FSVO: SEMI-DIRECT MONOCULAR VISUAL ODOMETRY USING FIXED MAPS

Zhiheng Fu[1], *Yulan Guo*[1,2], *Zaiping Lin*[1], *Wei An*[1]

1. College of Electronic Science and Engineering, National University of Defense Technology, Changsha, China
2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

ABSTRACT

We propose a fixed-map semi-direct visual odometry (FSVO) algorithm for Micro Aerial Vehicles (MAVs). The proposed approach does not need computationally expensive feature extraction and matching techniques for motion estimation at each frame. Instead, we extract and match ORiented Brief (ORB) features between keyframes and assist-frames. We replace the incremental map generation step in traditional algorithms with fixed map generation at keyframe and assist-frame only in our algorithm, resulting in reduced storage memory and higher flexibility for relocalization. Based on the fixed-map, we design a new keyframe selection criterion and a relocalization step. Our algorithm has no limit on the orientation of the camera and reduces drifting effectively. Experimental results on the EuRoC and KITTI datasets show that our algorithm achieves higher precision and robustness than the SVO algorithm.

Index Terms— Visual Odometry, Fixed-Map, Relocalization

1. INTRODUCTION

Micro Aerial Vehicles (MAVs) play an important role in various applications including disaster management, industrial inspection, and environment conservation [1] [2]. Localization works as a core technique for MAV systems for precise and autonomous operations. In complicated environments, GPS signals might be unavailable. Therefore, alternative localization solutions for MAVs are highly required. Due to the limitation in power and load capacity of MAVs, solutions using a single camera and an Inertial Measurement Unit (IMU) have been investigated [3, 4, 5, 6, 7].

A Visual Odometry (VO) system can estimate the camera position and attitude from a video recorded by the camera. Most monocular VO systems for MAVs are feature-based. A typical feature-based approach first extracts a sparse set of salient image features (e.g. points, lines) in each image and then match them in successive frames using invariant feature descriptors. Next, it robustly recovers the camera motion and structure using epipolar geometry, and finally refines the pose and structure by reprojection error minimization. A representative feature-based VO algorithm is ORB-SLAM [8], which achieves high estimation accuracy. However, its speed is too low for MAV applications. For RGB-D and stereo SLAM systems, direct methods based on photometric error minimization [9, 10] are becoming increasingly popular. Direct methods [11] estimate the structure and motion directly from all the intensity values in an image, including the areas with small gradients. These methods have been shown to outperform feature-based methods in terms of robustness for texture-less scenes [12] or images with camera defocusing and motion blur [13]. The computational cost for photometric error minimization is higher than reprojection error minimization, as it involves warping and integration of large image regions. However, since direct methods operate directly on the intensity values of an image, the time for feature detection and description can be avoided. Recently, LSD-SLAM [14] outperforms other direct methods. Alternatively, several semi-direct methods have been proposed to combine feature-based and direct approaches. Semi-direct methods extract features in keyframes only and track these features in subsequent frames. They then recover both camera motion and structure using photometric error minimization. Finally, the pose and structure are refined by reprojection error minimization. A popular monocular visual odometry for MAVs is Semi-Direct Visual Odometry (SVO) [15]. It achieves both high computational efficiency and accuracy. However, this algorithm is designed for down-looking camera only. Besides, its memory storage is high.

Inspired by incremental map and the SVO algorithm, we propose a new semi-direct method for MAVs, i.e., Fixed Map Visual Odometry (FSVO). We extract ORiented BRIEF (ORB) features [8] in each keyframe and its next frame (namely, assist-frame) and track the ORB features in the keyframe using the Lucas-Kanade Tracking (KLT) algorithm [16]. Then, with the positions of feature points in the assist-frame, we find their corresponding features in a small region around these positions. Finally, we eliminate outliers using the RANdom SAmple Consensus (RANSAC) [17] algorithm. Next, we build a fixed map using the inliers and insert keyframe and assist-frame into the fixed map. The initial camera pose is estimated using the sparse model-based image alignment algorithm [18]. We continue to use point features only to refine pose and structure. When the next keyframe is arrived, the fixed map is cleared and the position of MAV is

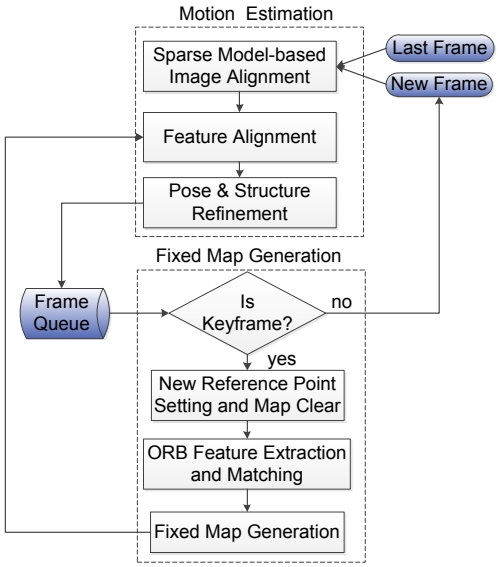

Fig. 1: The pipeline of our FSVO algorithm.

relocated. Consequently, the proposed algorithm can be used in scenes with occlusion and large motion. It can also work with forward-looking cameras.

The contributions of this paper can be summarized as follows. First, a novel semi-direct VO framework is proposed for MAVs using fixed maps to achieve improved motion estimation result. Second, the storage of keyframes for our VO framework is reduced. Third, there is no constraint on the orientation of the camera.

2. MOTION ESTIMATION

For visual odometry, we mainly focus on motion estimation rather than scene reconstruction. Therefore, we introduce a fixed map instead of an incremental map. Specifically, we first build a fixed map using feature-based methods. We then compute an initial value of relative camera motion using a direct method [14], as inspired by the SVO algorithm. We finally refine the motion using a feature-based nonlinear reprojection error. The pipeline of our FSVO algorithm is shown in Fig. 1, which consists of two parts, i.e., fixed map generation and motion estimation.

2.1. Feature-based Fixed Map Generation

We build a fixed map to replace the incremental map used in [19, 14, 15] using the keyframe and the assist-frame only, as shown in Fig. 2. To initialize our method, we set the first frame as the keyframe and extract ORB features from this image. To reduce computational time for feature detection and to improve robustness, each keyframe is evenly divided into several cells and only one feature with the highest response value is extracted from each cell. The same operation is also performed on the assist-frame. To reduce 3D structure estimation error, the assist-frame should have a certain disparity (i.e., 20% in our work) with the keyframe, so the second frame cannot always be considered as the assist-frame. Then, we track these features from the keyframe to the assist-frame using an image pyramid based KLT [16]. Specifically, the image is downsampled to its half size to create an image pyramid of three levels. The intensity residual is then optimized at the coarsest level until convergence. Subsequently, the optimization is initialized at the next finer level. In this way, our feature tracking can achieve a sub-pixel accuracy. Then, we perform feature matching by searching 8 neighbors around the cell produced by feature tracking. If a feature match is successful, we build correspondence between the matched features in two frames. Otherwise, we delete that feature. Next, we select the best points using RANSAC and calculate the essential matrix using them. Finally, we obtain an initial guess of the camera relative motion using essential matrix decomposition. With inliers and the initial camera position, we estimate the coordinates of 3D points by triangulation. We further use Local Bundle Adjustment (LBA) [20, 21, 22] to optimize the fixed map.

The process for fixed map generation is shown in Fig. 2, where $T_{k,k-1}$represents the relative transformation between the keyframe and the assist-frame, u_i and $u_i^{'}$ represent 2D points, and p represents a 3D point on the visible scene surface.

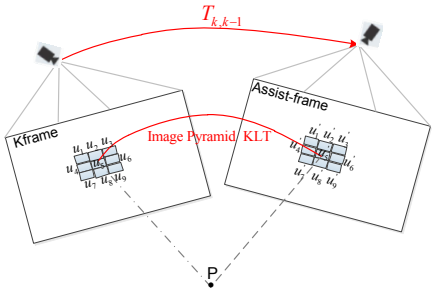

Fig. 2: Fixed map generation.

2.2. Motion Estimation Based on Semi-direct Method

We use sparse model-based image alignment, feature alignment, and pose and structure refinement to recover camera motion.

If the current frame is not a keyframe nor assist-frame, we estimate the camera motion with respect to the last frame using the steps discussed in Sec. 2.1. For sparse model-based image alignment, the maximum likelihood estimate of the rigid transformation $T_{k,k-1}$ between two consecutive camera poses minimizes the negative log-likelihood of the intensity residuals. That is:

$$T_{k,k-1} = \underset{\mathrm{T}}{\operatorname{argmin}} \iint_{\overline{R}} \rho[\delta I(T,u)]du. \qquad (1)$$

The intensity residual δI is defined as the photometric difference between pixels of the same 3D point. It can be computed by back projecting a 2D point u from the previous image I_{k-1} onto the current camera view:

$$\delta I(T,u) = I_k(\pi(T \cdot \pi^{-1}(u, d_u))) - I_{k-1}(u), \forall u \in \bar{R} \quad (2)$$

where the projection π is determined by the intrinsic camera parameters obtained from calibration, π^{-1} represents the inverse projection function, I_k represents the intensity image collected at time k, $\overline{R}$ is the image region where the depth d_u is known at time $k-1$ and the back-projected points are visible in the current image:

$$\overline{R} = \{u | u \in R_{k-1} \wedge \pi(T \cdot \pi^{-1}(u, d_u)) \in \Omega_k\} \quad (3)$$

where Ω is the image domain. For the sake of simplicity, we assume that the intensity residuals are normally distributed with a unit variance. We denote the small patch of 4×4 pixels around the feature point with vector $I(u_i)$, and find the camera pose to minimize the photometric error of all patches:

$$T_{k,k-1} = \arg\min_{T_{k,k-1}} \frac{1}{2} \sum \|\delta I(T_{k,k-1}, u_i)\|^2, \forall u_i \in \overline{R}. \quad (4)$$

Since Eq. (4) is nonlinear in $T_{k,k-1}$, it is solved by an iterative Gauss-Newton procedure. For feature alignment, it is an optimization step for $T_{k,k-1}$ estimated in the last step. Through back-projection, the obtained relative pose $T_{k,k-1}$ implicitly defines an initial guess for feature positions of all visible 3D points in the new image. However, these 3D point positions are not sufficiently accurate, i.e., the camera pose and the initial point position should be further improved. To reduce drifting, the camera pose should be aligned with the fixed map rather than its previous frame. Specifically, all 3D points of the map that are visible from the estimated camera pose are projected into the image, resulting in an estimate of the corresponding 2D feature positions u. We then optimize all 2D feature positions u' in the new image individually by minimizing the photometric error of the patch in the current image with respect to the reference patch in the reference frame r (keyframe and assist-frame):

$$u_i' = \underset{u_i}{\operatorname{argmin}} \frac{1}{2} \left\| I_k(u_i' - A_i \cdot I_r(u_i)) \right\|^2, \forall i \quad (5)$$

This alignment is solved using the inverse compositional Lucas-Kanade algorithm [16]. Different from the previous step, we apply an affine warping A_i to the reference patch, since a larger patch size is used (8×8 pixels) and the closest keyframe and assist-frame is typically further away than the previous image. In this step, our proposed algorithm is significantly faster than the SVO algorithm since our map is built only from two reference frames.

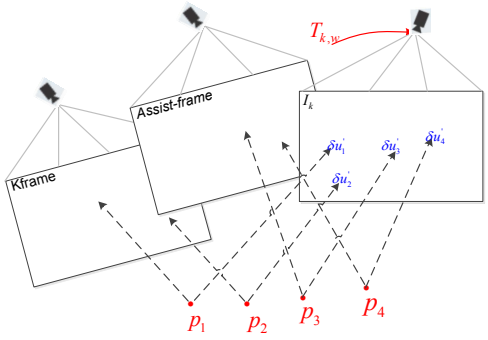

Fig. 3: Pose and structure optimization.

In this step, the camera pose and the structure (3D points) are optimized to minimize the reprojection error that has been established during the previous feature alignment step.

Actually, a reprojection residual that is not equal to zero is produced (See Fig. 3). In the final step, we again optimize the camera pose from the world coordinate frame to the camera frame of reference $T_{k,w}$ by minimizing the reprojection residuals. This is the well-known motion-only BA problem [23] and can efficiently be solved using an iterative non-linear least square minimization algorithm, such as Gauss Newton.

2.3. Relocalization

Relocalization is a part of fixed map generation. In each keyframe, we perform relocation and rebuild a fixed map. We select a keyframe by checking the overlap between the reference frames and the current frame. The overlap is defined as the number of tracked features and the number of 3D points projected from the fixed map. If one of these two numbers is less than a given threshold, we determine the current position of MAV using its previous position and select the next frame as a keyframe. In the proposed algorithm, we set new reference position at each keyframe if the position is lost, for example caused by image blur and large motion. Consequently, the robustness and the accuracy of the proposed algorithm can be improved.

3. EXPERIMENTS AND COMPARISONS

To test the performance of the proposed algorithm, we run our algorithm on the EuRoC dataset [24] and the KITTI dataset [25]. We use the same evaluation criterion as [24, 25] to test our method.

3.1. Results on the EuRoC Dataset

Since the FSVO algorithm is mainly proposed for MAVs, we therefore test our pipeline on the EuRoC MAV dataset. The EuRoC dataset is collected on-board by a MAV, it contains synchronized stereo images, IMU measurements, extrinsic and intrinsic calibrations and groundtruth. Given the estimated position (x_e, y_e, z_e) and attitude (θ_e, φ_e, ϕ_e), the

groundtruth position (x_g, y_g, z_g) and attitude $(\theta_g, \varphi_g, \phi_g)$, the position estimation error ε_p and the attitude estimation error ε_a are defined as follows:

$$\varepsilon_p = \sqrt{(x_e - x_g)^2 + (y_e - y_g)^2 + (z_e - z_g)^2} \quad (6)$$

$$\varepsilon_a = \sqrt{(\theta_e - \theta_g)^2 + (\varphi_e - \varphi_g)^2 + (\phi_e - \phi_g)^2} \quad (7)$$

In this section, we mainly test the performance of our method on images acquired by a downward-looking camera. The position and attitude error results are shown in Figs. 4-5. For comparison, we use the same parameters for FSVO and SVO. The average position/rotation estimation error of SVO and FSVO is 0.0783m/0.0262rad and 0.0554m/0.0223rad, respectively. That is mainly because that a more robust and strict keyframe selection criterion and relocalization step have been used in our algorithm. Also, fixed map plays a major role in the relocalization step.

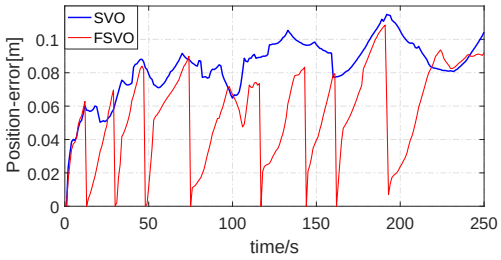

Fig. 4: Position estimation errors achieved on the EuRoC dataset.

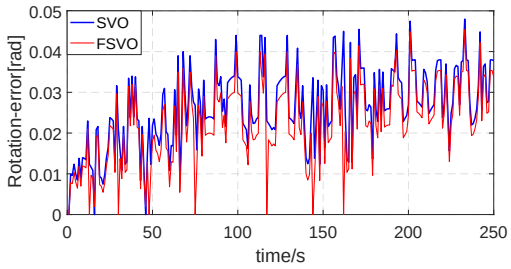

Fig. 5: Rotation estimation errors achieved on the EuRoC dataset.

3.2. Results on the KITTI Dataset

The odometry benchmark in the KITTI dataset consists of 22 stereo sequences, where 11 sequences for training have groundtruth trajectories and the other 11 sequences for evaluation do not have groundtruth. We use the frames acquired by the left camera from the first 11 stereo sequences and their corresponding groundtruth trajectories to test our method.

This experiment is mainly designed to test the performance of our method on images acquired by a forward-looking camera. The position and attitude errors are shown in Figs. 6-7. The average position/rotation estimation error of SVO and FSVO is 0.0996m/0.0711rad and 0.0662m/0.0464rad, respectively. On the KITTI dataset, SVO achieves worse performance as compared to that achieved on the EuRoC dataset. There might be two reasons for this observation. First, the keyframe selection criterion of the SVO algorithm is designed for downward looking cameras rather than forward looking cameras. Second, there are too many turns in the KITTI dataset where the scene cannot be timely reconstructed by the SVO algorithm. The keyframe selection criterion of our algorithm works well on the KITTI dataset and the relocalization step can reduce drifting when a turn occurs. Therefore, the proposed algorithm is more precise and robust than the SVO algorithm.

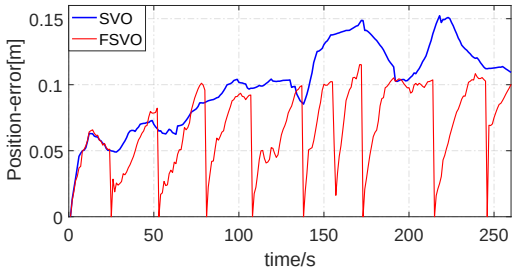

Fig. 6: Position estimation errors achieved on the KITTI Dataset.

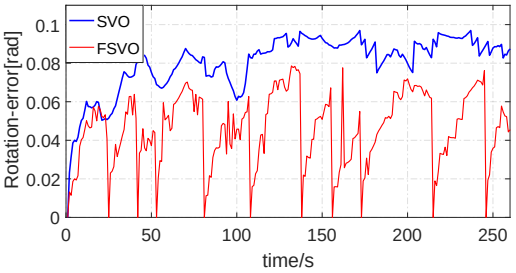

Fig. 7: Rotation estimation errors achieved on the KITTI Dataset.

4. CONCLUSION

In this paper, we propose a semi-direct VO framework using fixed maps. Our algorithm is based on fixed maps rather than incremental maps and proposes a new keyframe selection criterion and a relocation approach. It has no limit on the orientation of cameras. Experimental results on the EuRoC and KITTI datasets show that it is more accurate and robust than the SVO algorithm.

5. ACKNOWLEDGMENT

This work was partially supported by the National Natural Science Foundation of China (Nos. 61602499 and 61471371), and the National Postdoctoral Program for Innovative Talents (No.BX201600172).

6. REFERENCES

[1] C. Kanellakis and G. Nikolakopoulos, "Survey on Computer Vision for UAVs: Current Developments and Trends," *IJRS*, pp. 1–28, 2017.

[2] C. Forster, M. Faessler, F. Fontana, M. Werlberger, and D. Scaramuzza, "Continuous on-board monocular-vision-based elevation mapping applied to autonomous landing of micro aerial vehicles," in *IEEE ICRA*, 2015, pp. 111–118.

[3] S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular Vision for Long-term Micro Aerial Vehicle State Estimation: A Compendium," *J FIELD ROBOT*, vol. 30, no. 5, pp. 803–831, 2013.

[4] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip *et al.*, "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments," *IEEE ROBOT AUTOM MAG*, vol. 21, no. 3, pp. 26–40, 2014.

[5] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular slam with multiple micro aerial vehicles," in *IEEE/RSJ IROS*, 2013, pp. 3962–3970.

[6] C. Forster, M. Pizzoli, and D. Scaramuzza, "Air-ground localization and map augmentation using monocular dense reconstruction," in *IEEE/RSJ IROS*, 2013, pp. 3971–3978.

[7] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," in *IEEE ICRA*, 2010, pp. 21–28.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *IEEE ICCV*, 2011, pp. 2564–2571.

[9] T. Tykkälä, C. Audras, and A. I. Comport, "Direct iterative closest point for real-time visual odometry," in *IEEE ICCV*, 2011, pp. 2050–2056.

[10] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *IEEE ICRA*, 2013, pp. 3748–3754.

[11] M. Irani and P. Anandan, "About direct methods," in *International Workshop on Vision Algorithms*. Springer, 1999, pp. 267–277.

[12] S. Lovegrove, A. J. Davison, and J. Ibanez-Guzmán, "Accurate visual odometry from a rear parking camera," in *Intelligent Vehicles Symposium*, 2011, pp. 788–793.

[13] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *IEEE ICCV*, 2011, pp. 2320–2327.

[14] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *ECCV*. Springer, 2014, pp. 834–849.

[15] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE ICRA*, 2014, pp. 15–22.

[16] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *IJCV*, vol. 56, no. 3, pp. 221–255, 2004.

[17] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," in *Computer graphics forum*, vol. 26, no. 2. Wiley Online Library, 2007, pp. 214–226.

[18] J. M. Blackall, G. P. Penney, A. P. King, and D. J. Hawkes, "Alignment of sparse freehand 3-D ultrasound with preoperative images of the liver using models of respiratory motion and deformation," *IEEE Trans.Med.Imaging*, vol. 24, no. 11, pp. 1405–1416, 2005.

[19] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE T ROBOT*, vol. 31, no. 5, pp. 1147–1163, 2015.

[20] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *IEEE ICRA*, 2011, pp. 3607–3613.

[21] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustmenta modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.

[22] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Generic and real-time structure from motion using local bundle adjustment," *IMAGE VISION COMPUT*, vol. 27, no. 8, pp. 1178–1193, 2009.

[23] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" in *IEEE ICRA*, 2010, pp. 2657–2664.

[24] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The IJRR*, vol. 35, no. 10, pp. 1157–1163, 2016.

[25] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The IJRR*, vol. 32, no. 11, pp. 1231–1237, 2013.