# 3D Scene Reconstruction and Virtual Tour

Muhammad Eelian Khan
26100113@lums.edu.pk

Abdul Naafay
26100242@lums.edu.pk

December 6, 2025

## Abstract

This project presents a complete Structure-from-Motion (SfM) pipeline for reconstructing an indoor room and using it to create an interactive virtual tour. The system performs feature extraction, two-view geometry estimation, incremental camera pose recovery, triangulation, and bundle adjustment to obtain a consistent set of camera parameters and a 3D representation of the scene. While an initial sparse point cloud was generated from our own SfM reconstruction, its limited density and accumulated drift made it unsuitable for high-quality visualization. To ensure geometric accuracy and visual continuity, a dense point cloud produced by Agisoft Metashape was incorporated, yielding a significantly more stable and complete model of the environment. The recovered camera poses and dense geometry are then used to enable smooth viewpoint transitions and interactive navigation in a browser-based Three.js application. The final system provides a photorealistic and spatially coherent virtual walkthrough of the captured indoor space.

**Project Repository:** https://github.com/Abdul-Naafay/CV_Project_Group3

## 1 Introduction

Reconstructing real-world environments from imagery has long been a central problem in computer vision. Structure-from-Motion (SfM) provides a principled framework for recovering both camera poses and 3D scene structure from a set of unordered images. While SfM pipelines are well understood in outdoor or highly textured environments, indoor settings remain challenging due to low-texture surfaces, repeated patterns, and limited parallax. These characteristics often lead to unstable pose estimation, sparse reconstructions, and accumulated drift across disconnected regions of the scene.

This project aims to reconstruct an indoor room and render it as an interactive virtual tour accessible through a web interface. The pipeline integrates feature extraction, two-view geometry estimation, incremental pose recovery, triangulation, and global bundle adjustment to obtain a coherent reconstruction. Although an initial sparse point cloud was produced from our own SfM implementation, its density and geometric stability were insufficient for reliable visualization. To obtain a high-fidelity model suitable for rendering, a dense point cloud generated by Agisoft Metashape was adopted and aligned with the recovered camera poses. The final reconstruction is visualized in a Three.js-based virtual tour, where users can transition between viewpoints using smooth pose interpolation and image cross-fading.

## 2 Methodology

Our reconstruction pipeline follows a standard incremental Structure-from-Motion (SfM) formulation. Given an unordered set of images $\{I_i\}$, the goal is to recover camera poses $(R_i, t_i)$ and a consistent set of 3D points $\{X_j\}$ such that their projections agree with observed image measurements.

### 2.1 Feature Detection and Matching

For each image, local features are detected and described using a scale and rotation-invariant feature extractor. Each feature is represented by a descriptor vector $d \in \mathbb{R}^D$. Candidate correspondences between images $I_i$ and $I_j$ are identified using nearest-neighbor matching in descriptor space. To ensure robustness, we apply the ratio test to enforce descriptor distinctiveness and use symmetric matching to retain only mutually consistent correspondences.

Given corresponding points $x_k \leftrightarrow x'_k$ in normalized coordinates, these matches form the basis for estimating the epipolar geometry.

### 2.2 Two-View Geometry and Essential Matrix Estimation

The geometric relationship between two calibrated cameras is encoded by the essential matrix

$$E = [t]_\times R, \tag{1}$$

where $R \in SO(3)$ and $t \in \mathbb{R}^3$ represent the relative rotation and translation. We estimate $E$ from point correspondences using the five-point algorithm, followed by enforcing

the rank-2 constraint via singular value decomposition (SVD):

$$E = U\,\mathrm{diag}(1,1,0)\,V^\top. \tag{2}$$

The relative pose $(R,t)$ is recovered from $E$ using SVD decomposition and selecting the physically valid configuration through a cheirality check, ensuring that triangulated points lie in front of both cameras.

### 2.3 Triangulation

Given two camera projection matrices $P_i$ and $P_j$ and a corresponding feature pair $(x,x')$, the 3D point $X$ is estimated by minimizing the reprojection error:

$$X^* = \arg\min_X \left(\|x-\pi(P_iX)\|^2 + \|x'-\pi(P_jX)\|^2\right) \tag{3}$$

where $\pi(\cdot)$ denotes projection from homogeneous to image coordinates. A linear triangulation method is used to obtain an initial estimate, and points with large reprojection errors are discarded.

### 2.4 Incremental Camera Registration

After establishing an initial two-view reconstruction, remaining cameras are added incrementally. For a new image $I_k$, we identify 2D–3D correspondences by matching its features to existing 3D points. The camera pose $(R_k,t_k)$ is then estimated by solving the Perspective-n-Point (PnP) problem:

$$(R_k,t_k) = \arg\min_{R,t} \sum_{j\in\mathcal{V}_k} \|x_{kj}-\pi(RX_j+t)\|^2, \tag{4}$$

where $\mathcal{V}_k$ denotes the set of 3D points visible in image $I_k$. RANSAC is used to ensure robustness against outliers. Newly observed points are triangulated, expanding the reconstruction as additional views are introduced.

### 2.5 Bundle Adjustment

Bundle adjustment refines all camera poses and 3D points jointly by minimizing the global reprojection error:

$$\min_{\{R_i,t_i,X_j\}} \sum_{i,j} \|x_{ij}-\pi(R_iX_j+t_i)\|^2, \tag{5}$$

where $x_{ij}$ denotes the observation of point $X_j$ in camera $i$. This nonlinear optimization significantly reduces drift and improves metric consistency across the reconstruction.

### 2.6 Sparse vs. Dense Reconstruction

Our pipeline initially produced a sparse reconstruction in which 3D points correspond only to detected features. While sufficient for pose recovery, this representation was unsuitable for visualization due to low point density, noise on weakly textured surfaces, and noticeable drift between disconnected walls. To obtain a better model, we adopted a dense point cloud generated by Agisoft Metashape. Agisoft performs multi-view stereo (MVS) depth estimation, producing a dense geometric reconstruction that exhibits greater surface continuity and stability. The dense cloud is aligned with our estimated camera poses and used as the geometric substrate for the interactive virtual tour described in later sections.

## 3 Interactive Virtual Tour

The final objective of the project is to enable an interactive walkthrough of the reconstructed environment using the recovered camera poses and dense scene geometry. To achieve this, we implemented a browser-based viewer in Three.js that supports smooth transitions between viewpoints, realistic image rendering, and intuitive navigation. This section outlines the representation of the camera network and the mathematical framework used for pose interpolation.

### 3.1 View Graph Construction

Each reconstructed camera is represented as a node in a view graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ contains all camera poses and $\mathcal{E}$ encodes connectivity based on spatial proximity or image overlap. Two cameras $i$ and $j$ are connected if they observe a sufficient number of common 3D points, or if their centers satisfy

$$\|C_i - C_j\| < \tau, \tag{6}$$

where $C_i = -R_i^\top t_i$ denotes the camera center and $\tau$ is a distance threshold. This graph determines which transitions are allowed in the virtual tour and prevents users from teleporting between geometrically unrelated parts of the scene.

### 3.2 Pose Interpolation

When moving from camera $i$ to camera $j$, the virtual camera must follow a smooth path in space while maintaining consistent orientation. Let $(R_i,t_i)$ and $(R_j,t_j)$ denote the initial and target camera poses.

**Position Interpolation.** The camera position is interpolated linearly:

$$p(t) = (1-t)\,C_i + t\,C_j, \qquad t\in[0,1]. \tag{7}$$

**Orientation Interpolation.** Rotations are represented as unit quaternions $q_i$ and $q_j$. We apply spherical linear interpolation (SLERP) to achieve a constant-rate rotation:

$$q(t) = \frac{\sin((1-t)\theta)}{\sin\theta}q_i + \frac{\sin(t\theta)}{\sin\theta}q_j, \tag{8}$$

where $\theta = \arccos(q_i^\top q_j)$ is the angle between orientations. The interpolated quaternion is then converted back to a rotation matrix for rendering.

### 3.3 Image Cross-Fading

To produce visually smooth transitions, we blend the source and target images using an opacity parameter $\alpha(t)$:

$$I(t) = (1 - \alpha(t))\, I_i + \alpha(t)\, I_j, \tag{9}$$

where $\alpha(t)$ increases monotonically from 0 to 1. This removes the perceptual discontinuity that would otherwise occur during pose switching and provides a seamless photographic experience.

### 3.4 Handling Disconnected Reconstructions

Indoor environments often contain walls or corridors with limited feature overlap, leading to independently reconstructed components. Aligning these regions can be challenging due to drift or insufficient geometric constraints. Two strategies were considered for integrating these components into the viewer:

- **Python Pre-Processing:** Rigidly aligning partial reconstructions using manually selected correspondences or iterative closest point (ICP).

- **Three.js Scene Grouping:** Treating each independent reconstruction as a separate group and positioning them manually in the virtual environment.

In our implementation, the second approach was adopted. By organizing disjoint point clouds and camera sets into hierarchical Three.js groups, we achieved stable alignment and avoided compounding reconstruction errors. This strategy also offered greater flexibility during scene assembly and improved the consistency of the virtual tour.

### 3.5 Rendering Environment

The dense Agisoft point cloud is rendered as the geometric context for navigation, while each camera node stores the corresponding input image and pose. As the user traverses the view graph, the viewer updates the virtual camera's position and orientation using the interpolated pose, blends the relevant images, and displays the scene within the 3D environment. This combination of geometric fidelity, pose interpolation, and photometric cross-fading produces a smooth, immersive virtual walkthrough of the captured indoor space.

## 4 Experimental Results

This section evaluates the behavior of the incremental SfM pipeline, analyzes the impact of bundle adjustment, compares the sparse reconstruction to the dense MVS model, and discusses the performance of the final virtual tour.

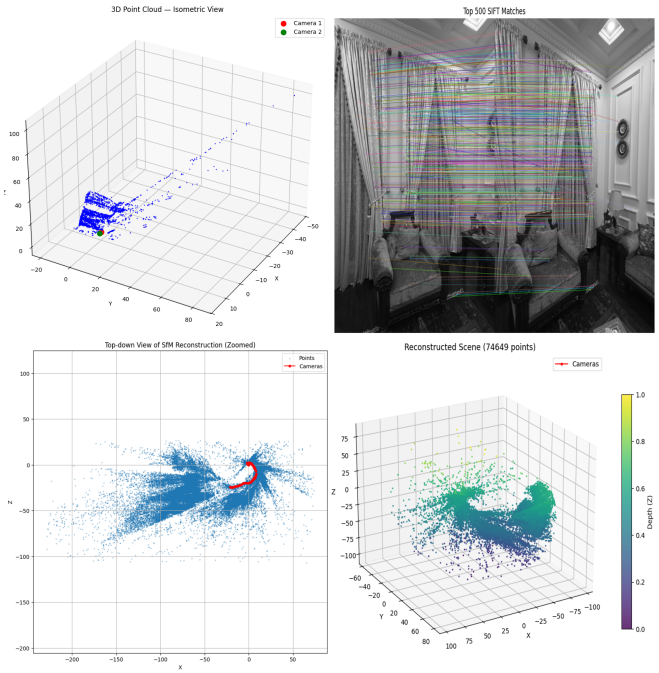


Figure 1: Sparse SfM reconstruction. Top: feature distribution and initial triangulated points. Bottom: camera poses before and after bundle adjustment.

### 4.1 Sparse Reconstruction

Figure 1 summarizes the outputs of the sparse SfM stage. The detected feature points (top-right) reveal strong activation in textured regions such as posters, edges, and furniture, while exhibiting noticeably weaker coverage on large planar walls. This uneven feature distribution directly affects the density and shape of the triangulated reconstruction shown in the top-left panel, where most walls appear incomplete or fragmented.

Before global refinement, the reconstructed camera trajectory drifted slightly along texture-poor walls. After bundle adjustment, the trajectory became significantly more coherent. Quantitatively, the global BA reduced the total reprojection cost from an initial value of $6.79 \times 10^6$ to $2.13 \times 10^6$ over 13 iterations. The final mean reprojection error was 11.43 pixels with a median of only 0.59 pixels, indicating that most points were well aligned while a minority of challenging regions (primarily flat walls) contributed to larger residuals. These results confirm that bundle adjustment substantially improved the global consistency of the recovered structure.

### 4.2 Dense Reconstruction

Figure 2 compares the sparse reconstruction with the dense Agisoft Metashape model. Our sparse pipeline produced approximately **77,000** 3D points, reflecting the limited number of keypoint-supported correspondences available indoors. In contrast, the Agisoft reconstruction produced approximately

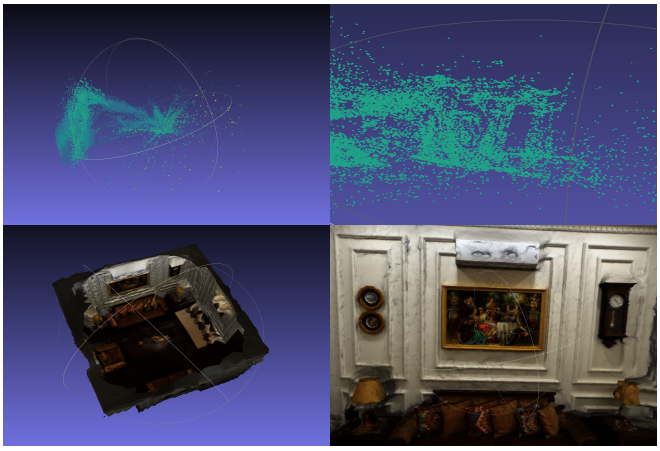

Figure 2: Comparison between sparse (top row) and dense (bottom row) reconstructions. The dense MVS model provides significantly higher surface continuity and geometric stability.

**470,000** points. The dense model exhibits far greater coverage of walls, ceiling regions, and object surfaces, producing a continuous geometric representation instead of the isolated clusters observed in the sparse reconstruction.

This disparity stems from methodological differences: sparse SfM triangulates only keypoints, whereas dense MVS estimates depth per pixel using multi-view photometric consistency. The dense model thus captures structure even in lightly textured regions where sparse keypoints fail. These results justify the integration of the Agisoft model for visualization in the virtual tour.

### 4.3 Feature and Matching Statistics

Across the 63 images, the number of detected SIFT keypoints ranged from approximately $2{,}000$ in low-texture images to over $14{,}000$ in highly textured views. Most images produced between $5{,}000$ and $10{,}000$ features. This variation strongly influenced match count and triangulation coverage: views facing blank walls contributed fewer points and increased uncertainty, whereas cluttered regions supported more robust pose estimation.

These statistics highlight the inherent challenges of indoor SfM: large uniform surfaces produce few keypoints, making reconstruction reliant on a relatively small subset of views.

### 4.4 Virtual Tour Rendering

Figure 3 shows rendering outputs from the interactive Three.js viewer. The use of the dense Agisoft cloud significantly enhances spatial immersion, ensuring correct parallax and providing a realistic sense of depth. Even though the underlying SfM reconstruction was sparse, the accurate camera poses combined with dense geometry result in a seamless and visually coherent virtual tour.

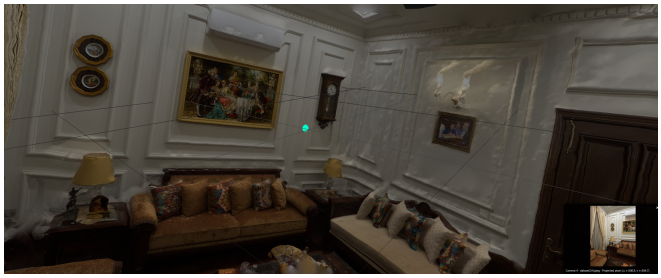

Figure 3: Virtual tour rendering results.

## 5 Discussion and Limitations

While the overall pipeline successfully recovered camera poses and enabled a functional virtual tour, the reconstruction process exposed several limitations rooted in both theoretical constraints of Structure-from-Motion (SfM) and practical challenges specific to indoor environments. This section critically examines why errors occurred, why some components underperformed, and what structural characteristics of the scene influenced the final outputs.

### 5.1 Drift and Instability in the Incremental Reconstruction

One of the most persistent issues was geometric drift observed before bundle adjustment. This drift manifested as a slight curvature of walls and a gradual deviation of the camera trajectory from the true physical layout. The primary cause lies in the incremental nature of pose estimation: each new PnP solution introduces small errors in rotation and translation, and these errors accumulate over the sequence.

The problem is worsened by the scene's geometry. Many portions of the room consist of large, textureless planar surfaces. When 2D–3D correspondences predominantly lie on a single plane, the PnP problem becomes poorly conditioned due to reduced geometric constraints. Theoretically, planar configurations introduce ambiguities in pose estimation because multiple camera motions produce similar reprojection errors. As a result, even slight noise in detection or matching leads to systematic drift.

Bundle adjustment corrected much of this accumulated error by optimizing all poses jointly. However, its ability to eliminate drift entirely was limited by the scarcity of reliable constraints on certain walls, making some sections of the reconstruction inherently underconstrained.

### 5.2 Feature Scarcity and Matching Failures

A major limitation arose from the uneven distribution of SIFT features. While some images contained more than $14{,}000$ keypoints, views facing blank or low-texture walls produced fewer than $1000$. Theoretical properties of the SIFT detector explain this behavior: keypoints are identified as extrema in

the Difference-of-Gaussians response, which requires significant local intensity variation. Uniform surfaces do not contain such variations, resulting in very few detectable features.

Low feature count has two crucial downstream consequences:

- **Poor feature coverage leads to unstable essential matrix estimation.** Small sets of correspondences, especially those concentrated on planar regions, reduce the reliability of two-view pose estimation.
- **Triangulated points cluster around textured areas**, leaving large holes in the sparse reconstruction and creating an incomplete geometric representation of the room.

Moreover, the reduced number of matches resulted in several PnP failures during incremental registration. Although robust estimators such as RANSAC mitigated outliers, they could not compensate for insufficient inliers. This explains why some camera poses required manual intervention or produced unusually high reprojection errors before bundle adjustment.

### 5.3 Runtime and Optimization Challenges

Global bundle adjustment was computationally expensive due to the scale of the problem: tens of thousands of 3D points and 63 camera poses produce a very large optimization space. The solver required multiple iterations to converge, and each iteration involved constructing and solving large sparse Jacobian systems.

Although the optimization successfully reduced the global error, the high runtime reflects a practical limitation of using full BA in student-scale implementations. Industrial SfM pipelines typically incorporate hierarchical strategies or keyframe selection to reduce computational cost. Our implementation optimized all parameters simultaneously, which, while accurate introduced runtime overhead.

### 5.4 Fragmented Sparse Geometry

Due to limited feature coverage, several walls and surfaces did not reconstruct fully in the sparse model. In many cases, only a thin vertical stripe of points appeared where wall corners or small textured artifacts existed. This fragmented geometry made it impossible to rely solely on the sparse reconstruction for building a coherent visualization.

Because of this, merging disconnected or poorly reconstructed walls into a coherent model was challenging. Automatic alignment based on point cloud overlap was unreliable due to insufficient geometric constraints. We therefore adopted a simpler approach in the virtual tour implementation: treating each independent reconstruction as a separate Three.js group and positioning them manually within the 3D scene. While effective in practice, this workaround highlights a fundamental limitation of sparse SfM in texture-poor indoor environments.

### 5.5 Limitations of the Virtual Tour Representation

Although the viewer provides smooth pose transitions and photorealistic image blending, several limitations remain:

- **The viewer depends heavily on accurate camera poses.** Any residual drift or misalignment is perceptually noticeable when interpolating between viewpoints.
- **The dense cloud is static and unstructured.** It provides visual realism but does not define surfaces or semantics, limiting interactive functionality.

Despite these limitations, the hybrid approach proved effective for producing a stable, visually realistic indoor tour.

### 5.6 Summary of Critical Insights

The project revealed several important lessons about the strengths and weaknesses of SfM for indoor reconstruction:

- Sparse SfM is extremely sensitive to texture availability and baseline width.
- Drift is inevitable when pose estimation is underconstrained; global optimization can reduce but not fully eliminate it.
- Dense MVS methods are essential for producing visually complete geometry.
- Indoor SfM benefits greatly from hybrid pipelines that separate pose estimation from dense reconstruction.

## 6 Conclusion

This project implemented a complete Structure-from-Motion pipeline for reconstructing an indoor environment and visualizing it through an interactive virtual tour. By integrating a dense MVS reconstruction from Agisoft Metashape, the system achieved the level of surface continuity and realism required for an immersive tour. The final viewer combined these dense geometric cues with the estimated camera poses to produce smooth, photorealistic transitions between viewpoints.

The project highlights the complementary strengths of sparse and dense reconstruction methods: sparse SfM excels at pose recovery, while dense MVS captures complete geometry. It also underscores the inherent challenges of indoor reconstruction, including feature scarcity, drift due to planar degeneracies, and the computational cost of global optimization.

Future work could explore automated alignment of disconnected components, mesh generation from the dense point cloud, and free-viewpoint navigation using neural rendering or depth-based reprojection.