

DynaVIG: Monocular Vision/INS/GNSS Integrated Navigation and Object Tracking for AGV in Dynamic Scenes

Ronghe Jin¹, Yan Wang¹, Zhi Gao², Xiaoji Niu¹, Li-Ta Hsu³, and Jingnan Liu¹

Abstract—Visual-Inertial Odometry (VIO) usually suffers from drifting over long-time runs, the accuracy is easily affected by dynamic objects. We propose DynaVIG, a navigation and object tracking system based on the integration of Monocular Vision, Inertial Navigation System (INS), and Global Navigation Satellite System (GNSS). Our system aims to provide an accurate global estimation of the navigation states and object poses for the automated ground vehicle (AGV) in dynamic scenes. Due to the scale ambiguity of the object, a prior height model is proposed to initialize the object pose, and the scale is continuously estimated with the aid of GNSS and INS. To precisely track the object with complex moving, we establish an accurate dynamics model according to its motion state. Then the multi-sensor observations are optimized in a unified framework. Experiments on the KITTI dataset demonstrate that the multi-sensor fusion can effectively improve the accuracy of navigation and object tracking, compared to state-of-the-art methods. In addition, the proposed system achieves good estimation of the objects that change speed or direction.

I. INTRODUCTION

Navigation and object tracking are two significant tasks in autonomous driving and robotics. Simultaneous Localization and Mapping (SLAM) using a monocular camera has low cost and high computational efficiency. The fusion of monocular SLAM with Inertial Navigation System (INS) and Global Navigation Satellite System (GNSS) can greatly improve the accuracy and robustness of navigation, the scale estimation enables monocular SLAM to obtain the capability of 3D measuring, which is similar to stereo or LiDAR. Object tracking can obtain the object's pose, allowing safety in automatic driving and physical interaction in augmented reality (AR)/virtual reality (VR). SLAM and object tracking are strongly correlated, some studies [1]-[3] have recently unified the problem of SLAM and object tracking and verified that they can benefit each other.

Many researchers studied Visual-Inertial Odometry (VIO) for the complementarity of the Inertial Measurement Unit (IMU) and SLAM. However, VIO has four unobservable directions [4] and suffers from drifting over long-time runs.

This work was supported by the National Key Research and Development Program of China under Grant 2016YFB0501804. (Corresponding author: Ronghe Jin.)

¹Ronghe Jin, Yan Wang, Xiaoji Niu, and Jingnan Liu are with GNSS Research Center, Wuhan University, No. 129 Luoyu Road, Wuhan 430079, China {huanhexiao, wstephen, xjniu, jnliu}@whu.edu.cn

²Zhi Gao is with the School of Remote Sensing and Information Engineering, Wuhan University, No. 129 Luoyu Road, Wuhan 430079, China gaozhinus@gmail.com

³Li-Ta Hsu is with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China lt.hsu@polyu.edu.hk

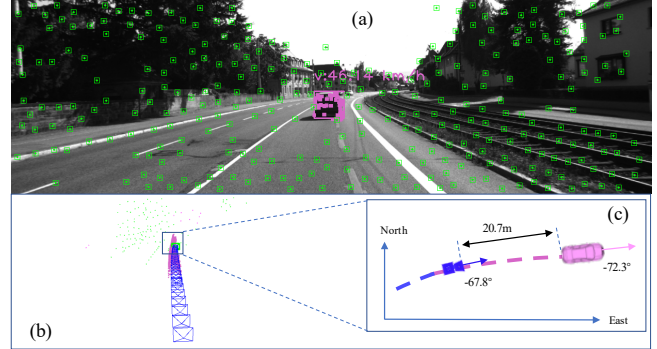


Fig. 1. One example on KITTI shows: (a) one object (pink) with its speed, and the static features (green). (b) 4D map corresponding to (a), trajectories of the camera (blue) and the object (pink). (c) The projection of the camera and the object on a 2D plane, the yaw angles and the object depth are given.

GNSS is an easy-obtained, drift-free, and global-aware observation that provides accurate long-term correction, thus Vision/INS/GNSS integration becomes attractive. The integration mainly includes loosely-coupled integration using GNSS position results and tightly-coupled integration using GNSS raw measurements [5], they achieve similar accuracy in open environments. Loosely-coupled integration is convenient to design the algorithm and configure the information matrix, while GNSS cannot provide results with less than 4 satellites. Tightly-coupled integration can work in challenging scenes using even 1 satellite, but the insufficient observations and multipath effect will seriously reduce the accuracy [6]. Moreover, the framework of tightly-coupled integration is complex, and the noise propagation needs careful handling. A general problem is that most VIO and Vision/INS/GNSS integrated systems neglect dynamic objects, which will reduce the performance in dynamic scenes.

To decrease the influence of dynamic objects, some works detect and eliminate them. However, the simple elimination may lose some available information about the objects. Some recent works [1]-[3], [7] have unified SLAM and object tracking and achieved a win-win for such two tasks, however, there are some shortcomings. The works with the monocular camera usually use the camera height to scale the map and object, but this needs a changeless camera height and an observable ground plane. The object pose of the 6 Degree of Freedom (DoF) definition fails to exploit the constraints of plane ground, while the 3 DoF definition ignores slopes. In addition, most works assume a constant velocity model of the camera and objects, which will affect the accuracy when they change speed or direction.

To address the above issues, we propose DynaVIG based on the Monocular Vision/INS/GNSS integration for navigation and object tracking. A loosely coupled GNSS is applied considering the complexity of object tracking and the availability of the KITTI dataset. The object pose is defined as 4 DoF to make better use of the ground constraint, and it is initialized via a prior height model due to the scale ambiguity. An accurate dynamics model of the object is constructed to process objects with complex motion, it is then combined with multi-sensor measurements to optimize the navigation states, map points, and object poses. Experiments of the KITTI tracking dataset are conducted for validation, one example is shown in Fig. 1. We highlight the contributions of our work as follows:

- A unified framework of navigation and object tracking is constructed based on the Monocular Vision/INS/GNSS integration;
- A prior height model and a precise dynamics model of the object are proposed for accurate object tracking;
- The experiments verify the improvements of the proposed system compared with existing methods;

II. RELATED WORK

A. Visual SLAM with Multi-Sensor Fusion

VIO is a widely researched topic [8]-[11] and obtains great improvements, but it suffers from drifts and unobservable directions. Scholars have studied the Vision/INS/GNSS integration to overcome the weaknesses of VIO. Earlier works are mainly loosely-coupled integrations. VINS-Fusion [12] couples GPS positions with VIO poses, but the result-level fusion depends heavily on the quality of GPS and VIO outputs. The work in [13] uses GNSS to couple with INS and vision, however, the GNSS simulated from the indoor dataset may limit the application. Recent works researched tightly-coupled integration to make better use of GNSS raw measurements. GVINS [4] is an excellent work of GNSS tightly-coupled integration with VIO, but it uses low-precision GNSS pseudorange measurements with meters of noise, and the ionospheric delay and troposphere delay using standard models may not be accurate enough. GAINS [14] uses GNSS pseudorange, Doppler frequency shift, and carrier phase measurements with a lightweight filter, which could be prone to the nonlinear error of SLAM. The main advantage of tightly-coupled integration is the ability to provide continuous service in challenging scenes, but the model is complicated and the accuracy may still be limited.

B. Scale Estimation for Monocular SLAM

Scale estimation is a critical topic for monocular SLAM. The work in [15] estimates the scale of monocular SLAM with a Bayesian filter, it uses the camera height to provide the initial scale and the object's prior height for correction. CubeSLAM [7] also uses the camera height and object size for scale. But different from [15], CubeSLAM constructs a framework for SLAM and objects to maintain a consistent scale. These approaches assume a given fixed camera height and an observable ground plane, which is easily influenced

by shaking, occlusion, and slope. Therefore, some scholars try to recover the monocular scale without prior information. The work in [16] uses some network architectures to estimate absolute distances between consecutive frames. The authors of [17] introduce the concept of "extent" to constrain the scale drift of SLAM and objects. These methods do not need constant camera height or planar roadway, but there is no information for physical scale estimation.

C. Dynamic SLAM with Object Tracking

To weaken the impact on SLAM, earlier works use geometric [18], [19] or learning-based methods [20], [21] to remove the dynamic objects. These works are effective but lose high-level information, failing to maximize the SLAM accuracy. Recently, researchers make efforts to couple the problems of SLAM and object tracking. CubeSLAM [7] generates the object's 3D bounding box using the 2D bounding box and vanishing points, then SLAM and object tracking are optimized together. CubeSLAM realizes 3D object detection with only one camera, but it is limited to stationary or slow-moving objects. VDO-SLAM [1] uses dense optical flow to ensure the robustness of object tracking, however, the calculation is very complicated. DynaSLAM II [2] proposes a tightly-coupled algorithm of SLAM and object tracking, but the object pose is defined as 6DoF without the constraint of the ground. TwistSLAM [3] uses plane ground assumption to constrain an object's movements, the performance shows great advantages over previous works, while the 3DoF of pose definition may not satisfy the slopes. Moreover, These works use a constant velocity model of the camera and object, which may be inaccurate in some cases. Some algorithms treat all objects as dynamic, resulting in fewer available features when the object is static.

Most multi-sensor integrated approaches are easily affected by dynamic objects, and the accuracy of SLAM and object tracking algorithms are usually limited, thus the navigation performance of the automated ground vehicle (AGV) could be seriously restricted in dynamic scenes. To this end, we aim to build an accurate global navigation and object tracking system using the Monocular Vision/INS/GNSS integration. By leveraging the drift-free GNSS and high-rate INS measurements, the system can eliminate the drift of SLAM and enable 3D object tracking with a monocular camera. The system can be used for AGV to precisely estimate the poses of the camera and objects.

III. METHOD

The structure of the proposed system is shown in Fig. 2. After being detected by YOLOv5, the objects are associated between frames by optical flow with the BRIEF descriptor. The object's motion state is rapidly determined, and the static ones are regarded as a part of the environment. IMU pre-integration and GNSS solutions can be calculated with parallel threads. Before the optimization, the prior height model is used to initialize the object pose, and the dynamics model is established according to its motion state. Then the

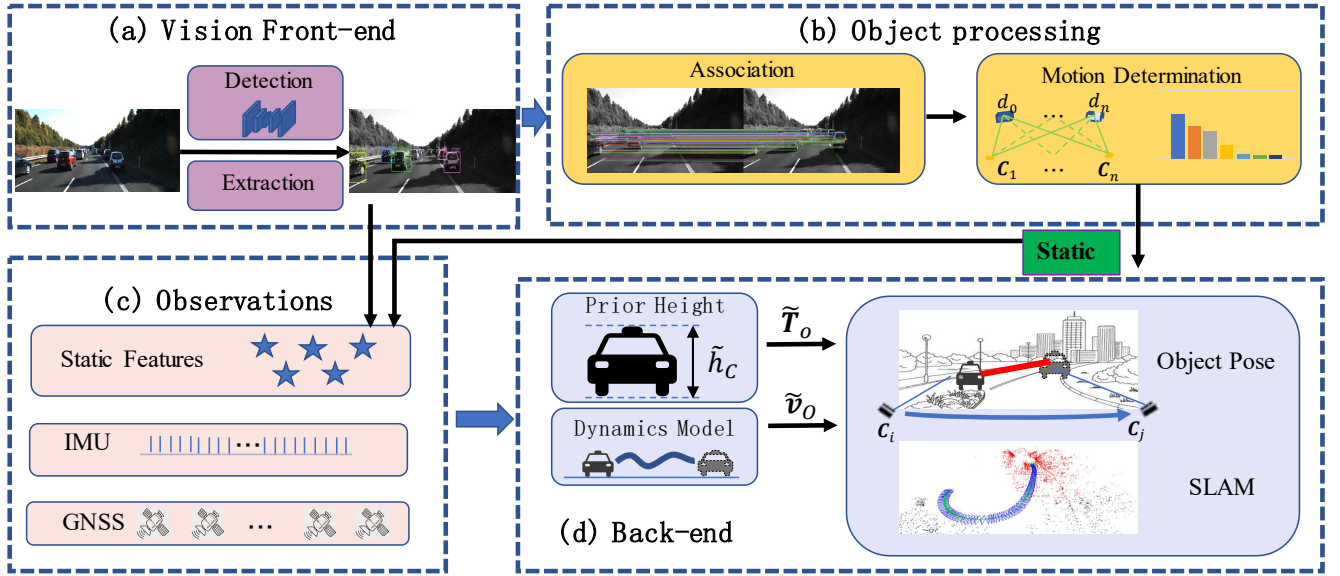


Fig. 2. Overview of our proposal. The front-end generates the objects with YOLOv5 and extracts Good Features, as shown in (a). (b) demonstrates the object association via optical flow with the BRIEF descriptor, its motion state is determined by the statistical characteristics of its depth sequence. (c) illustrates the observations including static features extracted in (a) and static objects in (b), as well as IMU and GNSS. (d) shows that given the prior height and dynamics models of the object, SLAM and object tracking are jointly optimized.

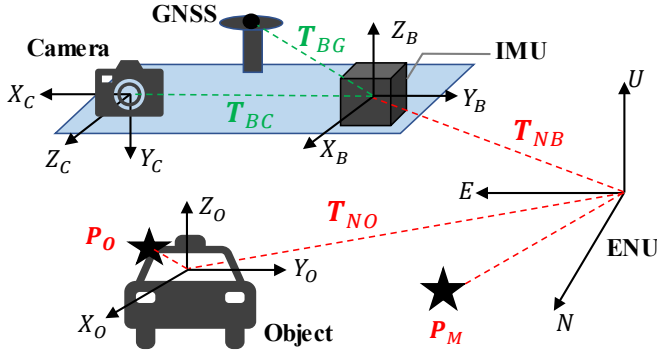


Fig. 3. Illustration of the coordinate frames, including the frame B , C , O , and N (ENU). The green symbols are the known extrinsic parameters between sensors, and the red symbols are the states.

multi-sensor measurements are optimized for the navigation and object tracking in a unified framework.

A. Notations

We define $T_{XY} \in SE(3)$ as the transformation from frame Y to X , $P_X \in \mathbb{R}^3$ as the point coordinate in frame X , and $v_{XY} \in \mathbb{R}^3$ as the translational velocity of Y in frame X . Four coordinate frames are defined in Fig. 3, including the Body frame B (aligned to the IMU frame), the camera frame C , the object frame O , and the navigation frame N . The frame N , also known as east-north-up (ENU), is the global reference for the system. The GNSS-IMU extrinsic parameter T_{BG} and the Camera-IMU extrinsic parameter T_{BC} have been calibrated. The body pose T_{NB} , the object pose T_{NO} , the map point P_M , and the object point P_O are the states to be estimated. The states also include the body velocity v_{NB} and the velocity v_{NO} , the yaw speed v_ψ , and

the scale s of the object.

B. Vision Front-End with Object Processing

The object's features are extracted by the method of [22], and the association via features matching uses high-efficiency Lucas-Kanade (LK) optical flow. However, it is not easy to track the object accurately even using multiscale pyramidal optical flow, due to the motion of objects. To improve the association robustness, we use a method named optical flow with the descriptor. Firstly a bidirectional optical flow is used for features matching, then the BRIEF descriptors [23] are calculated to select good matches via descriptor distance.

If the stationary objects are determined quickly, more available static features could be used for SLAM. As the multi-view geometry constraint does not satisfy the dynamic features, the standard deviation (STD) of the triangulated depth sequence $d = [d_0, d_1, \dots, d_m]$ can be used to determine the motion state. The STD should be small for static objects, while large for dynamic ones.

C. Prior Height Model for Object Parametrization

TwistSLAM [3] set the object pose as 3 DoF with plane road constraints, which may affect the accuracy on slopes. Since most small slopes (such as Fig. 1 shows) could lead to the long-term displacement of the z-axis but a slight change of pitch and roll, we define the object pose as 4 DoF, i.e., 3D translation and 1D rotation (yaw). The initial yaw ψ can be determined with $\psi = \tan^{-1}(v_n/v_e)$ [24], where v_n and v_e are the north and east components of the object's initial velocity respectively. The initial velocity can be calculated by position differential. Therefore, the initial position is the key to determining the initial state of the object. However,

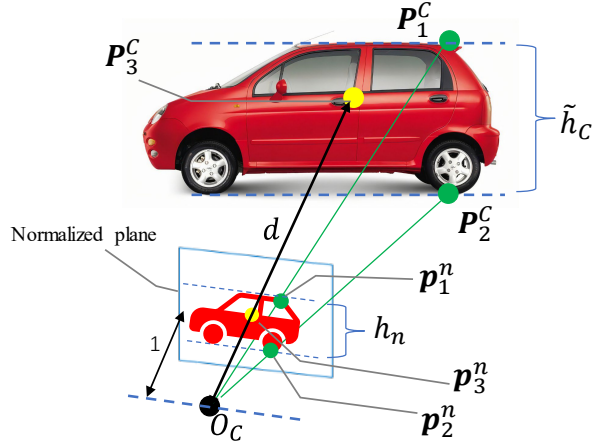


Fig. 4. Prior height Model of the object. O_C is the center of the frame C , P_i^C is the point of the object in the frame C , p_i^n is the projection of P_i^C on the normalized plane. The green points P_1^C and P_2^C are the top and bottom points of the object respectively, and the prior height \tilde{h}_C represents the distance of them. The yellow point P_3^C is projected to the center of the normalized plane. $O_C P_3^C = d$ is object depth in the frame C .

the object position can not be calculated by triangulation due to scale ambiguity.

To solve this problem, we propose a prior height model shown in Fig. 4. Assuming the height of a class (such as a person or car) is known, the initial position of the object in the frame C can be determined. Let $P_1^C(X_1^C, Y_1^C, Z_1^C)$ and $P_2^C(X_2^C, Y_2^C, Z_2^C)$ be the top and bottom point of the object in the frame C respectively, they are projected as $p_1^n(x_1^n, y_1^n)$ and $p_2^n(x_2^n, y_2^n)$ on the normalized plane. $p_i^n(x_i^n, y_i^n)$ can be converted via reprojection from the image observation (u_i, v_i) . According to the triangular similarity, we have:

$$\frac{1}{d} = \frac{y_2^n - y_1^n}{Y_2^C - Y_1^C} = \frac{y_2^n - y_1^n}{\tilde{h}_C} \quad (1)$$

where \tilde{h}_C is the prior height of the object, then we have the depth $d = \tilde{h}_C / (y_2^n - y_1^n)$. Hence, the coordinates of any object point k in the frame C can be calculated:

$$\begin{bmatrix} X_k^C & Y_k^C & Z_k^C \end{bmatrix} = \frac{\tilde{h}_C}{y_2^n - y_1^n} \cdot \begin{bmatrix} x_k^n & y_k^n & 1 \end{bmatrix} \quad (2)$$

Assuming the scale $s = \hat{h}_C / \tilde{h}_C$, where \hat{h}_C is the true height. s is added to the state vector for further refinement.

D. Factor Graph Optimization for Vision/INS/GNSS integration with Object Tracking

As mentioned in section III-A, the states can be defined as $\mathbf{X} = [\mathbf{T}_{NB}, \mathbf{v}_{NB}, \mathbf{P}_M, s, \mathbf{T}_{NO}, \mathbf{P}_O, v_\psi, \mathbf{v}_{NO}]$, Assuming all the observations are with Gaussian distribution, the factors can be processed with one optimizer. The loss function of DynaVIG is defined as follows:

$$\arg \min_{\mathbf{X}} = \left\{ \begin{aligned} & \sum \|e_{sta}\|_{\Sigma_s}^2 + \sum \|e_{IMU}\|_{\Sigma_I}^2 + \\ & \sum \|e_{GNSS}\|_{\Sigma_G}^2 + \sum \|e_{obj}\|_{\Sigma_O}^2 + \\ & \sum \|e_{dm}\|_{\Sigma_v}^2 \end{aligned} \right\} \quad (3)$$

where $e_{sta}, e_{IMU}, e_{GNSS}, e_{obj}$, and e_{dm} are the static feature, IMU pre-integration, GNSS, object feature, and object dynamics factors respectively, Σ is the covariance matrix. The traditional $e_{sta}, e_{IMU}, e_{GNSS}$ are as follows:

$$\begin{aligned} e_{sta} &= \pi((\mathbf{T}_{NB}\mathbf{T}_{BC})^{-1}\mathbf{P}_M) - \mathbf{p}_s \\ e_{IMU} &= f_{PI}(\Delta\mathbf{R}_{ij}, \Delta\mathbf{v}_{ij}, \Delta\mathbf{t}_{ij}, \mathbf{T}_{NB}, \mathbf{v}_{NB}) \\ e_{GNSS} &= (\mathbf{T}_{NB}\mathbf{T}_{BG})|_t - \mathbf{t}_G^N \end{aligned} \quad (4)$$

where π is the reprojection function, \mathbf{p}_s is the coordinate of the static feature on the normalized plane; f_{PI} is the IMU factor function, $\Delta\mathbf{R}_{ij}, \Delta\mathbf{v}_{ij}, \Delta\mathbf{t}_{ij}$ are the changes of rotation, velocity, and position pre-integrated using IMU measurements respectively; \mathbf{t}_G^N indicates the GNSS position solutions in the frame N .

Based on the reprojection π , e_{obj} can be reconstructed:

$$e_{obj} = \pi(s \cdot \mathbf{T}_{CN}\mathbf{T}_{NO}\mathbf{P}_O) - \mathbf{p}_O \quad (5)$$

where \mathbf{T}_{CN} is the inverse of the camera pose; \mathbf{P}_O is the coordinate of the object point, \mathbf{p}_O is its image observation.

For the object's dynamics model, we assume the velocities of the yaw and translation vary slowly. Let $\mathbf{v} = [v_\psi, \mathbf{v}_{NO}]$, it can be modeled as a random constant in a short time:

$$\begin{aligned} \dot{\mathbf{v}} &= \mathbf{0} \\ \mathbf{v}_{i-1} &= \mathbf{v}_i \end{aligned} \quad (6)$$

The two equations are the continuous and discrete forms of the random process of \mathbf{v} respectively, $i-1$ and i are two consecutive images. We propose to set the variance intensity of \mathbf{v} according to the motion complexity. Assuming the current velocity is the same as the previous, the variance intensity could be determined more accurately. Therefore, the random model and the optimization are cause and effect to each other. Then e_{dm} is defined as follows:

$$\begin{aligned} e_{dm} &= \mathbf{v}_i - \mathbf{v}_{i-1} \\ \Sigma_{e_{dm}} &= \exp(\|\mathbf{v}_{i-1}\| \cdot K_O) \end{aligned} \quad (7)$$

where K_O is the gain factor for velocity amplification.

IV. EXPERIMENTAL RESULTS

A. Experimental Setup

The proposed DynaVIG is evaluated on the KITTI Tracking dataset [25], which mainly contains cars and pedestrians, the ground truth of the camera and objects are provided. The IMU is extracted from the KITTI Raw dataset since only the raw IMU with a high rate (100Hz) is useful. The KITTI IMU may contain some sick ranges, including time stamp errors and duplicate records. Fortunately, these ranges are usually very short (within 0.1s), allowing us to fix them by interpolating their neighbors. The GNSS measurements are simulated by corrupting the trajectory with Gaussian noise, the sampling rate is 1Hz. The number of keyframes in the sliding window is limited to 10. The factor graph-based GTSAM [26] is used for optimization.

The major evaluation metrics are the absolute trajectory error (ATE) and relative pose error (RPE) [27]. The object scale error and the computational time are analyzed as well. We compare our results with state-of-the-art algorithms.

TABLE I
CAMERA POSE COMPARISON WITH EXISTING ALGORITHMS ON THE KITTI DATASET. ATE IS IN m , RPE_t IN m/f , RPE_R IN $^\circ/f$

seq	VDO-SLAM [1]			DynaSLAM II [2]			TwistSLAM [3]			Ours		
	ATE	RPE_t	RPE_R	ATE	RPE_t	RPE_R	ATE	RPE_t	RPE_R	ATE	RPE_t	RPE_R
00	-	0.07	0.07	1.29	0.04	0.06	-	0.04	0.05	0.03	0.02	0.04
01	-	0.12	0.04	2.31	0.05	0.04	-	0.04	0.03	0.05	0.03	0.05
02	-	0.04	0.02	0.91	0.04	0.02	-	0.03	0.03	0.05	0.03	0.06
03	-	0.08	0.03	0.69	0.06	0.04	-	0.06	0.02	0.02	0.02	0.05
04	-	0.11	0.05	1.42	0.07	0.06	-	0.06	0.04	0.04	0.02	0.05
05	-	0.09	0.02	1.34	0.06	0.03	-	0.06	0.02	0.02	0.01	0.03
06	-	0.02	0.05	0.19	0.02	0.04	-	0.02	0.04	0.01	0.01	0.04
07	-	-	-	3.10	0.05	0.07	-	0.04	0.04	0.09	0.04	0.04
08	-	-	-	1.68	0.10	0.04	-	0.07	0.03	0.07	0.05	0.08
09	-	-	-	5.02	0.06	0.06	-	0.05	0.04	0.02	0.01	0.04
10	-	-	-	1.30	0.07	0.03	-	0.07	0.03	0.03	0.02	0.04
11	-	-	-	1.03	0.04	0.03	-	0.03	0.02	0.08	0.02	0.04
13	-	-	-	1.10	0.04	0.04	-	0.03	0.04	0.17	0.07	0.08
14	-	-	-	0.12	0.03	0.08	-	0.03	0.06	0.03	0.02	0.07
18	-	0.07	0.02	1.09	0.05	0.02	-	0.04	0.02	0.05	0.05	0.03
19	-	-	-	2.25	0.05	0.03	-	0.03	0.03	0.29	0.02	0.01
20	-	0.17	0.03	1.36	0.07	0.04	-	0.04	0.03	0.07	0.04	0.06
mean	-	0.087	0.037	1.541	0.053	0.043	-	0.044	0.034	0.066	0.028	0.048
std	-	0.044	0.018	1.159	0.019	0.017	-	0.015	0.011	0.068	0.016	0.018

B. Camera Pose Estimation

This section analyzes the camera pose estimation. Table I shows the comparison of our method with the existing algorithms. The previous works all used stereo vision, and only DynaSLAM II calculated the ATE. We can see that DynaVIG achieves obvious improvements on the ATE and RPE_t , indicating that GNSS and INS have good effects on the camera translation. The RPE_R of TwistSLAM slightly outperforms DynaVIG. Since the KITTI sequences are usually too short for IMU bias to converge, the camera rotation depends mainly on the visual observations, thus DynaVIG using a monocular camera obtains lower rotation accuracy than stereo systems. As discussed in [27], the low precision of rotation will further deteriorate RPE_t , because RPE_t considers both translational and rotational errors. Therefore, the higher RPE_t accuracy of DynaVIG shows that multi-sensor fusion has a great advantage in translation estimation, compared with pure visual SLAM.

C. Object Tracking

For object pose, we use the sequences analyzed in DynaSLAM II and TwistSLAM, the results are shown in Table II. It demonstrates that DynaVIG mostly outperforms other algorithms on the ATE. This also benefits from the advantage of multi-sensor fusion in translation estimation, although the object translation of DynaVIG contains a scale error. This is because the estimation of object pose largely depends on the camera pose when optimized jointly, as the camera pose estimation has more sensors and a better geometry structure of features. DynaVIG achieves better RPE_R , which is greatly due to the proposed dynamics model, good examples are

objects with speed or/and direction changes, such as 10-0, 20-0, and 20-12 (sequence-id). Compared with TwistSLAM, the slightly worse RPE_t of DynaVIG is most likely due to the scale error.

The monocular scale errors of the objects are calculated in Table II, which is about 10% in most cases. The scale converging curves are shown at the bottom right of each column in Fig. 5, the scale of most objects can converge with time. However, there are some abnormal cases. Both 11-35 and 19-63 are static cars, and the camera of both sequences stayed stationary for a long time during the driving. The lack of translation leads to the inability to effectively scale estimation. 20-122 takes a small size of the image and is blocked for a short time, thus its scale may not converge.

D. Computational Time

In this section, we evaluate the computational time of our method. The experiments are carried out on a desktop PC with an Intel i3-4150 at 3.5GHz and 16-GB memory. To perform a fair comparison with DynaSLAM II, the front-end including the object processing of DynaVIG is treated as the tracking thread in DynaSLAM II, and the back-end is treated as the Local BA thread of DynaSLAM II. The results are listed in Table III. The front-end of DynaVIG spends about the same amount of time as DynaSLAM II, showing that both DynaVIG and DynaSLAM II could run in real time. However, the back-end of DynaVIG costs more much time than DynaSLAM II. Due to the different number of cameras and features, the influences on the front-end computation are hard to compare. As for the back-end, the number of features should be likely the main reason for the difference

TABLE II
OBJECT POSE COMPARISON WITH EXISTING ALGORITHMS ON THE KITTI DATASET. ATE IS IN m , RPE_t IN m/m , RPE_R IN $^\circ/m$

seq	id	DynaSLAM II [2]			TwistSLAM [3]			Ours			scale error
		ATE	RPE_t	RPE_R	ATE	RPE_t	RPE_R	ATE	RPE_t	RPE_R	
03	1	0.69	0.34	1.84	0.31	0.10	0.28	0.26	0.73	0.47	17.38%
05	31	0.51	0.26	13.50	0.35	0.19	0.58	0.18	0.24	0.32	3.66%
10	0	0.95	0.40	2.84	0.77	0.21	1.98	0.12	0.20	0.43	13.17%
11	0	1.05	0.43	12.51	0.17	0.23	0.23	0.16	0.74	0.46	5.06%
	35	1.25	0.89	16.64	0.10	0.03	0.11	0.07	0.02	0.27	-
18	2	1.10	0.30	9.27	0.21	0.27	0.66	0.05	0.32	0.29	1.14%
	3	1.13	0.55	20.05	0.15	0.21	0.56	0.26	0.17	0.22	10.22%
19	63	0.86	1.45	48.80	0.28	2.17	1.08	0.44	0.24	0.25	-
	72	0.99	1.12	3.36	0.16	0.05	0.34	0.11	0.01	0.08	93.10%
20	0	0.56	0.45	1.30	0.17	0.20	0.72	0.23	0.69	0.28	7.36%
	12	1.18	0.40	6.19	0.24	0.20	1.54	0.06	0.36	0.60	10.40%
	122	0.87	0.72	5.75	0.17	0.02	0.07	0.11	0.48	0.55	11.11%
mean		0.928	0.609	11.838	0.257	0.323	0.679	0.170	0.350	0.352	17.26%
std		0.240	0.369	13.140	0.177	0.588	0.587	0.114	0.258	0.151	27.07%

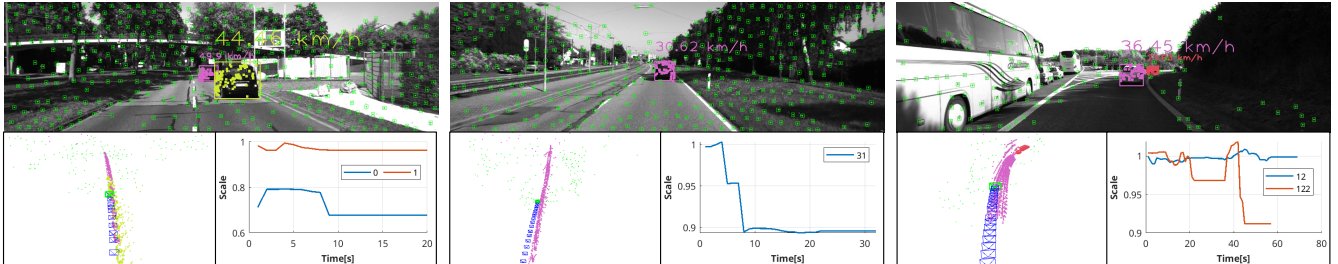


Fig. 5. Visualization of object tracking of the sequence 0003 (left), 0005 (Middle), and 0020 (Right) on the KITTI tracking dataset. (Top): Pink and yellow rectangles are objects with speed. Green points with rectangles are environment features; (Bottom left): 4D point clouds of the objects and map; (Bottom right): object scale converging curves with time.

TABLE III
COMPARISON OF COMPUTATIONAL TIME ($mSec$).

	3		20	
	front-end	back-end	front-end	back-end
DynaSLAM II	80.10	61.37	94.56	65.03
Ours	84.85	148.83	86.89	166.35

in computational time.

V. CONCLUSIONS

We propose DynaVIG, a navigation and object tracking system based on the Monocular Vision/INS/GNSS integration, which can eliminate the drift of traditional SLAM and realize 3D object tracking with a monocular camera. A prior height model is proposed for pose initialization and scale estimation of the object, and an accurate dynamics model

is constructed for precise tracking of objects with complex motion. Compared with the existing algorithms, DynaVIG achieves high-precision navigation and object tracking with real-time performance. In summary, DynaVIG is one of the state-of-the-art research of dynamic SLAM with object tracking. To the best of our knowledge, this is the first study using multi-sensor integration for accurate global navigation and object tracking.

ACKNOWLEDGMENT

Thanks to Tianyi Liu and Shaoquan Feng from Wuhan University, for their valuable suggestions of algorithm improvements and data processing.

REFERENCES

- [1] J. Zhang, M. Henein, R. Mahony, and V. Ila, "VDO-SLAM: a visual dynamic object-aware SLAM system," *arXiv preprint arXiv:2005.11052*, 2020.

- [2] B. Bescos, C. Campos, J. D. Tardós, and J. Neira, "DynaSLAM II: Tightly-Coupled Multi-Object Tracking and SLAM," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5191–5198, 2021.
- [3] M. Gonzalez, E. Marchand, A. Kacete, and J. Royan, "TwistSLAM: Constrained SLAM in Dynamic Environment," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6846–6853, 2022.
- [4] S. Cao, X. Lu, and S. Shen, "GVINS: Tightly Coupled GNSS–Visual–Inertial Fusion for Smooth and Consistent State Estimation," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2004–2021, 2022.
- [5] H. Tang, T. Zhang, X. Niu, J. Fan, and J. Liu, "IC-GVINS: A Robust, Real-time, INS-Centric GNSS-Visual-Inertial Navigation System for Wheeled Robot," *arXiv preprint arXiv:2204.04962*, 2022.
- [6] W. Wen, X. Bai, Y. C. Kan, and L.-T. Hsu, "Tightly Coupled GNSS/INS Integration via Factor Graph and Aided by Fish-Eye Camera," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10 651–10 662, 2019.
- [7] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D Object SLAM," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.
- [8] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [9] R. Mur-Artal and J. D. Tardós, "Visual-Inertial Monocular SLAM With Map Reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.
- [10] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1689–1696.
- [11] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [12] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," *arXiv preprint arXiv:1901.03642*, 2019.
- [13] G. Cioffi and D. Scaramuzza, "Tightly-coupled Fusion of Global Positional Measurements in Optimization-based Visual-Inertial Odometry," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5089–5095.
- [14] W. Lee, P. Geneva, Y. Yang, and G. Huang, "Tightly-coupled GNSS-aided Visual-Inertial Localization," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9484–9491.
- [15] E. Sucar and J.-B. Hayet, "Bayesian Scale Estimation for Monocular SLAM Based on Generic Object Detection for Correcting Scale Drift," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5152–5158.
- [16] D. Rukhovich, D. Mouritzen, R. Kaestner, M. Ruffi, and A. Velizhev, "Estimation of Absolute Scale in Monocular SLAM Using Synthetic Data," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 803–812.
- [17] D. Frost, V. Prisacariu, and D. Murray, "Recovering Stable Scale in Monocular SLAM Using Object-Supplemented Bundle Adjustment," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 736–747, 2018.
- [18] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [19] R. Mur-Artal and J. D. Tardós, "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, 2017.
- [20] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.
- [21] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, "Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment," *Robotics and Autonomous Systems*, vol. 117, pp. 1–16, 2019.
- [22] J. Shi and Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.
- [23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.
- [24] R. Jin, J. Liu, H. Zhang, and X. Niu, "Fast and Accurate Initialization for Monocular Vision/INS/GNSS Integrated System on Land Vehicle," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 26 074–26 085, 2021.
- [25] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.
- [26] F. Dellaert, "Factor graphs: Exploiting structure in robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 141–166, 2021.
- [27] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 573–580.