca *boto3* (SDK Python oficial da AWS). O serviço retorna uma lista de rótulos semânticos identificados na imagem, cada um acompanhado de um índice de confiança entre 0 e 100. Um frame é aceito se ao menos um dos rótulos configurados como alvo (`allow_labels`) for retornado com confiança igual ou superior ao limiar τ. Enquanto o primeiro estágio avalia apenas a qualidade técnica da imagem, este estágio fornece um critério semântico adicional para o cenário controlado do experimento, sinalizando frames compatíveis com o conteúdo de interesse no enquadramento.

Calibração dos rótulos e do limiar. A lista de rótulos-alvo e o limiar de confiança foram definidos empiricamente antes dos experimentos. Para essa calibração, foram selecionados três frames representativos: um com o objeto inteiro no enquadramento, um com o objeto parcialmente visível e um sem o objeto. Os três frames foram submetidos individualmente à operação *DetectLabels*, e os rótulos retornados, juntamente com suas respectivas confianças, foram comparados entre as três condições.

A seleção combinou dois critérios. Primeiro, foram eliminados os rótulos que apareciam exclusivamente no frame sem o objeto, por serem associados ao ambiente e não constituírem indícios de sua presença. Em seguida, os rótulos restantes foram avaliados manualmente quanto à coerência semântica com o objeto: somente termos que descreviam de maneira conceitualmente plausível sua aparência, seu material ou sua categoria foram mantidos. Assim, um rótulo semanticamente alheio ao objeto, como *Cloud*, seria

Figura 6: Exemplo real do Estágio 1 (dataset de validação, $\sigma_{\min} = 12{,}0$): frame aceito à esquerda (variância do Laplaciano $= 20,7$) e frame descartado à direita (variância $= 11,7$), ambos da mesma sessão e próximos na sequência de captura.

Fonte: Frame extraído de vídeo produzido pelo autor (2026).

descartado mesmo que satisfizesse numericamente o limiar de confiança. Essa etapa foi necessária porque a confiança, isoladamente, indica a compatibilidade atribuída pelo modelo ao rótulo, mas não garante que ele seja pertinente à finalidade específica da filtragem.

As confianças observadas nos frames com o objeto inteiro e parcialmente visível foram então comparadas com as respostas obtidas no frame sem o objeto. Essa comparação orientou a escolha de um corte operacional que preservasse a identificação nas duas condições de presença e rejeitasse respostas insuficientes na condição de ausência. Como resultado, para os experimentos de calibração foram adotados os rótulos-alvo `["Art", "Modern Art", "Box", "Cardboard", "Toy", "Paper"]` e o limiar $\tau = 50\%$. Esse procedimento configura uma verificação semântica contextual, e não o treinamento de um classificador específico para a caixa. Sua validade está restrita ao objeto e ao cenário controlado avaliados. A lista e o limiar foram posteriormente reajustados para o dataset de validação (Seção 4.8). O acesso à API é configurado via credenciais IAM com a política `AmazonRekognitionReadOnlyAccess`, seguindo o princípio do menor privilégio. A Figura 7 apresenta a saída real da API para um frame aceito.

Estágio 3 : Eliminação de duplicidade por *perceptual hash*. Cada frame

Figura 7: Saída real da API AWS Rekognition *DetectLabels* para um frame aceito (sessão 1, frame 002). O rótulo *Art* (91,6%) foi detectado acima do limiar $\tau = 50\%$ e pertence à lista alvo, resultando na decisão **ACEITAR** exibida na barra inferior.

Fonte: Frame extraído de vídeo produzido pelo autor (2026).

aprovado nos estágios anteriores é comparado por *perceptual hash* (pHash, Seção 2.2) com os frames já aceitos, e descartado se a similaridade s atingir ou superar o limiar $\rho_{\max}$. Esta etapa elimina frames redundantes decorrentes de movimentos orbitais lentos, nos quais frames consecutivos apresentam variação angular mínima e contribuição marginal para a reconstrução. O limiar adotado foi $\rho_{\max} = 90\%$. A Figura 8 mostra um par de frames consecutivos em que o segundo é descartado por redundância.

3.3.3 Structure from Motion (SfM)

A etapa de SfM é executada pelo software COLMAP em modo de reconstrução incremental. Nessa modalidade, a reconstrução começa a partir de um par de imagens com elevado número de correspondências e cresce incrementalmente à medida que novas imagens são registradas. A cada nova imagem incorporada, um ajuste fino (*bundle adjustment*) local é realizado para refinar as estimativas de posição da câmera e dos pontos 3D. Ao final, um ajuste fino global consolida a reconstrução completa.

A saída da etapa SfM é uma nuvem de pontos esparsa com dezenas de milhares de pontos, acompanhada das matrizes de câmera (intrínsecas e extrínsecas) estimadas para

Figura 8: Exemplo real do Estágio 3: par de frames consecutivos da mesma sessão com similaridade pHash de 90,6%, acima do limiar $\rho_{\max} = 90\%$. O segundo frame é descartado por redundância angular mínima em relação ao primeiro.

Fonte: Frame extraído de vídeo produzido pelo autor (2026).

cada imagem do conjunto filtrado.

3.3.4 Multi-View Stereo (MVS)

A etapa de Multi-View Stereo recebe como entrada as imagens originais e as posições da câmera estimadas pela etapa SfM, e produz uma nuvem de pontos densa, com maior fidelidade geométrica à superfície real do objeto. O MVS estima a profundidade de cada pixel em cada imagem por meio de análise de consistência fotométrica entre múltiplas vistas, gerando mapas de profundidade que são posteriormente fundidos em uma nuvem de pontos unificada e colorida.

A implementação utilizada neste trabalho é o *Dense Reconstruction* do COL-

MAP, baseado no algoritmo *PatchMatch Stereo*. As estimativas de profundidade e normais são propagadas espacialmente dentro de cada imagem e verificadas por consistência geométrica entre vizinhanças. A etapa de fusão (*depth map fusion*) aplica critérios de consistência entre as imagens para eliminar estimativas falsas e produzir uma nuvem de pontos final com alta densidade e baixo nível de ruído. A Figura 9 mostra a nuvem de pontos densa gerada pela etapa para a sessão 1.

Os parâmetros do MVS foram definidos a partir da varredura de parâmetros em duas rodadas descrita na Seção 4.4. Em relação aos padrões do COLMAP, dois critérios de filtragem dos mapas de profundidade foram tornados mais restritivos: correlação NCC mínima de 0,15 (padrão 0,1) e número mínimo de vistas consistentes igual a 3 (padrão 2). A motivação, observada na primeira rodada da varredura, é rejeitar pontos do fundo da cena: superfícies pouco texturizadas, como o piso, produzem correspondências de baixa correlação, confirmadas por poucas vistas, e o aperto desses critérios as descarta preferencialmente, preservando o objeto de interesse, visível em todas as imagens da órbita. Uma variante mais agressiva (NCC mínima de 0,2) foi avaliada na mesma rodada e descartada na inspeção visual comparativa das nuvens resultantes. Pela mesma razão, a etapa de fusão adota critérios estritos (mínimo de 12 pixels consistentes e erro máximo de normal de 3,0°), configuração vencedora da segunda rodada da varredura. O limiar de erro de reprojeção da verificação de consistência geométrica (Seção 2.4) foi mantido no valor padrão do COLMAP, de 3,0 pixels.

A etapa conta ainda com um mecanismo de contingência para a fusão. Quando o `stereo_fusion` não produz nenhum vértice, o que ocorre quando o SfM registra poucas imagens, a nuvem seria descartada e o processamento já realizado, perdido. Nesse caso, a pipeline retroprojeta diretamente para o espaço 3D os mapas de profundidade fotométricos já gerados pelo *PatchMatch Stereo*, usando as posições de câmera estimadas pelo SfM, e produz uma nuvem de pontos sem a verificação de consistência entre vistas. O objetivo é recuperar ao menos parte do trabalho computacional de uma execução que seria descartada. Na prática, a tentativa não trouxe o resultado esperado. As nuvens recuperadas por esse mecanismo apresentaram poucos pontos e não preservaram as características do objeto, servindo apenas como registro do comportamento da pipeline nos testes em que a fusão falhou (Seção 4.4).

3.3.5 Alinhamento Global e ICP

Em cenários onde múltiplas capturas do mesmo objeto são realizadas a partir de sessões ou perspectivas distintas, torna-se necessário alinhar as nuvens de pontos parciais em um único sistema de coordenadas antes da reconstrução de superfície. A pipeline implementada usa uma estratégia em dois estágios, ambos com módulos independentes e configuráveis: alinhamento grosseiro por *Global Registration* seguido de refinamento por Iterative Closest Point (ICP) [20]. Neste trabalho, a nuvem densa da sessão 2 é usada

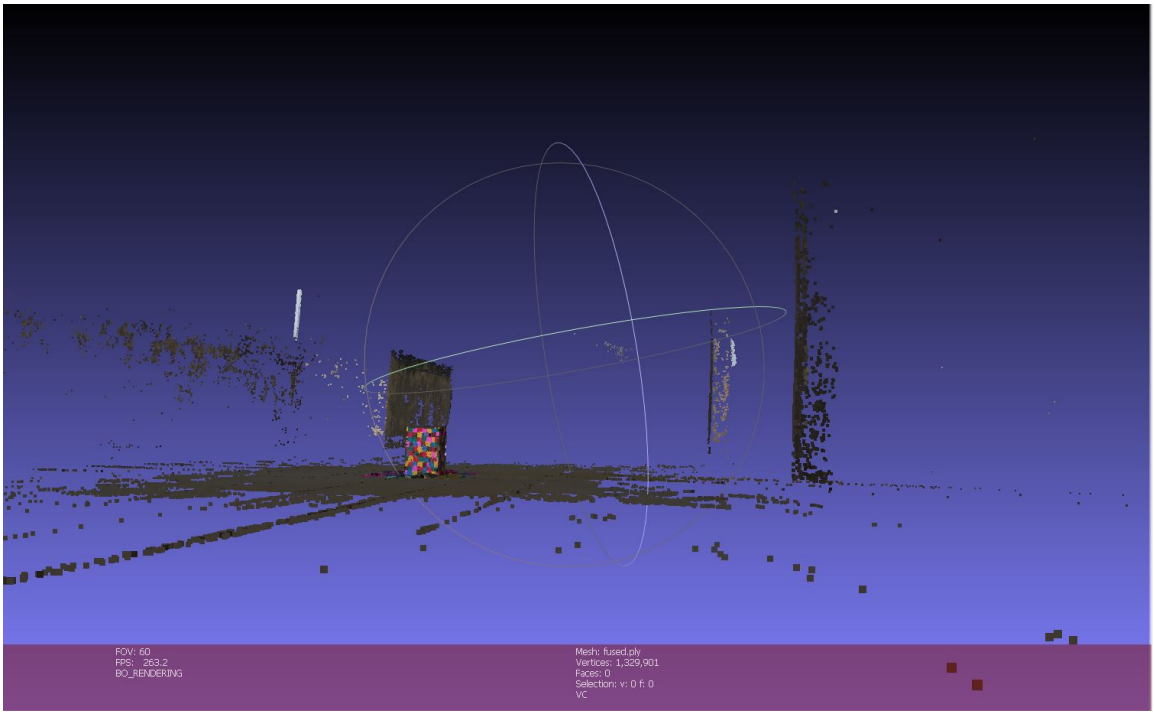


Figura 9: Nuvem de pontos densa gerada pela etapa MVS (COLMAP *PatchMatch Stereo*) a partir do conjunto de imagens filtradas, sessão 1.

Fonte: Captura de tela do MeshLab realizada pelo autor (2026).

como nuvem fonte e a da sessão 1 como nuvem de referência (alvo). A transformação rígida estimada leva a reconstrução da sessão 2 para o sistema de coordenadas da sessão 1, que permanece fixa.

A avaliação inicial do módulo de alinhamento usando somente ICP, sem inicialização, resultou em *fitness* nulo (Seção 4.5), as reconstruções do COLMAP para cada sessão partem de sistemas de coordenadas arbitrários e não guardam relação de escala ou orientação entre si, de modo que a transformação identidade usada como ponto de partida do ICP estava, em geral, longe da transformação correta. A tentativa original de usar o Super 4PCS [10] para o alinhamento grosseiro esbarrou na dificuldade de compreender e integrar a implementação de referência do algoritmo ao código da pipeline.

Como alternativa que não depende de compilação externa, implementou-se um módulo de *Global Registration* nativo do Open3D, baseado em RANSAC sobre descritores locais FPFH (*Fast Point Feature Histograms*), cada nuvem é subamostrada, tem normais e descritores FPFH calculados, e o RANSAC busca a transformação rígida que maximiza o número de correspondências geometricamente consistentes entre os dois conjuntos de descritores, sem exigir nenhuma inicialização. A transformação resultante é usada como inicialização do ICP (*point-to-plane*), que refina o alinhamento. Ambos os estágios têm seus limiares derivados da diagonal do *bounding box* de cada par de nuvens, em vez de valores fixos, compensando a escala arbitrária das reconstruções do COLMAP. O módulo é opcional e configurável, pode ser desativado quando o ICP sozinho já converge.

O ICP é executado na variante *point-to-plane* do Open3D, com no máximo 50 iterações. Como nenhum limiar é definido em valor absoluto, as distâncias derivam da diagonal do *bounding box* de cada par de nuvens. As nuvens são subamostradas com voxel equivalente a 0,3% dessa diagonal. A distância máxima de correspondência do ICP é de três vezes o voxel (cerca de 0,9% da diagonal), e o limiar de aceitação de correspondências do RANSAC no *Global Registration* é de 1,5 vez o voxel. Os raios de busca para estimativa de normais e de descritores FPFH correspondem a duas e cinco vezes o voxel, respectivamente, e o RANSAC executa até 100.000 iterações com semente aleatória fixa (`random_seed: 0`) para reduzir a variabilidade dessa etapa entre execuções. Essa medida não elimina variações numéricas oriundas de etapas anteriores.

3.3.6 Reconstrução de Superfície com *Screened Poisson*

A etapa final da pipeline converte a nuvem de pontos densa e alinhada em uma malha poligonal contínua (*mesh*), adequada para visualização, medida e exportação para ferramentas de modelagem 3D. Para essa finalidade, utiliza-se a técnica de *Screened Poisson Surface Reconstruction* [7], que formula o problema como a resolução de uma equação de Poisson sobre uma grade octree adaptativa.

A entrada do método é composta pelas posições dos pontos e pelas normais associadas a cada ponto, que na pipeline implementada têm origem na própria fusão do MVS (Seção 3.3.4). O parâmetro de profundidade da octree (*depth*) controla o nível de detalhe da malha resultante. Valores maiores produzem malhas mais detalhadas, porém com maior custo computacional e maior sensibilidade ao ruído.

O resultado é uma malha triangulada fechada (PLY), visualizada no MeshLab para inspeção geométrica.

3.4 Ambiente de Desenvolvimento e Ferramentas

As ferramentas utilizadas para o desenvolvimento da pipeline são descritas a seguir, e a Tabela 2 resume as especificações do ambiente computacional em que os experimentos foram executados:

- **AWS Rekognition**: serviço de visão computacional em nuvem da Amazon Web Services, acessado via SDK Python (*boto3*). Utilizado no Estágio 2 do módulo de filtragem (Seção 3.3.2), fornecendo rótulos semânticos empregados como critério contextual para o objeto de interesse em cada frame por meio da API *DetectLabels*.

- **COLMAP**: software de código aberto para reconstrução 3D baseada em SfM e MVS. Utilizado para geração de nuvem de pontos esparsa e densa.

- **Open3D**: biblioteca Python de código aberto para processamento de geometria 3D, incluindo nuvens de pontos e malhas. Utilizada na etapa de alinhamento (*Global*

Registration por RANSAC+FPFH e ICP) e na reconstrução de superfície (*Screened Poisson*). A integração do Super 4PCS foi tentada, mas não concluída pela dificuldade de compreender e integrar sua implementação de referência ao código da pipeline, o *Global Registration* nativo do Open3D foi adotado como alternativa (Seção 3.3.5).

- **MeshLab**: software de código aberto para visualização, inspeção e processamento de malhas e nuvens de pontos 3D. Utilizado para análise qualitativa dos modelos gerados em cada etapa da pipeline.
- **Python 3.x**: linguagem principal utilizada para orquestração da pipeline e integração das ferramentas.

Tabela 2: Especificações do ambiente computacional utilizado nos experimentos.

Componente	**Especificação**
Processador	AMD Ryzen 5 5600, 6 núcleos / 12 threads, 3,5 GHz
Memória RAM	16 GB DDR4 2400 MHz (2 × 8 GB)
Placa de vídeo	NVIDIA GeForce RTX 5060, 8 GB GDDR7
Armazenamento	SSD NVMe 1 TB (Reletech P600, M.2 PCIe 3.0)
Sistema operacional	Windows 10 Pro (64 bits)
Versão Python	3.10.11
Versão COLMAP	4.1.0 (com suporte CUDA)
Versão Open3D	0.19.0

Fonte: Elaborado pelo autor com base nas especificações do ambiente computacional utilizado nos experimentos (2026).

4 Resultados

Os experimentos foram realizados em duas etapas. Na primeira, anterior à integração do módulo de filtragem, foi executada uma varredura de 24 configurações em duas rodadas: a Rodada 1 variou parâmetros do MVS e a Rodada 2, configurações do SfM. Na segunda, após a integração da pipeline (filtragem, SfM, MVS, ICP e SPSR), foram executados os testes *baseline* (sem AWS) e *com filtragem AWS*, para isolar o efeito da filtragem semântica.

Todos os experimentos utilizam a mesma cena. As capturas foram realizadas em ambiente externo, sobre piso de cerâmica clara com rejuntes escuros, com parede de reboco irregular ao fundo e iluminação natural, sem controle de luz artificial. O objeto de interesse, a caixa de papelão revestida com padrão xadrez multicolorido (Figura 5), foi posicionado sobre recortes de papel colorido dispostos no piso. A câmera do smartphone

não foi calibrada previamente. Os parâmetros intrínsecos são estimados pelo próprio SfM durante a reconstrução, utilizando o modelo de câmera SIMPLE_RADIAL do COLMAP, refinado no ajuste global.

4.1 Eficiência da Filtragem de Imagens

A Tabela 3 apresenta os resultados detalhados do módulo de filtragem para cada sessão de captura nos dois cenários avaliados.

Tabela 3: Resultado do módulo de filtragem por estágio e sessão.

Estágio	**1m-Baseline**		**1m-AWS**	
	Sessão 1	**Sessão 2**	**Sessão 1**	**Sessão 2**
Frames de entrada	130	130	130	130
Descartados por desfoque (Laplaciano)	2	6	2	6
Descartados por ausência do objeto (AWS)	—	—	**5**	0
Descartados por redundância (pHash $\geq 90\%$)	18	47	17	47
Frames aceitos	**110**	**77**	**106**	**77**
Taxa de descarte	15,4%	40,8%	18,5%	40,8%

Fonte: Elaborado pelo autor com base nos resultados experimentais (2026).

O estágio de eliminação de duplicidade por *perceptual hash* foi o principal responsável pela redução do conjunto de entrada, descartando 18 e 47 frames nas sessões 1 e 2, respectivamente, na configuração sem AWS. A elevada taxa de descarte na sessão 2 (40,8%) é explicada pela maior sobreposição angular entre frames consecutivos nessa sessão, que tem duração de 35,6 s (menor que a sessão 1 de 54,3 s) com o mesmo número de frames extraídos, resultando em menor variação angular por frame.

A contribuição específica do AWS Rekognition foi a remoção de 5 frames adicionais na sessão 1, correspondendo a 3,9% do conjunto aceito pelo Laplaciano. Nesses frames, nenhum dos rótulos alvo foi retornado pela API no limiar de 50% configurado para o experimento. Na sessão 2, o AWS não descartou nenhum frame além dos já eliminados pelos estágios locais. Esses resultados descrevem o comportamento do critério semântico calibrado para esta cena controlada, sem constituir uma validação de classificação específica do objeto. A Figura 10 compara um frame aceito e um frame descartado por esse estágio, em exemplo real do dataset de validação.

4.2 Qualidade da Reconstrução SfM

A principal métrica de qualidade do SfM é o erro de reprojeção médio, sobre o conjunto de observações, da distância entre a projeção estimada de cada ponto 3D e sua observação 2D correspondente. Como referência de comparação, o próprio COLMAP [1]

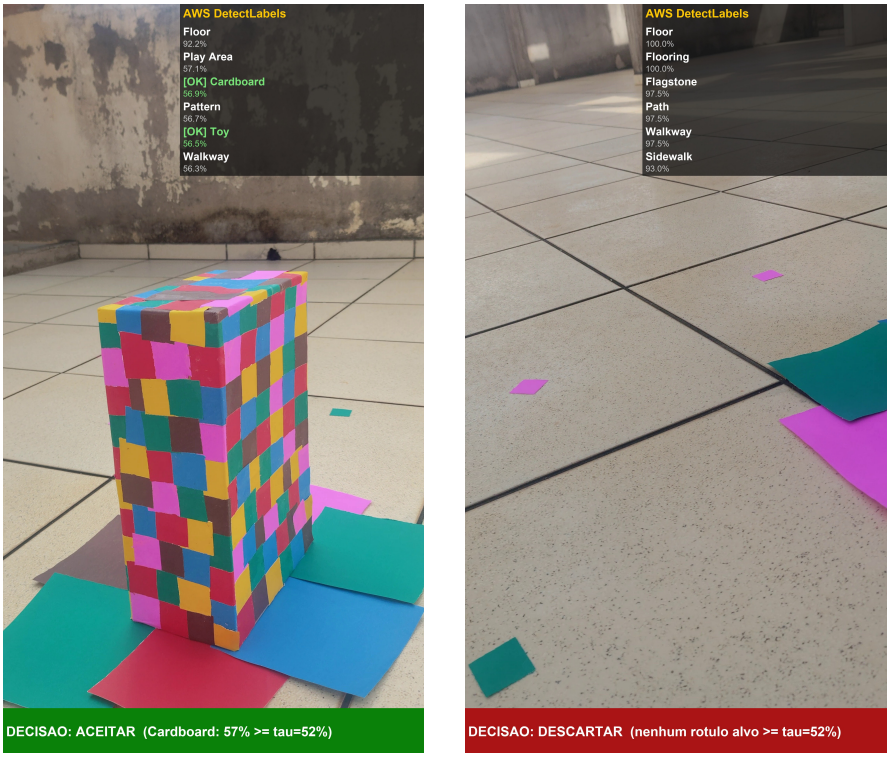


Figura 10: Comparação entre frame aceito (esquerda) e frame descartado (direita) pelo estágio 2 da filtragem, exemplo real do dataset de validação (AWS Rekognition *DetectLabels*, $\tau = 52\%$). No frame aceito, os rótulos *Cardboard* (56,9%) e *Toy* (56,5%) foram detectados acima do limiar e pertencem à lista alvo. No frame descartado, o objeto está fora do enquadramento e nenhum rótulo alvo atinge o limiar (apenas rótulos de piso/ambiente, como *Floor* e *Flagstone*), resultando no descarte antes da etapa SfM.

Fonte: Frame extraído de vídeo produzido pelo autor (2026).

Tabela 4: Resultados da etapa SfM nos cenários 1m-Baseline e 1m-AWS.

Sessão / Teste	**Entrada**	**Registradas**	**Taxa**	**Pontos 3D**	**Erro repr. (px)**
Sessão 1 — Baseline	110	110	100%	162.295	1,081
Sessão 1 — Com AWS	106	106	100%	156.015	1,073
Sessão 2 — Baseline	77	77	100%	107.856	1,040
Sessão 2 — Com AWS	77	77	100%	107.794	1,045

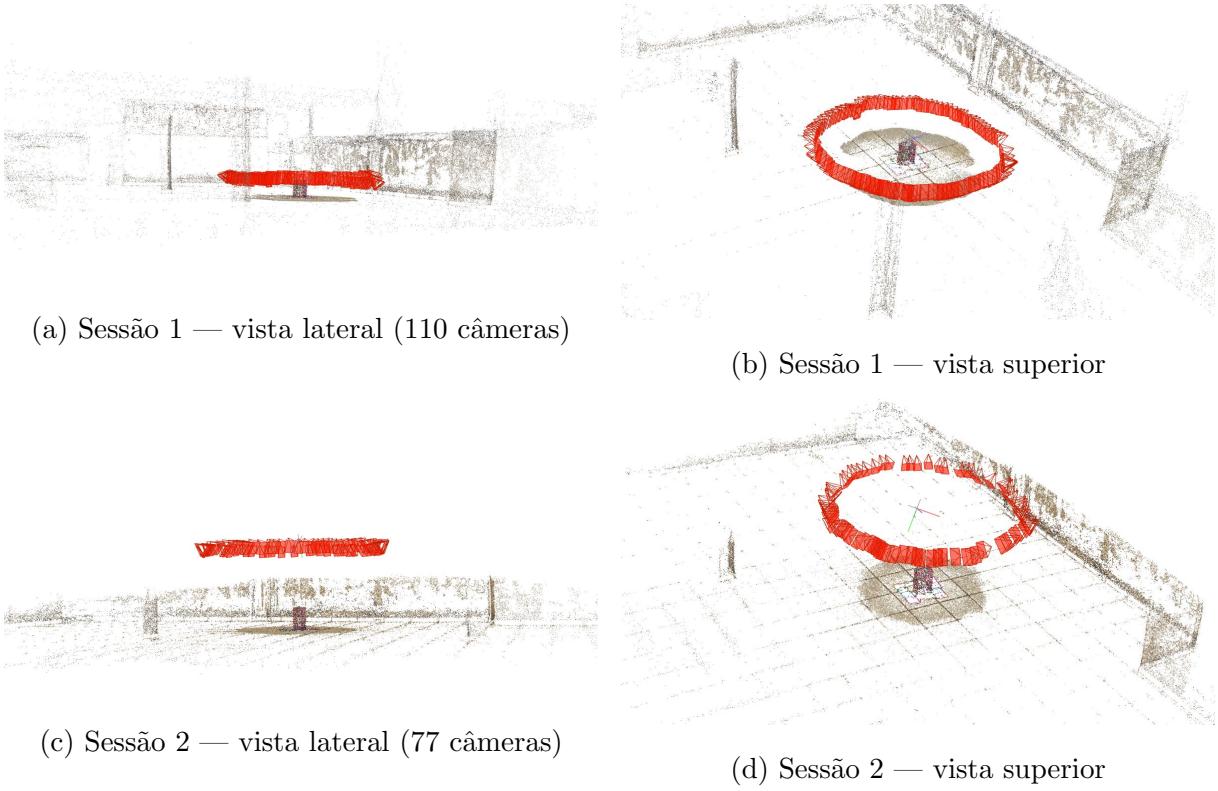
Fonte: Elaborado pelo autor com base nos resultados experimentais (2026).

reporta erro de reprojeção médio entre 0,60 px e 0,81 px em 15 conjuntos reais de fotos não ordenadas da internet.

A Tabela 4 apresenta os resultados da etapa SfM para os dois cenários avaliados.

Em ambos os cenários, o COLMAP registrou 100% das imagens de entrada, indicando que o conjunto filtrado manteve cobertura angular suficiente para o SfM incremental. Na execução reportada, a filtragem AWS reduziu em 3,7% o número de pontos 3D da sessão 1 (de 162.295 para 156.015). O erro de reprojeção médio ficou próximo

nos dois cenários e sessões, sem alteração numérica relevante associada à remoção dos 5 frames adicionais pelo AWS. Como há uma execução por cenário, essa comparação deve ser interpretada como descritiva. A Figura 11 mostra as nuvens esparsas e as trajetórias orbitais de câmera estimadas para as duas sessões.



(a) Sessão 1 — vista lateral (110 câmeras)

(b) Sessão 1 — vista superior

(c) Sessão 2 — vista lateral (77 câmeras)

(d) Sessão 2 — vista superior

Figura 11: Nuvem de pontos esparsa e trajetória orbital de câmeras estimadas pelo SfM (COLMAP) para as sessões 1 e 2, cenário 1m-AWS (com filtragem AWS ativa).

Fonte: Captura de tela do COLMAP realizada pelo autor (2026).

4.3 Resultados da Etapa MVS

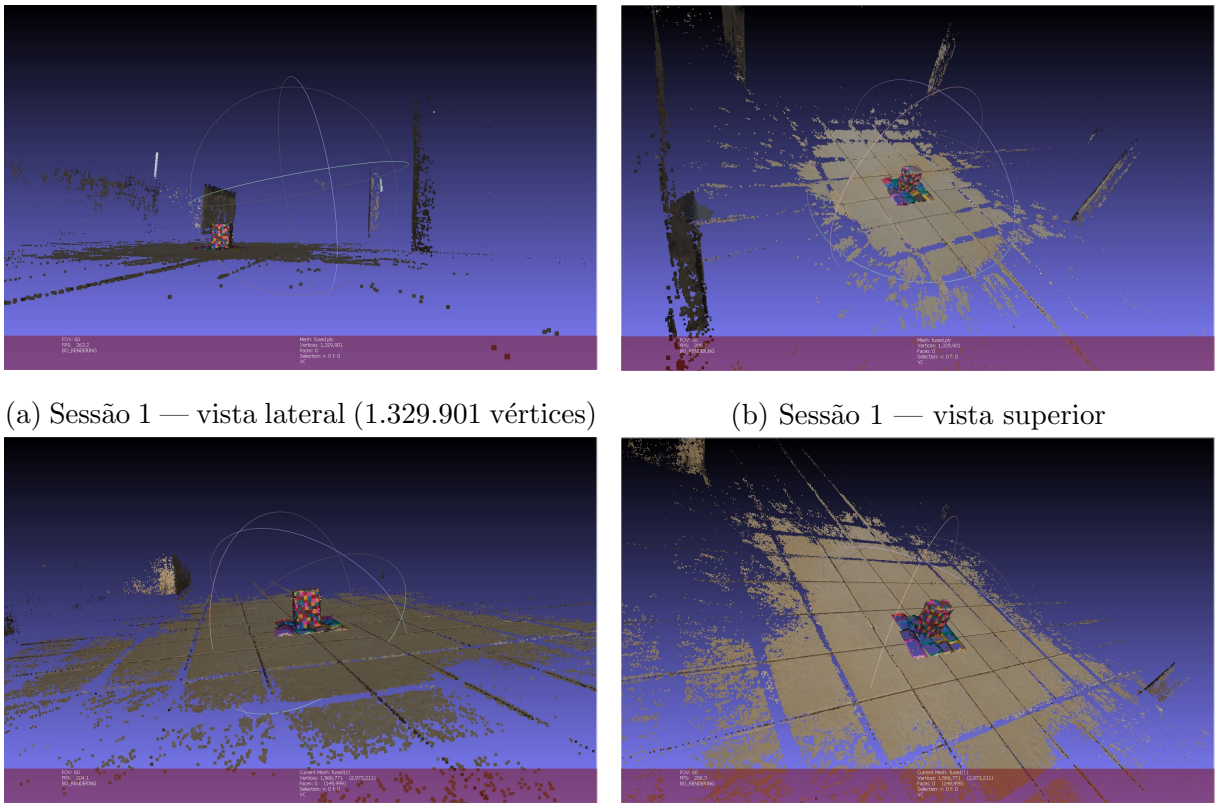
A Tabela 5 apresenta os resultados da etapa MVS, medidos pelo número de vértices da nuvem de pontos densa gerada pelo algoritmo *PatchMatch Stereo* com fusão geométrica (*stereo_fusion*) do COLMAP.

Nas execuções reportadas, a filtragem AWS produziu um incremento de 1,0% nos vértices fundidos da sessão 1 (1.329.901 vs. 1.316.237) e uma redução de 0,1% na sessão 2. As durações da etapa foram próximas entre os cenários, e o tempo total da pipeline diferiu em um minuto (Tabela 11). Esses valores sugerem que as chamadas à API *DetectLabels*, realizadas para 128 imagens na sessão 1 e 124 imagens na sessão 2, não dominaram o tempo total de processamento nas execuções observadas. Entretanto, sem repetições e sem medição isolada do tempo das chamadas à API, não é possível quantificar sua sobrecarga. A Figura 12 apresenta as nuvens de pontos densas geradas para as duas sessões.

Tabela 5: Resultados da etapa MVS nos cenários 1m-Baseline e 1m-AWS, com o número de vértices da nuvem densa e a duração da etapa para cada sessão.

Sessão / Teste	**Vértices fundidos**	**Duração MVS**
Sessão 1 — Baseline	1.316.237	≈ 2 h 17 min
Sessão 2 — Baseline	1.569.158	≈ 1 h 37 min
Sessão 1 — Com AWS	1.329.901	≈ 2 h 12 min
Sessão 2 — Com AWS	1.566.771	≈ 1 h 37 min

Fonte: Elaborado pelo autor com base nos resultados experimentais (2026).



(a) Sessão 1 — vista lateral (1.329.901 vértices)

(b) Sessão 1 — vista superior

(c) Sessão 2 — vista lateral (1.566.771 vértices)

(d) Sessão 2 — vista superior

Figura 12: Nuvens de pontos densas geradas pelo MVS (COLMAP *PatchMatch Stereo*) para as sessões 1 e 2, com filtragem AWS.

Fonte: Captura de tela do MeshLab realizada pelo autor (2026).

4.4 Análise de Parâmetros da Pipeline

As duas rodadas de varredura de parâmetros, executadas anteriormente à integração do módulo de filtragem, forneceram dados sobre o impacto de cada parâmetro na qualidade da reconstrução e no tempo de processamento. A Tabela 6 resume os principais resultados da Rodada 1 (variação de parâmetros MVS) e a Tabela 7 da Rodada 2 (variação de parâmetros SfM).

Tabela 6: Configurações selecionadas da varredura de parâmetros do MVS (Rodada 1), com o número de vértices das nuvens densas das sessões 1 (S1) e 2 (S2) e a duração total de cada teste. Em todos os testes válidos, as 130 imagens de cada sessão foram registradas pelo SfM. Valores marcados com * são produtos de contingência e não saídas MVS diretamente comparáveis.

Configuração	**S1 vértices**	**S2 vértices**	**Duração**	**Obs.**
Baseline (máx. qualidade)	8.368*	2.698.917	3 h 10 min	SfM S1 falhou
num_iterations = 5	1.515.290	2.557.537	4 h 21 min	
num_iterations = 3	1.582.961	2.466.571	2 h 53 min	
num_samples = 15	1.670.804	2.706.997	4 h 44 min	
geom_consistency = false	19.922	17.683	3 h 11 min	Qualidade inaceitável
max_image_size = 1500	1.316.532	1.865.623	4 h 09 min	
max_image_size = 1000	821.264	916.141	2 h 47 min	
Ultra performance (mín.)	12.203	13.198	18 min	Qualidade inaceitável
Ultra qualidade (máx.)	3.789.852	5.309.476	25 h 36 min	

* Produto de contingência obtido por retroprojeção de mapas de profundidade. Não corresponde a uma saída MVS concluída.

Fonte: Elaborado pelo autor com base nos resultados experimentais (2026).

O resultado mais significativo da Rodada 1 forneceu evidência empírica, nas configurações avaliadas, de que o parâmetro `geom_consistency` é crítico para esta pipeline: sua desativação reduziu a nuvem densa de ∼1,5 milhão para menos de 20.000 vértices por sessão, tornando o resultado inutilizável. Esse parâmetro habilita uma segunda passagem do PatchMatch Stereo com consistência geométrica entre vistas, que elimina estimativas falsas de profundidade. Todos os testes subsequentes mantiveram `geom_consistency = true`.

Tabela 7: Resultados selecionados da varredura de parâmetros do SfM (Rodada 2). Sessão 2 registrou 130 imagens em todos os testes válidos. Valores marcados com * são produtos de contingência e não saídas MVS diretamente comparáveis.

Configuração	**S1 vértices**	**S2 vértices**	**Duração**	**Obs.**
Baseline R2 (SIMPLE_RADIAL, *exhaustive*)	1.619.733	2.701.146	5 h 51 min	
Modelo RADIAL	8.108*	3.321.439	3 h 09 min	SfM S1 falhou
Modelo OPENCV	743*	3.314.604	3 h 17 min	SfM S1 falhou
Sequential matcher, *overlap*=10	1.621.456	2.706.220	5 h 39 min	
Sequential matcher, *overlap*=20	1.626.094	2.710.279	5 h 46 min	
Max matches = 32.768	1.621.886	2.711.657	8 h 04 min	
RADIAL + *sequential*, *overlap*=20	2.001.014	3.289.608	5 h 39 min	Maior S1

* Produto de contingência obtido por retroprojeção. Não corresponde a uma saída MVS concluída.

Fonte: Elaborado pelo autor com base nos resultados experimentais (2026).

Na Rodada 2, os modelos de câmera RADIAL e OPENCV não registraram adequadamente a sessão 1 com correspondência exaustiva (*exhaustive matching*), estratégia em que cada imagem é comparada com todas as demais do conjunto. Uma explicação possível é que a maior quantidade de parâmetros a otimizar no ajuste fino torne a convergência mais sensível à qualidade inicial das correspondências. A correspondência sequencial (*sequential matching*), que compara apenas imagens próximas na ordem de captura dentro

de uma janela temporal e por isso é adequada a capturas em vídeo, produziu resultados utilizáveis com menor duração de correspondência nas configurações avaliadas. A combinação RADIAL + sequencial com janela de 20 imagens produziu a maior nuvem densa da sessão 1 (2.001.014 vértices) nesta rodada exploratória. Essa configuração não foi adotada nos testes comparativos principais, que mantiveram SIMPLE_RADIAL com correspondência exaustiva para isolar exclusivamente o efeito da filtragem AWS.

4.5 Alinhamento das Nuvens de Pontos

O alinhamento entre as nuvens de pontos geradas nas duas sessões foi realizado com o algoritmo ICP (*Iterative Closest Point*), utilizando a implementação disponível na biblioteca Open3D [11]. As formulações clássicas propostas por Besl e McKay [31] e por Chen e Medioni [32] (Seção 2.5) descrevem o processo iterativo de minimização do erro entre pares de pontos correspondentes. As métricas *fitness* e RMSE utilizadas neste trabalho, entretanto, seguem a definição adotada pelo Open3D [12] e não fazem parte das formulações originais do algoritmo.

O *fitness* representa a fração dos pontos da nuvem **fonte** que encontra uma correspondência válida na nuvem alvo dentro da distância máxima $d_{\max}$. Essa métrica indica o grau de sobreposição entre as nuvens após o alinhamento e assume valores entre 0 e 1, sendo 1 o caso ideal. Já o RMSE corresponde à raiz do erro quadrático médio das distâncias entre os pares correspondentes, expresso nas unidades da própria reconstrução produzida pelo COLMAP.

Vale destacar que esse RMSE difere daquele normalmente empregado em *benchmarks* sintéticos, nos quais existe uma transformação de referência (*ground truth*) conhecida e é possível medir diretamente o erro em relação à solução correta, como em [20]. Neste trabalho, as reconstruções foram obtidas a partir de capturas reais e independentes, sem conhecimento prévio da transformação relativa entre elas. Nessas condições, a qualidade do alinhamento foi avaliada pela consistência entre as próprias nuvens de pontos, utilizando as métricas fornecidas pelo Open3D.

Tabela 8: Métricas de alinhamento nos cenários 1m-Baseline e 1m-AWS, conforme definidas pelo Open3D: *fitness* do ICP executado sem inicialização, *fitness* do *Global Registration* isolado, *fitness* do ICP inicializado pelo alinhamento global e RMSE final das correspondências.

Teste	**ICP sem init.**	***Global Reg.* (fitness)**	**ICP com init. global**	**RMSE final**
1m-Baseline	0,0000	0,6211	**0,7730**	0,127308
1m-AWS	0,6738	0,6101	**0,7516**	0,122819

Fonte: Elaborado pelo autor com base nos resultados experimentais (2026).

No cenário 1m-Baseline, o ICP executado sem uma transformação inicial não en-

controu correspondências válidas, resultando em *fitness* igual a zero. Esse resultado era esperado, já que cada reconstrução produzida pelo COLMAP é gerada em um sistema de coordenadas arbitrário. Assim, assumir a transformação identidade como estimativa inicial faz com que as nuvens estejam, em geral, muito distantes da solução correta. Como o ICP é um algoritmo de otimização local, ele não consegue convergir quando parte dessa condição [20].

A inclusão do estágio de *Global Registration*, baseado em RANSAC sobre descritores FPFH, forneceu uma estimativa inicial suficientemente próxima da solução. Após o refinamento pelo ICP, o *fitness* aumentou para 0,7730, com RMSE de 0,127. Embora a melhora seja expressiva do ponto de vista numérico, ela não garante, por si só, um alinhamento geometricamente correto do objeto de interesse, conforme discutido na Seção 4.6.

No cenário 1m-AWS, realizado com a filtragem AWS, o ICP apresentou convergência parcial mesmo sem uma estimativa inicial, alcançando *fitness* de 0,6738. A distância entre os centros das duas reconstruções considerando a transformação identidade permaneceu praticamente igual nos dois experimentos (6,84% da diagonal da cena no 1m-Baseline e 6,87% no 1m-AWS), indicando que a proximidade espacial não explica essa diferença de comportamento.

A principal diferença apareceu na quantidade de correspondências disponíveis antes do início das iterações. No 1m-Baseline, apenas 0,23% dos pontos da nuvem fonte encontravam um vizinho dentro da tolerância $d_{\max}$, enquanto esse valor aumentava para 0,98% no 1m-AWS. Embora ambos os percentuais sejam baixos, essa diferença alterou completamente a evolução do algoritmo.

No 1m-Baseline, o *fitness* chegou a aproximadamente 7,5% nas primeiras iterações, mas caiu para zero a partir da oitava iteração. Já no 1m-AWS, o valor aumentou de 0,98% para cerca de 37% logo na primeira atualização e estabilizou em torno de 67% por volta da quinta iteração.

Esse comportamento sugere um efeito cumulativo. Uma quantidade inicial ligeiramente maior de correspondências, desde que geometricamente consistentes, é suficiente para orientar a primeira atualização da transformação na direção correta. A partir desse ponto, cada nova iteração tende a produzir correspondências melhores, favorecendo a convergência. Uma explicação plausível é que a filtragem AWS tenha preservado principalmente pontos pertencentes ao objeto de interesse, que aparecem nas duas sessões, reduzindo a influência do piso, cuja geometria extensa e repetitiva favorece correspondências incorretas.

Mesmo nesse cenário, o *Global Registration* continuou trazendo ganhos, elevando o *fitness* final para 0,7516. Em nenhum dos experimentos essa etapa prejudicou o resultado obtido pelo ICP. No 1m-Baseline, ela foi essencial para possibilitar a convergência. No 1m-AWS, proporcionou um refinamento adicional. Por esse motivo, o alinhamento global foi incorporado como etapa padrão da pipeline, permanecendo opcional por configuração.

Como o RANSAC utilizado no *Global Registration* possui comportamento estocástico, a semente aleatória foi fixada (`random_seed: 0`) para reduzir a variabilidade dessa etapa entre execuções.

4.6 Limitação do Alinhamento Multi-Sessão: Dominância do Fundo

Apesar da melhora numérica do *fitness* do ICP (Tabela 8), a inspeção visual da malha resultante revelou um problema não capturado pelas métricas de alinhamento: a superfície do piso, presente em ambas as sessões devido ao protocolo de captura a aproximadamente 1 metro de distância (Tabela 1), domina tanto o *Global Registration* quanto o ICP. O piso é uma superfície de grande extensão espacial em comparação ao objeto de interesse, que ocupa uma fração pequena do total de pontos. Seu problema principal não é a planaridade em si, mas a baixa textura: os ladrilhos claros e visualmente repetitivos oferecem poucos pontos distintivos, produzindo descritores ambíguos e correspondências pouco confiáveis. Trata-se da mesma limitação que motivou a ampliação artificial da textura do objeto (Seção 3.3.1). A geometria planar agrava o quadro. Qualquer deslocamento lateral ao longo do plano continua satisfazendo a métrica de erro média, já que a maioria dos pontos (do piso) permanece coincidente independentemente desse deslocamento residual. O resultado é que o piso das duas sessões se sobrepõe corretamente, mas o objeto aparece deslocado e duplicado na malha final (Figura 13), mesmo com *fitness* = 0,7730.

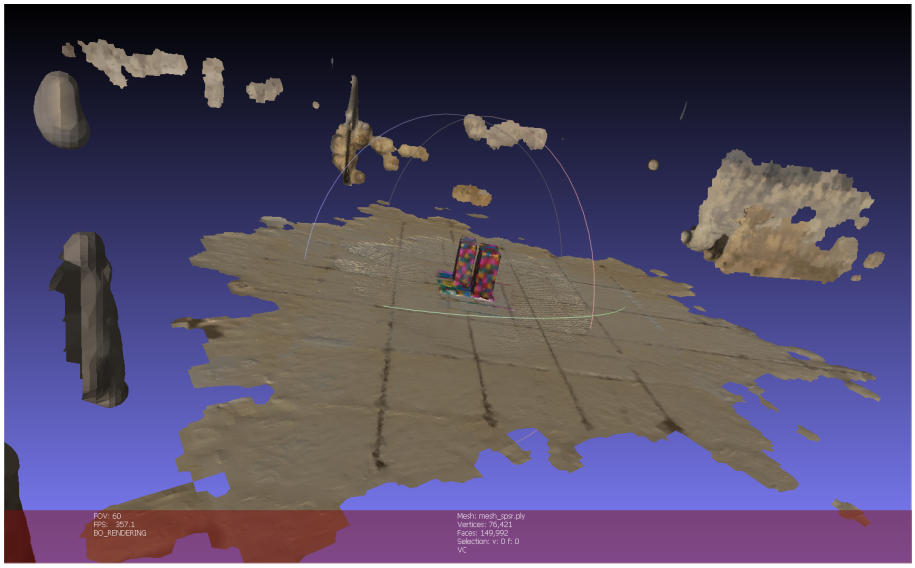


Figura 13: Exemplo de limitação do alinhamento multi-sessão no cenário 1m-AWS: malha SPSR obtida após combinar as nuvens das duas sessões. Embora o alinhamento apresente *fitness* numérico satisfatório, a predominância do piso faz com que o objeto de interesse apareça deslocado e duplicado.

Fonte: Captura de tela do MeshLab realizada pelo autor (2026).

4.6.1 Modularidade como mitigação: reconstrução por sessão

O problema a mitigar é o descrito acima: o objeto de interesse aparece deslocado e duplicado na malha combinada, ainda que as métricas numéricas de alinhamento sejam satisfatórias. A arquitetura modular da pipeline permite tratar a etapa de alinhamento como opcional: quando o registro multi-sessão não produz um resultado geometricamente confiável, cada sessão pode ser reconstruída de forma independente por SPSR, sem depender do ICP. Essa alternativa foi avaliada sobre o mesmo conjunto de dados: a sessão 1 (1.316.237 pontos de entrada) gerou uma malha de 150.000 triângulos, valor que corresponde ao limite de simplificação configurado na pipeline (Seção 4.7), e a sessão 2 (1.569.158 pontos) gerou igualmente 150.000 triângulos. Na inspeção visual, ambas as malhas foram coerentes com suas respectivas sessões de captura e não apresentaram o artefato observado na Figura 13. Embora a reconstrução combinada de duas perspectivas seja o objetivo original da pipeline, a possibilidade de recuar para reconstruções individuais demonstra na prática o valor da arquitetura modular proposta na Seção 3: uma falha isolada em uma etapa não compromete a pipeline inteira, apenas reduz o escopo do resultado final de "modelo combinado" para "modelos por perspectiva".

4.6.2 Causa raiz e direção de correção

O problema tem origem no protocolo de captura, não na implementação do alinhamento: a distância de aproximadamente 1 metro ao objeto (Tabela 1) inclui uma área substancial de piso e ambiente em cada frame, fazendo com que o fundo compartilhado entre as sessões seja espacialmente maior do que o próprio objeto de interesse. A partir desse diagnóstico, o protocolo de captura foi corrigido com a redução da distância para aproximadamente 40 cm, atacando a causa do problema na fonte, em vez de tentar corrigi-lo por pós-processamento. O efeito dessa correção é avaliado experimentalmente na Seção 4.8.

4.7 Reconstrução de Superfície

A reconstrução de superfície por *Screened Poisson Surface Reconstruction* (SPSR) [7] foi aplicada à nuvem de pontos combinada das duas sessões pelo alinhamento global e ICP (Seção 3.3.5). A técnica resolve uma equação de Poisson sobre uma grade octree adaptativa de profundidade $d = 10$, gerando uma malha triangulada fechada. A nuvem combinada foi submetida previamente à remoção de *outliers* estatísticos (limiar de 1,5 desvios-padrão, 30 vizinhos) e à estimativa de normais por PCA local. A Tabela 9 resume os resultados da etapa para os dois cenários.

Tabela 9: Resultados da etapa SPSR, a partir da nuvem combinada das duas sessões.

Teste	Pontos combinados	Após outlier removal	Triângulos gerados	Duração SPSR
1m-Baseline	2.885.395	2.820.538*	149.992	≈ 12 s
1m-AWS	2.896.672	2.833.991**	149.999	≈ 10 s

* 64.857 pontos removidos. ** 62.681 pontos removidos.

Fonte: Elaborado pelo autor com base nos resultados experimentais (2026).

Nos cenários 1m-Baseline e 1m-AWS, o número de triângulos reportado não é resultado espontâneo da reconstrução, mas o limite de simplificação configurado na pipeline (150.000 triângulos), aplicado por decimação quádrica do Open3D após o SPSR. A malha bruta gerada com profundidade $d = 10$ atinge cerca de 1,2 milhão de triângulos. Em avaliação comparativa com limites de 150 mil a 3 milhões de triângulos sobre a mesma nuvem, não se observou ganho perceptível de qualidade visual para o objeto reconstruído, enquanto o tamanho do arquivo cresce de aproximadamente 6 MB para 45 MB. O valor de 150.000 foi portanto adotado como padrão prático do projeto para as malhas de visualização. Nas execuções reportadas, as malhas dos dois cenários apresentaram aspecto visual comparável na inspeção qualitativa. Essa observação é compatível com a proximidade das métricas acompanhadas nas etapas anteriores, mas não constitui demonstração de equivalência geométrica entre as reconstruções. A Figura 14 mostra a malha gerada para o cenário 1m-AWS.

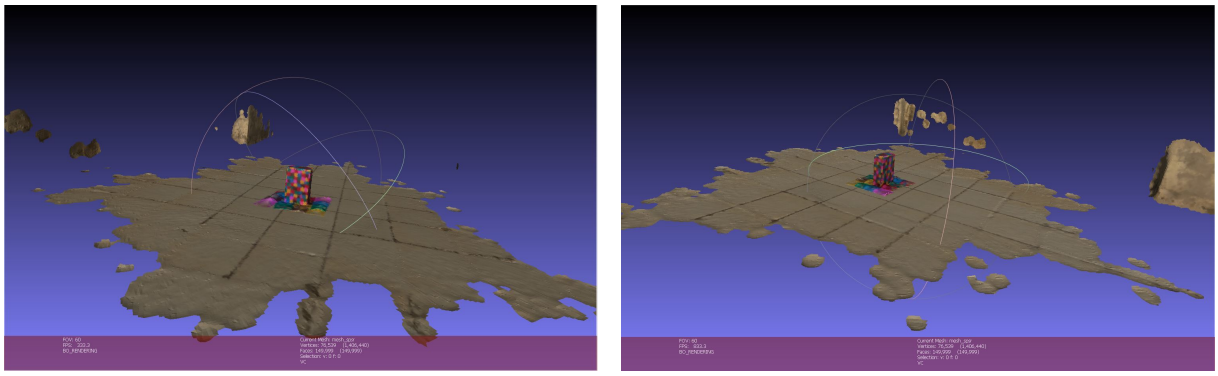


Figura 14: Malha SPSR reconstruída a partir de uma sessão individual no cenário 1m-AWS, apresentada em duas vistas. A reconstrução por sessão evita o artefato de deslocamento e duplicação associado ao alinhamento multi-sessão, mas não substitui a integração geométrica das duas perspectivas.

Fonte: Captura de tela do MeshLab realizada pelo autor (2026).

4.8 Validação com Protocolo de Captura Corrigido

Para verificar se a redução da distância de captura eliminaria a dominância do piso identificada na Seção 4.6, o mesmo objeto (caixa de papelão com padrão xadrez multicolorido) foi recapturado a aproximadamente 40 cm de distância, na mesma cena dos

experimentos anteriores, repetindo os dois cenários comparativos (*baseline* e *com filtragem AWS*) sobre o novo material. As duas novas sessões seguiram o procedimento orbital manual descrito na Seção 3.3.1, com distância, altura e inclinação aparentes semelhantes. Os parâmetros de filtragem foram recalibrados para o novo enquadramento: o limiar de nitidez $\sigma_{\min}$ foi reduzido de 25,0 para 12,0, o limiar de confiança do AWS Rekognition foi ajustado de 50% para 52%, o rótulo `"Paper"` foi removido da lista de rótulos-alvo por gerar falsos positivos sobre recortes de papel soltos no piso, e o número de frames extraídos por sessão foi ampliado de 130 para 200. A redução de $\sigma_{\min}$ foi necessária porque a distribuição da variância do Laplaciano no novo enquadramento é globalmente mais baixa (mediana de 21,7, contra 40,8 no dataset de calibração). Manter o limiar anterior descartaria a maior parte dos frames nítidos. Os vídeos das duas sessões foram regravados (sessão 1 com 51,7 s e 1539 frames totais, sessão 2 com 58,5 s e 1754 frames totais). A Tabela 10 detalha o resultado do módulo de filtragem sobre o novo dataset, e a Tabela 11 compara as métricas dos dois datasets em todas as etapas da pipeline.

Tabela 10: Resultado do módulo de filtragem no dataset de validação (˜40 cm).

Estágio	**40cm-Baseline**		**40cm-AWS**	
	Sessão 1	**Sessão 2**	**Sessão 1**	**Sessão 2**
Frames de entrada	200	200	200	200
Descartados por desfoque (Laplaciano)	1	0	1	0
Descartados por ausência do objeto (AWS)	—	—	12	2
Descartados por redundância (pHash $\geq 90\%$)	17	28	16	28
Frames aceitos	**182**	**172**	**171**	**170**

Fonte: Elaborado pelo autor com base nos resultados experimentais (2026).

Tabela 11: Comparação entre o dataset de calibração e o dataset de validação.

	Calibração (˜1 m)		**Validação (˜40 cm)**	
Métrica	**1m-Baseline**	**1m-AWS**	**40cm-Baseline**	**40cm-AWS**
Vértices fundidos — Sessão 1	1.316.237	1.329.901	575.051	571.509
Vértices fundidos — Sessão 2	1.569.158	1.566.771	514.048	518.497
Erro de reprojeção (px) — Sessão 1	1,081	1,073	1,2599	1,2733
Erro de reprojeção (px) — Sessão 2	1,040	1,045	1,3125	1,3100
Global Registration (fitness)	0,6211	0,6101	0,9247	0,9104
ICP final (fitness)	0,7730	0,7516	**0,9878**	**0,9959**
RMSE final	0,127308	0,122819	0,103948	0,103397
Duração total	4 h 08 min	4 h 07 min	7 h 51 min	7 h 50 min

Fonte: Elaborado pelo autor com base nos resultados experimentais (2026).

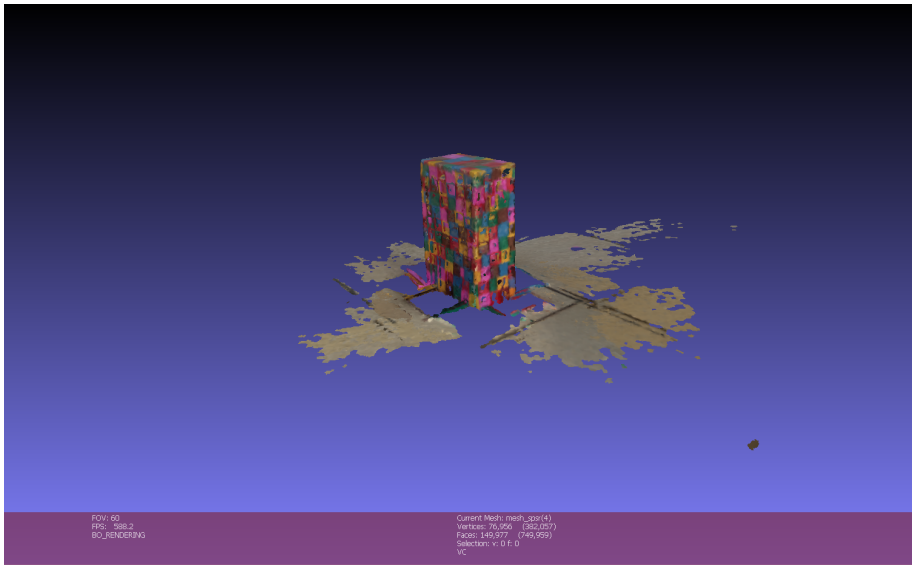


Figura 15: Malha SPSR gerada a partir da nuvem alinhada do dataset de 40cm-AWS.

Fonte: Captura de tela do MeshLab realizada pelo autor (2026).

A inspeção visual das malhas geradas a partir do dataset de validação (MeshLab, Figura 15) não identificou o deslocamento e a duplicação observados na Figura 13. Esse resultado é consistente com a hipótese de que reduzir a área de piso e ambiente no enquadramento dá maior peso relativo aos pontos do objeto nas etapas de *Global Registration* e ICP. Contudo, a comparação entre os datasets não isola exclusivamente a distância de captura: além de 1 m para aproximadamente 40 cm, foram alterados o número de frames extraídos, os limiares de filtragem e a lista de rótulos-alvo. Assim, os resultados apoiam a correção do protocolo como um todo, sem permitir atribuir a melhora somente à redução da distância. Como contrapartida, a duração total quase dobrou em relação ao dataset de calibração, passando de aproximadamente 4 h para aproximadamente 8 h. Esse aumento é compatível principalmente com a ampliação de 130 para 200 frames extraídos por sessão. A comparação entre os dois datasets não permite separar completamente os efeitos das demais alterações do protocolo.

4.9 Comparação com Ausência Total de Filtragem

Os dois cenários comparativos discutidos até aqui (*baseline* e *com filtragem AWS*) mantêm, em ambos os casos, os estágios locais de filtragem ativos, a variável isolada entre eles é apenas a presença ou ausência do estágio semântico via AWS Rekognition. Para isolar o efeito da filtragem como um todo, um terceiro cenário (*sem filtragem*) foi executado sobre o mesmo dataset de validação, com os três estágios desativados. Nesse cenário, todos os 200 frames extraídos por sessão seguem diretamente para o SfM, sem nenhum descarte. A Tabela 12 compara os três cenários.

Tabela 12: Comparação entre os três cenários de filtragem, sobre o mesmo dataset de validação.

Métrica	**40cm-SemFiltro**	**40cm-Baseline**	**40cm-AWS**
Frames aceitos — Sessão 1 / Sessão 2	200 / 200	182 / 172	171 / 170
Erro de reprojeção (px) — Sessão 1 / Sessão 2	1,2593 / 1,3183	1,2599 / 1,3125	1,2733 / 1,3100
Vértices fundidos — Sessão 1 / Sessão 2	635.343 / 646.167	575.051 / 514.048	571.509 / 518.497
Global Registration (fitness)	0,9148	0,9247	0,9104
ICP final (fitness)	0,9957	0,9878	0,9959
RMSE final	0,102688	0,103948	0,103397
Triângulos SPSR (nuvem combinada)	149.973	149.983	149.977
Duração total	**9 h 07 min**	7 h 51 min	7 h 50 min

Fonte: Elaborado pelo autor com base nos resultados experimentais (2026).

Nas execuções realizadas sobre esse dataset, a filtragem não apresentou diferença numérica relevante nas métricas quantitativas acompanhadas: o erro de reprojeção, o *fitness* final do alinhamento e o número de triângulos da malha combinada ficaram próximos entre os três cenários. Como foi realizada uma execução por cenário, esses resultados não permitem afirmar equivalência estatística. A inspeção visual das malhas, de caráter qualitativo, tampouco revelou diferença perceptível entre elas. A diferença prática observada está no tempo total de processamento. Sem filtragem, a pipeline processa 200 frames por sessão e leva 9 h 07 min. Com filtragem, as durações foram de 7 h 51 min e 7 h 50 min: reduções de 1 h 16 min e 1 h 17 min, equivalentes a aproximadamente 13,9% e 14,1% do tempo sem filtragem. De forma equivalente, o cenário sem filtragem levou cerca de 16% a mais que os cenários filtrados. Esse resultado refina a conclusão obtida no dataset de calibração (Seções 4.1 a 4.3): no escopo deste experimento, o benefício observável da filtragem é primariamente de eficiência computacional, por reduzir a quantidade de frames que chega ao SfM/MVS, e não de qualidade geométrica mensurada do modelo final.

5 Conclusão

Este trabalho desenvolveu e avaliou um módulo de filtragem automática de imagens em três estágios, composto por detecção de desfoque, verificação semântica contextual relacionada ao objeto de interesse por meio do AWS Rekognition e eliminação de duplicidade, como etapa de pré-processamento para reconstrução tridimensional a partir de vídeos capturados por dispositivos convencionais. O fluxo de reconstrução em si apoia-se em ferramentas consolidadas e homologadas pela literatura. O COLMAP realiza o SfM e o MVS, e o Open3D realiza o alinhamento e a reconstrução de superfície. Essa escolha permite isolar a contribuição específica do trabalho: a integração de um serviço de visão computacional em nuvem como filtro semântico prévio ao SfM, aplicação praticamente ausente da literatura acadêmica de reconstrução 3D, e a medição controlada de seu impacto sobre o custo computacional e a qualidade do modelo gerado. Neste trabalho, a

detecção de desfoque e a remoção de imagens duplicadas foram implementadas localmente e integradas ao módulo de filtragem.

A pipeline foi validada experimentalmente em duas etapas sobre vídeo 4K orbital real de uma caixa colorida: uma etapa de calibração (260 frames extraídos, captura a 1 m) e uma etapa de validação final com protocolo de captura corrigido (400 frames extraídos, captura a 40 cm).

A hipótese central foi apoiada no escopo experimental, sobretudo pela comparação com ausência total de filtragem (Seção 4.9), e não apenas pela comparação entre os cenários com e sem AWS. No dataset de validação, filtrar com Laplaciano e pHash, com ou sem o estágio AWS, reduziu o número de frames processados de 200 para 170–182 por sessão e o tempo total de execução de 9 h 07 min para 7 h 51 min ou 7 h 50 min. Isso corresponde a reduções de aproximadamente 14% em relação ao cenário sem filtragem. Nas execuções reportadas, não foram observadas diferenças numéricas relevantes no erro de reprojeção, no *fitness* final do alinhamento ou no número de triângulos da malha final. Contudo, a ausência de repetições impede afirmar equivalência estatística. Assim, o principal ganho observado da filtragem foi de eficiência computacional, e não uma melhora mensurada da qualidade geométrica.

A contribuição específica do estágio semântico do AWS, isolada da filtragem local (desfoque e eliminação de duplicidade), é mais modesta do que a da filtragem como um todo. No dataset de calibração, o AWS removeu apenas 5 frames adicionais além dos já descartados pelo Laplaciano (3,9% do conjunto aceito), sem alterar mensuravelmente o erro de reprojeção (1,081 px para 1,073 px) ou os vértices fundidos (aumento de 1,0%). No dataset de validação, o *fitness* do *Global Registration* no cenário com AWS (0,9104) foi inclusive ligeiramente inferior ao do cenário apenas com filtragem local (0,9247), embora o *fitness* final do ICP tenha sido marginalmente maior (0,9959 contra 0,9878). Esses resultados indicam que a maior parte do benefício de filtragem já é obtida pelos estágios locais. No cenário controlado avaliado, o valor específico do AWS está associado ao critério semântico configurado para a cena, e não a uma melhoria consistente de qualidade geométrica ou à validação de um classificador específico do objeto.

Vale registrar também uma limitação identificada durante a validação: o erro médio de reprojeção do SfM foi ligeiramente maior no dataset de 40 cm (≈1,26–1,32 px) do que no de 1 m (≈1,04–1,08 px). Ambos os valores também ficaram acima da faixa de 0,60–0,81 px reportada no artigo Structure-from-Motion Revisited [1] para datasets de referência (Seção 4.2). Apesar de os erros de reprojeção terem sido superiores aos relatados pelos autores, a reconstrução esparsa apresentou qualidade suficiente para ser processada pela etapa de MVS, resultando em uma nuvem de pontos densa e em um modelo tridimensional considerado aceitável.

O alinhamento entre sessões via ICP inicialmente não convergiu nos testes comparativos (*fitness* = 0,0 no cenário baseline), já que o algoritmo é sensível à inicialização e

as reconstruções independentes do COLMAP para cada sessão partem de sistemas de coordenadas arbitrários, sem relação de escala ou orientação entre si. A integração do Super 4PCS como etapa de alinhamento grosseiro, planejada na arquitetura original, esbarrou na dificuldade de compreender e integrar a implementação de referência do algoritmo ao código da pipeline. Em substituição, implementou-se um módulo de *Global Registration* nativo do Open3D, baseado em RANSAC sobre descritores FPFH. Essa abordagem resolveu o problema de convergência. No cenário baseline, o *fitness* do ICP subiu para 0,7730. Esse valor predominou em cinco das seis execuções de verificação, enquanto uma execução convergiu para 0,6987. A semente fixa não elimina totalmente a variabilidade numérica introduzida por etapas anteriores sem controle de semente, como a estimativa de normais. No cenário com AWS, o *fitness* subiu para 0,7516. Com isso, foi possível reconstruir a malha final a partir da nuvem combinada das duas sessões, em vez de utilizar apenas uma sessão isolada. Ainda assim, a inspeção visual da malha combinada revelou que esse *fitness* numericamente satisfatório mascarava um problema geométrico real: o piso, presente em grande proporção no protocolo de captura a 1 metro, dominava o alinhamento e deixava o objeto de interesse deslocado e duplicado (Seção 4.6). As evidências experimentais indicaram a distância de captura como o principal fator do protocolo associado a esse problema, e a redução para aproximadamente 40 cm motivou os demais ajustes de captura. Na validação sobre o novo dataset (Seção 4.8), o *fitness* do ICP foi de 0,9878 (baseline) e 0,9959 (com AWS), e a inspeção visual não identificou duplicação ou deslocamento do objeto na malha final. Como o protocolo mudou em conjunto, esse resultado corrobora a adequação do novo protocolo, sem isolar exclusivamente o efeito da distância.

6 Trabalhos futuros

As extensões mais relevantes identificadas são:

1. Comparação do *Global Registration* adotado com o Super 4PCS, com tempo dedicado à compreensão e integração de sua implementação de referência, avaliando se um alinhamento grosseiro baseado em geometria pura traz ganhos adicionais sobre o método baseado em descritores locais.

2. Comparação do custo-benefício da filtragem AWS frente a abordagens locais baseadas em redes neurais leves (MobileNet, EfficientDet).

3. Extensão do módulo de filtragem para suportar filtragem em lote via AWS S3, eliminando a limitação de tamanho por requisição.

4. Treinamento e avaliação de um modelo específico para o objeto de interesse, para substituir o critério contextual baseado em rótulos genéricos por uma classificação diretamente validada.

Referências

[1] Schönberger, J. L.; Frahm, J.-M. *Structure-from-Motion Revisited*. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, EUA, 2016. p. 4104–4113.

[2] Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[3] Triggs, B.; McLauchlan, P. F.; Hartley, R. I.; Fitzgibbon, A. W. Bundle Adjustment — A Modern Synthesis. In: TRIGGS, B.; ZISSERMAN, A.; SZELISKI, R. (Eds.). *Vision Algorithms: Theory and Practice*. Berlin: Springer-Verlag, 2000. (Lecture Notes in Computer Science, v. 1883). p. 298–372.

[4] Fischler, M. A.; Bolles, R. C. *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*. Communications of the ACM, v. 24, n. 6, p. 381–395, 1981.

[5] Samavati, T.; Soryani, M. *Deep learning-based 3D reconstruction: a survey. Artificial Intelligence Review*, Springer, v. 56, p. 9175–9219, 2023.

[6] Liu, S.; Yang, M.; Xing, T.; Yang, R. *A Survey of 3D Reconstruction: The Evolution from Multi-View Geometry to NeRF and 3DGS. Sensors*, MDPI, v. 25, n. 18, art. 5748, 2025.

[7] Kazhdan, M.; Hoppe, H. *Screened Poisson Surface Reconstruction. ACM Transactions on Graphics*, v. 32, n. 3, art. 29, 2013.

[8] Park, Y.; Lee, T.; Hahn, J.; Kang, M. *p-Poisson Surface Reconstruction in Curl-Free Flow from Point Clouds*. In: *Advances in Neural Information Processing Systems (NeurIPS 2023)*, 2023. p. 60077–60098.

[9] Andrade-Loarca, H.; Hege, J.; Cremers, D.; Kutyniok, G. *Neural Poisson Surface Reconstruction: Resolution-Agnostic Shape Reconstruction from Point Clouds*. arXiv preprint arXiv:2308.01766, 2023.

[10] Mellado, N.; Aiger, D.; Mitra, N. J. *Super 4PCS Fast Global Pointcloud Registration via Smart Indexing. Computer Graphics Forum*, v. 33, n. 5, p. 205–215, 2014.

[11] Zhou, Q.-Y.; Park, J.; Koltun, V. *Open3D: A Modern Library for 3D Data Processing*. arXiv preprint arXiv:1801.09847, 2018.

[12] Open3D. *RegistrationResult — Open3D Python API*. Documentação oficial da biblioteca Open3D, módulo `open3d.pipelines.registration`. Disponível

em: https://www.open3d.org/docs/release/python_api/open3d.pipelines.registration.RegistrationResult.html. Acesso em: jul. 2026.

[13] Open3D. *TriangleMesh.create_from_point_cloud_poisson — Open3D Python API*. Documentação oficial da biblioteca Open3D, módulo `open3d.geometry`. Disponível em: https://www.open3d.org/docs/release/python_api/open3d.geometry.TriangleMesh.html. Acesso em: jul. 2026.

[14] OpenCV. *Image Filtering — Laplacian()*. Documentação oficial da biblioteca OpenCV, v. 4.x. Disponível em: https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html. Acesso em: jul. 2026.

[15] OpenCV. *cv::VideoCapture Class Reference*. Documentação oficial da biblioteca OpenCV, v. 4.x. Disponível em: https://docs.opencv.org/4.x/d8/dfe/classcv_1_1VideoCapture.html. Acesso em: jul. 2026.

[16] Buchner, J. *ImageHash*. Biblioteca Python para *perceptual hashing* de imagens. Disponível em: https://pypi.org/project/ImageHash/. Acesso em: jul. 2026.

[17] Agarwal, S.; Mierle, K.; Others. *Ceres Solver: Modeling Non-linear Least Squares Problems*. Documentação oficial, seção *Loss Functions*. Disponível em: http://ceres-solver.org/nnls_modeling.html. Acesso em: jul. 2026.

[18] Amazon Web Services. *Amazon Rekognition Developer Guide*. Disponível em: https://docs.aws.amazon.com/rekognition/latest/dg/. Acesso em: jun. 2026.

[19] Pertuz, S.; Puig, D.; Garcia, M. A. *Analysis of focus measure operators for shape-from-focus*. *Pattern Recognition*, Elsevier, v. 46, n. 5, p. 1415–1432, 2013.

[20] Zhang, J.; Yao, Y.; Deng, B. *Fast and Robust Iterative Closest Point*. arXiv preprint arXiv:2007.07627, 2021.

[21] Fu, Z.; Zhang, E.; Sun, R.; Zang, J.; Zhang, W. *Research on 3D point cloud alignment algorithm based on SHOT features*. *PLOS ONE*, v. 19, n. 3, art. e0296704, 2024.

[22] Li, Q.; Yang, G.; Gao, C.; Huang, Y.; Zhang, J.; Huang, D.; Zhao, B.; Chen, X.; Chen, B. M. *Single drone-based 3D reconstruction approach to improve public engagement in conservation of heritage buildings: A case of Hakka Tulou*. *Journal of Building Engineering*, Elsevier, v. 87, art. 108954, 2024.

[23] Yang, L.; Liu, K.; Ou, R.; Qian, P.; Wu, Y.; Tian, Z.; Zhu, C.; Feng, S.; Yang, F. *Surface Defect-Extended BIM Generation Leveraging UAV Images and Deep Learning*. *Sensors*, MDPI, v. 24, n. 13, art. 4151, 2024.

[24] Wang, S.; Leroy, V.; Cabon, Y.; Chidlovskii, B.; Revaud, J. *DUSt3R: Geometric 3D Vision Made Easy*. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, EUA, 2024. p. 20697–20709.

[25] da Silva, E. A.; Apaza-Agüero, K. P. *Improved low-cost 3D reconstruction pipeline by merging data from different color and depth cameras*. *Revista Brasileira de Computação Aplicada*, Universidade de Passo Fundo, v. 14, n. 2, p. 56–66, 2022.

[26] Wu, Q.; Shen, Y.; Ke, J. *A general computationally-efficient 3D reconstruction pipeline for multiple images with point clouds*. In: *MICCAI 2023 Workshops*, Lecture Notes in Computer Science, v. 14393, p. 193–202. Springer, Cham, 2023.

[27] Shang, Z.; Shen, Z. *Single-pass inline pipeline 3D reconstruction using depth camera array*. *Automation in Construction*, Elsevier, v. 140, art. 104368, 2022.

[28] Zhu, L. *A pipeline of 3D scene reconstruction from point clouds*. Tese (Doutorado) — Aalto University School of Engineering, Espoo, Finlândia, 2015.

[29] Lowe, D. G. *Distinctive Image Features from Scale-Invariant Keypoints*. *International Journal of Computer Vision*, Springer, v. 60, n. 2, p. 91–110, 2004.

[30] Schönberger, J. L.; Zheng, E.; Pollefeys, M.; Frahm, J.-M. *Pixelwise View Selection for Unstructured Multi-View Stereo*. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, Holanda, 2016. p. 501–518.

[31] Besl, P. J.; McKay, N. D. *A Method for Registration of 3-D Shapes*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 14, n. 2, p. 239–256, fev. 1992.

[32] Chen, Y.; Medioni, G. Object Modeling by Registration of Multiple Range Images. In: *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*. Sacramento, CA: IEEE, abr. 1991. p. 2724–2729.

[33] Ahmed, N.; Natarajan, T.; Rao, K. R. *Discrete Cosine Transform*. IEEE Transactions on Computers, v. C-23, n. 1, p. 90–93, 1974.

[34] Hamming, R. W. *Error Detecting and Error Correcting Codes*. The Bell System Technical Journal, v. 29, n. 2, p. 147–160, 1950.