Large Scale Direct monocular Slam (LSD SLAM) - Mid Term Report

Maitrey Gramopadhye (160050049)
Neelesh Verma (160050062)
Swadha Sanghvi (16D070037)

Summary of the Project

- Introduction -
  We propose to implement parts of the research paper - https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwiX7dzAhorhAhULMo8KHWCTBQMQFjAAegQICBAC&url=https%3A%2F%2Fvision.in.tum.de%2F_media%2Fspezial%2Fbib%2Fengel14eccv.pdf&usg=AOvVaw115jAxxjnrw1np7r8GKY2Z. It comprises of a featureless Monocular SLAM algorithm to generate the semi-dense depth maps obtained by filtering over a large number of pixelwise small-baseline stereo comparisons. Then the 3D environment is reconstructed in real-time as pose-graph of keyframes.
- Initialisation
  The KeyFrame of the graph is initialised with the current frame and a random depth map, the transform is initialised as identity. Now, the depth map, transform and covariance are updated with each incoming frame.
- The Complete Method -
  - Tracking
  - Depth map estimation
  - Map optimization
- Tracking -
  After initialising a keyframe, the new frame's variance-normalized photometric error is optimized to find its relative camera pose, the method for doing so is described in Section 2.2 of the paper. Also, with each incoming frame, the uncertainty of it being of the same environment as the reference frame increases, captured by the covariance matrix which is updated as described in Section 2.3 of the paper. When the covariance falls below a threshold, a new KeyFrame is initialised, and an edge is added between the two KeyFrames. This edge holds the relative camera pose and covariance matrix. Note, no prior information is available for the frame, so minimisation is performed for a rigid body transform, this can lead to scale drift in the graph.
- Depth map estimation -

It uses a stereo-based depth update. The research paper refers to another research which we are in the process of reading.

- Map optimization -
  As we receive frames from the streaming video, we need to decide whether these frames would be initialised as a new Keyframe otherwise contributing to the refinement of the existing Keyframe, We perform a similarity transform to check for a threshold and decide the rest. This constitutes the second stage of our project. We have finished reading upon it, however are in the process of fully understanding it as this too is referred from another paper.
- Testing -
  We will test our implementation on the publicly available datasets referenced in the paper - [25] TUM RGB-D benchmark and [12] two simulated sequences.
- End Result -
  We will display the depth map of the frames (Stage 1) and 3D map of surrounding (Stage 2), being computed in real time, as a video sequence, along with the actual video sequence.

Work Done So far

Keeping the first stage of our project in mind, we are focusing on depth map creation, and, if completed, post that will move on 3D map creation. We have initialised the Keyframes and its corresponding attributes. Assuming the depth maps to be known, we have coded the optimization functions for refining the transform of the keyframe. Tracking further requires depth maps which we are still reading upon. Depth map generation is referred to from another research paper which we are currently in the process of reading. We are using python for our project and will be using numpy and openCV libraries.

Future Plans

We plan to complete our study of depth map generation and implement the Stage 1 of our project first (Real time depth map creation and KeyFrame graph generation). The remaining work in this stage apart from depth map generation is debugging the code.

We then plan to move on to the Stage 2, if time permits. Stage 2 is the 3D map generation. In this we would utilise the Keyframe graph generated as the 3D map. We will focus on optimizing the map, analysing loop closures and correcting scale drifts in the graph.