Computer Vision

Homework 1 & 2

Francesco Setti

Fall 2025

General Objective

In this homework, you will:

1. **Calibrate a real camera**, using checkerboard images acquired by yourself.
2. Use the calibration to **reconstruct a 3D object** from two images of the same scene, via:
   - feature extraction and matching,
   - estimation of the epipolar geometry,
   - triangulation of the corresponding points.

Part 0 – Data Acquisition (Mandatory)

0.1 Acquisition of calibration images (checkerboard)

You must **acquire your own images** for calibration, using:

- a smartphone, a webcam, or a digital camera;
- always the **same camera** for all images.

Requirements:

- Use a **printed checkerboard** (or a known planar pattern) with:
  - square cells of known size (e.g., 20–30 mm per side; report the exact value in your report),
  - at least a 6×7 grid of internal corners (or similar).
- Acquire **at least 10 images** of the checkerboard:
  - with **different poses** (rotations, translations, tilts),

- covering different image regions (center, borders, corners),
- at varying distances (near/far),
- avoiding strong reflections and excessive blur.

- Save the images in a common format (e.g., PNG or JPEG).

0.2 Choice and Acquisition of the Object / Scene to Reconstruct

You must also choose **an object or a small rigid scene** to reconstruct (examples: a box, a book, a small tabletop scene, a chair, etc.).
Requirements:

- The object should:
  - have sufficient **texture / visual detail** (avoid perfectly uniform or highly reflective surfaces),
  - remain **rigid and static** between the two acquisitions.
- Acquire **at least two images** of the same scene:
  - using the **same camera** employed for calibration,
  - with a **non-zero baseline** (change the camera position, not just a very small rotation),
  - keeping the object fully visible in both views.

You may use more than two images, but the homework requires at least the reconstruction from **one pair of views**.

Part 1 – Camera Calibration

1.1 Pattern Detection

- Automatically detect the **checkerboard corners** (or pattern points) in each calibration image.
- Assign to each image point its corresponding 3D coordinate in the pattern reference frame (e.g., plane $z = 0$, coordinates in millimeters).

1.2 Estimation of Calibration Parameters

- Using the opencv calibration toolbox, estimate:
  - the camera intrinsic matrix $\mathbf{K}$,
  - distortion coefficients (radial / tangential),
  - for each image: rotation $\mathbf{R}$ and translation $\mathbf{t}$ (extrinsic parameters).
- Compute the **mean reprojection error** (in pixels) over the calibration images.

Part 2 – 3D Reconstruction from Two Views

2.1 Feature Extraction and Matching

- Detect local features on the two images of your scene (e.g., SIFT, ORB, or other justified choice).
- Compute descriptors and find matches between the two views.
- Apply a match filtering strategy (e.g., ratio test, cross-check, preliminary RANSAC) to obtain a set of reliable correspondences.

2.2 Epipolar Geometry Estimation

- From the filtered matches, estimate the **fundamental matrix** F (e.g., using RANSAC).
- Using the intrinsic matrix K estimated in Part 1, compute the **essential matrix** E. (You can also directly compute matrix E if you prefer)
- Decompose E into the possible pairs $(\mathrm{R}, \mathbf{t})$ describing the **relative pose** of the second camera with respect to the first.
- Select the correct solution using the condition that the triangulated 3D points must lie **in front of both cameras** (cheirality).

2.3 Triangulation and 3D Reconstruction

- Construct the **projection matrices** P and P' for the two views, using K, R, and $\mathbf{t}$.
- Triangulate a subset of inlier matches to obtain 3D coordinates of scene points.
- Visualize the reconstructed 3D points (e.g., with a 3D scatter plot) and comment on:
  - spatial distribution,
  - presence of clearly wrong points (outliers),
  - qualitative correspondence with the real object/scene.

Submission Instructions

Each student must submit:

1. **Source code** in Python with clear comments (possibly in the form of a Jupyter notebook)
2. **Images** with correct relative paths used in the Python code.

3. A short **report** (approximately 6–8 pages in standard latex article format) including:
   - a brief description of the acquisition setup:
     - type of camera (smartphone, webcam, etc.),
     - checkerboard cell size,
     - chosen object/scene for reconstruction;
   - calibration results:
     - estimated intrinsic parameters: focal lengths, principal point, and (if present) skew.
     - distortion coefficients (if estimated).
     - mean reprojection error and standard deviation.
     - one or more images showing the detected checkerboard corners and the reprojected points using the estimated parameters (to visually assess calibration quality).
   - concise description of the feature matching and F/E estimation pipeline;
   - relative pose results $(\mathbf{R}, \mathbf{t})$ and visualization of the epipolar geometry;
   - 3D reconstruction visualization and qualitative discussion:
     - Number of detected features and number of matches, before and after filtering.
     - Estimated fundamental matrix F and essential matrix E.
     - Relative pose parameters $(\mathrm{R}, \mathbf{t})$ between the two views.
     - Visualization of the epipolar geometry (e.g., some epipolar lines in the second image corresponding to selected points in the first image)
     - Visualization of the reconstructed 3D point cloud (e.g., screenshot of the 3D figure).
     - Optional: a quantitative reconstruction error measure (e.g., back-projection error of 3D points into the two images).
   - any difficulties encountered (e.g., pattern not detected in some images, low number of matches, baseline too small, etc.).