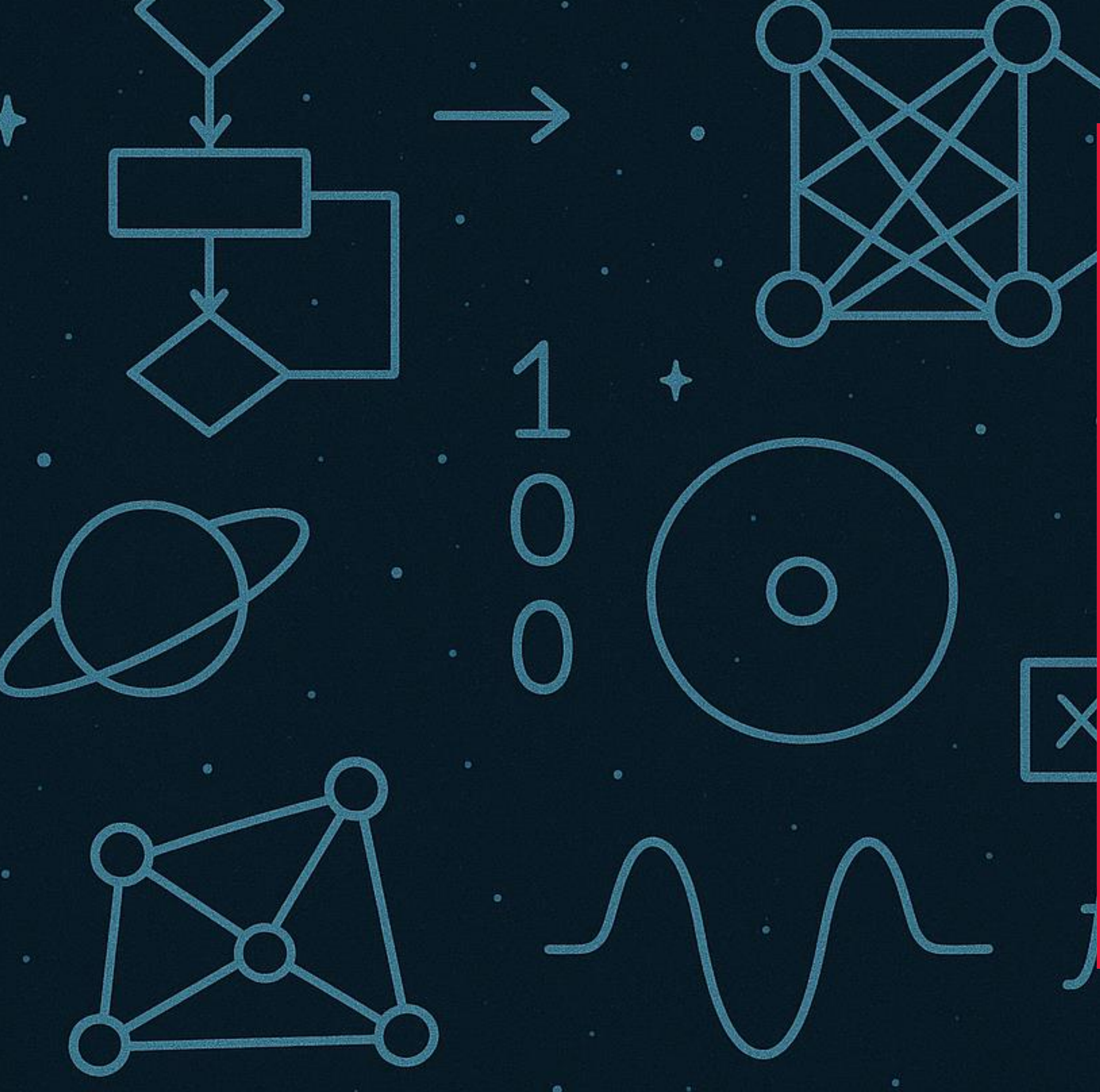


LE/ESSE 2220 Algorithmic and Computational Methods

Lab 8: Feature Matching in Satellite Images (BFMatcher & FLANN)

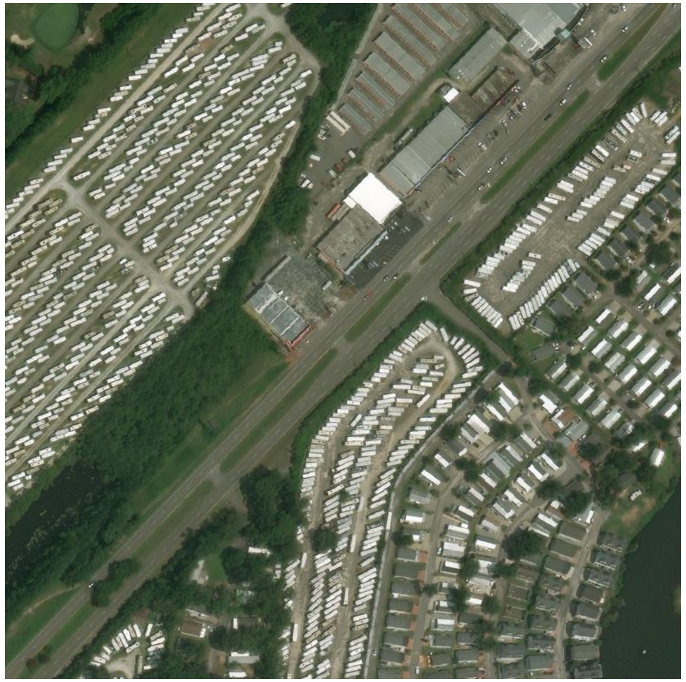
(Fall 2025-2026)

Z. ARJMANDI (ZAHRARJ@YORKU.CA)

YORK U

Lab 8: Feature Matching in Satellite Images

BFMatcher
and
FLANN



Dataset Background: xView2 (Building Damage Assessment)

- **What is xView2?**
  - A large satellite imagery dataset created to assess building damage after natural disasters.
- **What it contains:**
  - High-resolution **before/after** satellite images
- **Why it matters:**
  - Supports research in disaster response, change detection, and automated damage assessment.
- **Connection to this lab:**
  - You will use satellite image pairs from xView2 to practice keypoint detection and feature matching between pre- and post-event scenes.

Review

OpenCV Matchers

- **What All Feature Matching Algorithms Do:** Compare feature descriptors to find the most similar points between images.

- **1- BFMatcher (B**rute-**F**orce**)**
  - Compares every descriptor in one image with every descriptor in another image to find the best match.
  - It's simple but computationally expensive
  - Ideal for small or medium images.

- **2- FLANN-Based Matcher** (**F**ast **L**ibrary for **A**pproximate **N**earest **N**eighbors)
  - Designed for **large datasets** and **high-dimensional descriptors** (like SIFT or SURF).
  - Uses **approximate nearest neighbor search,** much faster than brute force
    - uses algorithms like:
      - **KD-Trees** (for float descriptors like SIFT, SURF)
      - **LSH (L**ocality **S**ensitive **H**ashing**)** (for binary descriptors like ORB)
  - Especially useful for matching thousands of descriptors efficiently.

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)

Lab 8

Lab 7: Steps

- **What to do (right now):**
  - Download the **starter Python code** and **your assigned satellite image pair** from eClass.
    - (Pick the pair that corresponds to the **last digit of your student number**.)
  - Open the script and begin filling out every TODO **in order**.
- **How to work (step-by-step):**
  - Copy/paste **one block at a time** so your code remains runnable at all times.
  - When applying a matching method or filter, **uncomment only one option**, run it, and save the resulting output image.
  - Verify that each saved image (raw BF, raw FLANN, distance filter, ratio test, symmetry, etc.) appears correctly in your folder.
- Use the **official OpenCV documentation**
- Repeat the full process **twice**:
  - Once with **SIFT**
  - Once with **ORB**

Report Format (short, personal, verifiable)

- 1. Dataset Pair (**1 mark)**
  - a) State which satellite pair you downloaded (based on your student number).
  - b) Include the two original images in your report (“before” and “after”).
  - c) In one sentence, describe what kind of scene it shows (urban, rural, coastal, etc.).

- 2. Summary Table of All Methods **(2 marks)**
  - For each row, fill in the exact number of matches your code produced.
  - “Observation” is for 1–2 comments about the quality of the matches (e.g., noisy, clean, too few, many false matches, etc.).

- 3. Questions About Your Table **(2 marks)**

  Answer the following based on your completed table:
  - a) Which combination produced the largest number of raw matches? Why?
  - b) Which filtering method removed the most mismatches?
  - c) Which method (BF or FLANN) gave more stable results between your two satellite images?
  - d) Which detector (SIFT or ORB) worked better for your pair?

Matcher	Detector	Raw Matches	Distance Filter	Ratio Test	Symmetry	Observation
BF	SIFT					
BF	ORB					
FLANN	SIFT				N/A	
FLANN	ORB				N/A	

- 4. Feature Extraction (SIFT and ORB) – Timing + Descriptors **(1 marks)**
  - a) Include extraction time for each image and detector.
  - b) In one sentence, explain why ORB is typically faster than SIFT.

- 7. Filtering Methods – BFMatcher **(1.5 marks)**
  - Per filter, describe what type of mismatches it removes.

- 8. Filtering Methods – FLANN **(2.5 marks)**
  - Explain in one sentence why FLANN might produce slightly different results each run.
  - Per filter, describe what type of mismatches it removes.

Appendix Requirements

- In the Appendix of your report, you must include:

1. **All 14 required output images**, each with a **proper title**:
   - SIFT: raw_bf, raw_flann, bf_distance, bf_ratio, bf_symmetry, flann_distance, flann_ratio
   - ORB: raw_bf, raw_flann, bf_distance, bf_ratio, bf_symmetry, flann_distance, flann_ratio
2. **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.