Assignment 3

COS 470/570: Image Processing and Computer Vision

Objective:

The goal of this assignment is to apply and gain a deep understanding of feature matching algorithms such as SIFT and ORB for keypoint detection and matching between image pairs. This task also includes an exploration of feature matching's theoretical applications in depth estimation from images.

Tasks:

First, download "pairs.zip" from BrightSpace. Then, you have two tasks:

Task 1: Feature Detection and Matching (70 points).

Use any keypoint detection algorithms (SIFT, ORB) to detect and match keypoints between several pairs of images. You are provided with initial images (see "pairs.zip") where you will apply these algorithms.

1. **Keypoint Detection**: Select any keypoint detection algorithm (SIFT, ORB) to detect keypoints in each image from the provided "pairs.zip". Fine-tune your algorithm's parameters to enhance the accuracy of the matches.
   For example, if you use SIFT as the feature detector, then you should be able to tune the parameters including nfeatures, nOctaveLayers, contrastThreshold, edgeThreshold and sigma. You can get more detailed information about these parameters from this link https://docs.opencv.org/4.x/d7/d60/classcv_1_1SIFT.html. If you decide to use the ORB as the feature detector, read this link before you start to code: https://docs.opencv.org/4.x/d1/d89/tutorial_py_orb.html
2. **Keypoint Matching:** Use FLANN or BFMatcher to match keypoints between each pair of images. Read this link (What is feature matcher? https://docs.opencv.org/3.4/dc/dc3/tutorial_py_matcher.html) before coding.
3. **Visualization:** Draw lines connecting the matched keypoints across the image pairs (You may need this function **cv2.drawMatches()**).
4. **Evaluation of Matches:**
   - Good Matches Definition: A match is considered 'good' if the Nearest Neighbor Distance Ratio (NNDR) is below a preset threshold (e.g., 0.8). This ratio compares the distance to the nearest neighbor with that of the second nearest neighbor, effectively filtering out weaker matches.

You can find the code example to calculate the “good match” from this link: https://docs.opencv.org/3.4/dc/dc3/tutorial_py_matcher.html

- Calculation: Report the number of good and bad matches and determine the accuracy rate for each image pair based on these counts. Describe how parameter tuning affected the results.

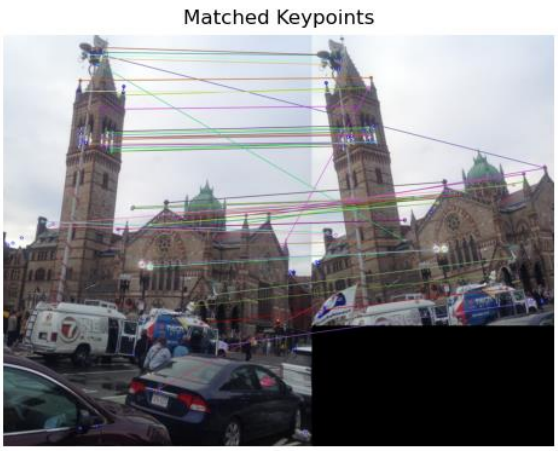
Your report should include visual outputs showing keypoints and their matches between image pairs, statistical analysis and discussion on the parameter tuning and its impact on the results.

Bonus opportunities: Implementing multiple combinations of keypoint detection and matching techniques can earn additional points. For example, using SIFT with BFMatcher earns base points, while adding ORB with BFMatcher with parameter adjustments earns bonus points (+10 for each additional solution).

Example for this task:

I provided an example as a reference, you can follow this to write the report:

I worked on the “Boston” image pairs as an example (you need to work on all these 4 pairs), my solution is SIFT+FLANN matcher. I got the following results:



Configurations	Total matches	Good matches	Bad matches	Accuracy
nfeatures=100, nOctaveLayers=4, contrastThreshold=0.06, edgeThreshold=15, sigma=1.9 NNDR = 0.8	101	67	34	66.34%

I did not fine-tune the parameters, so I believe you can easily beat my performance. Good luck!

Task 2: Application of Feature Matching for Depth Estimation (30 points).

Learning Objective: Understand how matched features can be used to solve a real-world problem: calculating depth information from images. This theoretical understanding will form the basis for practical applications in the field.

Read the provided materials on how feature matching can be utilized to calculate depth information from images. Summarize the theoretical steps involved in using matched features to estimate distances.

1. Study the provided reading materials (4 parts, read them all): https://docs.opencv.org/4.x/d9/db7/tutorial_py_table_of_contents_calib3d.html
2. Summarize the key concepts and methodologies discussed in the materials regarding the calculation of depth from matched features.
3. Explain the theoretical basis and the step-by-step process of how depth information can be inferred from feature matches.

Your summary should clearly articulate the connection between feature matching and depth estimation.

Submission:

Submit your Python scripts along with a report detailing your code and approach. Ensure your report provides enough evidence (screenshots) to prove that your code is functional. The deadline of this assignment is October 16 at 11:59 PM.