TDT4265 Computer Vision - Project Proposal

Group 11
Martin Kvisvik Larsen

March, 2019

1 Project Proposal

1.1 Project Goal

My goal for this project is to be able to implement a pose estimation system, mainly from scratch, in order to do trajectory estimation for a moving camera. Additionally I want to evaluate the performance of the system by using the online RGBD SLAM evaluation tool provided by Technische Universität Munichen (TUM). If possible, I want to compare the performance of my system with other systems and look at difference in the methods that has been used and their performance.

1.2 Relevant Literature

The literature that I will be using in the project is:

- R. Szeliski - "Computer Vision: Algorithms and Applications", [link]
- UiO Lecture Notes - "Corner Features", [link]
- UiO Lecture Notes - "The Perspective Camera Model", [link]
- UiO Lecture Notes - "Pose in 2D and 3D", [link]
- PSU Lecture Notes - "Harris Corner Detector", [link]
- J. Nocedal & S. Wright - "Numerical Optimization", [link]

1.3 System Architecture

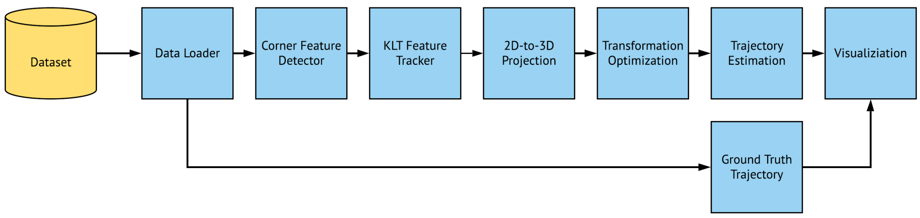

Figure 1: Planned system architecture.

1.4 Datasets

Throughout the project I plan to use several different datasets, depending on the stage of the project. All of the datasets are from TUM and can be found at the official university website [link]. The datasets contains RGB-D data from a Kinect [link] in the form of 640×480 RGB PNG images for the colour channels and 640×480 monochrome PNG images for the depth channel. The sequence of images is defined by two .txt files, one for the colour channels and one for the depth channel, containing timestamps and the corresponding image file at each timestamp. In addition to the images and the image sequence files, each dataset also contains ground truth data in the form of vectors of the following form:

$$\mathrm{gt}_i = [\mathrm{t}_i, \mathrm{tx}_i, \mathrm{ty}_i, \mathrm{tz}_i, \mathrm{qx}_i, \mathrm{qy}_i, \mathrm{qz}_i, \mathrm{qw}_i]$$

Here t_i is the time stamp, $[\mathrm{tx}_i, \mathrm{ty}_i, \mathrm{tz}_i]$ are the translation of the camera with respect to the world origin and $[\mathrm{qx}_i, \mathrm{qy}_i, \mathrm{qz}_i, \mathrm{qw}_i]$ is a unit quaternion defining the orientation of the optical center of the camera with respect to the world origin. Additionally, the datasets contain accelerations from an inertial measurement unit (IMU) aswell. However, the plan is not to use the IMU data.
The specific datasets from TUM that I plan to use are "fr1/xyz", "fr1/rpy", "fr1/desk" and "fr2/large_no_loop". The "fr1/xyz" and "fr1/rpy" datasets will be used during the implementation of the system for debugging purposes. Then when the system has been developed, the "fr1/desk" and "fr2/large_no_loop" datasets will be used to test the performance of the system. TUM also has an online evaluation tool for the estimated trajectory, which can be found on their website [link]. I will use the evaluation tool to get a objective performance measurement of the system.

1.5 Implementation

Initially, the plan is to implement the system in figure fig. 1 in C++. The idea is to build most parts of the system from scratch, except for the transformation optimization. The transformation optimization part will be handed about by the project's TA as Python code, so I will have to translate the code to C++. In order to reduce the amount of code that needs to be implemented I will use the OpenCV C++ library [link] to provide the most basic functionality of the system. OpenCV will allow me to use the built-in matrix class and the different linear algebra operations, so that I will not have to implement it myself.
If the implementation of the system in C++ proves to be to slow I might go for a backup plan and implement the system in Python. If this is the case I will use OpenCV-Python and NumPy for the basic functionality.