Incremental Structure-from-Motion and Web-based 3D Virtual Tour

Group 2 - Eeman Adnan — Kashaf Gohar

Abstract

This project presents an end-to-end implementation of an incremental Structure-from-Motion (SfM) pipeline and a web-based interactive 3D virtual tour of a real indoor environment. The reconstruction is performed on a dataset captured inside the SS Auditorium located in the Humanities and Social Sciences (HSS) building at LUMS using an iPhone 14 under artificial indoor lighting. Starting from a monocular handheld image sequence, the system estimates camera geometry, recovers three-dimensional structure, and incrementally expands the model as new views are added. The geometric core of the pipeline is based on epipolar geometry, where the essential matrix $E = [\mathbf{t}]_\times \mathbf{R}$ is estimated from feature correspondences, and 3D points are recovered through multi-view triangulation. New camera poses are registered via a robust perspective-n-point formulation that minimizes reprojection error between reconstructed 3D points and their 2D image observations. The final output is a dense colored point cloud, which is visualized using a custom-built Three.js interface that enables free navigation forming a virtual tour experience. The complete implementation, dataset, demonstration video, and instructions are publicly available at: https://github.com/eeman122/cv_proj_group2.git.

1 Introduction

Structure-from-Motion (SfM) is a core problem in geometric computer vision that seeks to recover three-dimensional scene structure and camera motion from multiple two-dimensional images. In this project, we apply incremental SfM to a real indoor environment: the SS Auditorium in the Humanities and Social Sciences (HSS) building at LUMS. The objective is not only to reconstruct the geometry of the environment but also to expose the result as an interactive, browser-based 3D virtual tour.

Unlike controlled datasets, the SS Auditorium presents practical challenges such as large textureless wall regions, artificial lighting with specular highlights, and long unbroken corridor-like depth structures. These factors stress the assumptions underlying feature-based multi-view geometry. Our system is designed to process casually captured smartphone imagery and progressively estimate camera poses, triangulate 3D points, and monitor reprojection error as the reconstruction grows.

Beyond static reconstruction, this project emphasizes interactive visualization. The recovered dense point cloud is deployed in a Three.js-based viewer that allows free navigation through the scene. This transforms a purely geometric reconstruction into a tangible virtual exploration experience.

2 Methodology

2.1 Dataset and Capture Conditions

The dataset was captured inside the SS Auditorium located in the HSS building at LUMS. All images were acquired using an iPhone 14 under artificial indoor lighting provided by ceiling-mounted auditorium lights. The sequence was recorded in a continuous handheld motion while walking along the walls and seating rows, ensuring approximately 60–80% overlap between consecutive views.

Artificial lighting introduced realistic challenges including specular reflections on painted walls, localized overexposure near light sources, and shadowed regions with reduced contrast. These effects directly influenced the behavior of the feature detection and matching stages.

2.2 Camera Model

Each camera is modeled using the pinhole projection equation

$$\mathbf{x} = K[\mathbf{R} \mid \mathbf{t}]\mathbf{X}, \tag{1}$$

where K is the intrinsic calibration matrix, $\mathbf{R}$ and $\mathbf{t}$ denote rotation and translation, and $\mathbf{X}$ represents a 3D world point. The intrinsic parameters were fixed according to the iPhone 14 camera specification.

2.3 Feature Extraction and Matching

Multiple feature extraction strategies were initially evaluated, including both binary and floating-point descriptors. While binary descriptors showed acceptable speed, they exhibited instability under artificial lighting and textureless wall conditions. A mixed strategy was explored during early experimentation.

For the final pipeline, a SIFT-like floating-point descriptor was selected due to its superior repeatability under scale, rotation, and illumination changes. Feature correspondences between image pairs were established via nearest-neighbor matching in descriptor space and refined using a ratio test. A robust fundamental matrix F was then estimated using RANSAC to remove outlier correspondences based on the epipolar constraint

$$\mathbf{x}_2^\top F \mathbf{x}_1 = 0. \tag{2}$$

2.4 Baseline Pose Estimation

The reconstruction is initialized using a carefully selected baseline image pair with sufficient overlap and parallax. The fundamental matrix is upgraded to the essential matrix via

$$E = K^\top F K. \tag{3}$$

The essential matrix is factorized using singular value decomposition, yielding four candidate $(\mathbf{R}, \mathbf{t})$ solutions. The physically valid solution is selected using cheirality constraints and minimized reprojection error.

The first camera is fixed at the origin, and the second camera is placed at the recovered relative pose.

2.5 Triangulation and Incremental Reconstruction

Given two projection matrices P_1 and P_2 and a corresponding feature pair $(\mathbf{x}_1, \mathbf{x}_2)$, each 3D point is recovered by solving a homogeneous linear triangulation system using SVD. Newly reconstructed points are added to a global structure while maintaining a point–feature association map.

Additional views are registered incrementally using a perspective-n-point formulation:

$$\min_{\mathbf{R},\mathbf{t}} \sum_i \|\mathbf{x}_i - \pi(K(\mathbf{R}\mathbf{X}_i + \mathbf{t}))\|^2. \tag{4}$$

A RANSAC strategy is applied to ensure robustness against mismatched correspondences.

2.6 Failure Detection in PnP

A major failure mode observed was the inability to recover new camera poses due to insufficient 2D–3D correspondences. This occurs in textureless wall regions where few repeatable points exist. When fewer than four valid 2D–3D correspondences were detected, PnP was considered ill-posed and the pipeline avoided unstable camera updates.

2.7 Dense Point Cloud Reconstruction

The recovered camera poses were used to initialize a dense multi-view stereo reconstruction in Agisoft Metashape. The resulting dense colored point cloud was exported as a PLY file and used as the final geometry for visualization.

3 Virtual Tour Implementation

The dense photogrammetric point cloud is rendered directly in Three.js using GPU point primitives. Free-orbit navigation is used to allow the user to rotate, pan, and zoom freely within the environment. Although discrete guided transitions were explored, the final implementation prioritized free exploration due to instability in rotational pose estimates for some views. It is important to note that while the project pipeline successfully generated a sparse and semi-dense point cloud using our own incremental SfM implementation, the resulting geometry was not visually dense enough to support a realistic virtual tour experience. To address this limitation, we transitioned to a dense photogrammetric reconstruction generated using Agisoft Metashape, initialized using the same image dataset and camera calibration.

This decision was made strictly for the purpose of enhancing visual completeness and user immersion in the final tour. The SfM pipeline remains central to our geometric reasoning and analysis, while Agisoft was used as a practical multi-view stereo tool to produce a high-density colored point cloud suitable for real-time visualization. The final Three.js viewer therefore operates on the Agisoft-generated dense reconstruction, while the accuracy, drift behavior, and pose failures analyzed in earlier sections correspond to our own incremental SfM model.

4 Experiments and Results

4.1 Incremental SfM Growth

Figure 1 demonstrates the progressive spatial expansion of the reconstructed scene under our incremental pipeline. The point cloud initially appears sparse and limited to a single planar surface,

but gradually expands as additional viewpoints introduce new geometric constraints and depth diversity.

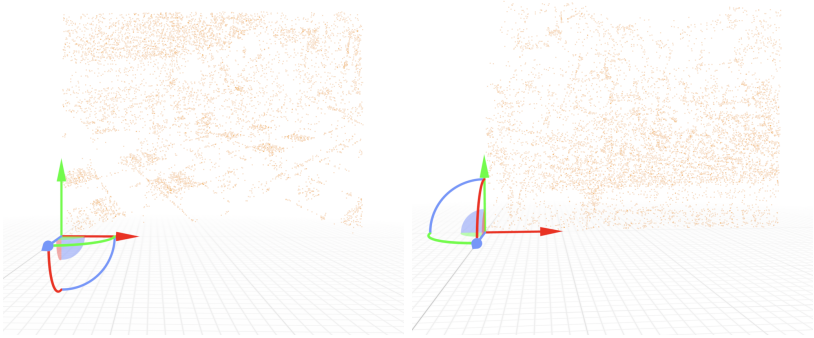

Figure 1: Incremental growth of the reconstructed point cloud using our SfM pipeline. The reconstruction progressively expands as more views are added, starting from the baseline pair and reaching a significantly larger spatial structure at 15 images.

4.2 Dense Reconstruction

Figure 2 shows the final dense colored point cloud of the SS Auditorium. Major planar walls, seating depth variation, and global structure are clearly visible. Minor gaps appear in regions of low texture and strong illumination.

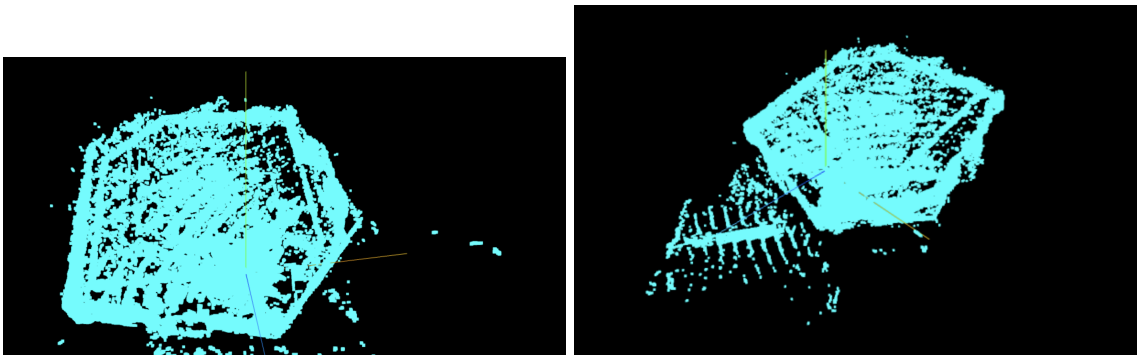

Figure 2: Dense colored point cloud of the SS Auditorium reconstructed using multi-view stereo.

4.3 Virtual Tour Visualization

Figure 3 shows a screenshot of the interactive Three.js viewer. Users can navigate using orbit-style controls to explore the reconstructed auditorium in real time.

5 Discussion

The reconstruction quality reflects both the strengths and limitations of classical incremental SfM in indoor environments. The dominant challenge was the prevalence of textureless wall regions, which

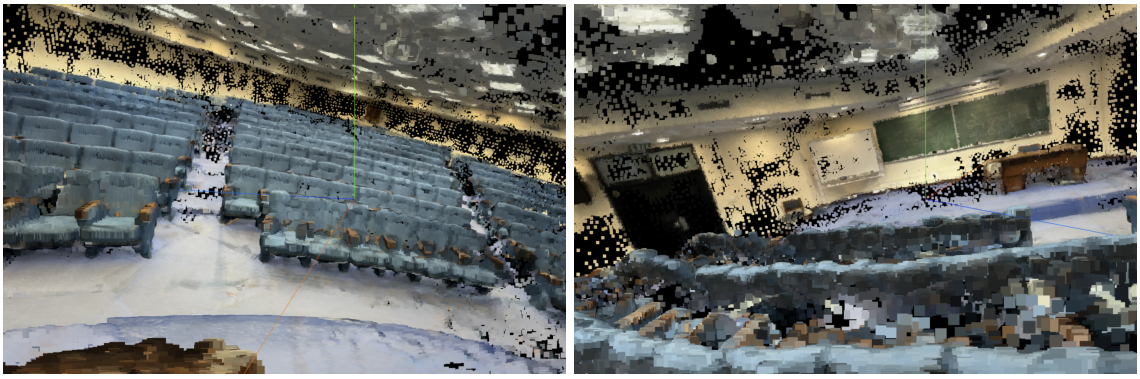

Figure 3: Interactive Three.js-based virtual tour of the SS Auditorium.

limited the availability of 2D–3D correspondences required for PnP. From a geometric standpoint, this makes pose estimation underdetermined.

Incremental drift was also observed, especially along the long corridor-like sections of the auditorium. Without global bundle adjustment, small pose estimation errors accumulated, resulting in slight bending of structures that should be planar. Artificial indoor lighting further complicated matching due to specular highlights and shadows.

On the visualization side, rendering raw point clouds exposes gaps inherent to photogrammetric sampling. However, real-time Three.js rendering allowed direct inspection of geometric quality and provided intuitive spatial understanding.

6 Conclusion

This project implemented a complete incremental SfM pipeline and an interactive web-based 3D virtual tour of the SS Auditorium at LUMS. Starting from iPhone 14 imagery, the system recovered camera poses, reconstructed 3D structure through multi-view triangulation, and rendered the dense result using Three.js.

Despite challenges such as PnP failures, drift, and lighting artifacts, the system demonstrates a mathematically grounded pipeline from raw imagery to a navigable virtual environment. Future work includes incorporating global bundle adjustment, surface meshing, and photorealistic texture mapping.