CS 5970/6970
Fall 2025
Assignment 2 – Feature Manipulation
Due: 10/06/2025 11:59 pm

This assignment builds on your understanding of geometric transformations and introduces you to real-world applications of feature matching and homography. You will use concepts from Lecture 6 and beyond, such as keypoint detection, perspective transformations, and image warping, to solve a visual puzzle automatically. This is where geometry meets computer vision magic!

Task 1: Localize a Hidden Object in the Scene [25 Points]

Your goal is to automatically find a hidden object (e.g., an image of Samford Hall) in a real-world scene using feature matching. Then, using geometric reasoning, you will highlight where the object appears in the scene.

Instructions:

1. Download the provided images (`Part1_scene.png` and `auburn.jpeg`)
2. Use the following techniques:
    a. SIFT keypoint detection and descriptor extraction (via OpenCV)
    b. Keypoint matching using Brute-Force or FLANN matcher
    c. Lowe's ratio test to filter noisy matches
    d. Homography estimation using RANSAC
    e. Perspective transformation to map the template's bounding box into the scene
    f. Draw the result:
        i. Overlay a bounding box (4-point polygon) on the scene image where the template appears
3. Hint: To estimate the homography matrix H, you need a set of matched keypoints between the template and scene image, define the 4 corners of the template image, and project them into the scene.

Task 2: Geometric Puzzle Solver v2 **[25 Points]**

In the previous assignment, you manually tested combinations of transformations (rotation, flipping) to piece together the correct image. This time, you will automatically identify the correct orientation and location of each tile by matching it to the original image using SIFT features and homography.

Instructions:

1. Download the following images:
    a. puzzle_scrambled.png: A 256×256 image made up of 16 scrambled 64×64 tiles
    b. puzzle_reference.png: The original (solved) version of the same image for comparison
2. Write a script to:
    a. Reconstruct the scrambled image by identifying and applying the correct geometric transformation to each tile so that it aligns with its matching position in the reference image.
3. Steps to implement:
    a. Split the scrambled image into 64×64 tiles
    b. For each tile, extract SIFT features and find its best matching location in the reference image using feature matching and homography estimation.
    c. Use the homography to warp each tile to the correct location in the reference image.
    d. Combine the warped tiles to reconstruct the solved image.
4. Hints:
    a. You can loop over each tile and match it against the entire reference image.
    b. Use cv2.perspectiveTransform() to map the tile corners using the homography.
    c. You do not need to manually test all flips/rotations — SIFT is rotation and scale invariant.

Deliverables:

1. An IPYNB notebook that can run on Google Colab.
    a. You can use the same notebook for each part. Make sure to label them correctly with a description using a text cell.
2. Your code that you used to uncover the message
    a. Provide proper documentation on how to run your code, any dependencies, etc. as a text cell *before* your code cell.
3. A description of your approach, what worked and what didn't and GenAI usage declaration, and a statement of contribution for each member in your team. If it is an individual project, then just say "This was an individual project and I did everything in it."