# IC614 : Computer Vision

# Programming Assignment 1
# Structure from Motion(SfM)

**YANG DONGJAE(202522027)**
Department of Electrical Engineering and Computer Science, DGIST
Daegu, South Korea

**Abstract**

This project aims to explore the mechanism of Structure from Motion (SfM), a classical computer vision technique for reconstructing 3D structures from multiple 2D images taken from varying viewpoints. In this assignment, the intrinsic camera matrix (K) was provided by the professor, along with 32 images captured from different perspectives. The reconstruction process is implemented using classical vision algorithms and is further applied to a set of raw images captured by the student, demonstrating the generalizability of the approach to unprocessed data.

Except for my data applied section, all contexts were written down based on provided data by the professor.

**Keywords:** SfM, Essential Matrix, Fundamental Metrix, Intrinsic **K**, RANSAC, triangulation, PnP

## Traditional Approach

- Spatial Invariant Feature Extraction  matching
- Preserve inliers and remove outliers by using RANSAC
- Estimate Fundamental Matrix
- Estimate Essential Matrix
- Extract [R|t] from decomposed Essential matrix with SVD
- Perform Triangulation using Cheirality condition
- Estimation of pose using PnP
- (IDEA) Apply L2-norm-based filtering to reject outlier 3D points whose reprojected distances to epipolar lines exceed a threshold. This geometric consistency check improves robustness of triangulated point cloud.

### Spatial Invariant Feature Extraction  Matching

In Structure from Motion problems, matching points between images taken from different viewpoints are essential. To achieve this, features must first be extracted from each image and then compared across images. Among various feature extraction techniques, SIFT (Scale-Invariant Feature Transform) is particularly powerful. It provides well-defined features that are invariant to scale, rotation, and translation. As a result, SIFT offers high repeatability and robustness in image processing tasks.

Keypoints are detected as local extrema in the DoG scale-space by comparing each pixel with 26 neighbors. Unstable extrema are discarded as follows:

- **Low-contrast removal:**

$$\hat{\mathbf{x}} = -H^{-1}\nabla D$$

  Discard if $|D(\hat{\mathbf{x}})|$ is below threshold.
- **Edge response elimination:** Compute Hessian:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

  Remove if:

$$\frac{(\mathrm{Tr}(H))^2}{\det(H)} > \frac{(r+1)^2}{r}, \quad r = 10$$


Figure 1: Gaussian scale-space pyramid across 4 octaves and multiple scales. Each row shows different octaves, and each column represents increasing scale levels.

### Fundamental Matrix Estimation and Outlier Rejection using RANSAC

Estimating a robust fundamental matrix **F** is essential in the SfM pipeline, especially in real-world scenarios where data may contain outliers due to incorrect feature matches, occlusions, or repetitive patterns. To address this, we employ the 8-point algorithm within a RANSAC (Random Sample Consensus) framework, which is a widely adopted method in computer vision for robust model fitting in the presence of noise.

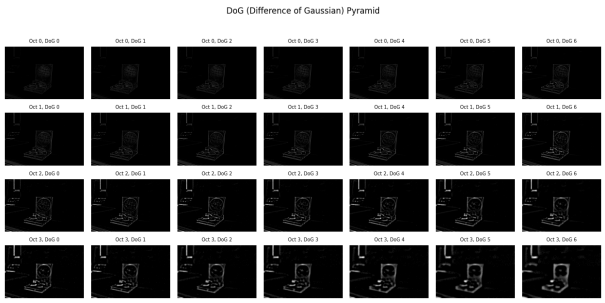


Figure 2: DoG (Difference of Gaussian) pyramid constructed by subtracting adjacent Gaussian-blurred images in each octave.

**(Step II) What is RANSAC and why use it?** RANSAC is an iterative algorithm designed to estimate parameters of a mathematical model from a dataset that may contain outliers. Rather than fitting a model to all data points, which risks bias due to incorrect correspondences, RANSAC iteratively samples random subsets, estimates candidate models, and evaluates their performance by counting the number of inliers—data points that agree with the estimated model within a predefined tolerance. This approach allows the algorithm to converge on a model that best fits the majority of inlier data, thus ensuring robustness against noisy observations.

In the context of fundamental matrix estimation, RANSAC improves the reliability of epipolar geometry by discarding correspondences that violate the epipolar constraint. The process is summarized as follows:

1. **Sampling Correspondences:**
   From the computed matches, randomly select 8 pairs of corresponding points to estimate the initial fundamental matrix.
2. **Normalization of Coordinates:**
   Prior to computing the initial matrix, normalize the image coordinates by translating the centroid to the origin and scaling the average distance to $\sqrt{2}$. The fundamental matrix is estimated using the following epipolar constraint:

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = x'^T \mathbf{F} x = 0$$

   This leads to a homogeneous linear system $\mathbf{Af} = 0$, where:

$$\mathbf{A} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_8x_8 & x'_8y_8 & x'_8 & y'_8x_8 & y'_8y_8 & y'_8 & x_8 & y_8 & 1 \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix}$$

   The fundamental matrix $\mathbf{F}$ is then obtained by solving this system using Singular Value Decomposition (SVD), followed by enforcing the rank-2 constraint by setting the smallest singular value to zero.
3. **Error Evaluation:**
   The computed fundamental matrix is validated by applying it to all point correspondences and measuring the epipolar constraint error:

$$\text{Error} = |x'^T \cdot \mathbf{F} \cdot x|$$

   This evaluates how well each point pair satisfies the epipolar geometry.
4. **Inlier Selection:**
   A point pair is classified as an inlier if its error is below a predefined threshold. The number of inliers indicates the quality of the estimated fundamental matrix.
5. **RANSAC:**
   Among multiple estimations, the fundamental matrix with the highest inlier count is chosen as the final result, ensuring a consensus-based fit that is resilient to outlier contamination.

   The computed fundamental matrix $\mathbf{F}$ is written below:

$$\mathbf{F} = \begin{bmatrix} -1.56141771 \times 10^{-7} & 4.67922289 \times 10^{-6} & 1.24921905 \times 10^{-3} \\ -3.56235464 \times 10^{-6} & 5.73683875 \times 10^{-7} & -1.84278529 \times 10^{-2} \\ -1.19858616 \times 10^{-3} & 1.64844577 \times 10^{-2} & 1.00000000 \times 10^{0} \end{bmatrix}$$

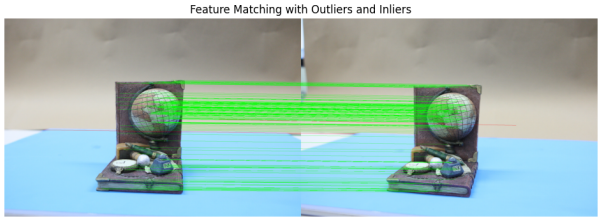


Figure 3: DoG (Difference of Gaussian) pyramid constructed by subtracting adjacent Gaussian-blurred images in each octave.

## Estimation of Essential Matrix

If the cameras are calibrated, the essential matrix $\mathbf{E}$ can be computed as $\mathbf{E} = \mathbf{K}^T\mathbf{F}\mathbf{K}$. It encodes the relative pose and is further decomposed to extract [R|t].(for this data, the K values were provided by the professor)

The computed Essential matrix is :

$$\mathbf{F} = \begin{bmatrix} -0.23588422 & 7.39827066 & 3.16451259 \\ -4.55234015 & 0.79158117 & -29.50975636 \\ -2.7545586 & 29.16520175 & 0.4445501 \end{bmatrix}$$

## Extract [R|t] from decomposed Essential matrix with SVD

The essential matrix $\mathbf{E}$ encodes the relative rotation and translation between two calibrated camera views. By applying singular value decomposition (SVD), four possible configurations of $[R|t]$ are obtained. The correct pose is selected using the cheirality condition.

**(Step III) Why Are There Four Camera Poses?** The decomposition of the essential matrix $\mathbf{E}$ into relative camera motion yields four possible configurations of rotation ($\mathbf{R}$) and translation ($\mathbf{t}$). This ambiguity arises from the properties of SVD and the inherent symmetry of the epipolar geometry.

**(Step III) Why Should We Apply W Between U and V for R?** When decomposing the essential matrix using SVD ($\mathbf{E} = \mathbf{U}\Sigma\mathbf{V}^T$), the resulting rotation matrix must be a valid element of the special orthogonal group —meaning it should satisfy the properties of a rotation matrix: orthogonality and $\det(\mathbf{R}) = +1$.

To enforce this, we introduce a predefined matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplying $\mathbf{U}$ and $\mathbf{V}^T$ by $\mathbf{W}$ or $\mathbf{W}^T$ ensures that the resulting matrix $\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^T$ (or $\mathbf{U}\mathbf{W}^T\mathbf{V}^T$) is a proper rotation matrix with $\det(\mathbf{R}) = +1$. Without $\mathbf{W}$, the product $\mathbf{U}\mathbf{V}^T$ may yield an invalid rotation (e.g., reflection with $\det = -1$).

Thus, the insertion of $\mathbf{W}$ corrects the sign and enforces the geometric validity of $\mathbf{R}$ as a physically plausible camera rotation.

Specifically, when $\mathbf{E} = \mathbf{U}\Sigma\mathbf{V}^T$ is decomposed, we can derive two possible rotation matrices using a fixed matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_1 = \mathbf{U}\mathbf{W}\mathbf{V}^T, \quad \mathbf{R}_2 = \mathbf{U}\mathbf{W}^T\mathbf{V}^T$$

For each rotation matrix, the translation vector $\mathbf{t}$ is obtained from the third column of $\mathbf{U}$ (i.e., $\mathbf{t} = \pm\mathbf{U}[:, 3]$). Combining these gives four pose hypotheses:

- $(\mathbf{R}_1, +\mathbf{t})$
- $(\mathbf{R}_1, -\mathbf{t})$
- $(\mathbf{R}_2, +\mathbf{t})$
- $(\mathbf{R}_2, -\mathbf{t})$

All four configurations are mathematically valid, but only one is physically feasible. The cheirality condition is applied to select the correct pose—this condition ensures that the reconstructed 3D points lie in front of both cameras (i.e., positive depth in both camera coordinate systems).

## Perform Triangulation using Cheirality condition

I reconstruct 3D points via triangulation for each $[R|t]$ pair to determine the correct camera pose among the four candidates. The correct pose is selected by applying the cheirality condition, which ensures that the reconstructed points lie in front of both cameras. This is commonly referred to as the depth positivity constraint.

Given projection matrices $\mathbf{P}_1 = [\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}_2 = [\mathbf{R}|\mathbf{t}]$, a 3D point $\mathbf{X}$ is triangulated from corresponding points $(x_1, x_2)$ using:

$$\mathbf{A}\mathbf{X} = 0, \quad \text{where } \mathbf{A} = \begin{bmatrix} x_1 P_1^{(3)} - P_1^{(1)} \\ y_1 P_1^{(3)} - P_1^{(2)} \\ x_2 P_2^{(3)} - P_2^{(1)} \\ y_2 P_2^{(3)} - P_2^{(2)} \end{bmatrix}$$

Then, for each solution $\mathbf{X}$, we verify:

$$Z_1 > 0 \quad \text{and} \quad Z_2 > 0$$

i.e., the 3D point must lie in front of both cameras.

## Estimation of Camera Pose using PnP

After triangulating a set of 3D points from the first two views, we estimate the camera pose of subsequent views using the Perspective-n-Point (PnP) algorithm. Given 2D–3D correspondences and the intrinsic matrix $\mathbf{K}$, we use OpenCV's `solvePnP` function to recover the camera's rotation and translation. This method allows us to incrementally register new views into the global coordinate frame based on already reconstructed 3D landmarks.

To validate this approach, we tested our SfM pipeline on both 2-view and 32-view image sets. As shown in Fig. 4, the reconstruction from only two images results in a sparse and noisy 3D point cloud. However, when applying the same pipeline to all 32 views (Fig. 5), the reconstructed scene becomes significantly denser and more recognizable. This demonstrates the effectiveness of the incremental PnP-based pose estimation in building a coherent structure by progressively adding new views.

**(Step V) What Are the Differences Between Epipolar Geometry and PnP?** Epipolar geometry and the Perspective-n-Point (PnP) problem address two different but related tasks in the context of camera pose estimation.

**Epipolar Geometry:** This describes the intrinsic projective geometry between two views of the same scene. It is governed by the fundamental or essential matrix, which encodes the relative rotation and translation between two cameras. The goal is to find correspondences between two images and use them to recover the camera motion and triangulate 3D points.

Key characteristics:

- Requires 2D–2D correspondences between two views.
- Outputs the relative pose (R, t) and reconstructs 3D points.
- Used in the early stages of SfM to initialize 3D reconstruction.

**PnP (Perspective-n-Point):** In contrast, PnP estimates the pose of a camera given a set of known 3D points and their corresponding 2D image projections. This is typically used once some 3D structure has been established.

Key characteristics:

- Requires 3D–2D correspondences.
- Estimates the absolute pose of a new camera with respect to a known 3D structure.
- Used in the incremental stages of SfM to register additional views.

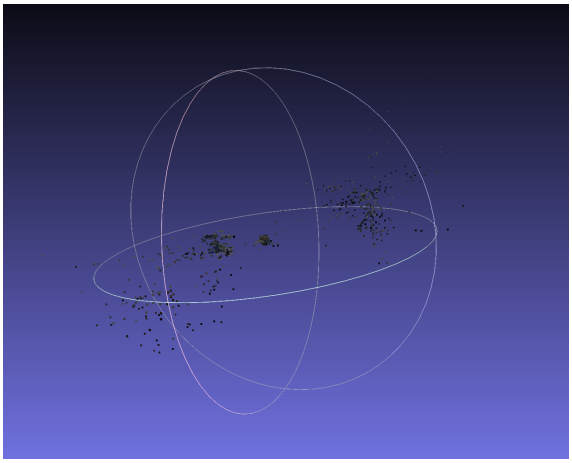

Figure 4: Sparse 3D reconstruction using 2 images.

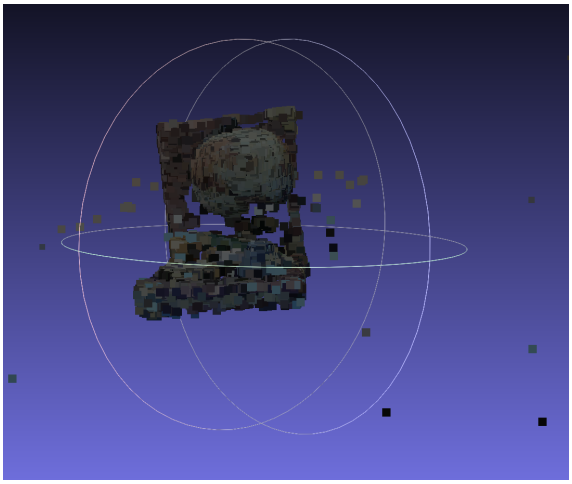

Figure 5: Dense 3D reconstruction using 32 images.

# IDEA: Geometric and Density-Based Outlier Rejection in SfM

## Challenge: Spatial Outliers in SfM Reconstruction

Although our SfM pipeline successfully reconstructs a point cloud from multiple views, a critical challenge remains: many triangulated points are spatially noisy and widely dispersed. As shown in Fig. 6, outliers caused by mismatches, low-parallax baselines, or ill-conditioned triangulation may corrupt the overall structure, making it ambiguous and unreliable for downstream tasks such as meshing or object recognition.

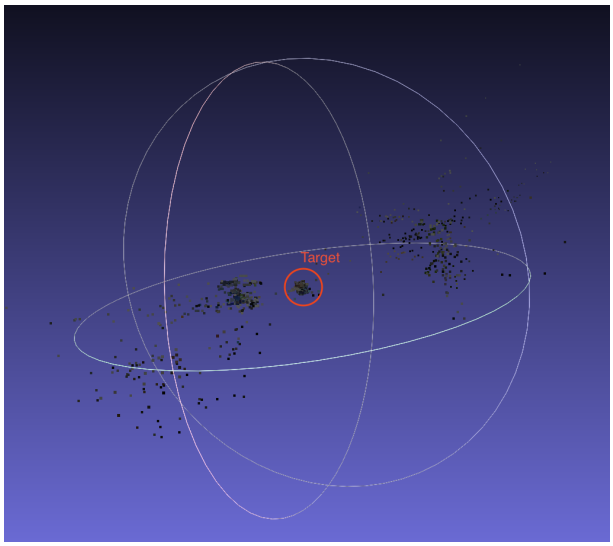


Figure 6: Raw point cloud output from SfM. Many points are scattered far from the actual object, making the shape ambiguous.

## L2-Based Centroid Filtering (Global Constraint)

A straightforward yet effective filtering approach is to exploit the global spatial distribution of reconstructed points using Euclidean distance. Let $\{\mathbf{X}_i \in \mathbb{R}^3\}_{i=1}^N$ be the reconstructed 3D points. We compute their centroid:

$$\bar{\mathbf{X}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{X}_i$$

Then, we retain only the points that are close to this centroid:

$$\mathcal{X}_{\text{inlier}} = \left\{ \mathbf{X}_i \;\middle|\; \|\mathbf{X}_i - \bar{\mathbf{X}}\|_2 < \delta \right\}$$

where $\delta$ is a tunable distance threshold. This acts as a global geometric filter that eliminates spatially distant outliers.

## DBSCAN-Based Local Density Filtering

While L2-based filtering captures global outliers, it does not consider local density, which is often more informative

in real-world scenes. Therefore, we additionally apply DBSCAN (Density-Based Spatial Clustering of Applications with Noise) to cluster 3D points based on local density.

Given a set of 3D points, DBSCAN defines core points that have at least 'min_samples' neighbors within an $\epsilon$-radius:

$$\mathcal{N}_\epsilon(\mathbf{X}_i) = \{\mathbf{X}_j \mid \|\mathbf{X}_i - \mathbf{X}_j\|_2 < \epsilon\}$$

If $|\mathcal{N}_\epsilon(\mathbf{X}_i)| \geq$ min_samples, then $\mathbf{X}_i$ is a core point and included in a cluster. Otherwise, it is considered noise. This allows DBSCAN to effectively identify dense object regions while rejecting isolated noise.

**Why Not K-Means?** We explicitly avoid using K-Means clustering for this task for several reasons:

- K-Means requires the number of clusters $k$ to be predefined, which is difficult to determine for unstructured point clouds.
- It assumes spherical, equally sized clusters, which does not hold for irregular or elongated 3D shapes.
- K-Means is sensitive to outliers, which can distort centroid computation.

DBSCAN, by contrast, is parameterized by density rather than shape or count, and thus fits well with noisy SfM point clouds.

### Implementation in Our Pipeline

In our pipeline:

- We apply DBSCAN with $\epsilon = 0.2$ and `min_samples` = 10 to the final 3D reconstruction.
- Points marked as inliers are saved and colored; outliers are discarded.
- The result is a significantly cleaner and denser point cloud centered on the actual object structure.

### Result: Denoised and Accurate 3D Structure

The effect of applying both global ($L_2$) and local (DBSCAN) filters is shown in Fig. 7. Compared to the raw point cloud, the filtered result is more compact, visually coherent, and better aligned with the true geometry of the object.

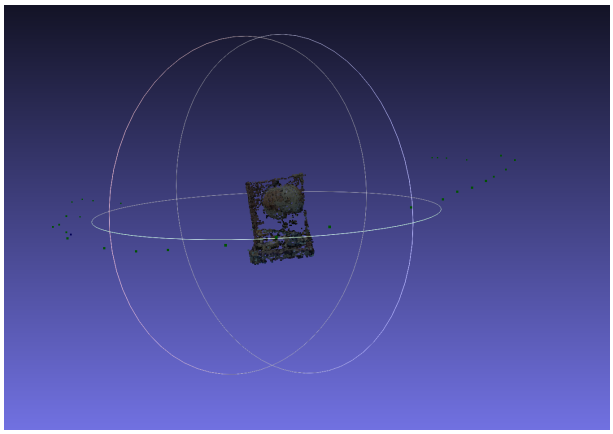

Figure 7: Filtered point cloud after applying $L_2$-based and DBSCAN-based outlier rejection. Most spatial noise has been removed.

## Application to Own Dataset and Learned Limitations

since now, we talking about what is SfM and to remove noise how I approach it. And now we talk about own dataset to apply SfM. After establishing the SfM pipeline and noise filtering strategy, I applied the framework to a series of images captured by my own mobile phone. To do this reliably, intrinsic calibration was necessary.

### Camera Calibration and Dataset Overview

To obtain the intrinsic matrix $\mathbf{K}$ of my phone camera, I conducted a checkerboard calibration using 53 images taken from different angles. This calibration provided the internal camera parameters necessary for precise SfM reconstruction.

Following calibration, I tested the pipeline using eight distinct datasets, including both my own captured images and online resources. The target objects were:

- Salad
- Optical system
- Laptop
- Figure
- Cup
- Stone pagoda
- Sculpture ()
- Beverage bottle

Among these, the sculpture dataset produced the most stable and accurate 3D reconstruction results. Therefore, only the reconstruction results for the sculpture are included in this report.

**(Step VI) What Is the Meaning of Each Element in Intrinsic Matrix K?** The intrinsic matrix $\mathbf{K}$ encodes the internal parameters of a camera that govern the projection of 3D points in the camera coordinate system onto the 2D image plane. It is a $3 \times 3$ matrix typically of the form:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Each element of this matrix has a specific geometric meaning:

- $f_x$, $f_y$: Focal lengths in the x and y directions, measured in pixels. These define how much a unit of distance in the scene is magnified in each direction.
- $c_x$, $c_y$: Coordinates of the principal point (image center) in pixels, indicating where the optical axis intersects the image plane.
- The last row $[0 \quad 0 \quad 1]$ enforces homogeneous coordinates for projective transformation.

Understanding these parameters is essential for accurate 3D reconstruction, camera calibration, and projection operations. After processing to get a K own my Iphone 12 Pro, the result K is :

$$\mathbf{K} = \begin{bmatrix} 1126.8243099308 & 0.0000000000 & 548.8477108217 \\ 0.0000000000 & 1120.2767872161 & 744.4018858846 \\ 0.0000000000 & 0.0000000000 & 1.0000000000 \end{bmatrix}$$

### Importance of SIFT in Practical Applications

From repeated experiments, I found that the success of SfM heavily relies on the quality of SIFT features. Well-localized and consistent SIFT keypoints lead to robust epipolar geometry estimation and thus better triangulation results.

One limitation I discovered is that SfM pipelines strongly depend on the visual richness and feature consistency of the target object. Objects lacking texture or repetitive patterns resulted in sparse or failed reconstructions.

### Attempted Enhancement: SIFT Refinement via Bounding Box Filtering

To overcome inconsistent SIFT detection, I attempted an object-detection-based preprocessing strategy: detect the object first and apply SIFT only within the bounding box (BBOX). However, this approach introduced significant noise and unstable matches, likely due to reduced spatial context or background texture being misclassified as valid features.

### Conclusion: SIFT-Compatibility of Target Objects Matters

From these findings, I conclude that the inherent SIFT-detectability of a target object is a critical factor in SfM performance. The more robust and repeatable the SIFT features are, the better the reconstruction. Thus, careful image acquisition—such as controlling lighting, avoiding blur, and choosing feature-rich objects—is crucial for real-world SfM applications.

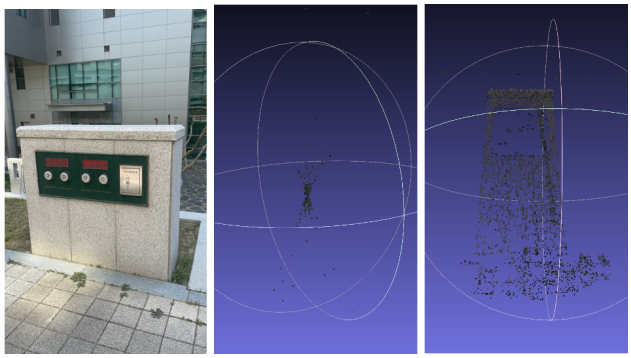

Figure 8: (Left) Input image. (Middle) 2-view SfM reconstruction. (Right) 3-view SfM reconstruction.

## Conclusion

This report explored a complete Structure from Motion pipeline, including traditional two-view geometry, noise filtering using both global and local strategies, and application to real-world data. Through experiments with my own dataset, I discovered practical limitations of SfM—particularly its reliance on good feature extraction. This insight emphasizes the importance of camera calibration, careful image capture, and scene selection when applying SfM in uncontrolled environments.

## References


[1] Byeol's SfM GitHub Repository. https://github.com/byeol3325/Structure-from-motion

[2] Naitri SFM Github Repositry https://github.com/naitri/SFM

[3] Woochan's Blog on SfM and Reconstruction. https://woochan-autobiography.tistory.com/944