A VINS Combined With Dynamic Object Detection for Autonomous Driving Vehicles

XIRU WU1, FENGTANG HUANG1, YAONAN WANG2, (Senior Member, IEEE), AND HUI JIANG1

1College of Electronic and Automation, Guilin University of Electronic Technology, Guilin 541004, China
2College of Electrical and Information Engineering, Hunan University, Changsha 410114, China

Corresponding author: Fengtang Huang (fengtanghuang520@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61863007, in part by the Guangxi Natural Science Foundation under Grant 2020GXNSFDA238029, and in part by the Innovation Project of Guilin of Electronic Technology (GUET) Graduate Education under Grant 2022YCX155 and Grant 2022YCX149.

ABSTRACT This paper proposes a visual-inertial navigation system (VINS) combined with a dynamic object detection (DOD) algorithm, to improve the localization and state estimate accuracy of autonomous driving vehicles (ADV) in dynamic environments. Firstly, based on the YOLOv5 network, we train the proposed DOD model to detect dynamic objects in the road environment. Secondly, by removing the feature points in the dynamic object regions, we track the remaining feature points to eliminate the influence of the dynamic object. Furthermore, we model the global positioning system (GPS) measurement as a general factor and introduce its residual factor into the cost function to eliminate the cumulative error. Finally, we validate the performance of the proposed method on public datasets and real-world experiments. The results show that the proposed method can effectively eliminate the influence of the dynamic object and eliminate the cumulative error. It provides theoretical guidance for ADV navigation in dynamic or large-scale outdoor environments.

INDEX TERMS Visual-inertial navigation system (VINS), autonomous driving vehicles (ADV), object detection, global positioning system (GP).

I. INTRODUCTION

In the past decade, ADV has been widely applied in the military and civilian fields [1]. Continuous and accurate positioning of vehicles is a primary requirement of the ADV system. As a basic auxiliary unit for ADV positioning, VINS has attracted the attention of many research institutions and scholars because of its simple hardware layout, small size, and low cost [2], [3], [4]. During VINS operation, visual observation and inertial measurement unit (IMU) are tightly coupled to estimate the state accurately. Therefore, VINS can achieve high-precision positioning results in an ideal environment with stable illumination conditions and sufficient texture information [5], [6], [7].

The performance of VINS is heavily dependent on the assumption that visual features are static [8], [9]. This assumption is difficult to satisfy in the actual dynamic environment. Dynamic objects such as pedestrians and vehicles in a typical urban environment are shown in Fig. 1. Since VINS hardly distinguish between static objects and dynamic ones, breaking the static-world assumption may lead to significant estimation errors. Therefore, robust methods are utilized to detect and discard outlier [10], [11]. However, the robust methods are still challenging to work stably in a...
dynamic environment. In summary, a dynamic environment undoubtedly limits the application of VINS on ADV.

Furthermore, the positioning errors accumulated largely over time limit the application of VINS too. The typical solution is loop detection and re-localization to mitigate the accumulative error. But such methods both increase computational complexity and memory requirements. Providing accurate, consistent localization by selectively fusing global pose information and local pose estimation is another solution [11]. GPS can be integrated into VINS to improve the localization accuracy and solve the drifting problem of ADV. In this paper, we fuse the position information of GPS in the global reference frame and the pose of VINS in the local reference.

Motivated by the above discussions, this paper mainly focuses on the two unsolved problems of dealing with dynamic objects in the environment and eliminating the cumulative error in a large-scale outdoor environment. To improve the localization and state estimation accuracy of ADV in a large-scale outdoor dynamic environment, this paper proposes an algorithm that combines VINS with DOD. The primary contributions of the paper are:

- According to the YOLOv5 network, we trained the proposed DOD model to detect dynamic objects.
- By combining VINS with DOD, we removed the feature points in the dynamic object regions and tracked the remaining feature points to eliminate the influence of the dynamic object.
- To eliminate the cumulative error in a large-scale outdoor environment, we model the GPS measurement as a general factor and introduce its residual factor into the cost function.

The rest of this paper is organized as follows. Section 2 briefly reviews various VINS achievements in a dynamic environment. Section 3 elucidates the architecture of our algorithm. We introduce the methodology of VINS combining GPS sensors and YOLOv5 network in Section 4. Section 5 shows the qualitative and quantitative results of the proposed method. In the end, Section 6 concludes the paper.

II. RELATED WORK

A. VINS IN DYNAMIC ENVIRONMENT

The main research directions to eliminate the influence of dynamic objects can be divided into three parts: DOD based on motion tracking [12], [13], [14], semantic segmentation and removal based on deep learning [15], [16], [17], and eliminating the influence of dynamic objects using robust methods [8]. Most VINS algorithms treat dynamic feature points as outliers. These points scarcely participate in localization and mapping.

To reduce the influence of dynamic objects on state estimation, one research direction is to manipulate features [12], [13], [14]. Reference [12] propose a sparse motion removal model that detects the dynamic and static regions based on a Bayesian framework. Later in [13], the authors design a new dynamic region detection method. The detection process is completed in a Bayesian framework and then pre-processes the input frame. Further, [14] proposed an optical flow-based motion detection method to remove irrelevant information afterwards.

Another research direction is using semantic segmentation to remove dynamic feature points [15], [16], [17]. The semantic-based approach uses semantic information to segment objects and remove outliers from tracking, such as...
RS-SLAM [15] and Dynamic-SLAM [16]. DynaSLAM [17] pixel-wise segments the priori dynamic objects in frames instead of extracting features on dynamic objects.

B. DYNAMIC OBJECT DETECTION

There are various different object detection algorithms based on deep learnin, such as faster region-based convolutional neural network (R-CNN) [18] and Mask R-CNN [19]. The R-CNN detector is a two-stage detecto. It utilizes a CNN to find potential regions that contain object. Redmon et al. [20] proposed another series of excellent detectors, YOLO. YOLOv5 can even reach 140 frames per secon becaus it utilizes a single convolutional network for the entire image at once.

C. VINS COMBINE WITH GPS

Inertial measuremen and visual are suitablefor obtaining accurate local state, but large drifts will accumulate during long-term navigation. As a result, global positional information can be fused with visual and inertial measurement to achieve accurate localization.

In [11], GPS measurements are fused with VIO (visual inertial odometry) estimates in a pose graph optimization model. Besides, Mohamed et al. [21] presented an integrated navigation system that depends on integrated sensors to enhance navigation in bad weather conditions. These systems are loosely couple, which means that the relative pose update is estimated by the VIO algorithm independently of the GPS measurements. Then the pose graph is optimized to align with the global frame.

Unlike previous loose couple works, tightl coupled method allows exploiting the correlation amongst the measurements. In [22], an adaptive fusion GNSS and VIO system were proposed to achieve accurate global positioning. Similar to [22] and [23] propose a methodology to tightly coupled fuse GNSS information with visual-inertial measurements in a nonlinear optimization estimator.

III. SYSTEM OVERVIEW

The structure of VINS can be divided into four parts: measurement preprocessing, features tracking, system initialization, and global optimization. The block diagram is shown in Fig. 2. Pictures in block diagrams are colored for better presentation to the reader.

A. FEATURES TRACKING

The Kanade-Lucas-Tomasi (KLT) [24] algorithm tracks features between two frames for the monocular camera. Fig. 3 shows the process of feature tracking.

After the DOD model obtains the bounding box of the dynamic object, the front-end removes the feature points in the object regions. Then we track the remaining feature points to eliminate the influence of the dynamic objects. It is worth noting that when the proportion of dynamic objects in the image is large, that is, when too many feature points are deleted, the algorithm will not generate drift due to the existence of IMU and GPS sensors. The schematic diagram of processing features points is shown in Fig. 4.

B. SYSTEM INITIALIZATION AND GLOBAL OPTIMIZATION

VINS utilize the IMU information to form motion constraints between consecutive image frames and provide the absolute scale of the motion. Furthermore, VINS requires exact initialization to perform accurate state estimation [25]. To lead the subsequent nonlinear optimization, the initialization procedure provides the value of pose, gravity vector, velocity, bias, feature location, etc.

GPS sensor is employed to provide drift-free measurements in a global frame, ultimately eliminating the cumulative error of VINS. We model the GPS measurement as a general factor. GPS factor and its related states constitutepart of the global pose graph. The global pose graph optimization module eliminates the cumulative error and drift.

IV. METHODOLOGY

A. VINS COMBINED WITH GPS

1) PROBLEM DEFINITION

A tightly coupled, nonlinear optimization-based method is used to obtain highly accurate state estimate by fusing GPS measurements, preintegrated IMU measurements and feature observations. The state vector $\mathbf{X}$ is defined as follows:

$$\mathbf{X} = [\mathbf{p}_0, \mathbf{R}_0, \mathbf{p}_1, \mathbf{R}_1, \ldots, \mathbf{p}_n, \mathbf{R}_n, \mathbf{x}_{cam}, \mathbf{x}_{imu}, \mathbf{x}_{gps}]$$

where $\mathbf{p}$ and $\mathbf{R}$ are the position and orientation of the body, respectively. $n$ is the total number of keyframes consider for optimization.

$$\mathbf{x}_{cam} = [\lambda_0, \lambda_1, \ldots, \lambda_I]$$

$\mathbf{x}_{cam}$ includes the depth of each feature. $I$ is the total number of features in the sliding window.

$$\mathbf{x}_{imu} = [\mathbf{v}_0, \mathbf{b}_{a_0}, \mathbf{b}_{g_0}, \mathbf{v}_1, \mathbf{b}_{a_1}, \mathbf{b}_{g_1}, \ldots, \mathbf{v}_n, \mathbf{b}_{a_n}, \mathbf{b}_{g_n}]$$

$\mathbf{x}_{imu}$ is consists of gyroscope bias $\mathbf{b}_g$, acceleration bias $\mathbf{b}_a$, and velocity $\mathbf{v}$.

$$\mathbf{x}_{gps} = [x_{gps_0}, y_{gps_0}, z_{gps_0}, x_{gps_1}, y_{gps_1}, z_{gps_1}, \ldots, x_{gps_m}, y_{gps_m}, z_{gps_m}]$$
x_{gps} is the GPS-related variable, x, y, and z are mean longitude, latitude and altitude, respectively. m represents the set of valid GPS measurements.

The maximum a posteriori method is used to solve the optimization problem of the pose graph. The maximum a posteriori is defined by:

\[ \mathcal{X}^* = \arg \max_{\mathcal{X}} \prod_{t=0}^{n} \prod_{k \in S} p \left( z^k_t \mid \mathcal{X} \right) \]

where, \( x_t = \{ p_t^w, q_t^w \} \). \( q_t^w \) and \( p_t^w \) are orientation quaternion and position vector, respectively. S is a set of measurements that come from sensors. The uncertainty of measurements is Gaussian distributed, \( p \left( z^k_t \mid \mathcal{X} \right) \sim \mathcal{N} \left( z^k_t, \Omega^k_t \right) \). The negative log-likelihood of equation (5) is as follows:

\[ \mathcal{X}^* = \arg \max_{\mathcal{X}} \sum_{t=0}^{n} \sum_{k \in S} \left\| z^k_t - h^k_t(\mathcal{X}) \right\|_{\Omega^k_t}^2 \]

where \( \Omega^k_t \) represent the information matrix for Gaussian Distribution \( p \left( z^k_t \mid \mathcal{X} \right) \) and \( h^k_t(\mathcal{X}) \). \( \| \cdot \|_{\Omega} \) represents the Mahalanobis norm. \( h(\cdot) \) denote the sensor model.

Then the bundle adjustment is performed. That is, the state estimation is transformed into a nonlinear least-squares problem. As a result, the entire navigation problem can be expressed as a nonlinear optimization problem.

2) SENSOR FACTORS

The camera factor is constructed from the feature in each frame, and it reprojects the first observation into the following frames. The camera residual of the observation in the image \( t \) is described as:

\[ z^c_t - h^c_t(\mathcal{X}) = z^c_t - h^c_t \left( R_t, p_t^c, R_t, p_t, \lambda_t \right) \]

\[ = \left[ \bar{u}_t \ \bar{v}_t \right] - \pi_c \left( T^b_t T^c_t \pi_c^{-1} \left( \bar{u}_t \ \bar{v}_t \right) \right) \]

\[ \pi_c^{-1} \text{ and } \pi_c \text{ are the back-projection and projection functions, respectively.} \]

\[ T \text{ denotes the } 4 \times 4 \text{ homogeneous transformation. } \Omega^k_t \text{ represents the covariance matrix of reprojection error.} \]

This camera factor is common to both stereo camera and monocular camera.

IMU residuals come from the IMU measurement model of VINS-Fusion. VINS-Fusion uses the IMU pre-integration [26], [27] algorithm to construct the IMU factor. The IMU residual is written as:

\[ z^imu_t \] - h^imu_t(\mathcal{X}) = \left[ \begin{array}{c} \alpha_t^{t-1} \left( p_t - p_{t-1} + \frac{1}{2} g dt^2 - v_{t-1} dt \right) \\ R_{t-1}^{-1} ( v_t - v_{t-1} + g dt ) \\ R_{t-1}^{-1} R_t \\ b_{a_t} - b_{a_{t-1}} \end{array} \right] \]

Within two time instants, the preintegration produces relative rotation \( \gamma_t^{t-1} \), velocity \( \beta_t^{t-1} \), and position \( \alpha_t^{t-1} \). \( \oplus \) denotes the minus operation on manifold. \( g \) represents the gravity vector and its standard value is approximately 9.81. \( b_a \) is acceleration bias of IMU and \( b_\omega \) is gyroscope bias of IMU.

The GPS measurement does not contain the cumulative error. Therefore, GPS constraint added as a global constraint could relieve the cumulative error. The GPS residual could be expressed as:

\[ z^{gps}_t \] - h^{gps}_t(\mathcal{X}) = \left[ x^{gps}_t - h^{gps}_t \left( x_t \right) = p^{gps}_t - p^w_t \right] \]

where \( p^{GPS}_t \) is calculated by the state and position transformation of the previous moment.

3) GPS/VIO OPTIMIZATION

During the entire state estimation period, the VIO provides the main contribution because GPS pseudo-range measurements were significantly biased in some parts of the path with bad signal conditions. On certain road segments with good signal conditions, the constraint of GPS measurements can rectify the accumulated drift in VIO and improve the overall accuracy. The fused result of GPS/VIO has a smaller drift.

The GPS/VIO optimization module first needs to process the VIO data. The pose of VIO is converted to the coordinate system of GPS. The conversion expression is written as:

\[ T_{GPS} = T_{GPS->VIO^*} T_{VIO} \]
The external parameter is optimized by the *ceres solver* [28]. The calculation formula is described as:

\[
T_{\text{GPS} \rightarrow \text{VIO}} = T_{\text{body} \rightarrow \text{GPS}}^{-1} T_{\text{body} \rightarrow \text{VIO}}
\]  

(11)

The GPS preprocessing module first processes GPS pseudo-range measurements. This module converts the longitude, latitude, and height to the world coordinate system and puts it into the global variable. Latitude and longitude are the earth’s coordinates, but the earth is spherical and needs to transform into a plane coordinate system. We need to convert Longitude-Latitude-Altitude (LLA) coordinates into East-North-Up (EN) coordinates. The module first converts the LLA coordinate system to the Earth-Centered-Earth-Fixed (ECEF) coordinate system, after which it converts to the ENU coordinate system. For a certain point in space, the transformation relationship from the LLA coordinate system to the ECEF coordinate system is as follows:

\[
\begin{align*}
X &= (N + z) \cos y \cos x \\
Y &= (N + z) \cos y \sin x \\
Z &= [N (1 - e^2) + z] \sin y
\end{align*}
\]

(12)

In formula (12), \(x\) is longitude, \(y\) is latitude, and \(z\) is altitude. \(N\) represents the radius of curvature, \(N = a/\sqrt{1 - e^2 \sin^2 y}\). \(e\) is the long half axis, \(e\) is the short half axis, and \(N\) is the reference ellipse’s first eccentricity in the LLA coordinate system.

The transformation from the ENU coordinate system to the ECEF coordinate system is as follows:

\[
\begin{bmatrix}
X_1 \\
Y_1 \\
Z_1
\end{bmatrix}_{\text{ECEF}} = R \cdot \begin{bmatrix}
E \\
N \\
U
\end{bmatrix}_{\text{ENU}} + \begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

(13)

where

\[
R = \begin{bmatrix}
- \cos x \sin y - \sin x \cos y \\
- \sin x \sin y - \cos x \sin y \\
\cos y & 0 & 0
\end{bmatrix}
\]

Make

\[
\begin{bmatrix}
\hat{X} \\
\hat{Y} \\
\hat{Z}
\end{bmatrix}_{\text{ECEF}} = \begin{bmatrix}
X_1 \\
Y_1 \\
Z_1
\end{bmatrix}_{\text{ECEF}} - \begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]

get the ENU coordinate system:

\[
\begin{bmatrix}
E \\
N \\
U
\end{bmatrix}_{\text{ENU}} = R^{-1} \begin{bmatrix}
\hat{X} \\
\hat{Y} \\
\hat{Z}
\end{bmatrix}
\]

(14)

Finally, we get

\[
\begin{align*}
E &= - \sin x \cdot \hat{X} + \cos x \cdot \hat{Y} \\
N &= - \cos x \sin y \cdot \hat{X} - \sin x \sin y \cdot \hat{Y} + \cos y \cdot \hat{Z} \\
U &= \cos x \cos y \cdot \hat{X} + \sin x \cos y \cdot \hat{Y} + \sin y \cdot \hat{Z}
\end{align*}
\]

(15)

A schematic representation of the transition matrix between the coordinate systems is shown in Fig. 5 [22], [23].

The GPS/VIO optimization schematic diagram is shown in Fig. 6.
values of the predicted bounding box relative to the original image, the algorithm divides the \((b_x, b_y, b_w, b_h)\) by the feature map size and then multiplies it by the size of the original picture.

The coordinates of the corresponding bounding box for a given object are shown in Fig. 7. Here, each grid cell is predicting the coordinates of the center \((b_x, b_y)\) \((c_x, c_y)\) is the upper-left coordinate of the grid cell in the feature map. Besides, \((p_w, p_h)\) is the width and height of the predicted bounding box. \(t_x\) and \(t_y\) is scale factor. \(\sigma\) is sigma function.

V. EXPERIMENTAL RESULTS

The proposed method is based on VINS-Mono [10], VINS-Fusion [11], [22], [23] and YOLOv5. We compare the proposed method with state-of-art VINS-Fusion to verify the effectiveness and improvement of the method. This paper has evaluated the proposed method on the public KITTI dataset [31] and real-world. Each configuration was run five times and the RMSE median value is reported. The running environment and the sensor’s parameter are shown in Table 1.

Horn’s method [33] is utilized to align the estimated trajectory with the ground truth. This paper evaluated accuracy by absolute trajectory error (ATE) and absolute pose error (APE), where ATE and APE are calculated by EVO tools [33]. We employed the absolute trajectory root mean square error (RMSE) for quantitative evaluations. ATE comprises the absolute distance between the translation components of the estimated and ground truth trajectory. The computation of the ATE at a time step \(i\) is shown in equation (16):

\[
ATE_i = E_i^{-1} T G_i
\]

where \(T\) is the transformation of aligning two trajectories, \(G\) is the ground truth, and \(E\) represents the estimated trajectory. For a sequence of \(N\) poses, the RMSE of ATE is given
by equation (17):

$$RMSE_{ATE_{1:N}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \| \text{trans}(ATE_i) \|^2}$$  (17)

**A. DOD MODEL**

The model training results are shown in Fig. 8. It can be seen from Fig. 9 that when iterating 300 times, the average accuracy of the DOD model is 94%, and the loss function is
TABLE 4. RMSE in KITTI dataset.

| Sequences /Scenes | Dynamic Proportion | Length (m) | Average Speed (m/s) | VINS-Fusion RMSE(m) | OURS RMSE(m) |
|-------------------|--------------------|------------|---------------------|---------------------|--------------|
|                   |                    |            | mono +gps | stereo +gps | stereo +loop | stereo+imu +gps | mono+gps | stereo+gps | stereo+loop | stereo+imu+gps |
| 00/urban          | 1.63%              | 3724.19    | 8.2      | 11.99     | 6.74       | 6.38       | 5.12       | 7.94 (133.8%) | 4.02 (†40.4%) | 6.17 (14.2%) | 4.87 (14.9%) |
| 01/highways       | 9.18%              | 2453.20    | 21.0     | 23.43     | 5.52       | 8.37       | 6.83       | 21.83 (168.6%) | 4.64 (†15.9%) | 8.50 (1.6%) | 6.95 (1.8%) |
| 02/rural          | 4.14%              | 5067.23    | 10.9     | 16.54     | 8.44       | 9.98       | 10.43      | 14.28 (137.5%) | 7.12 (†15.6%) | 8.54 (14.4%) | 10.32 (11.1%) |
| 03/rural          | 3.50%              | 560.89     | 7.0      | 8.27      | 3.41       | 2.13       | 2.13       | 7.83 (53.3%) | 3.17 (†7.0%) | 1.96 (†7.9%) | 2.05 (†3.8%) |
| 04/highways       | 13.33%             | 393.65     | 14.1     | 7.99      | 3.90       | 3.92       | 3.78       | 3.68 (53.9%) | 3.80 (†0.3%) | 3.86 (11.5%) | 3.70 (12.1%) |
| 05/urban          | 3.60%              | 2205.58    | 8.0      | 12.87     | 6.07       | 4.92       | 5.20       | 6.82 (†47.0%) | 3.95 (†34.9%) | 4.72 (†4.1%) | 4.91 (15.6%) |
| 06/urban          | 2.55%              | 1232.88    | 11.2     | 15.15     | 5.45       | 3.75       | 6.44       | 8.60 (43.2%) | 3.15 (†42.2%) | 3.54 (†5.6%) | 6.18 (14.0%) |
| 07/urban          | 30.00%             | 694.70     | 6.3      | 10.43     | 3.06       | 3.24       | 4.25       | 6.29 (139.7%) | 2.05 (†33.0%) | 3.10 (†4.3%) | 4.06 (14.5%) |
| 08/urban          | 3.39%              | 3222.80    | 6.22     | 19.03     | 9.43       | 10.85      | 9.39       | 17.29 (91.9%) | 8.30 (†12.0%) | 9.82 (95.9%) | 9.30 (10.0%) |
| 09/rural          | 0.00%              | 1705.05    | 10.7     | 10.19     | 2.98       | 6.30       | 4.74       | 9.60 (58.8%) | 2.97 (†70.3%) | 5.95 (75.6%) | 4.67 (11.5%) |
| 10/rural          | 0.00%              | 919.52     | 7.5      | 8.80      | 3.30       | 4.08       | 3.75       | 7.25 (13.3%) | 3.31 (†0.3%) | 4.24 (3.9%) | 3.80 (11.3%) |

0.5. The curve shows that the model has high accuracy speed and fast detection.

Table 2 shows the evaluation indexes of DOD, which shows that the model trained in this paper yields good results for all indexes. Timing statistics are show in Table 3.

B. EXPERIMENT WITH KITTI DATASET

This paper uses the KITTI dataset to evaluate the proposed method. This dataset provided ground truth for the sequences 00-10. Fig. 9 is a reconstructed map of sequence 00-10.

The KITTI dataset is collected from various scene data in the real-world, such as rural, highways, and urban. We mainly consider sequence 00 and sequence 05-0, because there are moving vehicles and pedestrians in these urban sequence. The aligned estimated trajectory, APE, and Roll-Pitch-Yaw (RPY) of sequences 00 and 05-0 are shown in Fig. 10. To better demonstrate the effectiveness of the proposed method, we only show four pure visual trajectory error curves. It can be seen from Fig. 10 that after adding the DOD model (yolo in the figure), the location error, rotation error, and translation error are significantly reduced. The results show that adding the DOD model to VINS improves the accuracy of the algorithm.

The AT’s RMSE of sequences 00-10 are shown in Table 4. We can observe that the state estimation accuracy has significantly improved using our proposed method. Compared with VINS-Fusion, the proposed method uses the mono+GPS or stereo+GPS combination has reduced the estimation error by about 40% on average in the dynamic urban sequences (sequence 00, 05, 06, 07). In addition, the proposed method using the stereo+loop or stereo+IMU+GPS combination has reduced the estimation error by about 5% on average.

Clearly, our algorithm outperforms VINS-Fusion in all urban sequences. There is often no loop in the actual ADV applications, only stereo+gps adds loop detection, and the other three combinations do not have loop detection to reduce the computational complexity.

It is worth noting that the error of stereo+imu+gps in Table 4 is larger than that of stereo+gps. It stands to reason that adding IMU sensors can improve the accuracy and robustness of the system. However, during the experiment, we found that the IMU measurement in the KITTI data source is not aligned with other measurements and some data are missing, resulting in a decrease in the accuracy of state estimation. In addition, the proposed method reduces the accuracy in some high-speed (Sequence 01) and static environments (Sequence 10). In a high-speed environment (Sequence 01), the feature points are seriously lost because the vehicle moves too fast. The feature tracking module requires a sufficient number of stable feature points to ensure high-precision pose estimation. In a static environment (Sequence 10), since the DOD model removes the feature points within the dynamic objects that remain stationary, deleting the available feature points will also cause a drop in accuracy.

C. EXPERIMENT IN REAL-WORLD

In the real-world experiment, the dataset was collected from an onboard sensor suite, and this suite is presented in Fig. 11. We drive the car around the outdoor dynamic environment. At the same time, we collected images and synchronized IMU measurements with an MYNTEYE-S1030 camera. GPS measurements are collected using the GPS-U-Blox. To provide an error evaluation of this experiment, we set 16 control points.
through a real-time kinematics GPS sensor, control points as the ground truth points. We align estimated trajectories with the control points to get the absolute error. The experiment environment is presented in Fig. 12. The estimated trajectories of the real-world are shown in Fig. 13(a). Fig. 13(b) shows the absolute error between the estimated trajectory and the control points. Table 5 shows the comparison of control points and statistical errors. It can be seen from Table 5 that the proposed method reduces the localization error by about 30% compared with VINS-Fusion.

In addition, we evaluate the running time of the proposed method. Compared to an average of 61ms/frame for VINS-Fusion, the increase of our system is just 15ms, which shows that our implementation efficiently introduces DOD model in the sliding window front-end. Our method increases the running time by about 25% on average compared with state-of-art VINS-Fusion. However, we believe that it is worthwhile to sacrifice 25% of the time cost to increase the accuracy by more than 25%. In conclusion, the proposed method can operate smoothly and reliably in a large-scale or dynamic outdoor environment.

VI. CONCLUSION
Achieving accurate localization via VINS in large-scale or dynamic outdoor environments is challenging. This paper proposes an algorithm that combines VINS and DOD for ADV to improve the accuracy and robustness of navigation. The proposed method removes the influence of dynamic objects by combining the DOD model and eliminates the cumulative error by fusing the GPS sensor.

Experiments on the KITTI dataset and real-world show that our method improves the localization accuracy by about 40% compared to state-of-art VINS-Fusion in the large-scale dynamic urban environment. The contrast experimental results showed that our algorithm enhanced accuracy and robustness in dynamic environments.

In the future, we will draw attention to designing better navigation algorithms and exploring more robust SLAM systems.

REFERENCES
[1] L. Xiong, R. Kang, J. Zhao, P. Zhang, M. Xu, R. Ju, C. Ye, and T. Feng, “G-VIDO: A vehicle dynamics and intermittent GNSS-aided visual-inertial state estimator for autonomous driving,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 8, pp. 11845–11861, Sep. 2021.

[2] A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, and V. Sze, “Navion: A 2-mW fully integrated real-time visual-inertial odometry accelerator for autonomous navigation of nano drones,” IEEE J. Solid-State Circuits, vol. 54, no. 4, pp. 1106–1119, Apr. 2019.
K. Minoda, F. Schilling, V. Wuest, D. Floreano, and T. Yairi, “VIODE: A simulated dataset to address the challenges of visual-inertial odometry in dynamic environments,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1343–1350, Apr. 2021.

H.-W. Chae, J.-H. Choi, and J.-B. Song, “Robust and autonomous stereo visual-inertial navigation for non-holonomic mobile robots,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9613–9623, Sep. 2020.

R. Mur-Artal and J. D. Tardós, “Visual-inertial monocular SLAM with map reuse,” *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.

K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, “Robust stereo visual inertial odometry for fast autonomous flight,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 965–972, Apr. 2018.

J. Lee and S.-Y. Park, “PLF-VINS: Real-time monocular visual-inertial SLAM with point-line fusion and parallel-line fusion,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7033–7040, Oct. 2021.

L. Cui and C. Ma, “SDF-SLAM: Semantic depth filter SLAM for dynamic environments,” *IEEE Access*, vol. 8, pp. 95301–95311, 2020.

X. Bai, W. Wen, and L. Hsu, “Robust visual-inertial integrated navigation system aided by online sensor model adaption for autonomous ground vehicles in urban areas,” *Remote Sens.*, vol. 12, no. 10, pp. 1686–1708, Jan. 2020.

T. Qin, P. Li, and S. Shen, “VINS-Mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

T. Qin, J. Pan, S. Cao, and S. Shen, “A general optimization-based framework for local odometry estimation with multiple sensors,” 2019, arXiv:1901.03638.

J. Cheng, C. Wang, and M. Q.-H. Meng, “Robust visual localization in dynamic environments based on sparse motion removal,” *IEEE Trans. Auton. Sci. Eng.*, vol. 17, no. 2, pp. 658–669, Apr. 2020.

J. Cheng, H. Zhang, and M. Q.-H. Meng, “Improving visual localization accuracy in dynamic environments based on dynamic region removal,” *IEEE Trans. Auton. Sci. Eng.*, vol. 17, no. 3, pp. 1585–1596, Jul. 2020.

W. Xie, P. Liu, and M. Zheng, “Moving object segmentation and detection for robust RGBD-SLAM in dynamic environments,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–8, 2021, doi: 10.1109/TIM.2020.3026803.

T. Ran, L. Yuan, J. Zhang, D. Tang, and L. He, “RS-SLAM: A robust semantic SLAM in dynamic environments based on RGB-D sensor,” *IEEE Sensors J.*, vol. 21, no. 18, pp. 20657–20664, Sep. 2021.

J. Jiao, C. Wang, N. Li, Z. Deng, and W. Xu, “An adaptive visual dynamic-SLAM method based on fusing the semantic information,” *IEEE Sensors J.*, early access, Jan. 14, 2021, doi: 10.1109/JSEN.2021.3051691.

B. Bescos, J. M. Facil, J. Civera, and J. Neira, “DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes,” *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.

S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2018.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

H. Mohamed, H. Khatr, and K. Moussa, “An intelligent combined visual navigation brain model/GPS/MEMS–INS/ADSCF method to develop vehicle independent guidance solutions,” *Microachines*, vol. 2, no. 6, pp. 718–735, Jun. 2021.

Z. Gong, P. Liu, F. Wen, R. Ying, X. Ji, R. Miao, and X. Xue, “Graph-based adaptive fusion of GNSS and VIO under intermittent GNSS-degraded environment,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–16, 2021, doi: 10.1109/TIM.2020.3039640.

Z. Gong, R. Ying, F. Wen, J. Qian, and P. Liu, “Tightly coupled integration of GNSS and vision SLAM using 10DoF optimization on manifold,” *IEEE Sensors J.*, vol. 19, no. 24, pp. 12105–12117, Dec. 2019.

B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proc. DARPA Image. Understand. Workshop*, Apr. 1981, pp. 121–130.

K. Eckenhoft, Y. Yang, P. Geneva, and G. Huang, “Tightly-coupled visual-inertial localization and 3-D rigid-body target tracking,” *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1541–1548, Apr. 2019.

J. Jiang, J. Yuan, X. Zhang, and X. Zhang, “DVIO: An optimization-based tightly coupled direct visual-inertial odometry,” *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 11212–11222, Nov. 2021.

J. Xiao, D. Xiong, Q. Yu, K. Huang, H. Lu, and Z. Zeng, “A real-time sliding-window-based visual-inertial odometry for MAVs,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4049–4058, Jun. 2020.

S. Agarwal and K. Mierle, *Ceres Solver*. Accessed: 2012. [Online]. Available: http://ceres-solver.org/

F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, “BDD100K: A diverse driving dataset for heterogeneous multitask learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2633–2642.

A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving?” The KITTI vision benchmark suite,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

I. A. Sucan, M. Moll, and L. E. Kavraki, “The open motion planning library,” *IEEE Robot. Autom. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012.

B. K. P. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *J. Opt. Soc. Amer. A Opt. Image Sci.*, vol. 4, no. 4, pp. 629–642, 1987.