# 3D Computer Vision

## Homework #2: Two-View Geometry and Epipolar Geometry

November 15, 2025

Amirkabir University of Technology
Computer Engineering Department

**LLM Usage Policy Notice**
You may use AI tools for assistance, but you must be able to explain every line of your code and may be asked to implement parts from scratch in a face-to-face meeting with the TAs.

---

## Objective

Implement a complete two-view geometry pipeline that establishes correspondences between two images, estimates the fundamental matrix, visualizes epipolar geometry, recovers camera motion, and triangulates 3D points. You will work with a stereo pair of images to understand epipolar constraints, fundamental and essential matrices, and basic stereo reconstruction.

## Data and Tools

**Required Dataset:** You must use a stereo image pair from the Middlebury Stereo 2014 dataset (`https://vision.middlebury.edu/stereo/data/scenes2014/`). This dataset provides high-quality stereo pairs with ground truth disparities and calibration information. Choose one of the training datasets (with ground truth available) or test datasets. Each dataset includes:

- Left and right views (`im0.png` and `im1.png`)
- Calibration file (`calib.txt`) containing camera intrinsics and baseline information
- Ground truth disparity maps (`disp0.pfm` and `disp1.pfm`) for training sets

**Example Input:** Figure 1 shows an example stereo pair that will be used throughout this assignment to demonstrate the two-view geometry pipeline.

Figure 1: Example stereo image pair for two-view geometry analysis

Implement in Python with OpenCV/NumPy. You will need to use feature detectors (e.g., SIFT), feature matchers, and robust estimation methods (e.g., RANSAC). The calibration file format is described in the dataset documentation and can be parsed to extract camera intrinsics $K$ and baseline information needed for essential matrix recovery and triangulation.

1. **Image Pair Selection and Feature Detection**
   Select a stereo image pair from the Middlebury Stereo 2014 dataset. Use SIFT (Scale-Invariant Feature Transform) to detect keypoints and compute descriptors for both images. Visualize the detected keypoints overlaid on both images. Discuss the distribution of keypoints across the images.

   **Note:** Refer to the appendix for conceptual explanations of SIFT feature detection and matching.

2. **Feature Matching and Correspondence**
   Match features between the two images using a suitable distance metric (e.g., L2 norm for SIFT descriptors). Use ratio test (e.g., Lowe's ratio test with threshold 0.7-0.8) to filter out ambiguous matches. Visualize the initial matches and the filtered matches. Report the number of matches before and after filtering.

   Then, apply RANSAC (Random Sample Consensus) to robustly estimate the fundamental matrix and remove outliers from your correspondences. Visualize the inlier matches after RANSAC filtering. Report the number of inliers and outliers, and the percentage of matches that survived RANSAC.

   **Note:** Refer to the appendix for conceptual explanations of RANSAC robust estimation.

3. **Fundamental Matrix Estimation and Epipolar Geometry**
   Estimate the fundamental matrix $F$ from the inlier correspondences using the normalized 8-point algorithm. You can use OpenCV's implementation, but you must explain the method:

- Form the homogeneous system from point correspondences $(x, x')$: $x'^{\top} F x = 0$
- Solve the linear system using SVD
- Enforce the rank-2 constraint on $F$

Report the estimated fundamental matrix $F$. Verify the epipolar constraint: for several correspondences, compute $x'^{\top} F x$ and show that it is close to zero for inlier pairs.

Visualize epipolar geometry:

- For 5-10 selected points in the first image, draw their corresponding epipolar lines in the second image
- Show the epipoles in both images (where the epipolar lines converge)
- Overlay epipolar lines and epipoles on both images

Discuss what the epipolar constraint means geometrically and how it relates to camera geometry.

4. **Essential Matrix and Camera Motion Recovery**
Extract camera intrinsics $K_1$ and $K_2$ from the `calib.txt` file provided with your Middlebury dataset. The calibration file contains camera matrices in the format `cam0` and `cam1`. Extract the focal length $f$ and principal points $(c_x, c_y)$ to construct $K$. For the Middlebury dataset, you can typically assume $K_1 = K_2 = K$ (same camera intrinsics for both views). Recover the essential matrix:

$$E = K_2^{\top} F K_1$$

Decompose the essential matrix $E$ to extract the relative camera motion. Use SVD: $E = U\text{diag}(1, 1, 0)V^{\top}$. The rotation $R$ and translation $t$ can be extracted, giving four possible solutions. Use the cheirality constraint (positive depth) to disambiguate and select the correct solution.

Report:

- The essential matrix $E$
- The relative rotation $R$ (as a rotation matrix and optionally as axis-angle or Euler angles)
- The relative translation $t$ (normalized, up to scale)
- Which of the four solutions was selected and why

5. **Triangulation and 3D Reconstruction**
Given the camera matrices $P_1 = K[I|0]$ and $P_2 = K[R|t]$, triangulate the 3D points from the inlier correspondences using linear triangulation or the midpoint method.

For linear triangulation, for each correspondence $(x_1, x_2)$:

- Form the system: $x_1 \times P_1 X = 0$ and $x_2 \times P_2 X = 0$
- Solve using SVD to find the 3D point $X$

Visualize the reconstructed 3D points in a 3D plot. Color-code the points (e.g., by depth or match confidence). Project the 3D points back to both images and visualize the reprojections. Report the mean reprojection error.

Discuss the limitations: What causes ambiguity in depth? Why is translation only recovered up to scale? How does the baseline (distance between cameras) affect reconstruction quality?

# Submission

Submit a concise PDF report (3–5 pages) containing: a description of the methods and equations you used, clear figures for each stage (detected keypoints, matches before/after RANSAC, epipolar lines and epipoles, 3D reconstruction), numerical results (fundamental matrix, essential matrix, rotation, translation, reprojection errors), and thorough discussion. Provide runnable code (Python + OpenCV/NumPy) and a script to reproduce all figures.

# A Appendix: Feature Detection and Matching Concepts

## A.1 Scale-Invariant Feature Transform (SIFT)

SIFT detects distinctive keypoints in images and describes each keypoint with a 128-dimensional vector that captures the local image appearance around that point. These descriptor vectors are designed to be invariant to scale, rotation, and illumination changes, making them robust for matching across different views of the same scene.

To match features between two images, the descriptor vectors from both images are compared using a distance metric (typically L2 norm). For each keypoint in the first image, we find the closest matching keypoint in the second image based on descriptor similarity. Lowe's ratio test is then applied to filter ambiguous matches: if the distance to the best match is not significantly smaller than the distance to the second-best match, the correspondence is considered unreliable and discarded. The remaining matches provide point correspondences needed for estimating the fundamental matrix in two-view geometry.

Figure 2 demonstrates SIFT keypoint detection on our example stereo pair. Keypoints are shown as circles with their scale and orientation indicated. Figure 3 shows the complete feature matching pipeline: (a) initial matches after descriptor comparison, (b) matches after applying Lowe's ratio test to filter ambiguous correspondences, and (c) final inlier matches after RANSAC filtering.

**Important observation:** In this stereo pair, the camera mainly moved horizontally. This means that correct correspondences should appear on the same horizontal scanline (same y-coordinate) in both images. Any matching line that is not horizontal indicates an incorrect correspondence. This geometric constraint is why RANSAC is particularly effective here—it can identify and remove matches that violate the epipolar constraint, which in this case corresponds to matches that are not horizontally aligned.

Figure 2: SIFT keypoint detection on the stereo pair

## A.2 Robust Estimation with RANSAC

Feature matching inevitably produces incorrect correspondences (outliers) due to repetitive textures, occlusions, or similar-looking regions. RANSAC (Random Sample Consensus) is a robust estimation method that can fit a model to data even when many outliers are present.

RANSAC works by repeatedly: (1) randomly selecting a small subset of correspondences, (2) fitting a model (e.g., fundamental matrix) to this subset, and (3) counting how many other correspondences are consistent with this model (inliers). The model with the most inliers is kept. For fundamental matrix estimation, correspondences are considered inliers if they satisfy the epipolar constraint within a threshold, meaning the points lie close to their corresponding epipolar lines. This process effectively separates correct matches from incorrect ones, allowing robust estimation of the fundamental matrix even when many feature matches are wrong.

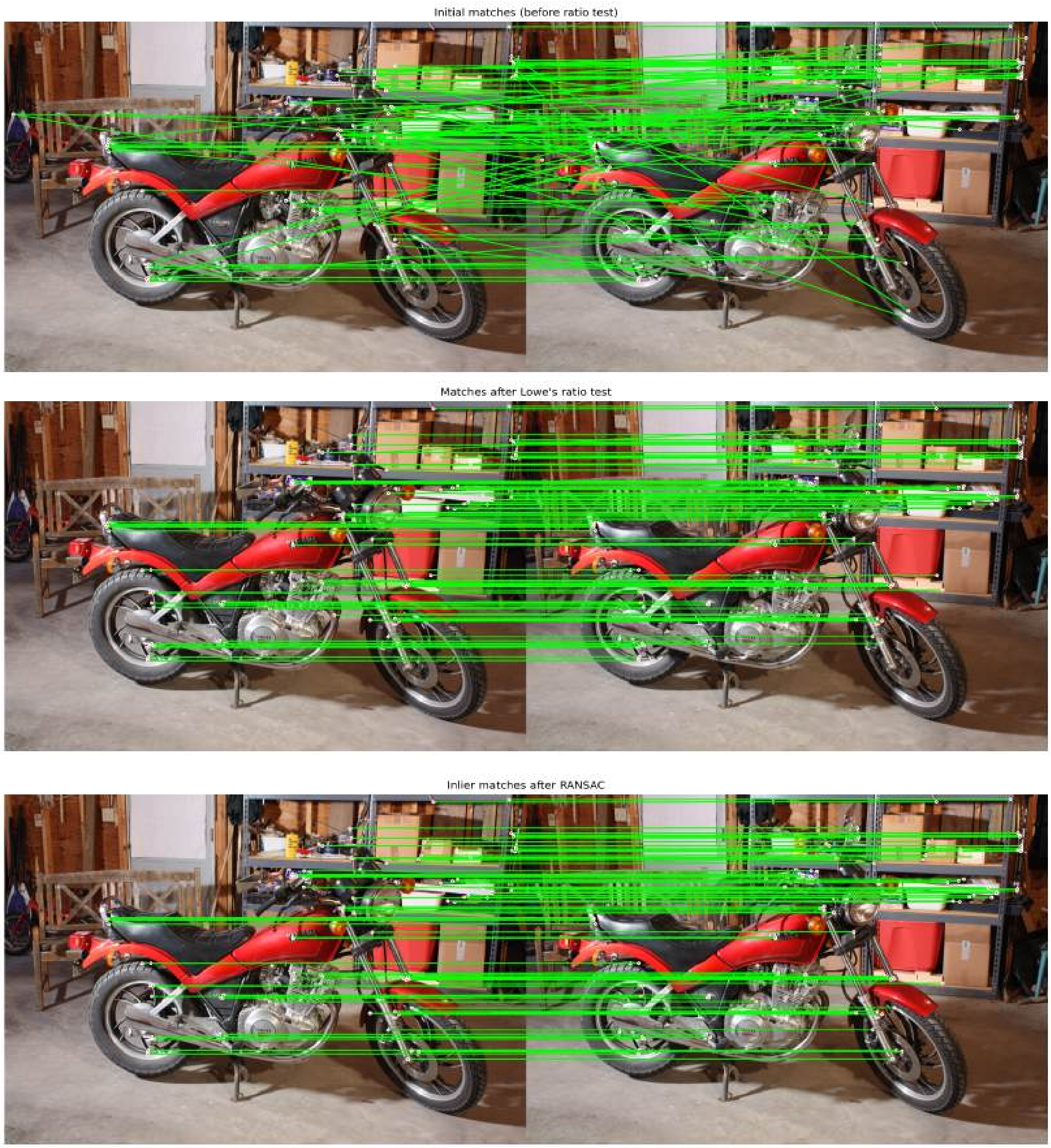


Figure 3: Feature matching pipeline: (a) initial matches, (b) matches after Lowe's ratio test, (c) final inlier matches after RANSAC filtering