LE/ESSE 2220 Algorithmic and Computational Methods

Lab 9: Homography & RANSAC

(Fall 2025-2026)

Z. ARJMANDI (ZAHRARJ@YORKU.CA)

YORK U

Lab 8: Feature Matching in Satellite Images

BFMatcher
and
FLANN

YORK U

Review

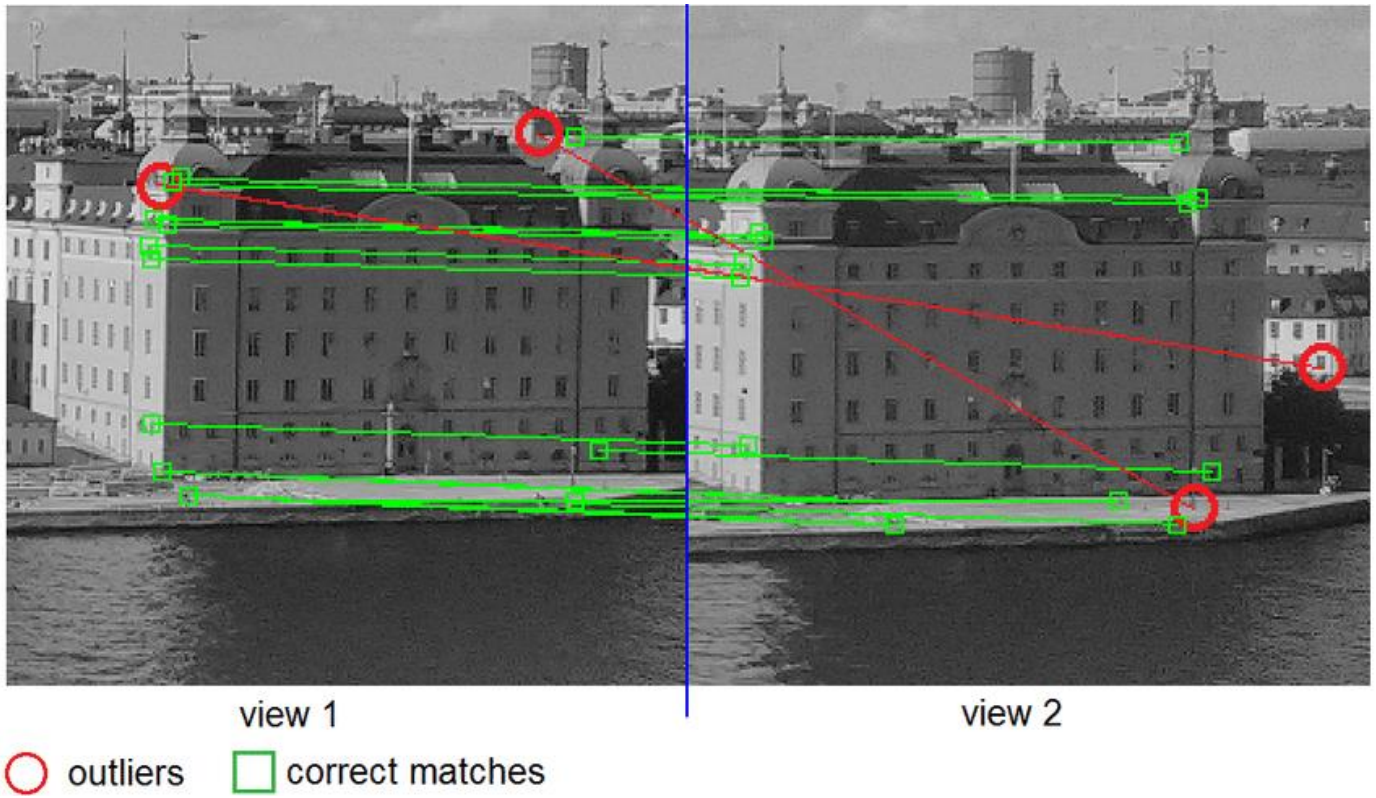
Main Filtering Methods Overview

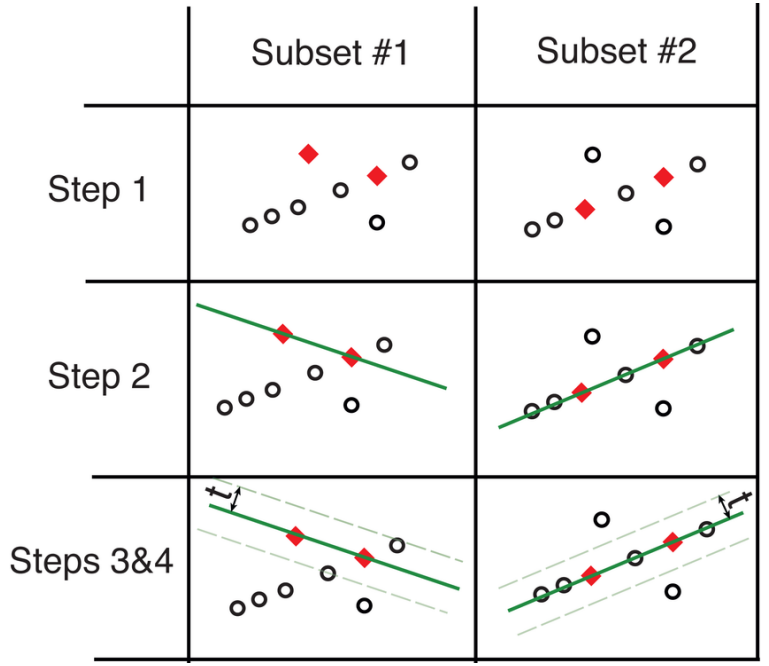
Category	Method Name	Idea / What It Does
A. Feature Filtering (before matching)	Size / Response Threshold	Keep only strong, stable keypoints (kp.response or kp.size)
	Spatial Constraints	Keep keypoints within certain ROI or coordinates
B. Match Filtering (after matching)	Distance Threshold	Keep matches with distance < certain value
	Lowe's Ratio Test	Accept match only if the best match is much better than the 2nd-best
	Symmetry Check	Keep matches that agree in both directions (A→B and B→A)
	RANSAC Filtering	Keep matches consistent with a geometric model (e.g., homography)

B4: RANSAC Filtering

Why It's Useful

- Removes mismatches caused by noise, clutter, duplicate features, and repetitive textures.
- Keeps only feature pairs that make **sense geometrically** (same rotation, scale, perspective).
- Essential for: image stitching, panorama creation, AR, camera pose estimation.





Homography

All Transformations Represented by a Homography (H – 3×3 Matrix)

- **1) Translation** $\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}$ Shifts image in x and y.
- **2) Rotation (in the image plane)** $\begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$ Standard 2D rotation.
- **3) Scaling** $\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$ Zoom in or out.
- **4) Shear (Skew)** $\begin{pmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ Tilts the axes.
- **5) Affine Transformation** $\begin{pmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix}$ All previous transformations *combined*, but without perspective distortion.
- **6) Pure Projective (Perspective)** $\begin{pmatrix} - & - & - \\ - & - & - \\ h_{31} & h_{32} & 1 \end{pmatrix}$ perspective distortion

A **homography** is a **2D projective transformation**.

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}$$

YORK U

Lab 9

Steps

Your code must be written by you. Submissions that are fully or mostly generated by AI will not be marked.

- **1. Load Images + Detect Features**
  - Load both images in grayscale.
  - Detect keypoints and descriptors (SIFT or ORB).
- **2. Create Raw Matches**
  - Use a matcher (BFMatcher, no ratio test).

- **3. Pipeline A: No Location Filter**
  - **A1 Distance Filter**
    - Remove matches using distance threshold
    - Save filtered-match image.
  - **A2. RANSAC**
    - Compute the homography using RANSAC.
    - Save **inliers only** and **outliers only** match images.
    - Record the homography matrix H_A.

- **4. Pipeline B: With Location Filter**
  - **Similar to A but add one location filter for <u>keypoints</u>**
    - Remove all keypoints that fall on the **right and left side of the image**, so only the cathedral keypoints remain.
    - Choose an appropriate cutoff value for Image 1 and a separate cutoff value for Image 2.

YORK U

Questions

- 1. Compare the Two Homographies
  - a) Copy Both **Homography Matrices** (1 mark)
  - b) Compare $\boldsymbol{H_A}$ (no location filter) and $\boldsymbol{H_B}$ (with location filter). (3 mark)
    - **Translation terms**: which homography shifts the image more? Why?
    - **Perspective terms**: which one shows less projective distortion? Why?
    - Comment on what the remaining entries indicate
  - c) How removing keypoints (left and right) changed the match quality and RANSAC inlier count. (1 mark)
  - d) Which pipeline produces a homography that better reflects the **actual camera motion** for the main building? (1 mark)

- 2. For each pipeline (A and B), report: (2 mark)
  - Number of **raw matches** (after removing keypoints for B)
  - Number of **matches after the distance filter**
  - Number of **RANSAC inliers**
  - Number of **RANSAC outliers**
  - **RANSAC Inlier ratio** = inliers / total filtered matches

Appendix Requirements

- 3. In the Appendix of your report, you must include: (3 mark)

  a) Images to Include (8 Total)

  - Pipeline A:
    - **A_raw_matches,** all raw matches
    - **A_filtered_matches,** after distance filter
    - **A_inliers,** RANSAC inliers only
    - **A_outliers,** RANSAC outliers only
  - Pipeline B:
    - **B_raw_matches,** after removing left/right-side keypoints
    - **B_filtered_matches,** after distance filter
    - **B_inliers,** RANSAC inliers only
    - **B_outliers,** RANSAC outliers only

  b) Your complete Python code, exactly as used to generate the results.

  - **Do not delete any sections when switching methods.**
  - Simply **comment out** the parts you are not running.