Objective:

Create 3D point cloud using Visual SLAM algorithm.

Approach 1

CNN Slam

Description: This is a method where CNN-predicted dense depth maps are naturally fused together with depth measurements obtained from direct monocular SLAM. This is an open source Github Repository which is consist of total three parts –

I. Monocular depth prediction
II. CNN depth prediction
III. Fuse together and after optimization obtain 3D point cloud

The repository uses quite old version of python (2.x) and TensorFlow libraries. At first the libraries were being updated to the compatible libraries of the current version. It was a lengthy and time-consuming process, but in the end the part 1 and part 2 parts of the previously mentioned list were completed successfully. But while fusing them, a library of older version made it difficult to complete as no compatible library of this one could not be found. Later, it was found that the library has become obsolete in the latest version.

That's why, in the second attempt the versions of python and TensorFlow were downgraded in a virtual environment and that particular problem was solved. Nevertheless, new problems were still arriving and they were consuming a lot of time to be solved. So, it was decided later to drop this repository and start working with an approach that has used more recent versions of python and TensorFlow.

Paper: https://arxiv.org/abs/1704.03489

Github Repository: https://github.com/iitmcvg/CNN_SLAM

Approach 2

RTAB-Map (Real-Time Appearance-Based Mapping):

Description: This is an RGB-D, Stereo and Lidar Graph-Based SLAM approach based on an incremental appearance-based loop closure detector. It is an Open-Source Lidar and Visual SLAM Library. From the installation guideline, RTAB-Map API was installed and as it uses comparatively newer version of libraries, it was not difficult to get results from it. It can produce point cloud from RGB-D data or Stereo Camera data. At first, point cloud from a video that was captured with a stereo vision camera was used as input, and the result was quite satisfactory. After that, point cloud was created from left camera image and right camera image of stereo vision camera. These two approaches are given in the tutorials that can be spotted in the RTAB-Map repository. Calibration data of the camera is also an important pre-requisite for producing result. In the mentioned two cases, the parameters values were given in the tutorials.

After working with stereo images, similar result was produced using the RGB-D data. The dataset was collected from the TUM dataset (RGB and Depth images are given and taken from Kinect). They also provide the calibration parameter values for the used data set.

Later, it was tried to produce point cloud only from RGB image, without any depth information provided. But it turns out that, RTAB-Map is not capable of doing so. So, it was planned to produce the depth images of the given RGB data set externally and then use the depth image to produce point cloud. Basically, it is following the method that has been used in the RGB-D case. To produce point cloud 3 approaches were taken –

- Using CNN SLAM which is discussed in the 'Approach 1' section. As the first two steps is to create depth images, using that method depth images were created from monocular RGB images. But the result was significantly poor.
- Using Self-supervised Monocular depth prediction, which is an open source Github Repository. The result was still poor but, comparatively improved from CNN SLAM.
  Github Repository: https://github.com/nianticlabs/monodepth2
- Using Monocular Depth Estimation using Zero-shot Cross-dataset Transfer, which is also an open source Github Repository. In this case, the result of the depth images was quite good, but the result of point cloud was not up to the mark.
  Github Repository: https://github.com/islorg/MiDaS

After several attempts, as not getting closer to the desired result, this idea was dropped.

Paper:
https://introlab.3it.usherbrooke.ca/mediawikiintrolab/images/7/7a/Labbe18JFR_preprint.pdf

Github Repository: https://github.com/introlab/rtabmap/wiki

Approach 3

PUTSLAM

Description: After getting good result in RTAB-Map, an interest was taken into PUTSLAM. This is an RGBD-based Simultaneous Localization and Mapping technique, that is developed at Poznań University of Technology (PUT). It is also available as an open source Github Repository and this repo is 5 years older and the ubuntu version 14.04 is used. No further update has been provided.

At first, it was tried with COLAB as the user machine was using Windows. But, as COLAB uses the latest version of Ubuntu, this repository could not be run for the version issues. After that, older version of ubuntu was installed as virtual machine and tried to run the Repository. But there were too many version and module errors, even more than the first approach – CNN SLAM. So, after trying few days, this approach was dropped eventually.

Github Repository: https://github.com/LRMPUT/PUTSLAM