Semi-Direct Visual Odometry and Mapping with RGB-D Camera

Xinliang Zhong, …Authors put here

***Abstract*— We propose a …**

# I. INTRODUCTION

The problem of simultaneous localization and mapping(SLAM) is one of the hotspots in the field of robotics and computer vision community over the past decade. Precise positioning is the basis for robot control and navigation in GPS-denied environments. Especially for micro aerial vehicle(MAV) working in complex and cluttered unknown indoor environments. They need to constantly update their position at high rates and low latency for position and orientation control. At the same time, they can only carry limited weight and power consumption of the sensor and processor. While previously many SLAM systems relied on expensive and heavy laser scanners. RGB-D cameras based on structured light provide a powerful alternative and well suited for such application. For instance, the Asus Xtion sensor provides both color and depth images directly in real-time. As the sensor weighs only 77 grams and consumes less than 2.5 watt, it can be easily used for localization, mapping and navigation of MAVs.

To our knowledge, most Visual Odometry(VO) or SLAM systems are feature based, which typically extract sparse keypoints from the camera image and then estimate the motion from the consecutive frame. In contrast to featured based methods, direct methods which use all the pixels based on photometric error minimization are becoming increasingly popular.

In this paper we propose, to our best knowledge, a semi-direct VO and mapping system which inherits feature based and direct methods tightly to improve the precision and robustness. We track the pixels with strong gradient and choose keyframes which are used for mapping and loop closure. A robust sensor model based on the t-distribution [\*], and an error function which mixes the depth error and photometric error are used to our SLAM system. In contract to some VO-only based system [\*], we propose a hybrid approach that combines the state-of-art loop closure method **—** Bag of Words(BoW) and method based on spatial location constraints. Our method achieves higher robustness and precision just using CPU, which can be easily migrated to embedded devices and applied to MAVs.

The main contributions of this paper are:

• The first one put here…

• The second put here…

• The third put here…

Section II provides… and Section III, thereafter, introduces… Section IV and Section V explain …

# II. RELATED WORK

Visual SLAM approaches, also referred to as “structure and motion estimation” (SFM) [\*] compute the robot’s motion and the map using cameras as sensors. A series of works for camera pose estimation and optimization using RGB-D data have been published over the past few years. According to the implementation, we classify them as following categories.

a) *Feature-Based Methods:* The standard approach is to extract feature from consecutive images by keypoint detectors and descriptors such as SIFT [\*], SURF [\*] and ORB [\*]. Camera pose can be estimated by matching the keypoints to last frame [\*], [\*], [\*]. Local Bundle Adjustment (BA) [\*] and Random sample consensus (RANSAC) [\*] are applied to refining the matches and restricting the outliers to ensure the precise transformation between frames. The first RGB-D SLAM system was proposed by Henry et al. [\*] who extract SIFT features by SIFTGPU in combination with ICP algorithm [\*].

They created and optimized the pose graph by a sparse BA method. Similarly, in order to make the system more general, Endres et al. [\*] used SURF and ORB features to realize the VO. They accomplished the pose-graph optimization by g2o framework instead of BA. Yet, all of these approaches ignore most of the image information, because a few part of (typically 200-500) interest points are extracted to use.

b) *Direct Methods:* Direct methods aim at using the whole image to estimate the structure and motion. Koch [\*] showed that the camera pose can be estimated efficiently by minimizing the photometric error which can be seen as an extension of the Lukas-Kanade tracker [\*] between the observed images. KinectFusion [\*] is an impressive approach that using a variant ICP for image model alignment. Drawbacks of ICP algorithm include the dependency on a good initial guess to avoid getting stuck in a local minimum. Similarly, DTAM is another SLAM approach which based on photometric error. However, too much GPU memory and extensive computational power are required to make both KinectFusion and DATM run in real-time. A robust odometry estimation were proposed by Kerl et al. [\*]. They use a robust photometric error function based on t-distribution that reduces the influence of large residuals between consecutive images. In their further study, an entropy-based criterion is proposed for keyframes selection and pose graph optimization. Such method is also applied in monocular SLAM algorithm like LSD-SLAM [\*]. Need to add something…

c) Semi-*Direct Methods:* Recently, Semi-Direct method approach is popular in monocular SLAM system. Engel et al. [\*] introduced an efficient epipolar search method, enabling real-time VO and semi-dense map reconstruction. An efficient Semi-Direct monocular VO (SVO) [\*] approach was proposed by Forster et al. Their algorithm is applied to MAV state estimation in GPS-denied environments and run at 55 frames per second on the onboard embedded computer. However, the success of SVO depend on a good depth initialization result from the probabilistic depth filter. Furthermore, it is only a monocular VO system without loop closure, which means drift is more likely to occur. Bu et al. [\*] improve it with loop closure detection and extend the approach with RGB-D cameras.

In contrast to all previous work on RGB-D SLAM system, our method inherits the advantages of feature-based and direct method tightly. Put the main contribution here!!!

# III. APPROACH

1. *System Overview*

In general, a visual SLAM system can be divided into three modules [\*]: Frontend, backend and final map representation. The Frontend is mainly focus on computing the camera’s motion from a sequence of images. As the existence of inherent and estimation uncertainty, backend is used to optimize the camera’s pose by a maximum likelihood solution [\*] or some other methods. With the trajectory between frames, we can generate a map (usually point clouds) for what we have observed. Figure 1 shows an overview of our system. The algorithm uses three parallel threads, one for estimating the camera motion, and the second one for optimizing the camera’s poses and the third one for mapping.

1. *Motion Estimation*

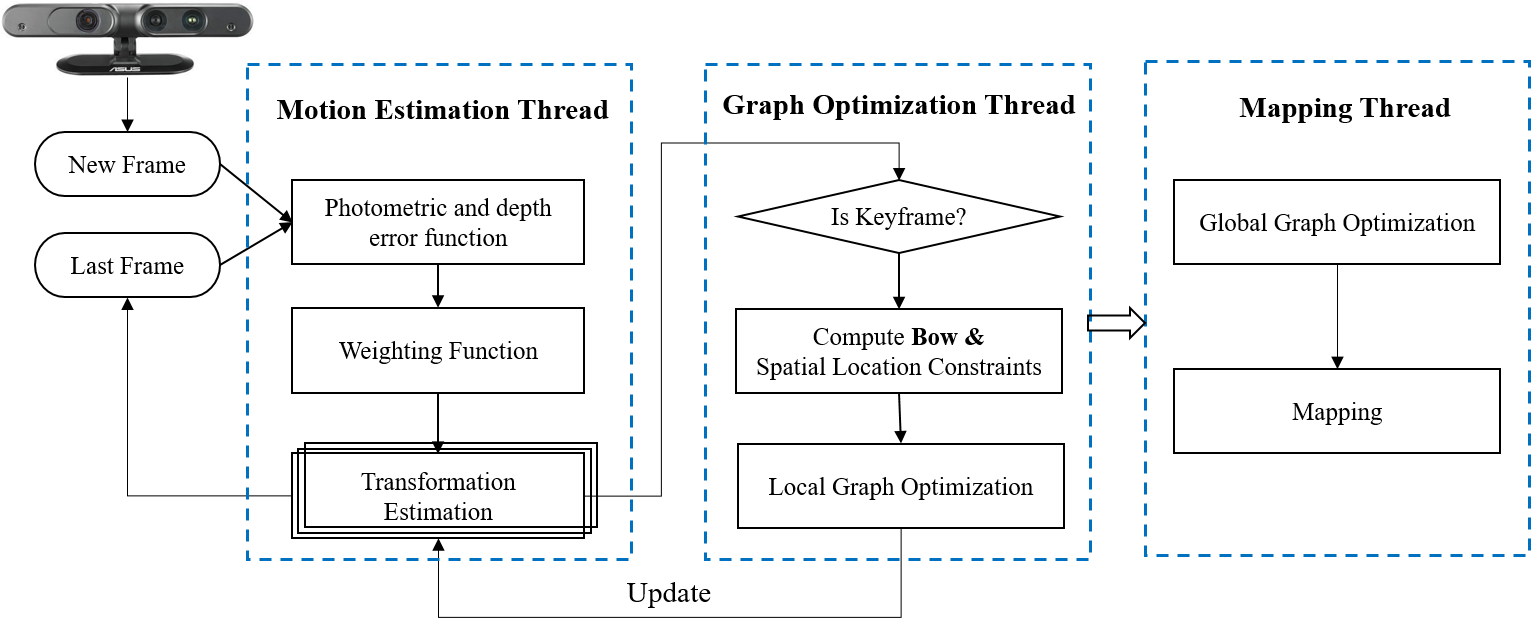


Fig. 1: Semi-Direct Visual Odometry and mapping pipeline

