2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Monocular Visual-Inertial Odometry Based on Sparse Feature Selection with Adaptive Grid

Zhiao Cai, Ming Yang, Chunxiang Wang and Bing Wang

***Abstract*— For sparse feature based visual-inertial odometry, feature selection is vital to the performance. The selected features should be evenly distributed in the image and be appropriate for tracking. This paper presents a visual-inertial odometry approach based on sparse feature selection with the adaptive grid to improve the performance in different environments. In the proposed approach, FAST corner detection is employed in every grid, the size of which is adaptively adjusted. Features in the same grid are ranked and selected based on scores to ensure good feature quality. Subsequently, the selected features are tracked by the KLT sparse optical flow with local intensity normalization. Finally, the inertial and visual measurements are jointly optimized by sliding window based nonlinear optimization to achieve six degrees of freedom motion estimation. The proposed method is validated on the public dataset and real-world experiments, which shows our approach reaches state-of-the-art performance.**

I. Introduction

Localization is one of the critical problems in autonomous navigation of robots. In general, localization can be divided into relative (local) localization and absolute (global) localization [1]. Relative localization estimates translation and orientation based on the initial position of robots and the state of motion measured by sensors. Odometry, which is one of the types of relative localization technology, provides the variety of robot poses relative to the preset initial pose. Wheel encoders and inertial measurement units (IMU) are the most common sensors used for odometry. While wheel encoders cannot be installed on every type of robots, and IMU suffers from large drift after long-time integration.

Recently, odometry based on light detection and ranging (LIDAR) and camera can reach accurate and real-time performance due to the development of algorithm and computer resources. Since camera is low-cost, meanwhile it can provide rich information of surrounding environments, odometry based on camera becomes a proper solution to relative localization, which is usually called as visual odometry (VO). Moreover, visual and inertial sensors are complementary with each other, which can compensate for the drift errors measured respectively by the two kinds of sensors. Odometry which uses combined information from visual and inertial sensors is commonly known as visual-inertial odometry (VIO).

This work was supported by the National Natural Science Foundation of China (U1764264). Ming Yang is the corresponding author.

Zhiao Cai, Ming Yang, Chunxiang Wang and Bing Wang are with the Department of Automation, Shanghai Jiao Tong University, Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, 200240, CN (phone: +86-21-34204533; email: MingYANG@sjtu.edu.cn).

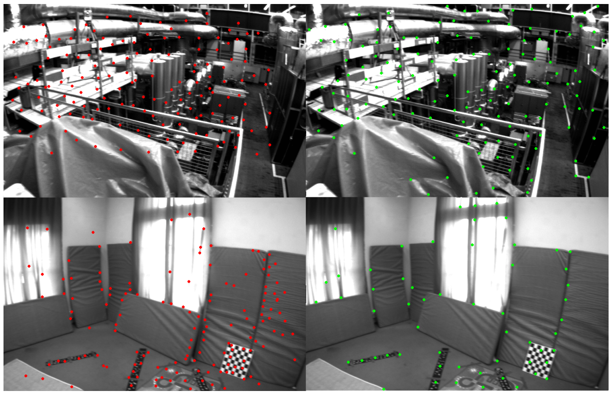


Fig. 1. Comparison between adaptive grid and fixed grid. Red points are detected in adaptive grids, and green points are detected in fixed girds. In environments with rich texture, the two methods are similar. While in texture-less environments, the method based on adaptive grid can adjust the grid size adaptively to detect more features in relatively uniform distribution, at the same time, ensure the good quality of features by score ranking. The pictures are selected from the EUROC dataset.

The process of visual-inertial odometry has two main parts: front-end and back-end [2]. In front-end, the data association of consecutive frames is obtained by either feature based methods or direct methods. Feature-based methods normally utilize feature detectors or descriptors, such as Harris, SURF, FAST and ORB, to detect and match sparse points between frames [3]. While direct methods straightforward use pixel intensities to achieve data association [4]. Feature-based methods suffer from incorrect correspondences, which can be partly eliminated by IMU data. Therefore, in the research field of VIO, feature-based methods are more commonly used. The back-end is usually filter based or optimization based, which aims to reduce the error of motion estimation.

Visual-inertial odometry is based on the integration of motion estimation, which will inevitably have drift. Even though successful loop closure can decrease the drift, low drift is one of the most essential characteristics of VIO. To reduce the drift, the data association achieved in front-end is vital, which means proper features selection and tracking is needed.

This work is inspired by a monocular visual-inertial state estimator based on sliding window optimization, VINS-mono [5], which adopts typical Harris corner detector [6] to select features and basic KLT tracker to track features.

In this paper, a visual-inertial odometry method focused on sparse feature selection in adaptive grids is proposed to reduce the drift. The main idea is to adjust the grid size

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

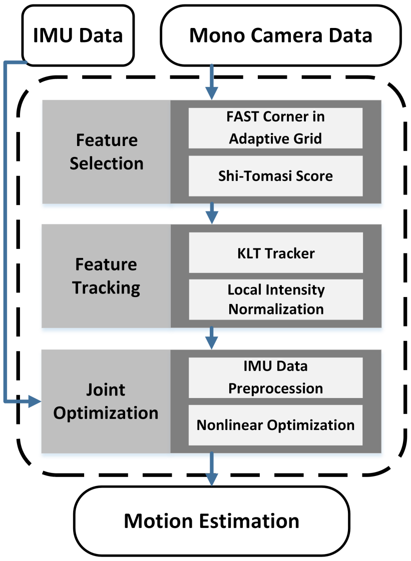


Fig. 2. Framework of the proposed algorithm

based on the number and quality of features iteratively. Fig. 2 shows the framework of the proposed method. The FAST corner detector with adaptive grid and Shi-Tomasi scoring is adopted to select features. A pyramidal implementation of the KLT tracker with local intensity normalization is employed to track selected features. The feature corresponding and inertial measurements are jointly optimized to estimate motions. The proposed approach is validated on EUROC MAV Visual-Inertial Datasets [7] and the outdoor experiment. The experiment result shows that carefully designed feature selection and tracking algorithm can improve the performance of VIO.

The remainder of this paper is organized as follows. In section II, related work is discussed. Section III describes feature selection and tracking method, which is the core part of this paper. Section IV describes the nonlinear optimization based on the sliding window. Section V shows the experiment results. Finally, the conclusion is made in Section VI.

II. Related Work

Visual-inertial odometry shares many same characteristics with visual odometry. The term visual odometry was first mentioned in [8]. PTAM [9] is one of the most famous visual odometry, which first splits tracking and mapping into two parallel threads so that optimization based back-end could operate in real time. [3] presents a comprehensive SLAM system for monocular, stereo and RGB-D cameras, which uses ORB descriptor for all SLAM tasks. [4] presents a novel visual SLAM system which utilizes direct information of images rather than traditional features.

Many contributions have been made to improve the accuracy and robustness of visual odometry and SLAM. However, robust performance is still an unsolved question of visual SLAM [2]. Since the visual and inertial sensors can provide complementary measurements to achieve robust performance, visual-inertial odometry technique is introduced.

An extended Kalman filter (EKF)-based, tightly coupled VIO algorithm is introduced in [10], which also proposes bounded sliding window method for state estimation. OKVIS [11], an optimization based, tightly coupled VIO approach is presented, which is based on nonlinear optimization to improve the performance. The back-end of OKVIS is similar to our work, while the front-end of it is based on typical FAST corner detection and BRISK descriptor, which is more time consuming and less robust to the lack of features.

Since feature-based methods and direct methods have pros and cons respectively, recent works trend to combine these two types of method. In SVO [12], features are selected by grid-based FAST corner, then a direct method is employed to align features and simultaneously estimate motions. Another work, DSO [13], aligns sparse features in the framework of direct method rather than dense features. These two approaches are similar to our work in feature selection and tracking. Features are detected by a feature-based method and tracked directly based on intensity. But the targets of optimization are difficult. SVO and DSO minimize photometric error, while the proposed algorithm minimizes reprojection error, making our method more robust to illumination change.

More recently, a robust and versatile monocular visual-inertial state estimator, VINS-mono, is introduced in [5]. VINS-mono employs sliding window based nonlinear estimation to jointly optimize visual measurements and inertial measurements, while the approach of VINS-mono to achieve data association is basic. In this paper, the approach of feature selection and tracking is especially designed, aiming to improve the accuracy of motion estimation.

III. Feature Selection and Tracking

This section describes the algorithm of feature selection and tracking, which is the core part of the proposed method. A feature selection approach is designed to ensure the balance between uniform distribution and the quality of features. KLT sparse flow is employed to track features. As a result, the selected features benefit the accuracy and robustness of motion estimation.

A. Feature Selection

For accurate motion estimation, the selected features should be evenly distributed in the whole image. In works such as [3], [5], [12], features are selected in grids with a fixed size to ensure the even distribution. While selecting features in this way may cause that selected features in texture-less grids are not suitable for tracking. To solve the problem, features are detected in grids with adaptive sizes.

The input image is evenly divided into grids with size of $s \times s$, where s is the current grid size. In initial, the current size is set to $\sqrt{H \times W/N}$, where H and W is respectively the height and width of the input image, N is the number of total required features.

Then, FAST corner detector [14] with non-maximum suppression is employed in every single grid. To ensure the uniform distribution of all feature points in the image, the detected points are first excluded according to the position. For one detected point, if the distance between it and an existed point is less than half of current grid size $s/2$, it will be eliminated.

The remaining detected features in the same grid are sorted based on Shi-Tomasi score [15]. Since features are tracked by KLT tracker, Shi-Tomasi score is a better evaluation criteria than FAST corner response. After sorting, the strongest point in every grid is selected in the current iteration.

If the number of selected features in the current iteration is less than the required number, the gird size is reduced in the next iteration. This procedure will be repeated until the maximum number of iterations is reached or the added features are enough. The detailed pseudocode of aforementioned algorithm is shown in Algorithm 1.

Algorithm 1 Feature selection based on adaptive grid

```
Input: grid size($grid\_size$), adjustment step($step$), existed
    feature($P_{existed}$), image($I$), max number of features($N$)
Output: grid size($grid\_size$), selected feature($P_{selected}$)
if not initialized then
    $grid\_size = \sqrt{sizeof(I)/N}$
end if
$n = N - sizeof(P_{existed})$
while $num\_detected < n$ and $iter < MAX\_ITER$ do
    divide I into square grids($grid\_size \times grid\_size$)
    clear $P_{selected}$
    $num\_selected = 0$
    for every girds do
        detect FAST corners
        for every detected points do
            if distance($currentpoint, P_{existed}$) $>$ $grid\_size/2$
            then
                calculate Shi-Tomasi score
                push to $P_{grid}$
            end if
        end for
        sort $P_{grid}$ by Shi-Tomasi score
        the strongest point push to $P_{selected}$
        $num\_detected = num\_detected + 1$
    end for
    if $num\_detected < n$ then
        $grid\_size = grid\_size - step$
        if $grid\_size < MIN\_SIZE$ then
            $grid\_size = MIN\_SIZE$
            break
        end if
    end if
end while
if $num\_detected >= n$ then
    $grid\_size = grid\_size + step$
end if
```

B. Feature Tracking

The selected features are tracked in consecutive frames, hence the following two hypotheses are generally satisfied, which are:

- (a) The displacement of two consecutive frames is small.
- (b) The local intensity varies little.

Therefore, KLT tracker [16] can be adopted to track features. The major benefit of achieving feature correspondences by KLT tracker, is that the feature descriptor is not necessary, which reduces the time consumption. Feature correspondences can be obtained in high frequency, which will naturally improve the performance of KLT tracker according to the aforementioned hypotheses.

The pyramidal implementation of KLT tracker is employed to track features in the image pyramid, hence it can deal with the situations which not well satisfy the hypothesis (a).

While it cannot deal with illumination change, for example, the change of exposure time. Therefore, before the standard KLT sparse algorithm is employed, a basic local intensity normalization method is applied to deal with such situations. The gain α and bias β of intensity in each pair of corresponding windows are normalized.

$$\alpha = \sqrt{\frac{\sum_{X \in \Omega_1} I_1^2(X)}{\sum_{Y \in \Omega_2} I_2^2(Y)}} \tag{1}$$

$$\beta = \frac{\sum_{X \in \Omega_1} I_1(X)}{|\Omega_1|} - \alpha \frac{\sum_{Y \in \Omega_2} I_2(Y)}{|\Omega_2|} \tag{2}$$

where I_1 and I_2 denote two consecutive frames. Ω_1 and Ω_2 denote the window to be aligned in I_1 and I_2 respectively, with a fixed size of $n \times n$. The calculated α and β is used to adjust the intensity in the second window.

$$I_{2norm}(Y) = \alpha I_2(Y) + \beta, Y \in \Omega_2 \tag{3}$$

For every pair of corresponding windows, the gain and bias are calculated and the intensity in two windows is normalized. Subsequently, KLT sparse algorithm is employed to get the feature corresponding. Due to the normalization of intensity, the tracker is less lighting sensitive.

After all the features are tracked in the next frame, the successfully tracked features are tested in the 8-point [17] RANSAC step with fundamental matrix model [18]. The lost features in tracking and outliers in the RANSAC step are rejected from the existed features.

The remaining features are transformed to the projective space based on the intrinsic camera parameters. The coordinates of these features in the projective space and their unique ids are the final result of feature selection and tracking, which serves as the visual measurement in the nonlinear optimization to estimate the motion.

IV. Tightly-Coupled Nonlinear Optimization

In this section, to estimate the motion, the visual measurement and inertial measurement are jointly optimized in the framework of sliding window based nonlinear optimization. The visual measurement is derived from section III. The inertial measurement is achieved based on IMU pre-integration method [19]. The nonlinear optimization problem is solved by Ceres Solver [20].

A. Initialization

Initialization is one of the key parts of monocular VIO. Since the scale can be indirectly derived from inertial measurement, the initialization of monocular VIO is much easier than that of monocular visual odometry. A loosely-coupled visual-inertial alignment approach [21] is adopted to initialize the system, which contains visual structure from motion (SFM) in sliding window and visual-inertial alignment.

As a result, the following information is derived in the initialization procedure: the initial bias of gyroscope and accelerometer; the initial absolute scale and velocity; the gravity vector.

B. Motion Estimation

1) Nonlinear Optimization Problem: The nonlinear optimization problem is based on the keyframes in the current sliding window, and the keyframes are selected based on the average parallax of tracked features. To deal with distant features, inverse depth parametrization is adopted to represent the position of features. The mathematical form of state vector to be estimated is shown as following:

$$\begin{aligned}\mathbf{X} &= [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x}_c^b, \lambda_0, \lambda_1, \dots, \lambda_m] \\ \mathbf{x}_i &= [\mathbf{p}_{b_i}^w, \mathbf{v}_{b_i}^w, \mathbf{q}_{b_i}^w, \mathbf{b}_a, \mathbf{b}_g] \\ \mathbf{x}_c^b &= [\mathbf{p}_b^w, \mathbf{q}_b^w]\end{aligned} \tag{4}$$

where $\mathbf{x}_i$ is the state of the frames in sliding window, which contains position in the world coordinate $\mathbf{p}_{b_i}^w$, velocity in the world coordinate $\mathbf{v}_{b_i}^w$, rotation in the world coordinate $\mathbf{q}_{b_i}^w$, acceleration bias $\mathbf{b}_a$ and gyroscope bias $\mathbf{b}_g$. $\mathbf{x}_c^b$ is the camera-IMU extrinsic, including translation and rotation. λ_j is the inverse depth of the j^{th} feature when it is first observed. n and m are the number of keyframes and the number of features in current sliding window respectively.

Maximum posteriori (MAP) estimation is employed to estimate the state vector. The residuals of all measurements are minimized, which is expressed as a nonlinear cost function:

$$\begin{aligned}\underset{\mathbf{X}}{\text{minimize}}\{&\|\mathbf{r}_p - \mathbf{H}_p\mathbf{X}\|^2 + \sum_{i\in B}\|\mathbf{r}_B(\hat{\mathbf{z}}_{b_{i+1}}^{b_i}, \mathbf{X})\|^2_{\mathbf{P}_{b_{i+1}}^{b_i}} \\ &+ \sum_{(l,k)\in C}\|\mathbf{r}_C(\hat{\mathbf{z}}_l^{c_k}, \mathbf{X})\|^2_{\mathbf{P}_l^{c_k}}\}\end{aligned} \tag{5}$$

In Eq. (5), $\|\cdot\|$ is the Mahalanobis norm, and the residuals are divided into three parts. The first part is the residual for the model of prior information, where $\mathbf{r}_p$ and $\mathbf{H}_p$ are prior information from marginalization, which will be discussed later. The second part is the residual for inertial measurements, where $\mathbf{r}_B$ is the model of IMU described in [19]. The third part is the residual for visual measurements, where visual model $\mathbf{r}_C$ is the reprojection error in the unified unit sphere [5]. Furthermore, to improve the robustness to the outliers of feature corresponding, the Cauchy loss function is employed to the residual for visual measurements. The Cauchy loss function is defined as:

$$\rho(s) = log(1+s) \tag{6}$$

2) Marginalization: The meaning of marginalization is to remove the state of frames, containing related features and IMU measurements, from the sliding window. When marginalizing out a frame, directly removing all the features and IMU measurements related to it will influence the stability of optimization [22], degrading the performance.Therefore, the marginalized states are used to construct the prior information in Eq. (5). In general, through marginalization, computation complexity is bounded, meanwhile, the valuable information of the removed frame is retained.

V. Experiment Results

In this section, two experiments are carried out to evaluate the proposed method. The first one is the comparison between our approach and the state-of-the-art algorithms based on the public dataset, the absolute trajectory error is used to analyze the accuracy. The second one is the outdoor experiment with our own visual-inertial sensor. Since ground truth is not available in the outdoor experiment, the start and end point error is adopted to evaluate performance.

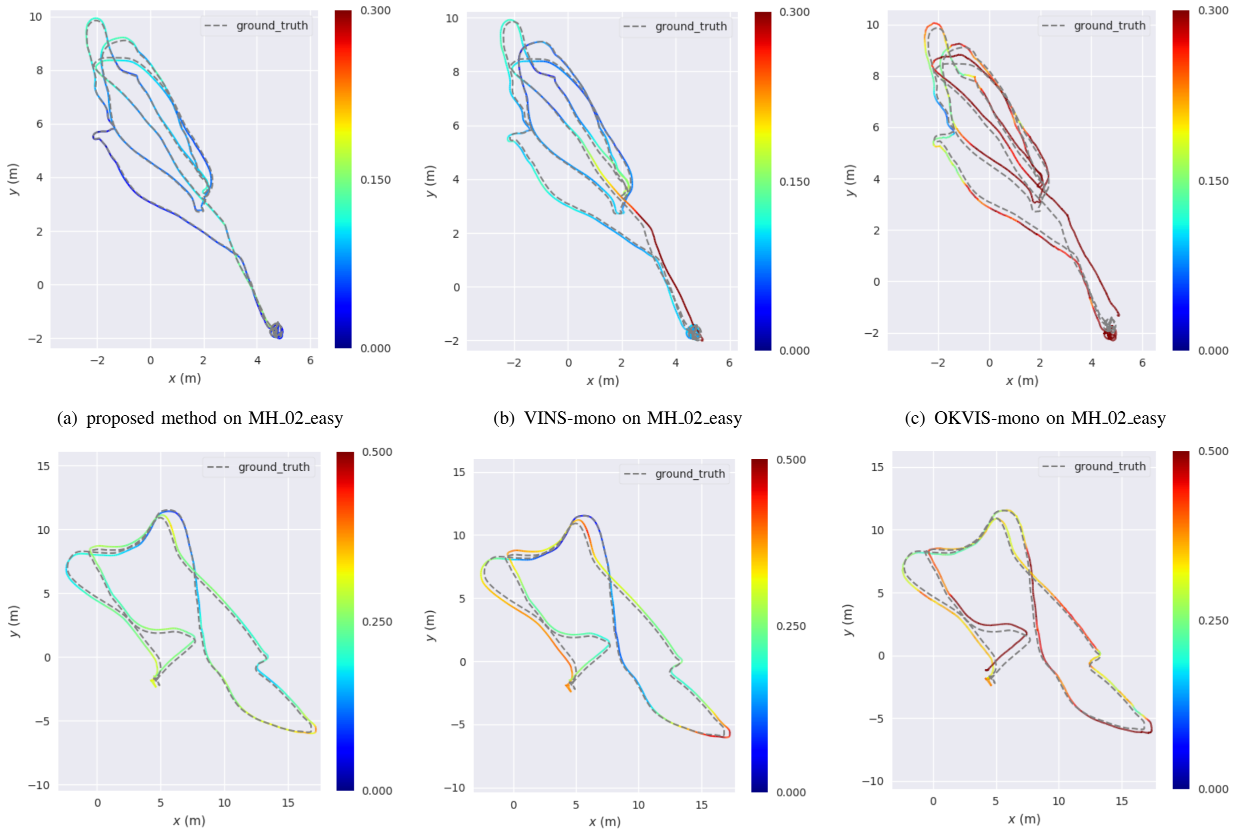
A. Dataset Comparison

The proposed algorithm is evaluated on EUROC MAV Visual-Inertial Datasets [7]. The datasets contain stereo images (Aptina MT9V034 global shutter, 20 FPS, WVGA monochrome), and accurately synchronized IMU data (ADIS16448, accelerometer and gyroscope, 200 Hz). The sensors are installed on a micro aerial vehicle. Meanwhile, millimeter accurate position ground truth is provided by the system of VICON or Leica MS50. Since the proposed method is a monocular visual-inertial odometry system, only the left image of the stereo camera is used.

The proposed method is compared with other two optimization based state-of-the-art algorithms: VINS-mono [5] and OKVIS [11]. These two methods are both open source and originally can be tested on EUROC datasets, so their default settings for EUROC datasets are used. For the sake of fairness, the same parameter setting is used for all the sequences in our method.

The proposed method and OKVIS are purely visual-inertial odometry, while VINS-mono is more complex with loop closure. To evaluate the performance of visual-inertial odometry, in this experiment, the loop closure part is removed from VINS-mono. OKVIS can work with monocular and stereo cameras, we use the result of OKVIS working with the monocular image of the left camera. It should be noticed that all the algorithms run in real time.

The root-mean-square error (RMSE) of absolute trajectory error (ATE) is adopted to represent the accuracy. To evaluate the performance equitably, the translation and rotation between the trajectory of visual-inertial and the ground truth trajectory are aligned [23]. The absolute trajectory error is calculated after alignment. The final result is shown in Table I. Furthermore, the estimated trajectories and the corresponding errors of two sequences are shown in Fig. 3. The experiment result shows that, in most of the sequences,



(a) proposed method on MH_02_easy (b) VINS-mono on MH_02_easy (c) OKVIS-mono on MH_02_easy

(d) proposed method on MH_05_difficult (e) VINS-mono on MH_05_difficult (f) OKVIS-mono on MH_05_difficult

Fig. 3. The absolute trajectory error expressed as color maps on two sequences of the EUROC dataset

the absolute trajectory error of the proposed method is lower than that of VINS-mono and OKVIS-mono.

B. Outdoor Experiment

The sensor used in the outdoor experiment is Intel RealSense ZR300, which has a fisheye camera (30 FPS, VGA, monochrome) and an inertial measurement unit (accelerometer and gyroscope, 200Hz). The camera-IMU extrinsic and time offset are calibrated by the calibration toolbox introduced in [24]. In the outdoor experiment, the proposed method is compared with VINS-mono. The loop closure part of VINS-mono is removed.

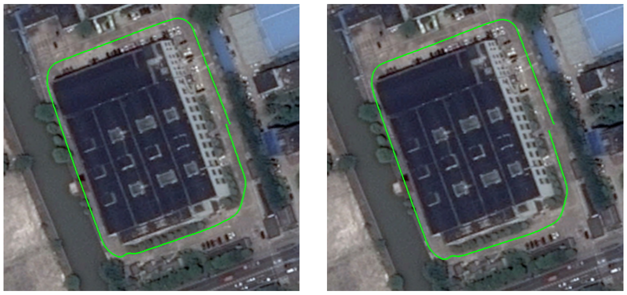
We held the sensor in hand and walked around to acquire the data of outdoor experiment. The whole trajectory is about 430 meters. To analyze the performance, the start point and end point of the trajectory is nearly the same point. Therefore, the distance between the start point and end point is adopted to evaluate the accuracy. Furthermore, the estimated trajectory is drawn on Google map. Fig. 4 shows the estimated trajectory of the proposed method and VINS-mono.

It should be noticed that the parameter settings in the outdoor experiment of these two methods are same as that in the dataset experiment. The translation between the start point and end point of our method is $[-2.504, 0.427, -1.008]$ m, which is 0.63% of the whole length. And the final

TABLE I

ABSOLUTE TRAJECTORY ERROR (RMSE) IN METERS OF THE EUROC DATASET AFTER TRANSLATION AND ROTATION ALIGNMENT WITH THE GROUND TRUTH TRAJECTORY

ATE(m) Method Sequence	Proposed Method	VINS-mono	OKVIS-mono
MH_01_easy	**0.10548**	0.14028	0.34099
MH_02_easy	**0.08287**	0.14219	0.31693
MH_03_medium	0.12093	**0.09447**	0.29895
MH_04_difficult	**0.23456**	0.26135	0.29939
MH_05_difficult	**0.25246**	0.30199	0.43321
V1_01_easy	0.05925	0.05923	N/A
V1_02_medium	**0.06687**	0.09141	0.17529
V1_03_difficult	**0.12241**	0.14512	0.22872
V2_01_easy	**0.07911**	0.08365	0.25555
V2_02_medium	**0.06482**	0.08773	0.20546
V2_03_difficult	0.17867	0.17869	0.29102



(a) proposed method

(b) VINS-mono

Fig. 4. The estimated trajectory on Google map

translation of VINS-mono is $[-4.328, -3.055, -1.925]$ m, which is 1.31% of the whole length.

C. Discussion

The feature distribution in outdoor environments is not as uniform as indoor environments, such as the EUROC dataset. That is probably the reason why the performance of our method is better than VINS-mono in the outdoor experiment, since our approach selects features in adaptive grids, while in VINS-mono, features are selected in grids with a fixed size.

Furthermore, the proposed method achieves higher accuracy in the EUROC dataset evaluation. One of the possible reasons is that selecting features in adaptive grids ensures the stable number and quality of tracked features, which can benefit the effectiveness and stability of the optimization. And the influence of the various exposure time of EUROC dataset can be reduced by the local intensity normalization.

All the experiments run on a typical laptop with Intel i5-4200H CPU. The timing analysis is shown in Table II, which shows the proposed method can run in real time on a laptop.

TABLE II
TIME ANALYSIS

	Feature Selection and Tracking	Motion Estimation
Propose Method	14.3 ms	60.8 ms
VINS-mono	10.2 ms	50.1 ms

VI. CONCLUSIONS

In this paper, a visual-inertial odometry approach based on sparse feature selection with adaptive grid is proposed to ensure the balance between uniform distribution and quality of features in various environments. The experiment shows that the proposed method improves the accuracy of motion estimation compared with the state-of-the-art methods.

REFERENCES

[1] P. Goel, S. I. Roumeliotis, and G. S. Sukhatme, "Robust localization using relative and absolute position estimates," in *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems.*, vol. 2, 1999, pp. 1134–1140 vol.2.

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec 2016.

[3] R. Mur-Artal and J. D. Tards, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct 2017.

[4] J. Engel, T. Schöps, and D. Cremers, *LSD-SLAM: Large-Scale Direct Monocular SLAM*. Cham: Springer International Publishing, 2014, pp. 834–849.

[5] T. Qin, P. Li, Z. Yang, and S. Shen, "Vins-mono," https://github.com/HKUST-Aerial-Robotics/VINS-Mono, 2017.

[6] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[7] 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 International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[8] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, June 2004, pp. I–652–I–659 Vol.1.

[9] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, Nov 2007, pp. 225–234.

[10] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 3565–3572.

[11] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visualinertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[12] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, April 2017.

[13] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[14] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, vol. 1, May 2006, pp. 430–443.

[15] J. Shi and C. Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jun 1994, pp. 593–600.

[16] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81, San Francisco, CA, USA, 1981, pp. 674–679.

[17] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, Jun 1997.

[18] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[19] F. D. C. Forster, L. Carlone and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Robotics: Science and Systems (RSS)*, Rome, 2015.

[20] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[21] Z. Yang and S. Shen, "Monocular visual-inertial state estimation with online initialization and camera imu extrinsic calibration," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 39–51, Jan 2017.

[22] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, 2010.

[23] M. Grupp, "evo: Python package for the evaluation of odometry and slam," https://github.com/MichaelGrupp/evo.

[24] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1280–1286.