Lab 6 Report: EXERCISES

Yilin Zhang [1]. Lab Group: 31.

Deliverable 1 - Nister's 5-point Algorithm

Read the paper and answer the questions below: Read the following paper.

[1] Nistér, David. "An efficient solution to the five-point relative pose problem." 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Vol. 2. 2003. link here.

Questions:

1 Outline the main computational steps required to get the relative pose estimate (up to scale) in Nister's 5-point algorithm.
2 Does the 5-point algorithm exhibit any degeneracy? (degeneracy = special arrangements of the 3D points or the camera poses under which the algorithm fails)
3 When used within RANSAC, what is the expected number of iterations the 5-point algorithm requires to find an outlier-free set?
Hint: take same assumptions of the lecture notes

Deliverable 2 - Designing a Minimal Solver

Can you do better than Nister? Nister's method is a minimal solver since it uses 5 point correspondences to compute the 5 degrees of freedom that define the relative pose (up to scale) between the two cameras (recall: each point induces a scalar equation). In the presence of external information (e.g., data from other sensors), we may be able use less point correspondences to compute the relative pose.

Consider a drone flying in an unknown environment, and equipped with a camera and an Inertial Measurement Unit (IMU). We want to use the feature correspondences extracted in the images captured at two consecutive time instants t_1 and t_2 to estimate the relative pose (up to scale) between the pose at time t_1 and the pose at time t_2. Besides the camera, we can use the IMU (and in particular the gyroscopes in the IMU) to estimate the relative rotation between the pose of the camera at time t_1 and t_2.

You are required to solve the following problems:

1 Assume the relative camera rotation between time and is known from the IMU. Design a minimal solver that computes the remaining degrees of freedom of the relative pose.
Hint: take same assumptions of the lecture notes
2 **OPTIONAL**: Describe the pseudo-code of a RANSAC algorithm using the minimal solver developed in point a) to compute the relative pose in presence of outliers (wrong correspondences).

[1]Yilin zhang, OUC id: 23020036094, is with Faculty of Robotics and Computer Science, Ocean University of China and Heriot-Watt University, China Mainland zyl8820@stu.ouc.edu.cn

Appendix B
Team

Deliverable 3 - Initial Setup

Before we go to motion estimation, an important task is to calibrate the camera of the drone, i.e., to obtain the camera intrinsics and distortion coefficients. Normally you would need to calibrate the camera yourself offline to obtain the parameters.

However, in this lab the camera that the drone is equipped with has been calibrated already, and calibration information is provided to you! (If you are curious about how to calibrate a camera, feel free to check this ROS package)

As part of the starter code, we provide a function `calibrateKeypoints` to calibrate and undistort the keypoints. Make sure you use this function to calibrate the keypoints before passing them to RANSAC.

Deliverable 4 - 2D-2D Correspondences

Given a set of keypoint correspondences in a pair of images (2D - 2D image correspondences), as computed in the previous lab 5, we can use 2-view (geometric verification) algorithms to estimate the relative pose (up to scale) from one viewpoint to another.

To do so, we will be using three different algorithms and comparing their performance.

We will first start with the 5-point algorithm of Nister. Then we will test the 8-point method we have seen in class. Finally, we will test the 2-point method you developed in Deliverable 2. For all techniques, we use the feature matching code we developed in Lab 5 (use the provided solution code for lab 5 if you prefer - download it here). In particular, we use SIFT for feature matching in the remaining of this problem set.

We provide you with a skeleton code in `lab6` folder where we have set-up ROS callbacks to receive the necessary information.

We ask you to complete the code inside the following functions:

1 **`cameraCallback`: this is the main function for this lab.**
2 **`evaluateRPE`: evaluating the relative pose estimates**
3 **Publish your relative pose estimate**

Deliverable 5 - 3D-3D Correspondences

The rosbag we provide you also contains depth values registered with the RGB camera, this means that each pixel location in the RGB camera has an associated depth value in the Depth image.

In this part, we have provided code to scale to bearing vectors to 3D point clouds, and what you need to do is to use Arun's algorithm (with RANSAC) to compute the drone's relative pose from frame to frame.

1 **`cameraCallback`: mplement Arun's algorithm**

Performance Expectations: What levels of rotation and translation errors should one expect from using these different algorithms?

Summary of Team Deliverables: For the given dataset, we require you to run **all algorithms** on it and compare their performances.