MCEN90032 Sensor Systems

Workshop 3(b): Camera Data Processing

Prepared by: Nandakishor Desai and Marimuthu Palaniswami

---

1 Introduction

Autonomous cars use video cameras to see and understand the objects in the road. By equipping cars with the cameras at every angle, the vehicles can be made to maintain a 360° view of their external environment providing a broader picture of the traffic conditions around them. Camera attached to the vehicle can capture sequence of images through its movement. The sequence of images can be used to determine recover the camera pose and determine its movement relative to starting point. Generally, steps to perform pose recovery include extracting image feature descriptors, which are then matched across images to determine features corresponding to each other. Subsequently, motion estimation is employed to estimate the transformation between the corresponding features and determine vehicle movement between two locations. This process is generally known as visual odometry.

In this task, you will write simple MATLAB program scripts to understand different steps of visual odometry, including extracting the feature descriptors and finding their correspondences between images. Further, you will estimate the homography matrix to determine the geometric transformations between image frames and finally recover the camera pose relative to its initial point.

- The workshop assignment is worth **6 marks**.
- You will need to perform this work individually.
- There is no report for this workshop. You need to just submit your well-documented codes in a zipped folder through LMS.
- The demonstrations are due by (23/05/2022 during week 12). You need to explain the working of your code to demonstrators and answer their questions.
- The codes are to be submitted by 27/05/2021 before 11:59 PM.

1.1 Learning Objectives

- Understand the keys steps of visual odometry.
- Extracting the feature descriptors from images.
- Determining feature correspondences between the images.

1.2 Component Required (minimum)

1. A MATLAB account, if you don't have one please go to https://matlab.mathworks.com/
2. Ensure that you have computer vision toolbox and a MATLAB version of 2021b.

2 Project Instructions

Part 1: Feature Extraction

Scale invariant feature transform (SIFT) is a local feature detector and descriptor that is constructed based on the difference of the Gaussian (DoG) scale-space pyramid. SIFT descriptor is constructed by using crudely localised information and the distribution of gradient features. It includes fixing a reproducible orientation based on information from a circular region around the interest point and then extracting the descriptor from a square region constructed in alignment to the selected orientation.

(a) Extract SIFT feature keypoints from the given image file1.png and display the 10 strongest (based on the metric returned in the SIFTpoints object) keypoints by overlaying them on the image. Assume the number of layers in octaves to be 1 and sigma of the Gaussian applied to the input image at the zeroth octave to be 1.6.

(b) Repeat (a) by increasing the number of layers within an octave to 5. Compare the 10 strongest keypoints from (a) with (b) and comment (you can write this in the MATLAB code as a comment).

Part 2: Feature Matching

Feature matching is an essential step in several computer vision applications,including visual SLAM and odometry. Feature matching involves establishing correspondences between two images of the same object or scene. The first step is to detect a set of interest points associated with image descriptors. Once the features and descriptors are extracted from the sequence of images, the next step involves establishing matching between feature descriptors to establish correspondences.

(a) Extract SIFT feature descriptors from file1.png and file2.png. Assume number of layers in octaves to be 3 and sigma of the Gaussian applied to the input image at the zeroth octave to be 1.6. Write a MATLAB script (**without directly using the matchfeatures command from MATLAB**) to establish unique correspondences between feature descriptors of the two images employing exhaustive search with sum of squared differences as distance metric. Unique correspondences implies that the features matched should be one-one (i.e., feature mapping from image 1 to image 2 should be same as image 2 to image 1). Ensure that your features are l_2 normalized before performing the feature matching.

(b) Implement feature matching using the inbuilt MATLAB function to establish the unique correspondences between the feature descriptors (assume MatchThreshold to be 100 and MaxRatio to be 1.0). Verify with your matched features from (a). Display the matched feature points overlaid on the images.

Part 3: Recovering Camera Pose

Homography is defined as the mapping between two image plane projections. It determines the transformation between the images capturing the planar world scenes and helps understand the movement of the camera and recover its pose

(a) Estimate the homography matrix between the two image frames using the matched feature points from the previous question. You can use the '*estimateGeometricTransform2D*' command in the MATLAB with appropriate arguments.

(b) Once the homography matrix is computed in (a), use the '*relativeCameraPose*' MATLAB command to recover the pose of the second camera relative to the first. You can assume the focal lengths of the camera to be (517.3,516.5) pixels. The optical centres of the camera are at (318.6, 255.3) pixels.

Notes and tips

- You do not have to implement any of the algorithms from scratch (**except 2(a)**) and can use existing MATLAB commands.
- Pay attention to arguments inside a MATLAB command to configure your code for the task.
- Refer to lecture 17-19 for basic knowledge of image formation, feature extraction, feature matching and homography estimation.
- Please talk to your demonstrators if you are unclear on anything.

3 Submission

Submit your codes as a zip file containing '.m' files. Name the report and zip file with your student ID and submit to Canvas before the due date. Late penalties apply at a rate of -10% (of the whole assignment marks) per day. Assignments submitted later than 5 working days after the due date will not be accepted and will receive zero marks. Only the last submission will be assessed.

4 Academic Integrity

We take academic integrity seriously. Please note that while students may discuss and help each other in the thinking process, you should work on your assignments separately. Details about academic integrity can be found at http://academicintegrity.unimelb.edu.au/. Please check with the tutors or the lecturer if you are in doubt. Ignorance is not a valid reason for academic misconducts.

References

[1] Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012, October). A benchmark for the evaluation of RGB-D SLAM systems. In 2012 IEEE/RSJ international conference on intelligent robots and systems (pp. 573-580). IEEE.