`

Objective :

To Implement a feature detection on any synthetic dataset or any open source dataset (Middlebury) of stereo images. Implement any feature matching mechanism and integrate that to filter matches based on Lowe’s ratio test. Finally, estimate the camera pose and compare with the ground truth data.

Also, Calculate error metrics to evaluate the accuracy of the pose estimation and present your thoughts on how we can improve the camera pose estimation.

Dataset Overview :

Chess stereo images from Middlebury Dataset

Image 1

Image 2

`

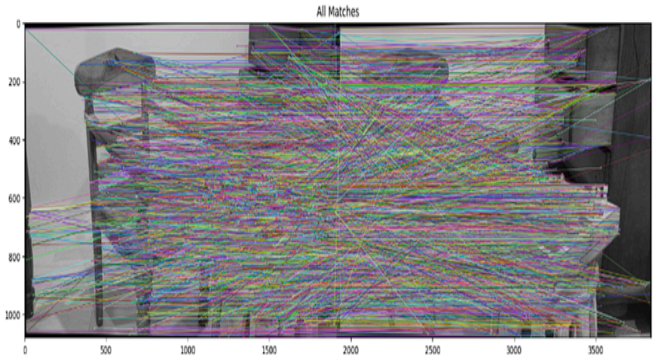
Algorithm

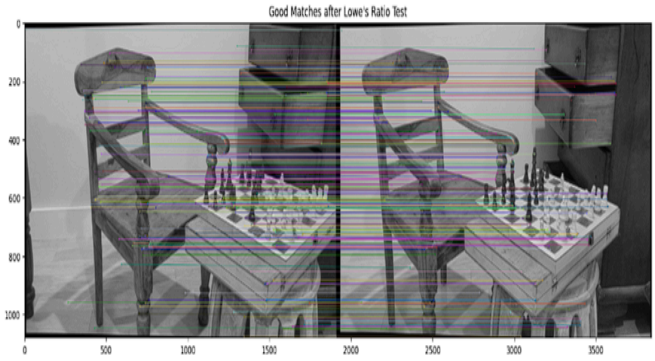
The algorithm basically involves use of SIFT (Scale invariant feature transform) for feature detection and matching techniques.

The Scale-Invariant Feature Transform (SIFT) algorithm is used to detect keypoints and compute descriptors for both images.

Features between the two images are matched using the BFMatcher with the knnMatch method. This process involves finding the two nearest matches for each descriptor, which is a common step before applying a ratio test to filter out less reliable matches.

This test filters out matches based on the ratio of distances between the closest and the second-closest match. It helps to retain only those matches for which the distance ratio is below a certain threshold (e.g., 0.5), indicating a high likelihood of correctness.





The calibration data for the camera, which includes parameters like the focal length and the principal point, is parsed from a file. This data is crucial for subsequent geometric transformations and camera pose estimation.

Using the points from the good matches, the essential matrix is computed. This matrix encapsulates the epipolar geometry between

`

the two views and is used to estimate the relative rotation and translation between the cameras.

Estimated Rotation Matrix:

[[0.99994084 0.00205496 0.01068126]

 [-0.00203991 0.99999691 -0.00141977]

 [-0.01068415 0.0013979   0.99994195]]

Estimated Translation Vector:

[[-0.99983945]

 [0.01292074]

 [-0.01241483]]

The relative camera pose (rotation and translation) is recovered from the essential matrix. This step is crucial for applications like structure from motion (SfM) and stereo reconstruction.

Finally, the matches and pose estimation results are visualized and evaluated against some ground truth, if available, to assess the accuracy of the estimated parameters.

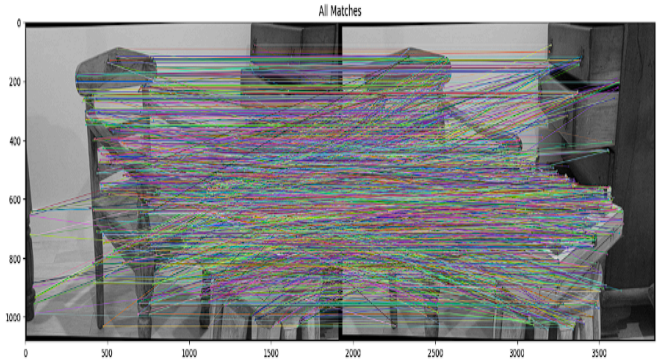
Rotation Error (Frobenius norm): 0.015511327969632206

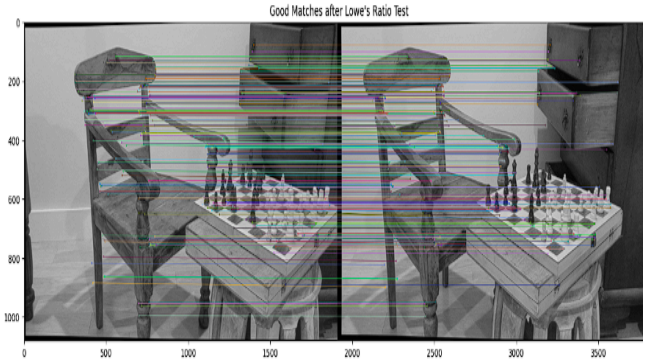
Translation Error (Euclidean distance): 125.85984072583346

Improvements

The AKAZE algorithm is used for detecting keypoints and computing descriptors in both images. AKAZE provides a set of features that are robust to changes in angle and illumination.

`





FLANN (Fast Library for Approximate Nearest Neighbors) is used for efficient matching of the AKAZE descriptors. The parameters indicate the use of the LSH (Locality Sensitive Hashing) algorithm for this purpose.

Matches are filtered using Lowe's ratio test to retain only those with a significant difference in distance between the best and second-best match, which helps reduce false positives.

The essential matrix and the relative pose between the cameras are computed using RANSAC, based on the good matches and camera calibration data. The essential matrix encodes information about the translation and rotation which are then extracted to estimate the camera pose.

Finally, the estimated rotation matrix and translation vector are displayed, and their errors compared to some ground truth values are calculated. This helps in assessing the accuracy of the pose estimation.

Estimated Rotation Matrix:

[[0.99998542 -0.00102394 0.00530213]

 [0.00100989 0.99999598 0.00265059]

 [-0.00530482 -0.00264519 0.99998243]]

Estimated Translation Vector:

[[-0.99940485]

 [-0.02422109]

 [-0.02456205]]

`

Rotation Error (Frobenius norm): 0.008505596988783486

Translation Error (Euclidean distance): 125.85940957249983

How can we improve??

1. **Enhanced Feature Matching:**
    - We can use more advanced feature detectors and descriptors (like ORB, BRISK, LightGlue) that might provide better performance in specific scenarios compared to SIFT or AKAZE.
    - Can Incorporate semantic information or machine learning to prioritize features based on their likelihood of belonging to static structures in the environment rather than transient ones.

2. **Filtering and Refinement:**
    - Experiment with different RANSAC parameters or alternatives like PROSAC, which prioritizes matches that are more likely to be correct, improving the robustness of fundamental matrix estimation.

3. **Deep Learning Approaches:**
    - **Learning-based Feature Matching:** We can Utilize neural networks trained specifically for the task of feature matching across challenging conditions, potentially integrated within a learned end-to-end system for pose estimation like LightGlue

Handling Moving Objects in Scenes

1. **Segmentation and Classification:**
    - **Dynamic Object Segmentation:** We can use segmentation networks (like Mask R-CNN, YOLO) to identify and mask out moving objects such as humans or animals. Only use features from static parts of the scene for pose estimation.

`

- **Semantic Classification:** Classify parts of the image into static and dynamic components. Exclude dynamic components from the pose estimation process.

2. **Machine Learning Approaches:**
    - **Train on Diverse Datasets:** We can Use datasets that include dynamic scenes for training machine learning models, which can help the model learn to ignore or segment out moving objects.
    - **Deep Feature Matching:** Some deep learning approaches can inherently distinguish between features from static background and moving objects by learning from large datasets.

For handling dynamic scenes, it might be useful to integrate different types of sensors (like LiDAR or depth sensors) which can provide additional data points that are less sensitive to issues like motion or poor lighting conditions.

By combining several of these strategies, it’s possible to significantly enhance the robustness and accuracy of camera pose estimation, especially in complex environments with moving objects.