Improved UKF-SLAM with Lie Group Operation and Robust Feature Tracking for Motion Vehicles

Xin Huang1,&, Zuoxian Liang1,&, Kai Zhang1 and Pingyuan Liu2,*

1 Shenzhen International Graduate School, Tsinghua University, Beijing, China.
2 Machine Patrol Management Center, Guangdong Power Grid Corporation Ltd., Guangzhou, China
(&These authors contributed equally to this work and should be considered co-first authors)

*Corresponding author email: plainliu@139.com

Abstract. This paper proposes an improved simultaneous localization and mapping (SLAM) algorithm based on tightly coupled camera images and IMU data, which provides accurate and robust localization for autonomous vehicles and unmanned aerial vehicles (UAV), especially for those in GPS-denied environments. Many research efforts have demonstrated the effectiveness of fusing camera images and inertial data with the Unscented Kalman filter (UKF), but there is still one tricky problem about the non-linearity of the kinematics of rotations. To address this issue, we propose a novel UKF-SLAM approach by rebuilding system and measurement models based on the Lie group and Lie algebra, which obtains state estimates with reasonably high accuracy. Besides, we also offer a new method to handle corner matching outliers, which only causes slightly additional computation costs but eliminates outliers and enhances corner tracking robustness. Results from extensive experimental data have validated the effectiveness of the proposed approach, and this method also achieves comparable precision to the state-of-art.

Keywords: SLAM; Kalman Filter; Sensor Fusion.

1. Introduction

High-precision positioning, a fundamental element of environment perception and motion planning, is one of the core modules of a self-driving car and an intelligent transportation system. Building a precise and robust localization system with low-cost sensors is still challenging despite many decades of research. Consumer-grade GPS signals may fail in some scenarios, and the LiDAR is more expensive than the camera. Camera images can work as the primary source of environmental information, but a pure visual localization system is usually vulnerable to some image changes due to uneven illumination or viewpoint changes. Therefore, the combination of low-cost visual and inertial measurements[1] is quite a reasonable solution to this problem.

We can recover rotary information of the moving body and the absolute scale of the environment from inertial measurements, which can be acquired by a low-cost and commercially widely used sensor. However, the estimated rotation is noisy and even diverges. A tightly coupled approach of both optical and inertial measurements is adopted to enhance the accuracy and robustness of the system to overcome this problem.

Recent studies have two primary methods to fuse visual observation and inertial measurement, including the Kalman filter and the optimization-based method. In the first group, either loosely-coupled[1] or tightly-coupled[2] based on Kalman filters at different sensor rates is adopted. In the second group, the problems are usually modelled as non-linear equations, costing heavy computation and generally solved...
by the Levenberg–Marquardt algorithm[3]. While iterative calculation can improve the accuracy, some extra filters must be employed to deal with inertial measurements. This paper aims to obtain accurate and reliable estimated poses of mobile vehicles based on the Kalman filter method. It is widely known that extended Kalman filter (EKF) achieves reasonably satisfactory performance in most cases of motion estimation. Nevertheless, it sometimes brings about unneglected estimation errors and even diverges the system because of the linearization errors. And Unscented Kalman filter (UKF) uses sigma points to fit the probability density function of state variables instead of linearization methodology[4]. Moreover, the kinematics of rotations, which are usually represented in quaternions, cause the models’ non-linearity [5]. The solution is to convert the quaternions to vectors with three elements by Lie group operation. Besides, visual information is of vital importance to the accuracy of the SLAM system, but it is susceptible to outliers. We make some modifications in optical flow tracking to make the system more practical in particular scenarios to address the issue. The main contributions of this paper are: Firstly, we show how to transform quaternions into a three-component vector using the Lie group operation. Secondly, we rebuild system and measurement models based on the Lie group and Lie algebra for UKF to cope with visual and inertial data. In addition, we propose a new optical flow tracking method to eliminate outliers and enhance corner tracking robustness. The structure of the paper is as below: Section 2 briefly gives some basic knowledge about the system, and then the design of UKF based on Lie group operation in detail is shown in Section 3. Section 4 describes the back projection descriptor optical flow tracking. Finally, Section 5 analysis the results in detail, and Section 6 concludes.

2. Preliminaries
We begin by briefly introducing those models used in propagation and update steps.

2.1. State Vector and System Model
Assuming that the biases of IMU can be modelled as random walks and there are $n$ landmarks that have been visually tracked. Then we define the state vector as below:

$$\chi = [R^T \quad v^T \quad x^T \quad b_g^T \quad b_a^T \quad p_1^T \ldots \quad p_n^T]$$

where $R \in SO(3), v \in \mathbb{R}^3, x \in \mathbb{R}^3$ represent the orientation, position, velocity of the body respectively, $b_g \in \mathbb{R}^3$ and $b_a \in \mathbb{R}^3$ indicate the IMU biases, $p_1, \ldots, p_n \in \mathbb{R}^3$ are the 3D positions of landmarks in the global frame. The kinematics of the system are:

$$\dot{R} = R \left( \omega - b_g + \eta_g \right)$$

$$\dot{v} = R \left( a - b_a + \eta_a \right) + g$$

$$\dot{x} = v, \quad \dot{b}_g = \eta_{bg}, \quad \dot{b}_a = \eta_{ba}$$

$$\dot{p}_i = 0, i = 1, \ldots, n$$

Generally, the IMU measurements $w$ and $a$ are inputs of the system with noise and biases, and $\eta = \begin{bmatrix} \eta_g^T & \eta_a^T & \eta_{bg}^T & \eta_{ba}^T \end{bmatrix}^T$ is the system noise.

2.2. Measurement Model
In this work, visual information from images usually contains a set of landmarks that have been observed and tracked during a specific period. Landmark $p_i$ observed from the standard perspective projection model can be written as follow:
where \( y_i \) is the measurement of the landmark expressed in a normalized pixel form corresponding to camera frame, that is,

\[
\begin{bmatrix}
u_i \\
v_j \\
\omega_i 
\end{bmatrix} = \mathbf{R}^B \mathbf{r} \left( \mathbf{p}_i - \mathbf{x}^B \right) - \mathbf{x}^B 
\]

The above equation formulates how to get the distance in camera frame between the observed landmark and camera, where \( \mathbf{R}^B \) and \( \mathbf{x}^B \) represent the previously computed rotation matrix and the translation from frame \{A\} to the frame \{B\} respectively. Besides, \( \mathbf{n}_{y_i} \) denotes the measurement noise, which is Gaussian distribution with zero mean error.

### 3. UKF-SLAM Based on Lie Group Operation

This section rebuilds the UKF-SLAM system on the ground of the above content.

#### 3.1. Conventional UKF

UKF is appropriate for dealing with non-linear problems, approximating the possible distribution of random variables by sigma points rather than linearizing the issues. In this way, it is easier to propagate the sigma points with the non-linear kinematics and recover the mean and covariance with the weighted sum.

The augmentation of the state vector is defined as

\[
\chi_a = [\chi^T \; \mathbf{n}_B]^T
\]

where the dimension of state vector \( \chi \) is \( S \) and the size of augmented state is \( N = S + 12 \) accordingly. So the sigma points are determined as

\[
\begin{align*}
\sigma_0 &= \chi_a \\
\sigma_i &= \chi_a + \left[ \sqrt{\lambda + N} \mathbf{P}_a \right]_{ii}, i = 1, 2, \ldots, N \\
\sigma_i &= \chi_a - \left[ \sqrt{\lambda + N} \mathbf{P}_a \right]_{ii}, i = N + 1, \ldots, 2N
\end{align*}
\]

With

\[
P_a = \begin{bmatrix} P_{5 \times S} & 0_{5 \times 12} \\ 0_{12 \times S} & Q_{12 \times 12} \end{bmatrix}, \quad Q = \begin{bmatrix} Q_g & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & Q_a & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & Q_{b} \end{bmatrix}_{12 \times 12}
\]

where \( \sqrt{P_a} \), the square root of the matrix \( P_a \), can be computed with Cholesky decomposition, operator \([\cdot]\) means to get the vector of the ith column of the matrix.

Each sigma point is propagated using the system model function \( f \) (augmented), which has been described in Section 3.1, then we can calculate a priori state estimate.

\[
\hat{x}_{k \mid k-1} = \sum_{i=0}^{2N} w_i f(\sigma_i), \quad \hat{P}_{k \mid k-1} = \sum_{i=0}^{2N} w_i \left( f(\sigma_i) - \hat{x}_{k \mid k-1} \right) \left( f(\sigma_i) - \hat{x}_{k \mid k-1} \right)^T
\]

Where

\[
w_0 = \frac{\lambda}{\lambda + N}, w_i = \frac{\lambda}{2(\lambda + N)}, \lambda = \alpha^2(N + \beta) - N
\]
Usually, $\alpha$ is set to be a small value ($10^{-3}$), and $\beta$ will be 2 if the state is Gaussian distribution. Similarly, the measurements are updated through measurement model function $h$ which has been described in Section 3.2, and some cross-covariance matrix is obtained as well.

$$\hat{z}_k = \sum_{i=0}^{2N} w_i h(\sigma_i)$$

$$P_{xz} = \sum_{i=0}^{2N} w_i \left( f(\sigma_i) - \hat{z}_{k|k-1} \right) \left( h(\sigma_i) - \hat{z}_k \right)^T$$

$$P_{zz} = \sum_{i=0}^{2N} w_i \left( h(\sigma_i) - \hat{z}_k \right) \left( h(\sigma_i) - \hat{z}_k \right)^T$$

Finally, the Kalman gain $K_k$ and the posterior state estimate $\chi_k$ as long as its corresponding covariance $P_k$ can be computed as

$$K_k = P_{xz} P_{zz}^{-1}$$

$$\chi_k = \hat{x}_{k|k-1} + K_k (z_k - \hat{z}_k)$$

$$P_k = \hat{P}_{k|k-1} + K_k P_{zz} K_k^T$$

### 3.2. Improved UKF with Lie Group Operation

A Lie group $G$, an n-dimensional matrix, is a special kind of group characterized by continuity, holding the following properties:

$$\forall G_1, G_2 \in G, G_1 \cdot G_2 \in G$$

$$\forall G_1, G_2, G_3 \in G, (G_1 \cdot G_2) \cdot G_3 = G_1 \cdot (G_2 \cdot G_3)$$

Also, rotation matrix $R \in SO(3)$, transformation matrix $T = \begin{bmatrix} R & p \\ \mathbf{0}^T & 1 \end{bmatrix} \in SE(3)$, where $SO(3) \in G, SE(3) \in G$.

A Lie algebra $\Phi$, an n-dimensional vector space, is computed from a Lie group $G$ and the link between a Lie group $G$ and its associated Lie algebra $\Phi$ is the exponential and logarithm map. More specifically, the logarithm map transforms a rotation matrix $R \in SO(3)$ to a skew-symmetric matrix, and the exponential map transforms the Lie algebra to the rotation matrix $R \in SO(3)$ inversely.

$$\phi^* = \log(R), \quad \Phi = \log(R)^*$$

The operator $\left( \cdot \right)^*$ represents the mapping from a 3-dimensional vector to a skew-symmetric matrix, and the operator $\left( \cdot \right)^*$ is the reverse.

It is necessary to pull in a new definition of Gaussian distribution from [6] to improve UKF with Lie group operation, that is

$$\Theta(X) = a e^{\frac{1}{2} \log_G \left( X^* \right) \log_G \left( \sigma^* \right)}$$

where $e = \log_G \left( X^* \right)$ is Gaussian distribution in traditional Euclidean space with mean $\mathbf{0}$ and covariance $P$. This equation relates the Gaussian distribution to the Lie group $G$ around the identity and with the exponential map,

$$X = \mu \exp_G (\hat{\epsilon})$$

Therefore, the Kalman filter can relate to $SE(3)$. 

The operator $\left( \cdot \right)^*$ represents the mapping from a 3-dimensional vector to a skew-symmetric matrix, and the operator $\left( \cdot \right)^*$ is the reverse.
\[
\begin{align*}
\chi_{SE(3)}^k &= T^w_C \left[ \exp_{SE(3)} \left( R^C_w \chi_{SE(3)}^{k-1} \right) \right] \\
\sigma_{SE(3)_h} &= \chi_{SE(3)} \\
\sigma_{SE(3)_h} &= \chi_{SE(3)} \exp_{SE(3)} \left( \sqrt{(\lambda + N) P_a} \right), i = 1, 2, \ldots, N \\
\sigma_{SE(3)_h} &= \chi_{SE(3)} \exp_{SE(3)} \left( -\sqrt{(\lambda + N) P_a} \right), i = N + 1, \ldots, 2N
\end{align*}
\]

Then these sigma points \( \sigma_{SE(3)_h} \) are propagated through the system model, but the mean is probably not unique, and we need to adjust the covariance to the manifold on \( SE(3) \).

\[
\hat{\mu}_{SE(3)} = \arg\min_{\rho \in SE(3)} \sum_{i=0}^{2N} w_i d^2 \left( f \left( \sigma_{SE(3)_h} \right), \rho \right)
\]

\[
\hat{P}_{SE(3)} = \sum_{i=0}^{2N} w_i \log_{\mu_{SE(3)}} \left( f \left( \sigma_{SE(3)_h} \right) \right) log_{\mu_{SE(3)}} \left( f \left( \sigma_{SE(3)_h} \right) \right)^T
\]

where operator \( d(\cdot) \) represents the geodesic distance on \( SE(3) \).

Besides, those cross-covariance matrices during the measurement update step in our filter should be transformed with the Lie group as below.

\[
\hat{2}_{SE(3)} = \arg\min_{\rho \in SE(3)} \sum_{i=0}^{2N} w_i d^2 \left( h \left( \sigma_{SE(3)_h} \right), \rho \right)
\]

\[
\hat{P}_{z \mid SE(3)} = \sum_{i=0}^{2N} w_i \log_{\mu_{SE(3)}} \left( f \left( \sigma_{SE(3)_h} \right) \right) log_{\mu_{SE(3)}} \left( h \left( \sigma_{SE(3)_h} \right) \right)^T
\]

Hence, we reformulate the UKF with Lie group operation.

\[
\begin{align*}
\hat{K}_{z \mid SE(3)} &= \exp_{SE(3)} \left( P_{z \mid SE(3)}^{-1} \hat{2}_{SE(3)} \right) \\
\chi_{k \mid SE(3)} &= \hat{\mu}_{SE(3)} + \exp_{SE(3)} \left( K_{z \mid SE(3)} \right) log_{\mu_{SE(3)}} \left( \hat{2}_{SE(3)} \right) \\
P_{k \mid SE(3)} &= \hat{P}_{SE(3)} + \exp_{SE(3)} \left( K_{z \mid SE(3)} P_{z \mid SE(3)} K_{z \mid SE(3)}^T \right)
\end{align*}
\]

A conceptual summary of the improved UKF on \( SE(3) \) mainly contains propagation and measurement steps as is below:

- Based on the current state vector, extend the state on \( SE(3) \) and generate corresponding sigma points.
- Propagate sigma points through the non-linear system function \( f \) and obtain a priori state estimate.
- Predict new locations of the current landmark tracks using measurement function \( h \) and determine the cross-covariance.
- Use the measurement residual and Kalman gain to correct the prior state estimate, which we call posterior state estimate.
- Repeat as necessary.

4. Back-projection-Descriptor Optical Flow Tracking

In this section, a new optical flow feature tracking method that combines back-projection and descriptor is proposed. Generally, there are two main categories in the front-end of the V-SLAM systems: conventional optical flow[7] and descriptor-based methods[8]. However, there are often wrong matches, even adopting some eliminating algorithms such as RANSC, and the procedure to calculate descriptors
are time-consuming, both of which have some sick impact on system performance. Yang et al.[9], based on the work of ORB-SLAM[10], proposed a sparse optical flow algorithm. It is inspiring to adopt ORB features to correct the corner coordinates in the image frame for optical flow tracking. In this work, we also choose the ORB descriptors since it is time-efficient. Different from Yang’s method, we conduct fast-tracking using optical flow before back-projection and calculating descriptors. The conventional optical flow is based on corner detection (usually Shi-Tomasi) and optical flow tracking, which we also employ in our work. After matching corners, we back-project corners onto the former image, calculate the feature angles for matched pairs, and compute the ORB descriptors. Nextly, ORB descriptor distances of feature pairs can be calculated, which can be employed to filter out wrong matches, as shown in Table 1.

Table 1. Step of algorithm 1.

| ALGORITHM 1 Back-projection-Descriptor Optical Flow Tracking Algorithm |
|-------------------------------------------------|
| **Input** serial image sequences |
| 1. Detect corner and track the optical flow |
| 2. Back-project matched pairs |
| 3. Calculate the feature angle and compute ORB descriptor |
| 4. Filter out wrong matches |
| For matched pairs with descriptor distances lower than the threshold, classify them as inliers. |
| For matched pairs with descriptor distances higher than the threshold, classify them as outliers. |
| **Output** matched features pairs |

5. Experimental Results
This section uses the public dataset EuRoC[11] to evaluate our approach qualitatively and quantitatively. Comparison between other SOTA methods and our method is performed using EuRoC on a PC, including a 3.60 GHz dual-core Intel Core i7 processor.

5.1. Feasibility Experimental Analysis
We run the proposed SLAM system, and experimental results show that the proposed method is feasible. Figure 1 shows the estimated trajectories (after alignment) and corresponding ground truth, and Figure 2 shows the position error of the XYZ axis of 4 EuRoC data. The proposed method is pretty good at positioning, and the estimated trajectories are roughly consistent with the ground truth.
Figure 1. Results of 4 EuRoC sequences and the yellow part represent the positioning error.

Figure 2. Position error of XYZ axis of 4 EuRoC sequences.
5.2. Experimental Analysis of Refinement of Optical Flow Tracking

To illustrate the refinement of optical flow tracking with back-projection-descriptor compared to the conventional ones, we run each EuRoC sequence 30 times. The average means and standard deviations are recorded in Table 2-4. After adopting back-projection and descriptor, the estimator is superior to the conventional one in most sequences, where outliers are eliminated effectively. As for the processing time in Table 5-7, it can be concluded that our new method only takes little extra computation. The processing time-varying from sequences to sequences strongly depends on the speed of motion. There are more features to track that can be used to calculate descriptors for slow motion, while the quick motion is the opposite.

The RMSEs for each sequence compared to optical flow tracking are as follows.

### Table 2. RMSEs (m) for MH sequence.

| Sequence       | MH_01_easy | MH_02_easy | MH_03_medium | MH_04_difficult | MH_05_difficult |
|----------------|------------|------------|--------------|-----------------|-----------------|
|                | mean       | std        | mean         | std             | mean            |
| conventional   | 0.179      | 0.069      | 0.207        | 0.058           | 0.244           |
| proposed       | 0.142      | 0.052      | 0.194        | 0.047           | 0.245           |

### Table 3. RMSEs (m) for V1 sequence.

| Sequence       | V1_01_easy | V1_02_medium | V1_03_difficult |
|----------------|------------|--------------|-----------------|
|                | mean       | std          | mean            |
| conventional   | 0.118      | 0.021        | 0.058           |
| proposed       | 0.094      | 0.013        | 0.065           |

### Table 4. RMSEs (m) for V2 sequence.

| Sequence       | V2_01_easy | V2_02_medium | V2_03_difficult |
|----------------|------------|--------------|-----------------|
|                | mean       | std          | mean            |
| conventional   | 0.173      | 0.056        | 0.327           |
| proposed       | 0.132      | 0.039        | 0.246           |

The average processing time of the proposed method for each sequence is as follows. The maximum feature number is set to 100.

### Table 5. The average processing time (ms) of the proposed method for MH sequence.

| Sequence       | MH_01_easy | MH_02_easy | MH_03_medium | MH_04_difficult | MH_05_difficult |
|----------------|------------|------------|--------------|-----------------|-----------------|
| Process time   | 0.8816     | 0.9854     | 0.8607       | 0.9002          | 0.7887          |

### Table 6. The average processing time (ms) of the proposed method for the V1 sequence.

| Sequence       | V1_01_easy | V1_02_medium | V1_03_difficult |
|----------------|------------|--------------|-----------------|
| Process time   | 0.8497     | 0.7775       | 0.6158          |

### Table 7. The average processing time (ms) of the proposed method for the V2 sequence.

| Sequence       | V1_01_easy | V1_02_medium | V1_03_difficult |
|----------------|------------|--------------|-----------------|
| Process time   | 0.8349     | 0.6892       | 0.4370          |

5.3. Comparison between Other Methods

In our experiments, those SOTA methods include VINS-MONO[12], ROVIO[13], and normal UKFSLAM[5]. The proposed method is entirely accurate compared to other methods, as shown in Table 8-10. VINS-MONO primarily performs best among tested algorithms, but it uses batch optimization, and the proposed method also achieves outstanding performance in some sequences. It is worth mentioning that quick motion (such as V1_03 and V2_03) makes it harder to track features continuously,
which leads to unsatisfying positioning results. The proposed method, Normal UKF-SLAM and ROVIO, are all filters for state estimation in essence, which can use IMU measurements only to initialize the system, while VINS-MONO should wait for more tracked features to relocate.

Table 8. RMSEs (m) of the state-of-art algorithms and our proposed method on EuRoC dataset for MH sequence.

| Sequence     | MH_01_easy | MH_02_easy | MH_03_medium | MH_04_difficult | MH_05_difficult |
|--------------|------------|------------|--------------|-----------------|-----------------|
| VIN-MONO     | 0.114      | 0.186      | 0.202        | 0.257           | 0.254           |
| ROVIO        | 0.172      | 0.521      | 0.386        | 0.522           | 0.418           |
| NormalUKF    | 0.284      | 0.199      | 0.246        | 0.490           | 0.351           |
| Proposed     | 0.139      | 0.170      | 0.243        | 0.323           | 0.261           |

Table 9. RMSEs (m) of the state-of-art algorithms and our proposed method on EuRoC dataset for V1 sequence.

| Sequence     | V1_01_easy | V1_02_medium | V1_03_difficult |
|--------------|------------|--------------|-----------------|
| VIN-MONO     | 0.072      | 0.089        | 0.117           |
| ROVIO        | 0.190      | 0.146        | 0.099           |
| Normal UKF-SLAM | 0.106    | 0.108        | 0.089           |
| Proposed     | 0.096      | 0.061        | 0.071           |

Table 10. RMSEs (m) of the state-of-art algorithms and our proposed method on EuRoC dataset for V2 sequence.

| Sequence     | V2_01_easy | V2_02_medium | V2_03_difficult |
|--------------|------------|--------------|-----------------|
| VIN-MONO     | 0.110      | 0.192        | 0.244           |
| ROVIO        | 0.542      | 0.440        | 0.182           |
| Normal UKF-SLAM | 0.113    | 0.185        | 0.240           |
| Proposed     | 0.140      | 0.258        | 0.166           |

6. Conclusions
This paper has refined the optical flow tracking procedure using back-projection and descriptor, eliminating wrong feature pairs effectively and taking little additional computation. It also reformulated UKF-SLAM based on Lie group operation to obtain state estimates with reasonably high accuracy. Results from experimental data have validated the effectiveness of the proposed approach, and this method also achieves comparable precision to the state-of-art. In the future, our work will focus on exploring the consistency properties of UKF and refining the initialization procedure that makes the algorithm more practical.

Acknowledgements
Supported by China Southern Power Grid Company Limited Science and Technology Project, Project Number: GDKJXM20190010(031000KK52190001).

References
[1] Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., & Furgale, P. (2015). Keyframe-based visual-inertial odometry using non-linear optimization. The International Journal of Robotics Research, 34(3), 314-334.
[2] Weiss, S., Scaramuzza, D., & Siegwart, R. (2011). Monocular-SLAM–based navigation for autonomous micro helicopters in GPS-denied environments. Journal of Field Robotics, 28(6), 854-874.
[3] Jones, E. S., & Soatto, S. (2011). Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. The International Journal of Robotics Research, 30(4), 407-430.
[4] Xiong, K., Zhang, H. Y., & Chan, C. W. (2006). Performance evaluation of UKF-based non-linear filtering. Automatica, 42(2), 261-270.
[5] Loianno, G., Watterson, M., & Kumar, V. (2016, May). Visual inertial odometry for quadrotors on SE (3). In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1544-1551). IEEE.

[6] Wang, Y., & Chirikjian, G. S. (2006). Error propagation on the Euclidean group with applications to manipulator kinematics. IEEE Transactions on Robotics, 22(4), 591-602.

[7] Engel, J., Schöps, T., & Cremers, D. (2014, September). LSD-SLAM: Large-scale direct monocular SLAM. In European conference on computer vision (pp. 834-849). Springer, Cham.

[8] Zheng, X., Moratto, Z., Li, M., & Mourikis, A. I. (2017, May). Photometric patch-based visualinertial odometry. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 3264-3271). IEEE.

[9] Yang, N., Wang, R., Gao, X., & Cremers, D. (2018). Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. IEEE Robotics and Automation Letters, 3(4), 2878-2885.

[10] Mur-Artal, R., & Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics, 33(5), 1255-1262.

[11] Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., ... & Siegwart, R. (2016). The EuRoC micro aerial vehicle datasets. The International Journal of Robotics Research, 35(10), 1157-1163.

[12] Qin, T., Li, P., & Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics, 34(4), 1004-1020.

[13] Bloesch, M., Omari, S., Hutter, M., & Siegwart, R. (2015, September). Robust visual inertial odometry using a direct EKF-based approach. In 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 298-304). IEEE.