Vision-Based Navigation
J. Stückler, V. Usenko, L. von Stumberg
Winter Semester 2017/2018

Computer Vision Group
Department of Informatics
Technical University of Munich

Exercise Sheet 2

Topic: Direct Dense Visual Motion Estimation

Submission deadline: Tuesday, 07.11.2017, 23:59 am
Hand-in via email to visnav_ws2017@vision.in.tum.de

General Notice

The exercises should be done in teams of two to three students. Each student in a team must be able to present the solution to the tutors during the exercise sessions on a lab PC in room 02.05.014. The presentations and solutions will be graded and will count for the final grade of the lab course. If you have not yet done so, please register yourself together with your team members on the team list in room 02.05.14.

We will use ROS Kinetic on Ubuntu 16.04 in this lab course. It is already installed on the lab computers. If you want to use your own laptop, you will need to install these versions of Ubuntu and ROS. Please read the ROS and OpenCV documentation for further reference.

Introduction

The goal of this exercise is to acquire practical experience with a visual motion estimation method for RGB-D images using direct dense image alignment.

Exercise 1: (16 points)

Download the code sample for this exercise provided on the course website:

- `https://vision.cs.tum.edu/teaching/ws2017/visnav_ws2017/material`

To get you started, it contains a basic implementation for dense direct image alignment. You will have to implement methods within this code in this exercise.

Also use the following bag file in this exercise:

- `https://vision.in.tum.de/rgbd/dataset/freiburg2/rgbd_dataset_freiburg2_desk.bag`

Please note that if you run into computation issues on your PCs you can play the bag files at reduces speed using -d option of rosbag play.

(a) Implement the gradient of the (photometric) residual using finite differencing (function deriveNumeric). Please note that you can use calculateError function to evaluate residual. (4 points)

(b) Implement the gradient of the (photometric) residual using analytic differentiation (function deriveAnalytic). You can find more information on analytic derivatives here: `http://ingmec.ual.es/~jlblanco/papers/jlblanco2010geometry3D_techrep.pdf` (4 points)

(c) By default the provided code uses Gauss-Newton method to minimize the error function (see useGN variable). Implement Gradient Descent and Levenberg-Marquardt optimization algorithms to minimize the error function. What differences in number of iterations and convergence properties do you observe? (4 points)

(d) Compare your motion estimates for finite differencing and analytic differentiation with the ground truth using the evaluate_rpe and evaluate_ate tools of the RGB-D benchmark dataset. For the evaluate_rpe tool, use the unit of 'frames' and a fixed difference of 1 frame to calculate the relative poses. What is the RMSE error in RPE and ATE of your method? Plot the ground truth trajectory and your result using the evaluate_ate tool. Also plot the RPE per frame using the evaluate_rpe tool. Does the accuracy increase if you drop frames for the processing? Evaluate ATE and RPE for different number of frame drops (1,2,4,8,16,32). (2 points)

(e) Estimate the pose covariance for the non-linear least squares solution and visualize it using RViz. Compare the results for finite differencing and analytic differentiation visually and provide a numerical example. (2 points)

Submission instructions

A complete submission consists both of a PDF file with the solutions/answers to the questions on the exercise sheet and a ZIP file containing the source code that you used to solve the given problems. Note all names and matriculation numbers of your team members in the PDF file. Make sure that your ZIP file contains all files necessary to compile and run your code, but it should not contain any build files or binaries. Please submit your solution via email to `visnav_ws2017@vision.in.tum.de`.