**Robot Navigation - the SLAM Problem**

**Introduction**

This is a technique review for SLAM technology. SLAM stands for simultaneous localization and mapping, which is a category of problems robots need to solve to navigate in the environment. The solution to the problem can help self-driving cars navigating around the city, exploration vehicles mapping the unknown environment, and etc.. The two tasks, localization and mapping, respectively localize the robot’s relative position in a given reference frame, and create a map, or representation, of the surrounding environment. Both tasks rely on each other, and a better result of one task will improve the result of the other. Though localization problem can be solved with GPS, there are situations that GPS is not applicable, such as space expedition, where GPS is not usable, and indoor localization, where GPS signal is too weak to be accurate, while the mapping problem is nontrivial.

Many solutions are created to solve the problem, and can be divided into two categories, key point tracking and optical flow tracking, using camera as the main, though other sensors, such as depth sensor or gyroscopic sensors may also be used. Localization is accomplished through predicting camera pose by tracking key point or optical flow, and mapping is created with the key point location. Since SLAM are for real-time systems, and the computing power for the solutions are demanding, there are few commercial implementations on the market currently, and most proposed solutions are from research institutions. Three state-of-art solutions will be addressed below.

**Solutions**

**SVO: Fast Semi-Direct Monocular Visual Odometry**

This solution is created for micro aerial vehicles, which can not provide as much computing power as laptops, in University of Zurich. Therefore, its design principle is to require less computing resource, while to keep the processing speed high, and error low. It has only a camera as sensor. As most system, it runs tracking and mapping in parallel to have a higher frame processing rate, and it uses optical flowing information as the main tracking factor. Optical flowing calculates the gradients of image pixel intensity to determine camera motion. Though this solution also uses key points in the local map to optimize camera pose as the second tracking step, key point extraction is done on the mapping thread, which saves time for motion estimation. Also as the solution name indicate, the main goal of the solution is pose estimation, so mapping is not well implemented as other solution discussed here. The system were tested on a laptop, which has eight cores, and 2.8GHz (but the system is only allowed two cores), and an embedded platform, which has four cores, and 1.6 GHz. The runtimes are 300 frames per second on the laptop, and 55 frames per second on the embedded platform, which is qualified for a real time system, since most real time tracking system needs to process a frame rate between 20-60Hz. The error for the localization can be archived below 0.006m/s and 0.5degree/s.

**LSD-SLAM: Large-Scale Direct Monocular SLAM**

This solution is created in Technical University Munich to have a better global mapping, compared to works before it. This system has three parts, one for tracking, one for local mapping, and one for global mapping. This system also uses optical flow for motion estimation, but different from SVO, it takes depth noise into account, through which better motion estimation can be obtained, and it doesn’t use key point tracking at all, which promises a higher frame processing rate, since tracking thread doesn’t need to extract features, or even match features. Local mapping creates local key frames, which contain key points to construct a local map, as SVO does. However, global mapping optimization is where LSA-SLAM surpasses its predecessors. A global map of key frames are kept and optimized all the time, and it helps detect loops, where robot comes back to place it has visited. This system has been tested on PC, and smartphone, though device specifications are not provided. Therefore, it should be able to run on a real time system. Also the accuracy is promising, with the error resolution at a magnitude of centimeter.

**ORB-SLAM**

ORB here stands for ORB features, which is a feature description. This solution is the only solution here use feature point tracking to estimate motions. As LSD-SLAM, ORB-SLAM also uses three threads, one for tracking, one for local mapping, and one for global optimization. The tracking procedure has two steps, like that of SVO. For the first step, ORB features are extracted for new frame, and are matched with key features in last frame to estimate a rough motion. Then key features in the local map are used to optimize camera pose estimation. The local mapping and global optimization of ORB-SLAM is similar to those of LSD-SLAM, with more optimization implemented to calculate a more accurate global map. However, since features are extracted at each frame, the frame processing speed is as low as 30 frames per second, lower than SVO, or LSD-SLAM. Also mapping optimization is computing power demanding, which means electrical power demanding. But the accuracy is 10 times than LSD-SLAM’s accuracy. Therefore, there is the trade off between speed and accuracy.

**Reference:**

|  |  |
| --- | --- |
| [1] | C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast semi-direct monocular visual odometry,” in ICRA, 2014, pp. 15–22. |
| [2] | J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular SLAM”, in ECCV, 2014, pp. 834-849 |
| [3] | R. Mur-Artal, J. Monteil, J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system", IEEE Transactions on Robotics (TRO), vol. 31, no. (5), pp. 1147-1163, 2015. |
| [4] | T. Malisiewicz, 2016, Jan. 13, “The Future of Real-Time SLAM and Deep Learning vs SLAM”. Available: <http://www.computervisionblog.com/2016/01/why-slam-matters-future-of-real-time.html> |
| [5] | S.Riisgaard, and M. R. Blas, *SLAM for Dummies*, Cambridge, MA: MIT |
| [6] | A. Handa, and A. Davison, “High Frame-Rate”. Available: http://wp.doc.ic.ac.uk/robotvision/project/high-frame-rate/ |
| [7] | S. Thrun, “Lession 6: SLAM(Simultaneous Localization and Mapping)”, Artificial Intelligence for Robotics. Available: <https://www.udacity.com/course/artificial-intelligence-for-robotics--cs373> |