SLAM for Indoor Parking: A Comprehensive Benchmark Dataset and a Tightly Coupled Semantic Framework

XUAN SHAO, YING SHEN, LIN ZHANG, and SHENGJIE ZHAO, School of Software Engineering, Tongji University, China
DANDAN ZHU, Artificial Intelligence Institute, Shanghai Jiao Tong University, China
YICONG ZHOU, Department of Computer and Information Science, University of Macau, China

For the task of autonomous indoor parking, various Visual-Inertial Simultaneous Localization And Mapping (SLAM) systems are expected to achieve comparable results with the benefit of complementary effects of visual cameras and the Inertial Measurement Units. To compare these competing SLAM systems, it is necessary to have publicly available datasets, offering an objective way to demonstrate the pros/cons of each SLAM system. However, the availability of such high-quality datasets is surprisingly limited due to the profound challenge of the groundtruth trajectory acquisition in the Global Positioning Satellite denied indoor parking environments. In this article, we establish BeVIS, a large-scale *Be*nchmark dataset with *V*isual (front-view), *I*nertial and *S*urround-view sensors for evaluating the performance of SLAM systems developed for autonomous indoor parking, which is the first of its kind where both the raw data and the groundtruth trajectories are available. In BeVIS, the groundtruth trajectories are obtained by tracking artificial landmarks scattered in the indoor parking environments, whose coordinates are recorded in a surveying manner with a high-precision Electronic Total Station. Moreover, the groundtruth trajectories are comprehensively evaluated in terms of two respects, the reprojection error and the pose volatility, respectively. Apart from BeVIS, we propose a novel tightly coupled semantic SLAM framework, namely VIS_{SLAM}-2, leveraging *V*isual (front-view), *I*nertial, and *S*urround-view sensor modalities, specially for the task of autonomous indoor parking. It is the first work attempting to provide a general form to model various semantic objects on the ground. Experiments on BeVIS demonstrate the effectiveness of the proposed VIS_{SLAM}-2. Our benchmark dataset BeVIS is publicly available at https://shaoxuan92.github.io/BeVIS.

CCS Concepts: • **Computing methodologies → Reconstruction;**

Additional Key Words and Phrases: Autonomous indoor parking, benchmark dataset, groundtruth trajectory acquisition, Electronic Total Station, semantic SLAM

This work was supported in part by the National Natural Science Foundation of China under Grants 61973235, 61972285, and 61936014, in part by the Natural Science Foundation of Shanghai under Grant 19ZR1461300, in part by the Shanghai Science and Technology Innovation Plan under Grant 20510760400, in part by the Dawn Program of Shanghai Municipal Education Commission under Grant 21SG23, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100, and in part by the Fundamental Research Funds for the Central Universities.
Authors' addresses: X. Shao, Y. Shen (corresponding author), L. Zhang (corresponding author), and S. Zhao, School of Software Engineering, Tongji University, 4800 Cao'An Highway, Shanghai, China, 201804; emails: {1810553, yingshen, cslinzhang, shengjiezhao}@tongji.edu.cn; D. Zhu, Artificial Intelligence Institute, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai, China, 200240; email: ddz@sjtu.edu.cn; Y. Zhou, Department of Computer and Information Science, University of Macau, Taipa University Road, Macau, China; email: yicongzhou@um.edu.mo.

1551-6857/2023/1-ART1 $15.00
https://doi.org/10.1145/3510856

ACM Reference format:
Xuan Shao, Ying Shen, Lin Zhang, Shengjie Zhao, Dandan Zhu, and Yicong Zhou. 2023. SLAM for Indoor Parking: A Comprehensive Benchmark Dataset and a Tightly Coupled Semantic Framework. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 1, Article 1 (January 2023), 23 pages.
https://doi.org/10.1145/3510856

1 INTRODUCTION

Autonomous parking without human intervention is one of the most demanded and challenging tasks for an autonomous vehicle. The key to this task is the precise real-time localization of the autonomous vehicle. Currently there are different approaches to this task according to the parking environment of the autonomous vehicle. In an outdoor parking environment, a **Global Navigation Satellite System (GNSS)**, especially the **Global Positioning System (GPS)**, is a natural choice for the localization mission. As the poor coverage of satellite signals caused by occlusions weakens the performance of GNSS-based approaches, alternative approaches to reliable localization in an indoor parking environment are necessary. Among them, the **Simultaneous Localization And Mapping (SLAM)** systems enable an autonomous vehicle to simultaneously build a map of the indoor parking environment and track its position using this built map, simply by its on-board sensors. Since both visual cameras and the **Inertial Measurement Units (IMU)** are cheap, ubiquitous, and complementary, researchers have shown great enthusiasm for building **VI-SLAM (Visual-Inertial SLAM)** systems and a number of relevant VI-SLAM systems for autonomous indoor parking have been presented [3, 21, 27, 41, 53].

To compare these competing VI-SLAM systems for autonomous indoor parking, it is necessary to have publicly available benchmark datasets to demonstrate the pros/cons of each VI-SLAM system in an objective way. However, the availability of such high-quality datasets is surprisingly limited. A profound challenge with building such datasets lies in the difficulty of the groundtruth trajectory acquisition in an indoor parking environment. For the groundtruth trajectory acquisition in an outdoor parking environment, a GNSS is usually combined with other high-precision systems, such as the **Inertial Navigation System (INS)**, to ensure a reliable groundtruth trajectory. Due to the poor coverage of satellite signals in an indoor parking environment, a motion capture system is commonly selected instead. But the equipment is costly and time-consuming to set up, and its coverage capability is limited.

Additionally, while existing VI-SLAM systems for autonomous indoor parking have been successfully demonstrated in specific circumstances by incorporating semantic information in the indoor parking environment, the performance of these VI-SLAM systems would be unsatisfactorily compromised in the case of unexpected changes of the environment. For instance, the presence of a moving vehicle might lead to a large localization error over time without effective data association among stable features. Nevertheless, semantic objects on the ground (parking-slots, speed bumps and parking-slot IDs) are stable and salient features for this specific application scenario of autonomous indoor parking, exhibiting strong semantic consistency. Unfortunately, few eminent VI-SLAM systems have fully explored these features at present.

As mentioned, although the task of autonomous indoor parking has been explored for some time, appropriate benchmark datasets with groundtruth trajectories and reliable, mature SLAM systems are still lacking. In this work, we attempt to address these issues and the main contributions of this work are summarized as follows:

(1) We establish a benchmark dataset called ***Be*nchmark dataset with *V*isual (front-view), *I*nertial and *S*urround-view sensors (BeVIS)** for evaluating the performance of SLAM systems developed for autonomous indoor parking. In BeVIS, the groundtruth trajectories

are obtained by tracking artificial landmarks scattered in the indoor parking environments, whose coordinates are recorded in a surveying manner with a high-precision **Electronic Total Station (ETS)**, ensuring objective evaluation of different SLAM systems. To the best of our knowledge, BeVIS is a large-scale dataset where both the raw data and the groundtruth trajectories are provided, which is the first of its kind. BeVIS has been released to the community, which will facilitate the relevant studies for autonomous indoor parking.

(2) The groundtruth trajectories of the vehicle in BeVIS are comprehensively evaluated in terms of two respects, the reprojection error and the pose volatility, respectively. The reprojection error is used to quantify how closely an estimate of a three-dimensional (3D) point recreates the point's true projection to the camera. As for the pose volatility, it reflects the fluctuation of estimated camera pose when the vehicle remains stationary. Results demonstrate the effectiveness of our proposed approach to the groundtruth trajectory acquisition (please refer to Section 3.2 for details).

(3) We propose a tightly coupled semantic SLAM system, namely VIS_{SLAM}-2, inspired by VIS_{SLAM} proposed in Reference [41], specially for the task of autonomous indoor parking. Compared with VIS_{SLAM}, VIS_{SLAM}-2 is the first framework attempting to provide a general form to model all semantic objects on the ground. Its superiority over its rivals has been corroborated by extensive qualitative and quantitative experiments.

The remainder of this article is organized as follows. Section 2 introduces the related work. The benchmark dataset BeVIS and the evaluation of the groundtruth trajectory acquisition approach are presented in Section 3. Section 4 details the proposed VIS_{SLAM}-2 for autonomous indoor parking. Section 5 reports the experimental results and Section 6 concludes the article.

2 RELATED WORK

In this section, we give an overview of the benchmark datasets for evaluating SLAM systems and the VI-SLAM systems for autonomous indoor parking.

2.1 SLAM Benchmark Datasets

To evaluate the performance of different SLAM systems, several benchmark datasets were established. According to different sensor setups when collecting the datasets, two categories of datasets are briefly reviewed in this subsection, visual SLAM datasets and visual-inertial SLAM datasets.

Visual SLAM Datasets. The TUM Mono VO dataset [11] and the TUM RGB-D dataset [42] are two typical benchmark datasets for evaluating visual SLAM systems. The TUM Mono VO dataset is collected for evaluating the monocular odometry. It contains sequences in indoor and outdoor environments, which have been photometrically calibrated with respect to the exposure time and the lens vignetting, and so on. But, the groundtruth trajectories are not provided in the dataset. The TUM RGB-D dataset has been extensively used by the research community for the evaluation of RGB-D SLAM systems. It totally provides 47 RGB-D sequences with groundtruth trajectories recorded with a motion capture system. However, due to the limited coverage of a motion capture system, it can only record groundtruth trajectories in a small part of the environment.

Visual-inertial SLAM Datasets. Apart from visual images, visual-inertial SLAM datasets also involve motion data from IMUs, offering additional information for building robust VI-SLAM systems. The Kitti [16] and Malaga Urban [4] datasets are two popular datasets collected in the outdoor environments, in which the groundtruth trajectories are also given. But the IMU measurements in both datasets fail to be time synchronized with the visual images. To fully utilize the data from different sensors, datasets with time-synchronized visual and IMU data have been established. Among them, the Urban@CRAS [15] and KAIST Urban [19] datasets are recorded on urban outdoor roads

Table 1. Comparisons of Benchmark Datasets for Evaluating Different SLAM Systems

Benchmark Dataset	Environment	Sensor	Groundtruth
TUM Mono VO [11]	indoor	V	×
TUM RGB-D [42]	in/outdoor	V	MCS
Kitti [16]	outdoor	V-I	GPS/INS
Malaga Urban [4]	outdoor	V-I	GPS
Urban@CRAS [15]	outdoor	V-I	GPS
KAIST Urban [19]	outdoor	V-I	GPS
Oxford Multimotion [20]	indoor	V-I	MCS
EuRoC MAV [7]	indoor	V-I	MCS
TUM VI [39]	in/outdoor	V-I	MCS
BeVIS	indoor	V-I-S	ETS

(S: surround-view system; MCS: motion capture system).

under various driving conditions. For the aforementioned four datasets, the Malaga Urban [4], Urban@CRAS [15], and KAIST Urban [19] datasets provide coarse groundtruth trajectories from a low-cost GPS, whereas the Kitti dataset [16] provides GPS/INS-based groundtruth trajectories with an accuracy within 10 cm. However, the GPS-based groundtruth trajectory acquisition approaches are not applicable in the indoor environments due to the poor coverage of satellite signals in such environments.

Typical datasets collected in the indoor environments are the Oxford Multimotion dataset [20], the EuRoC **Micro Aerial Vehicle (MAV)** dataset [7], and the TUM VI dataset [39]. The Oxford Multimotion dataset [20] provides sequences for the evaluation of the vehicle's localization accuracy in dynamic indoor environments with multiple moving objects. The EuRoC MAV dataset [7] includes 11 indoor sequences recorded with a Skybotix stereo VI sensor from a MAV. The TUM VI dataset [39] provides 20-Hz images and time-synchronized accelerometer and gyro measurements at 200 Hz. But the groundtruth trajectories in the above three datasets are acquired using a motion capture system.

To summarize, the approaches to the groundtruth trajectory acquisition used in existing datasets are not applicable in the GPS-denied indoor parking environments or fail to guarantee the integrity of the groundtruth trajectories. This article seeks to establish a benchmark dataset BeVIS with groundtruth trajectories provided with the benefit of an ETS, which is both affordable and applicable in the indoor parking environments. The differences between BeVIS and other datasets for evaluating SLAM systems are presented in Table 1.

2.2 VI-SLAM Systems

According to the types of sensor fusion, VI-SLAM systems can be roughly divided into two categories, loosely coupled approaches [25, 47] and tightly coupled ones [5, 8, 23, 27, 31–33]. It needs to be noted that maps constructed by these VI-SLAM systems only provide geometric information, lacking a semantic understanding of the environment. To acquire a semantic understanding of the surrounding environment, VI-SLAM systems have recently begun to incorporate semantic features to build semantic VI-SLAM systems. First attemps among this line include References [9, 10, 12, 38]. Salas-Moreno et al. [38] proposed an object-oriented SLAM++, where semantic objects are manually edited in advance and the Iterative Closest Point method is used to obtain the camera pose during driving. To reduce the scale ambiguity and drift during driving, Frost et al. [12] proposed a SLAM system in which semantic objects in the environment are incorporated in a

bundle adjustment-inspired framework. However, features used in these systems are handcrafted, limiting their application in complicated scenarios with irregular and unexpected objects.

The rapid proliferation of deep learning techniques [6, 17, 18, 34–36] has given rise to a growing number of robust feature extraction strategies, which have boosted the localization accuracy of numerous VI-SLAM systems. Yang et al. [50] proposed a real-time monocular plane SLAM system that extracts planar features from a 3D plane model in the low-texture environments. Sünderhauf et al. [43] correlated labels with semantic objects based on the nearest neighbor method. These labelled semantic objects are served as semantic landmarks to effectively improve the localization accuracy of the SLAM system. Yang et al. [49] proposed a general SLAM system CubeSLAM for monocular 3D object detection and mapping. In CubeSLAM, a joint camera-object-point optimization scheme is utilized to construct the pose and scale constraints for graph optimization, enabling object-level mapping and localization of the SLAM system. To deal with dynamic objects in the surrounding environment, Mask-SLAM [21] and DynaSLAM [3] systems were proposed based on the ORB-SLAM2 system [26]. All *a priori* dynamic objects are segmented out in Mask-SLAM and DynaSLAM systems by the multi-view geometry technology. Nicholson et al. [28] developed a factor graph-based SLAM system that jointly estimates the camera pose and a 3D landmark representation of the environment.

However, an apparent shortcoming of the aforementioned SLAM systems is that they are prone to tracking inconsistency during driving. Specifically, when building SLAM systems for autonomous indoor parking, correct data association is essential to improve their localization accuracy and robustness. But the high presence of dynamic objects in the environment, like a moving car or pedestrian, corrupts the quality of pose estimation by deceiving the data association in these SLAM systems. By contrast, semantic objects on the ground (parking-slots, speed bumps, and parking-slot IDs) embody the most stable and consistent information in the indoor parking environment. Unfortunately, few SLAM systems hold the ability to perceive such salient features on the ground. The first work that leverages objects detected on the ground is the one established in Reference [53]. Zhao et al. [53] detected parking-slots in the surround-view images and incorporated them to the SLAM system. However, artificial landmarks are used to facilitate localization in Zhao et al.'s system, whereas parking-slots contribute little for optimization of the system. To the best of our knowledge, the latest work that leveraged objects detected on the ground is the one established in Reference [41]. Shao et al. [41] proposed the VIS_{SLAM} system where parking-slots in the surround-views are incorporated during optimization. However, surround-view features selected in VIS_{SLAM} are parking-slot specific, resulting in tracking inconsistency in circumstances where parking-slots are occluded by a parked car. Besides, the property of two neighboring parking-slots used in VIS_{SLAM} are scenario specific, rather than being completely general when it comes to different indoor parking environments where there are no neighboring parking-slots. The differences between our proposed VIS_{SLAM}-2 and other SLAM systems for autonomous indoor parking are summarized in Table 2.

3 THE BENCHMARK DATASET BEVIS

BeVIS established in this work is a large-scale benchmark dataset for evaluating the performance of SLAM systems developed for autonomous indoor parking. In this section, we will present the way we establish BeVIS.

3.1 Pipeline to Establish BeVIS

The pipeline to establish BeVIS is shown in Figure 1. As can be seen from Figure 1, there are four major steps involved: platform establishment, sensor calibration, data collection and groundtruth trajectory acquisition. Platform establishment ensures a modified electric vehicle with the

Table 2. Comparisons of SLAM Systems for Indoor Parking

SLAM System	Localization	Mapping	MSO
Mur-Artal et al.'s work [27]	V + I	geometric	×
Mask-SLAM [21]	V + I	semantic	×
DynaSLAM [3]	V + I	semantic	×
Zhao et al.'s work [53]	V + I + T (Tags)	semantic	×
VIS_{SLAM} [41]	V + I + S	semantic	×
VIS_{SLAM}-2	V + I + S	semantic	√

(MSO: multiple surround-view objects).

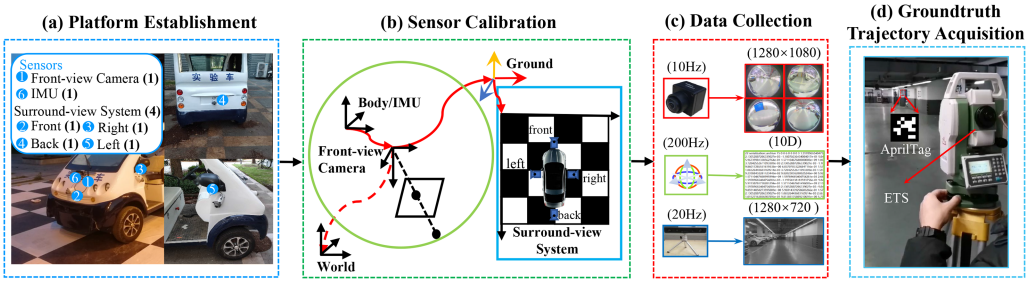

Fig. 1. Pipeline to establish BeVIS. There are four major steps involved, platform establishment, sensor calibration, data collection, and groundtruth trajectory acquisition. Platform establishment ensures a modified electric vehicle with the multi-sensor setup for the perception of the indoor parking environments. Sensor calibration is responsible for both the intrinsic and extrinsic calibration of all the sensors mounted on the vehicle. Afterwards, both the perceptual and navigational data are collected when driving the vehicle in different indoor parking environments. Finally, the groundtruth trajectories in BeVIS are obtained with the aid of an ETS.

multi-sensor setup for the perception of the indoor parking environments. Sensor calibration is responsible for both the intrinsic and extrinsic calibration of all the sensors mounted on the vehicle. Afterwards, both the perceptual and navigational data are collected when driving the vehicle in different indoor parking environments, followed by a classification process for the division of the difficulty level. Finally, the groundtruth trajectories in BeVIS are obtained with the aid of an ETS, which is both affordable and applicable in the indoor parking environments.

3.1.1 Platform Establishment. We selected an electric vehicle as our base platform for data collection. The vehicle was equipped with two types of sensors, perception sensors and navigation sensors. In the following, we briefly describe all the on-board sensors.

• *Front-view Perception Sensor:* The front-view perception sensor on the platform is a pinhole global-shutter visual camera, MYNTEYE D-1000-50, capturing images straight ahead of the vehicle. The camera can provide an image with the resolution of 1,280 × 720 at 30 **frames per second (fps)**. The HFOV, VFOV, and DFOV of the front-view perception sensor are 64°, 38°, and 70°, respectively. The Infrared Radiation module in the camera improves its adaptability to different lighting conditions of the indoor parking environments.

• *Surround-view Perception Sensor:* The surround-view perception sensor on the platform consists of four fisheye cameras, Leopard Imaging OV-10640-490, mounted on the front, left, back and right sides of the electric vehicle to form a surround-view camera system. The resolution, field-of-view, and acquisition frequency of each fisheye camera in the surround-view system are 1,280 × 1,080, 190° and 30 fps, respectively.

Table 3. The Data for the Calibration of All the Cameras on the Platform

Data \ Sensor	Surround-view System				Front-view	Total
	Front	Left	Back	Right		
intrinCalib-1	105	108	113	120	115	561
intrinCalib-2	84	98	86	100	95	463
extrinCalib-1	60	60	60	60	308	548
extrinCalib-2	312	312	312	312	1,353	2,601
extrinCalib-3	402	402	402	402	1,959	3,567

• *Navigation Sensor:* The navigation sensor on the platform is a consumer-grade, low-weight, 6 Degrees of Freedom IMU rigidly connected to the front-view camera. It can provide both the translation and orientation measurements through its accelerometers and gyroscopes, reflecting the ego-motion of the vehicle.

As seen in Figure 1(a), the orientation of each camera in the surround-view system is about 45° ground-oriented, capturing images of the ground around the vehicle. By calibrating the extrinsic parameters between the surround-view camera system and the ground plane, four fisheye images can be synthesized into a surround-view image from a top-down, bird's-eye view. The front-view camera was fixed higher than the front-view camera in the surround-view system, facing straight ahead to ensure a broad view of the camera.

3.1.2 Sensor Calibration. Sensor calibration consists of the intrinsic calibration and the extrinsic calibration of all sensors. The intrinsic calibration can be achieved in advance in an offline manner [48, 52].

For all the sensors, a schematic view of their coordinate systems is shown in Figure 1(b). In Figure 1(b), the dotted line indicates a temporally changing pose when moving the vehicle. The left green circle and the right blue rectangle contain the sensors that are rigidly connected to the vehicle, among which are a front-view camera, an IMU and a surround-view camera system. As for the extrinsic calibration, the rigid-body transformation matrix $\mathbf{T}_{BA}$, which allows the reprojection of any point from one coordinate system A to the other coordinate system B, is computed. According to different types of sensors, the extrinsic calibration can be categorized into three respects: camera-IMU calibration, camera-ground calibration and surround-view camera system calibration. Following References [13, 24], the transformation matrix $\mathbf{T}_{FB}$ from the IMU coordinate system B to the front-view camera coordinate system F and the transformation matrix $\mathbf{T}_{FG}$ from the ground coordinate system G to F can be obtained, respectively. Additionally, for the calibration of the surround-view camera system, it can be performed according to the methods proposed in References [40, 45].

Meanwhile, in BeVIS, we make the data for the calibration of all the cameras on the platform accessible such that users can perform their own calibration, even though we provide our calibration results. As seen in Table 3, the data can be divided into two categories as follows:

- intrinCalib-1/2: They are for the intrinsic calibration of the front-view camera and four fisheye cameras in the surround-view system. Images of a handheld 9×6 checkerboard were recorded by placing the checkerboard at different positions in front of each camera.
- extrinCalib-1/2/3: They are for the extrinsic calibration of all the cameras. One 10m $\times$ 10m calibration site on the ground was first established. This calibration site is with 10×10 squares and each square is 1 m in length. One point $\mathbf{P}$ was selected where all the cameras can see enough squares on the calibration site. By parking the electric vehicle on $\mathbf{P}$, images recorded by the surround-view camera system and the front-view camera were then simultaneously collected.

Table 4. The Overview of the Characteristics of Each Sequence in BeVIS

Char. / Sequence	Environment Characteristics			Trajectory Characteristics		
	DynaObj	Lighting	Feature	Scale	Speed	Initialization
SLAM-easy-01	small	bright	less	560s, 3 rounds	slow	long T, small R
SLAM-moderate-02	large	bright	abundant	263s, 1.5 rounds	fast	short T, big R
SLAM-difficult-03	moderate	changing	less	697s, 4 rounds	moderate	short T, small R
SLAM-difficult-04	small	dark	abundant	320s, 4.5 rounds	fast	short T, small R

(DynaObj: dynamic object; s: second; T: translation; R: rotation).

3.1.3 Data Collection. After sensor calibration, sequences in BeVIS were collected when driving the modified electric vehicle at around 10–40 km/h in four typical indoor parking sites. Note that for the front-view camera and the surround-view camera system, they are "software" synchronized by capturing images controlled by a multi-thread data collection function. For these cameras, each of them can capture images at 30 fps or higher theoretically. However, to ensure the image quality, the data collection frequency of each camera should be decreased to some extent. The sensors mounted on the vehicle would capture both perceptual and navigational data in a synchronized fashion during driving. We ensure the high quality of the collected images by the following two ways. First, for the selection of cameras, we chose the commonly used industrial cameras to ensure that all cameras can output high-resolution, high-quality images. Second, during data collection, we adjusted the collection frequency of each camera so that each frame can be successfully collected and saved.

Based on different characteristics of collected sequences in BeVIS, we manually classified them into three levels, "easy," "moderate," and "difficult." Specifically, both the environment and the trajectory characteristics are considered. The environment characteristics include the number of dynamic objects in the indoor parking environment, the illumination condition of the indoor parking environment, and the number of features in the indoor parking environment. The trajectory characteristics involve the total duration/rounds of each trajectory, the average speed of the vehicle, and the scale of initial movement (translation and rotation) of the vehicle. Note that if the initial translation and rotation of the vehicle is large, then all six axes of the IMU on the vehicle can be fully excited for better IMU initialization. Characteristics of each sequence in BeVIS are illustrated in Table 4.

3.1.4 Groundtruth Trajectory Acquisition. Actually, when establishing BeVIS, the groundtruth trajectories are crucial for the objective evaluation of different SLAM systems. Unfortunately, they are generally unavailable due to the fact that the current groundtruth trajectory acquisition approaches are unsuitable in the GNSS-denied indoor parking environments or fail to guarantee the integrity of the trajectories due to the high cost of a motion capture system. To address the problem, we provide an effective yet cost-efficient groundtruth trajectory acquisition approach. As can be seen in Figure 2, three steps are involved, artificial landmarks deployment, coordinates measurement and camera poses estimation. Specifically, artificial landmarks deployment ensures a tailored indoor parking environment with artificial landmarks that can be easily detected. Coordinates measurement is responsible for the 3D coordinates of these artificial landmarks to be measured by the ETS with a small measurement error. Afterwards, the front-view camera on the vehicle can be localized accordingly by tracking above artificial landmarks when driving the vehicle in the indoor parking environment.

Artificial Landmarks Deployment. The purpose of the artificial landmarks deployment is to ensure a tailored indoor parking environment with artificial landmarks that can be easily detected.

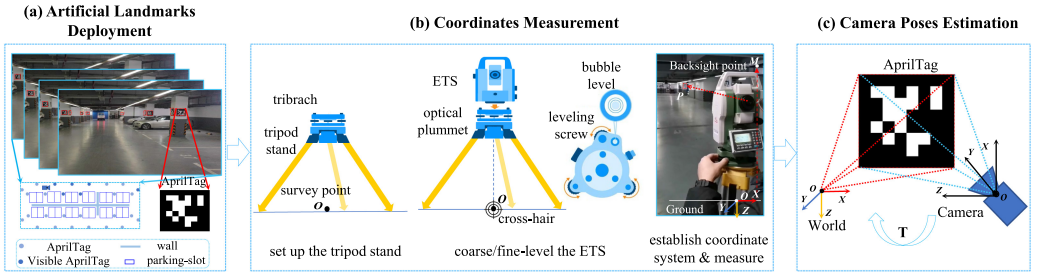

Fig. 2. Approach to groundtruth trajectory acquisition. Three steps are involved in this approach, artificial landmarks deployment, coordinates measurement, and camera poses estimation. Artificial landmarks deployment ensures a tailored indoor parking site with artificial landmarks. Coordinates measurement is responsible for the 3D coordinates of these artificial landmarks to be measured by the ETS. Afterwards, the front-view camera on the vehicle is localized accordingly by tracking above artificial landmarks.

In this step, AprilTags, the popular printable visual fiducial markers [29, 37, 46], are selected as the artificial landmarks. AprilTags are conceptually similar to QR Codes, but are designed to encode far smaller data payloads, allowing them to be detected more robustly in poor lighting conditions of the indoor parking environment. For each AprilTag, it is represented by one "quad," a valid four-sided region [29]. As seen in Figure 2(a), by evenly placing AprilTags in an indoor parking site, one can create a set of artificial landmarks scattered throughout the site.

Coordinates Measurement. Accurate coordinates of artificial landmarks are prerequisite for the high-precision camera poses estimation. To achieve this goal, the coordinates of four corners of the valid quad in each AprilTag are measured with an ETS, the most widely used equipment in the surveying field. An ETS is a compact, lightweight and portable equipment with an electronic theodolite for angle measurement and an Electromagnetic Distance Measuring instrument for distance measurement. With an ETS, an operator can take measurements of all the visible points' coordinates with accuracy within a couple of millimeters. As seen in Figure 2(b), there are five basic steps involved in coordinates measurement by an ETS.

• *Set up the Tripod Stand.* A survey point **O** on the ground is first selected where points in all directions can be observed as many as possible. Then the tripod stand is set up on **O** by extending the tripod legs to make the tripod head approximately level. By centering the tribrach on the tripod stand, both instruments are fastened together via a connecting screw.

• *Coarse-level the ETS.* Leveling the ETS must be accomplished in sufficient accuracy, otherwise it will not report results. Before attaching the ETS, the optical plummet on the tribrach is used to coarse-level the tripod stand. First, while holding two tripod legs, the operator should move the third tripod leg to keep the tripod as level as possible. And the optical plummet allows the operator to view the tribrach's center and place its cross-hair precisely over **O**. Then, the ETS can be secured to the top of the tribrach.

• *Fine-level the ETS.* Apart from the optical plummet, the tribrach features a bubble level and three leveling screws to fine-level the tripod as necessary. Viewing the bubble level, the operator can adjust the height of two tripod legs so that the bubble is close to the center of the bubble level. A more precise leveling result can be guaranteed by adjusting three leveling screws until the bubble is precisely in the center of the bubble level. By turning the ETS such that its face-plate is parallel with two leveling screws, the third screw is used to make the final fine adjustment.

• *Establish the Coordinate System.* To set up the coordinate system, three concurrent lines orthogonal with one another are required. Thus, four points are needed. First, **O** is defined as the origin of the coordinate system. By placing the ETS over **O** and designating a back-sight point

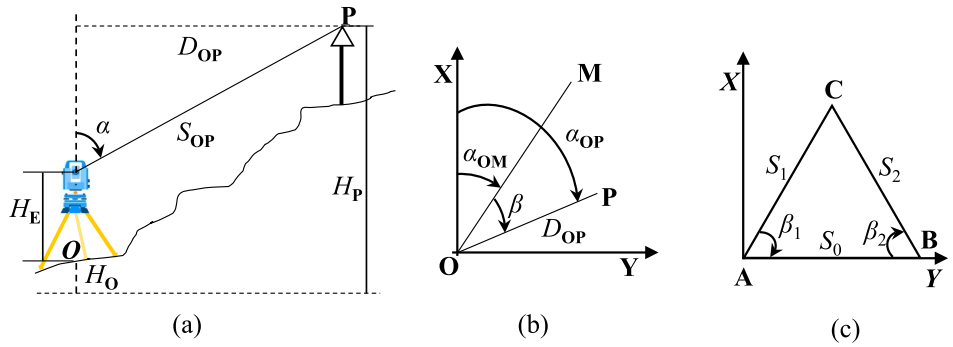

Fig. 3. Principles of (a) the computation of **Z** coordinate value, (b) the computation of **X** and **Y** coordinate values, and (c) the resection method.

M, the **X**-axis is then built pointing from **O** to the projection of **M** on the ground. The **Y**-axis is defined by a vector pointing from **O** that is orthogonal to the **X**-axis. And the **Z**-axis is perpendicular to both **X** and **Y** axes, which is directed vertically downward.

• *Measure the 3D Coordinate.* Based on the angle and distance measurements of the target point **P** by the ETS, **P**'s coordinate values, $X_{\mathbf{P}}$, $Y_{\mathbf{P}}$ and $Z_{\mathbf{P}}$, can be calculated accordingly. As can be seen in Figure 3(a), $Z_{\mathbf{P}}$ is represented by $H_{\mathbf{P}}$, the height of **P**,

$$Z_{\mathbf{P}} = H_{\mathbf{P}} = H_{\mathbf{O}} + S_{\mathbf{OP}} \cdot cos\alpha + H_{\mathbf{E}}, \tag{1}$$

where $H_{\mathbf{O}}$ is the height of **O**, $S_{\mathbf{OP}}$ is the distance from **O** to **P**, α is the vertical angle of $\overrightarrow{\mathbf{OP}}$, and $H_{\mathbf{E}}$ is the height of the ETS.

As seen in Figure 3(b), $X_{\mathbf{P}}$ and $Y_{\mathbf{P}}$ can be obtained by the following equation:

$$\begin{aligned} X_{\mathbf{P}} &= X_{\mathbf{O}} + D_{\mathbf{OP}} \cdot cos\alpha_{\mathbf{OP}}, \\ Y_{\mathbf{P}} &= Y_{\mathbf{O}} + D_{\mathbf{OP}} \cdot sin\alpha_{\mathbf{OP}}, \end{aligned} \tag{2}$$

where $D_{\mathbf{OP}}$ and $\alpha_{\mathbf{OP}}$ are defined as

$$\begin{aligned} D_{\mathbf{OP}} &= S_{\mathbf{OP}} \cdot sin\alpha, \\ \alpha_{\mathbf{OP}} &= \alpha_{\mathbf{OM}} + \beta, \end{aligned} \tag{3}$$

where $\alpha_{\mathbf{OM}}$ is the azimuth angle of $\overrightarrow{\mathbf{OM}}$ and β is the horizontal angle of $\overrightarrow{\mathbf{OP}}$.

By moving the ETS around, the coordinates of all artificial landmarks throughout the indoor parking site can be precisely acquired by the resection method [2]. As seen in Figure 3(c), assume **A** $(X_{\mathbf{A}}, Y_{\mathbf{A}})$ and **B** $(X_{\mathbf{B}}, Y_{\mathbf{B}})$ are two points whose coordinates are measured at **O**, and **C** is a new survey point. The slope distances of $\overrightarrow{\mathbf{AC}}$ and $\overrightarrow{\mathbf{BC}}$ are denoted by S_1 and S_2, respectively. $X_{\mathbf{C}}$ and $Y_{\mathbf{C}}$ can be formulated as

$$\begin{aligned} X_{\mathbf{C}} &= S_1 \cdot sin\beta_1, \\ Y_{\mathbf{C}} &= S_1 \cdot cos\beta_1, \end{aligned} \tag{4}$$

where β_1 can be obtained according to cosine theorem, i.e.,

$$\begin{aligned} S_2^2 &= S_1^2 + S_0^2 - 2S_1S_0cos\beta_1 \\ \rightarrow \beta_1 &= arccos\tfrac{S_1^2+S_0^2-S_2^2}{2S_1S_0}. \end{aligned} \tag{5}$$

Knowing the coordinates of **C**, the coordinates of points that are visible at **C** can be similarly acquired according to Equation (1) and Equation (2).

Camera Poses Estimation. When driving the vehicle in an indoor parking site, its mounted front-view camera could capture images of the site. **Perspective-n-Point (PnP)** can be applied to obtain the camera poses for each image.

• *PnP.* As seen in Figure 2(c), by aligning artificial landmarks with known 3D coordinates and their 2D projections into the front-view camera at time t with known pixel coordinates, the estimation of the camera pose $\mathbf{T}_{CW}^{t}$ at time t can be casted as solving a PnP problem, i.e.,

$$\mathbf{T}_{CW}^{t} = \arg\min_{\mathbf{T}_{CW}^{t}} \sum_{i=1}^{N} e_i^t = \arg\min_{\mathbf{T}_{CW}^{t}} \sum_{i=1}^{N} \left\| f\left(\mathbf{T}_{CW}^{t}, \mathbf{P}_W^i, \mathbf{D}\right) - \mathbf{p}_C^{t,i} \right\|_2^2, \tag{6}$$

where $\mathbf{D}$ is the set comprising the distortion coefficients of the camera, and $f(\ldots, \mathbf{D})$ is the camera distortion model that transforms each point $\mathbf{P}_W^i$ in the world coordinate system to the point on the camera's imaging plane $\mathbf{p}_C^{t,i}$ at time t. EPnP algorithm [22, 24] is adopted to solve the problem and several variants include DLT [1], P3P [14] and UPnP [30] can also be used to solve this problem. The optimal camera pose $\mathbf{T}_{CW}^{t}$ is acquired in an iterative manner for robustness and accuracy. Initially, $\mathbf{T}_{CW}^{t}$ is obtained using the RANSAC method and points with large reprojection errors are removed. Afterwards, $\mathbf{T}_{CW}^{t}$ is further refined using the remaining points until the number of the remaining points is stable. Make sure at least four AprilTags are visible in the front-view camera, providing a better tradeoff between speed and effectiveness.

3.2 Evaluation of the Groundtruth Trajectory Acquisition Approach

To comprehensively evaluate the performance of our proposed approach to the groundtruth trajectory acquisition, we define two metrics, **Re-Projection Error (*RPE*)** and **Pose Volatility (*PV*)**. *RPE* is a geometric error corresponding to the image distance between a projected point and its measured counterpart. It is used to quantify how closely an estimate of a 3D point recreates the point's true projection. As for *PV*, it reflects the overall fluctuation of estimated camera pose in $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ directions when the vehicle remains stationary, i.e.,

$$PV = \left(\frac{1}{N} \sum_{i=1}^{N} \left\| trans\left(\mathbf{T}_{CW}^{i}\right) - E\left(trans\left(\mathbf{T}_{CW}\right)\right) \right\|^2 \right)^{\frac{1}{2}}, \tag{7}$$

where N is the total number of images in each sequence, $trans(\mathbf{T}_{CW}^{i})$ is the translation part of the camera pose of the ith image, representing the camera's motion in the $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ directions. $E(trans(\mathbf{T}_{CW}))$ in Equation (7) is the average of translation parts of all camera poses in the sequence. When the vehicle remains stationary, the pose of its mounted camera is expected to be stably unchanged. Thus, a smaller *PV* indicates a higher accuracy of the estimated camera pose.

Meanwhile, nine image sequences were collected for the quantitative evaluation of the performance of the groundtruth trajectory acquisition approach. Among these sequences, there are five sequences that are collected in the "straight" areas of the indoor parking sites and four sequences in the "corner" areas, respectively. Characteristics of each sequence, the number of images in the sequence, the number of visible AprilTags in each image of the sequence, the number of survey points at which coordinates of these AprilTags are measured and the average distance of all these AprilTags to the camera, are detailed in Table 5.

4 VIS$_{SLAM}$-2 FOR AUTONOMOUS INDOOR PARKING

VIS$_{SLAM}$-2 is inspired by VIS$_{SLAM}$ proposed in Reference [41]. Designed for navigation in the indoor parking site, VIS$_{SLAM}$-2 is a tightly coupled semantic SLAM system that fully explores semantic objects detected in surround-views in its optimization framework. Specifically, to enhance the system's robustness against varying illumination and low-texture conditions, semantic objects

Table 5. Characteristics of Sequences for the Evaluation of Groundtruth Trajectory Acquisition Approach

Sequence \ Characteristic	Images	AprilTags	Survey Points	Avg. Distance
GT-straight-01	111	3	1	0.43 m
GT-straight-02	101	3	1	0.77 m
GT-straight-03	171	12	≥2	0.94 m
GT-straight-04	90	7	≥2	0.60 m
GT-straight-05	108	5	≥2	1.31 m
GT-corner-06	121	3	1	0.80 m
GT-corner-07	151	5	≥2	1.90 m
GT-corner-08	101	9	≥2	1.24 m
GT-corner-09	86	14	≥2	0.78 m

on the ground including parking-slots, speed bumps and parking-slot IDs are extracted from the surround-view images. And strong semantic constraints induced by these semantic objects are introduced in VIS_{SLAM}-2. Compared with VIS_{SLAM} that incorporates only adjacent parking-slots in indoor parking environments, VIS_{SLAM}-2 provides a general form to model various semantic objects on the ground. The joint optimization model of VIS_{SLAM}-2 will be detailed in this section with regard to its formulation and all error terms involved.

4.1 Joint Optimization Model Formulation

We first introduce the measurements and unknowns in VIS_{SLAM}-2. Given keypoints $\mathcal{Z}$ in the front-view image, IMU measurements $\mathcal{M}$ and semantic observations O in the surround-view image, the proposed joint optimization approach for VIS_{SLAM}-2 determines optimal camera poses $\mathcal{T}$, map points $\mathcal{P}$ matched with $\mathcal{Z}$ as well as surround-view landmarks $\mathcal{L}$, jointly. Such an optimization problem can be casted as,

$$\begin{aligned}
\{\mathcal{T},\mathcal{P},\mathcal{L}\}^* &= \arg\max_{\mathcal{T},\mathcal{P},\mathcal{L}} p(\mathcal{T},\mathcal{P},\mathcal{L}|\mathcal{Z},\mathcal{M},O). \\
&= \arg\max_{\mathcal{T},\mathcal{P},\mathcal{L}} p(\mathcal{T},\mathcal{P},\mathcal{L})p(\mathcal{Z},\mathcal{M},O|\mathcal{T},\mathcal{P},\mathcal{L}) \\
&= \arg\max_{\mathcal{T},\mathcal{P},\mathcal{L}} p(\mathcal{T},\mathcal{P},\mathcal{L})p(\mathcal{Z}|\mathcal{T},\mathcal{P},\mathcal{L})p(\mathcal{M}|\mathcal{T},\mathcal{P},\mathcal{L})p(O|\mathcal{T},\mathcal{P},\mathcal{L}) \\
&= \arg\max_{\mathcal{T},\mathcal{P},\mathcal{L}} \underbrace{p(\mathcal{T})p(\mathcal{P})}_{prior} \underbrace{p(\mathcal{Z}|\mathcal{T},\mathcal{P})p(\mathcal{M}|\mathcal{T})}_{visual-inertial\ term} \overbrace{\underbrace{p(\mathcal{L})}_{prior}\ \underbrace{p(O|\mathcal{T},\mathcal{L})}_{surround-view\ term}}^{observation}.
\end{aligned} \tag{8}$$

To find out optimal estimation of $\mathcal{T}$, $\mathcal{P}$ and $\mathcal{L}$, we jointly optimize visual, inertial and surround-view error terms, $\mathbf{E}_V$, $\mathbf{E}_I$, and $\mathbf{E}_S$, in a tightly coupled objective,

$$\{\mathcal{L},\mathcal{T},\mathcal{P}\}^* = \arg\min_{\mathcal{L},\mathcal{T},\mathcal{P}} \mathbf{E}_V + \mathbf{E}_I + \mathbf{E}_S + \mathbf{C}. \tag{9}$$

4.2 Surround-view Error Term Formulation

The visual-inertial error terms $\mathbf{E}_V$ and $\mathbf{E}_I$ can be modelled following References [27, 41], respectively. The error term $\mathbf{E}_S$ can be split into a prior error term $\mathbf{E}_{Prior}$ and an observation error term $\mathbf{E}_{Obs}$ corresponding to $p(\mathcal{L})$ and $p(O|\mathcal{T},\mathcal{L})$, respectively. Therefore, $\mathbf{E}_S$ can be defined as

$$\mathbf{E}_S = \mathbf{E}_{Prior} + \mathbf{E}_{Obs}. \tag{10}$$

4.2.1 Notation. Assuming that there exist M surround-view landmarks in the indoor parking environment. At time t, the vehicle obtains K_t surround-view observations, denoted by $O_t = \{\mathbf{O}_t^1; \mathbf{O}_t^2; \ldots; \mathbf{O}_t^{K_t}\}$. Data association of each observation is denoted by $y_t^i \in \{1; \ldots; M\}$. For example, at time $t = 1$, the surround-view camera system obtains two observations $O_1 = \{\mathbf{O}_1^1; \mathbf{O}_1^2\}$. And these two observations are from surround-view landmarks No. 2 and No. 4, then $y_1^1 = 2$ and $y_1^2 = 4$.

4.2.2 Semantic Object Detection and Localization. The semantic object detection framework used in VIS$_{\text{SLAM}}$-2 is inspired by a similar architecture dedicated to parking-slot detection published in our previous work [51]. Specifically, we proposed a two-stage object detection framework by first detecting marking-points of a parking-slot, the endpoints of a speed bump and the center of a parking-slot ID. Then, for parking-slot and speed bump detection, another classification module is used to tell if two marking-points/endpoints belong to the same parking-slot/speed bump. The position $\mathbf{L}_{y_t^i}$ of the ith surround-view landmark at time t in the world coordinate system can be obtained by the following equation, $\mathbf{L}_{y_t^i} = \mathbf{T}_t^{-1}\mathbf{O}_t^i$, where $\mathbf{O}_t^i$ is the position of the ith semantic observation at time t in the front-view camera coordinate system, which can be obtained by calibrating the transformation matrix between the front-view camera coordinate system and the ground coordinate system. $\mathbf{T}_t$ is the front-view camera pose in the world coordinate system at time t, which is returned by the visual odometry.

4.2.3 Prior Error Term. $p(\mathcal{L})$ models the prior distribution for positions of all surround-view landmarks, i.e.,

$$p(\mathcal{L}) = \prod_{t=1}^{T} p(\mathcal{L}_{y_t}),\ \mathcal{L}_{y_t} = \{\mathbf{L}_{y_t^i}\}_{i=0}^{N_t}, \tag{11}$$

where $p(\mathcal{L}_{y_t})$ is the prior distribution of the positions of surround-view landmarks at time t, $\mathbf{L}_{y_t^i}$ is the position of the ith surround-view landmark at time t, and N_t is the total number of surround-view landmarks at time t. $p(\mathcal{L}_{y_t})$ can be reformulated as

$$\begin{aligned} p(\mathcal{L}_{y_t}) &= p(\mathbf{L}_{y_t^1}) \prod_{i=2}^{N_t} p(\mathbf{L}_{y_t^i} | \mathbf{L}_{y_t^{i-1}} \mathbf{L}_{y_t^{i-2}} \ldots \mathbf{L}_{y_t^1}) \\ &\propto p(\mathbf{L}_{y_t^1}) \prod_{i=2}^{N_t} \prod_{j=1}^{i-1} p(\mathbf{L}_{y_t^i} | \mathbf{L}_{y_t^j}), \end{aligned} \tag{12}$$

where $p(\mathbf{L}_{y_t^1})$ follows a uniform distribution and $p(\mathbf{L}_{y_t^i} | \mathbf{L}_{y_t^j})$ is defined as

$$p(\mathbf{L}_{y_t^i} | \mathbf{L}_{y_t^j}) = \mathcal{N}\left(g\left(y_t^j\right), \Lambda_{i,t}\right), \tag{13}$$

where $\mathcal{N}(\,.\,,\,.\,)$ represents a normal distribution, $g(y_t^j)$ is the position of the ith surround-view landmark in the map induced by the jth surround-view landmark at time t, and $\Lambda_{i,t}$ models the uncertainty. As seen in Figure 4, when a semantic object on the ground is detected in the surround-view image, its coordinates in the ground coordinate system can be automatically obtained. So, the spatial relationship between any two observed semantic objects is naturally reflected in the surround-view image. Thus, $g(y_t^j)$ is defined as

$$\begin{aligned} g\left(y_t^j\right) &= \mathbf{L}_{y_t^j} + D_{y_t^i, y_t^j} \mathbf{s}_t^{i,j} \\ &where\ \ \mathbf{s}_t^{i,j} \quad // \quad \mathbf{O}_t^j \mathbf{O}_t^i, \end{aligned} \tag{14}$$

where $\mathbf{s}_t^{i,j}$ is a unit vector pointing to the ith surround-view observation from the jth surround-view observation at time t, and $D_{y_t^i, y_t^j}$ is the distance between two surround-view objects. Thus,

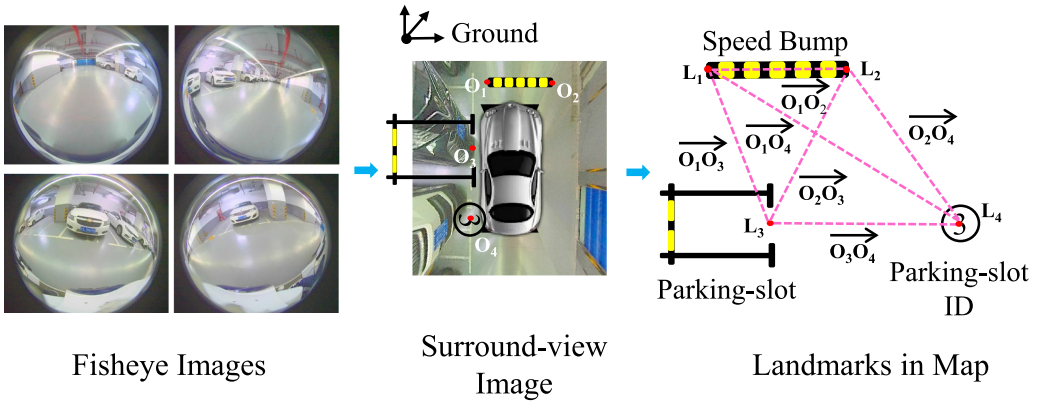

Fig. 4. Overview of the prior error term. When a semantic object on the ground is detected in the surround-view image, its coordinates in the ground coordinate system can be automatically obtained. So the spatial relationship between any two observed semantic objects on the ground is naturally reflected in the surround-view image.

the prior error term for the ith surround-view landmark at time t is given by

$$\mathbf{e}_{prior}^{i,t} = D_{y_t^i, y_t^j} \mathbf{s}_t^{i,j} - (\mathbf{L}_{y_t^i} - \mathbf{L}_{y_t^j}). \tag{15}$$

Intuitively, minimizing the prior error term implies iteratively tweaking each surround-view landmark in the map to ensure that its spatial relationship with any surround-view landmark conforms with that in the surround-view image.

4.2.4 Surround-view Error Term. Combining both the prior and the observation terms for all semantic objects, the surround-view error term $\mathbf{E}_S$ can be constructed, i.e.,

$$\begin{aligned} \mathbf{E}_S &= \mathbf{E}_{Prior} + \mathbf{E}_{Obs} \\ &= \sum_{t=1}^{T} \sum_{i=1}^{N_t} \left(\mathbf{e}_{prior}^{i,t}\right)^T \Lambda_{i,t} \mathbf{e}_{prior}^{i,t} + \sum_{t=1}^{T} \sum_{k=1}^{K_t} \left(\mathbf{e}_{obs}^{k,t}\right)^T \Phi_{k,t} \mathbf{e}_{obs}^{k,t}, \end{aligned} \tag{16}$$

where $\mathbf{e}_{obs}^{k,t}$ is the observation error term of the kth surround-view landmark observed at time t, whose definition is the same with the registration term in Reference [41]. Both $\Lambda_{i,t}$ and $\Phi_{k,t}$ in Equation (16) are in proportion to the detection confidence of each surround-view semantic object.

5 EXPERIMENTAL RESULTS AND DISCUSSIONS

5.1 Comparison of BeVIS and Its Counterparts

To facilitate the SLAM study for autonomous indoor parking, we have established and released BeVIS, which now can be publicly accessed at https://shaoxuan92.github.io/BeVIS. Actually, Shao et al. [41] released a dataset for autonomous indoor parking, the publicly available one in this field, and in this article it is referred to as Tongji Indoor Parking Dataset (TJIP for short). Information of both datasets is summarized in Table 6. It can be seen from Table 6 that from the perspectives of their scales and imaging conditions, BeVIS is much better than TJIP. Besides, compared with TJIP, the groundtruth trajectories are, for the first time, provided in BeVIS. Sequences for camera calibration and evaluation of the groundtruth trajectory acquisition approach are also provided. Thus, the following experiments were all conducted on BeVIS.

Table 6. Comparison of TJIP [41] and BeVIS. (GT: groundtruth)

Aspect \ Benchmark Dataset	TJIP	BeVIS
Number of Indoor Parking Sites	1	4
Number of Front-view Images/IMU data	5,000+	34,000+
Number of Surround-view Images	1,500+	12,000+
Imaging Conditions	simple	diverse
Groundtruth Trajectories	×	√
Camera Calibration Sequences	×	√
GT Trajectories Evaluation Sequences	×	√

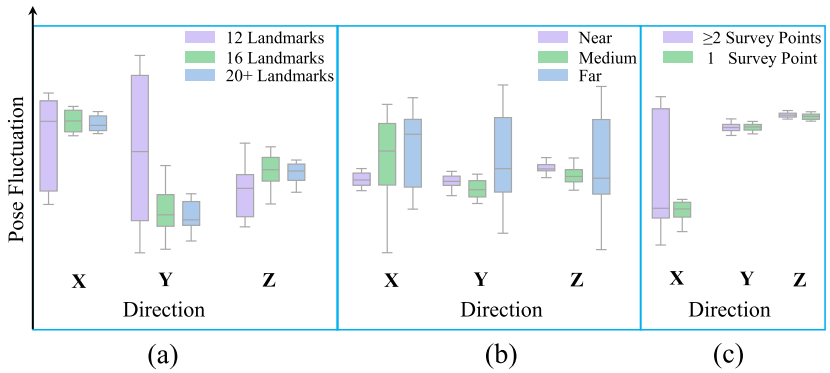

Fig. 5. Results of how selected factors influence the pose fluctuation in three directions. These factors are (a) the number of AprilTags, (b) the average distance to the camera of the AprilTags, and (c) the number of survey points at which AprilTags are measured. Note that "1.2e-3" and "6e-4" are two empirical thresholds for the factor D. When D > 1.2e-3, the average distance between the camera and the AprilTags is regarded as "near." When 6e-4 < D < 1.2e-3, the average distance between the camera and the AprilTags is regarded as "medium." When D < 6e-4, the average distance between the camera and the AprilTags is regarded as "far."

5.2 Factors Influencing the Performance of the GT Trajectory Acquisition Approach

Four factors, the number, the average size, and the average distance to the camera of the AprilTags as well as the number of survey points at which these AprilTags are measured, are selected to explore their influences on the performance of the groundtruth trajectory acquisition approach. Particularly, for the size and distance factors, we define a new factor D that simultaneously considers both factors,

$$D = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\sum_{j=1}^{n_i}\frac{W_q^{i,j}\cdot H_q^{i,j}}{W_I^i\cdot H_I^i}, \tag{17}$$

where m and n_i are the total number of images and the number of AprilTags detected in the ith image, respectively. $W_q^{i,j}$, $H_q^{i,j}$, W_I^i, and H_I^i in Equation (17) are the width and height of the jth AprilTag detected in the ith image as well as the width and the height of the ith image, respectively. In this experiment, the pose fluctuation in the **X**, **Y**, and **Z** directions is used as the performance measure, and the results are shown in Figure 5.

The Number of AprilTags. For each AprilTag, it is represented by four landmarks corresponding to its four corners. From Figure 5(a), we can see that as the number of landmarks increases,

Table 7. The Evaluation of the Accuracy of Groundtruth Trajectories in BeVIS

Sequence \ Metric	Landmarks	Pose Fluctuation			*PV*	*RPE*
		X	**Y**	**Z**		
GT-straight-01	12	0.99	3.88	0.93	4.11	0.25
GT-straight-02	12	1.32	3.07	0.87	3.45	0.30
GT-straight-03	20	1.71	4.60	6.44	8.10	0.57
GT-straight-04	14	0.26	0.40	0.28	0.56	0.28
GT-straight-05	8	9.3	21.08	8.96	24.96	0.52
GT-corner-06	20	0.70	0.74	0.71	1.24	0.39
GT-corner-07	20	2.24	0.44	0.60	2.37	0.52
GT-corner-08	12	2.05	1.79	2.02	3.39	0.26
GT-corner-09	16	0.26	3.03	3.07	4.32	0.24
Avg. (straight)	13	2.72	6.61	3.50	7.61	0.38
Avg. (corner)	17	1.31	1.50	1.60	2.83	0.35
Avg. (total)	15	2.09	4.34	2.65	5.49	0.37

the pose fluctuation in the **X**, **Y**, and **Z** directions drops by a large margin. Actually, 16 landmarks or more (4 AprilTags at least) can ensure an accurate estimation of the camera pose.

The Average Distance to the Camera of the AprilTags. Since all AprilTags are of the same size, a larger D indicates that the camera is much closer to the AprilTags. From Figure 5(b), we find that when D decreases, which means the average distance gets larger between the AprilTags and the camera, the approach would not report stable results. Additionally, to guarantee a stable outcome of the camera pose, D should be larger than 1.2e-3, an empirical threshold.

The Number of Survey Points at Which AprilTags Are Measured. From Figure 5(c), we find that if all AprilTags are measured at the same survey point, then a more consistent and stable camera pose can be guaranteed. Otherwise the camera pose would be unstable. Note that since we move the ETS roughly along the X direction, the cumulative errors in this direction will increase, leading to a relatively large value of the pose fluctuation in the X direction. These unstable camera poses will be filtered out, which will not affect the accuracy of the trajectory groundtruth.

In addition, we calculate both *RPE* and *PV* values for sequences from **GT-straight-01** to **GT-corner-09** to evaluate the accuracy of groundtruth trajectories in BeVIS. The results are detailed in Table 7. It can be seen from Table 7 that the average *PV* and *RPE* values are 5.49 cm and 0.37 px, respectively, both of which demonstrate the effectiveness of our proposed groundtruth trajectory acquisition approach. Besides, both *RPE* and *PV* values in the "straight" areas are larger than those in the "corner" areas. This is largely due to the fact that the average distance to the camera of the AprilTags in the "straight" areas is larger than that in the "corner" areas.

5.3 Quantitative Evaluation of VIS$_{SLAM}$-2

Four evaluation metrics are selected for quantitative evaluation of VIS$_{SLAM}$-2, the **revisiting error (RE)**, the **absolute trajectory error (ATE)**, the **distance of adjacent semantic objects (DAS)** and the **average processing time (APT)**.

(1) Revisiting Error. The revisiting error can be used to evaluate the localization accuracy of a SLAM system. It is valid in localization evaluation, because an autonomous parking system allows for an absolute localization error during driving. As long as the revisiting error is small enough, the vehicle will adopt a consistent driving strategy when it drives to the same position. We define

Table 8. The Revisiting Error of VIS_{SLAM}-2 in Sequences of BeVIS

Reference Point \ Revisiting Error				Δx	Δy	Δz	RE
S-e-01	−0.452	0.047	−0.250	0.01	0.01	0.01	0.017
	−5.591	−0.046	3.047	0.05	0.01	0.01	0.052
S-m-02	−19.70	−0.364	13.84	0.02	0.01	0.01	0.025
	−0.404	0.412	−7.98	0.03	0.01	0.01	0.033
S-d-03	−31.87	1.128	−0.643	0.01	0.01	0.01	0.017
	−42.96	0.592	6.370	0.03	0.01	0.01	0.033
S-d-04	−3.178	0.117	0.575	0.03	0.01	0.01	0.033
	−37.65	−0.201	19.78	0.05	0.02	0.01	0.055

(unit: meter).

the revisiting error as

$$e_{RE} = \left(\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left\| trans\left(\mathbf{P}_i^{j+1}\right) - trans\left(\mathbf{P}_i^{j}\right) \right\|^2 \right)^{\frac{1}{2}}, \tag{18}$$

where M denotes the total number of reference points, N is the total rounds, $\mathbf{P}_i^j$ denotes the camera pose of the ith reference point at the jth round, $\mathbf{P}_i^{j+1}$ is the camera pose of the reference point at the (j+1)-th round, and $trans(\,.\,)$ represents the translation part of the pose. From Table 8, we find that the average RE in BeVIS benchmark dataset is 0.033 m, demonstrating the effectiveness of our proposed VIS_{SLAM}-2.

(2) Absolute Trajectory Error. With the groundtruth trajectories acquired in Section 3, the absolute trajectory error can be used to evaluate the SLAM system's performance directly by comparing the difference between the estimated and the groundtruth trajectories, i.e.,

$$e_{ATE} = \left(\frac{1}{M} \sum_{i=1}^{M} \left\| trans\left(\mathbf{Q}_i^{-1}\mathbf{P}_i\right) \right\|^2 \right)^{\frac{1}{2}}, \qquad \mathbf{Q}_i, \mathbf{P}_i \in SE(3), \tag{19}$$

where M is the total number of frames in each sequence and $trans(.)$ represents the translation part of a camera pose. $\{\mathbf{Q}_i\}_{i=1}^M$ and $\{\mathbf{P}_i\}_{i=1}^M$ in Equation (19) are the aligned groundtruth and estimated poses of all frames, respectively. Note that the different coordination systems of the estimated trajectory and the groundtruth trajectory can be aligned with the Umeyama's method [44], which is a point cloud matching algorithm to transform the source cloud into the same coordinate system as the target cloud. We denote the original position of each point in the estimated trajectory by $\mathbf{P}_i^{'}$. A square root error can be established for the estimation of the transformation matrix between the estimated and the groundtruth trajectories,

$$e = \left(\frac{1}{M} \sum_{i=1}^{M} ||\mathbf{Q}_i - (s\mathbf{R}\mathbf{P}_i^{'} + \mathbf{t})||^2 \right)^{\frac{1}{2}}, \tag{20}$$

where $\mathbf{R}$, $\mathbf{t}$, and s are the rotation matrix, the translation matrix and the scale factor. By calculating the mean and variance of both trajectories as well as the SVD, the optimal $\mathbf{R}^*$, $\mathbf{t}^*$ and s^* can be obtained accordingly. When calculating the absolute trajectory error, $\mathbf{P}_i = \mathbf{R}^*\mathbf{P}_i^{'} + \mathbf{t}^*$. Similarly, e_{sATE}, which is the difference between the estimated and the groundtruth trajectories after being

Table 9. ATE and DAS of VIS_{SLAM}-2 in BeVIS w/o Surround-view Error Terms

Metric \ Sequence		S-e-1	S-m-2	S-d-3	S-d-4	Average
ATE	Without $\mathbf{E}_S$	0.520	0.608	0.943	0.872	0.736
	With $\mathbf{E}_S$	0.133	0.419	0.581	0.621	0.438
	Decrease	0.387	0.189	0.362	0.251	0.322
sATE	Without $\mathbf{E}_S$	0.512	0.201	0.243	0.193	0.287
	With $\mathbf{E}_S$	0.132	0.104	0.279	0.105	0.155
	Decrease	0.380	0.097	-0.036	0.088	0.132
DAS	Without $\mathbf{E}_S$	0.137	0.268	0.178	0.289	0.218
	With $\mathbf{E}_S$	0.078	0.118	0.104	0.144	0.111
	Decrease	0.059	0.150	0.074	0.145	0.107

(unit: meter).

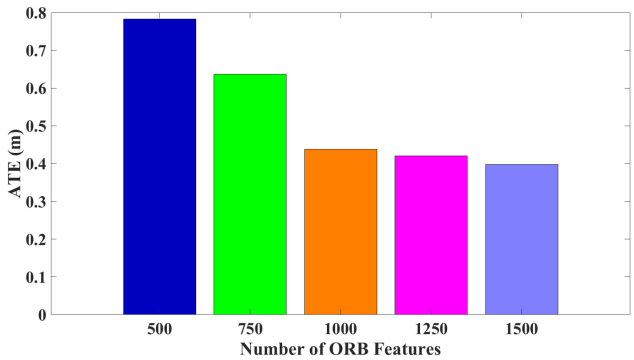

Fig. 6. The absolute trajectory error of VIS_{SLAM}-2 using different number of ORB features.

scaled to the same metric, can be defined as

$$e_{sATE} = \left(\frac{1}{M} \sum_{i=1}^{M} trans(\mathbf{Q}_i^{-1}(s^*\mathbf{R}^*\mathbf{P}_i^{'} + \mathbf{t}^*))||^2 \right)^{\frac{1}{2}}. \tag{21}$$

From Table 9, we find that the average ATE of all sequences is 0.438 m, which demonstrates the effectiveness of our VIS_{SLAM}-2. Besides, ATE in each sequence decreases after the optimization with $\mathbf{E}_S$. It can be also found in Table 9 that sATE, which is the difference between the estimated and the groundtruth trajectories after being scaled to the same metric, is smaller than ATE. But compared with ATE, sATE does not reflect the real trajectories differences, since the scale of the estimated trajectory is not the same with the groundtruth trajectory. Additionally, the relationship between ATE and the number of ORB features has been depicted in Figure 6. As shown in Figure 6, the ATE of VIS_{SLAM}-2 can be improved by increasing the number of extracted feature points. Actually, how to achieve a balance between speed and accuracy is a common practical engineering problem. It actually depends on which factor the end user attaches more importance to. In our system, when the number of used features is 1,000, the ATE can be as low as 0.438 m, and the processing speed can reach 14 fps. According to our experience, such mapping accuracy and processing speed can meet the needs of autonomous parking tasks.

Table 10. The Average Processing Time of VIS_{SLAM}-2 in BeVIS

Metric \ Sequence	S-e-1	S-m-2	S-d-3	S-d-4	Avg.
APT	0.068	0.072	0.067	0.078	0.071

(unit: second).

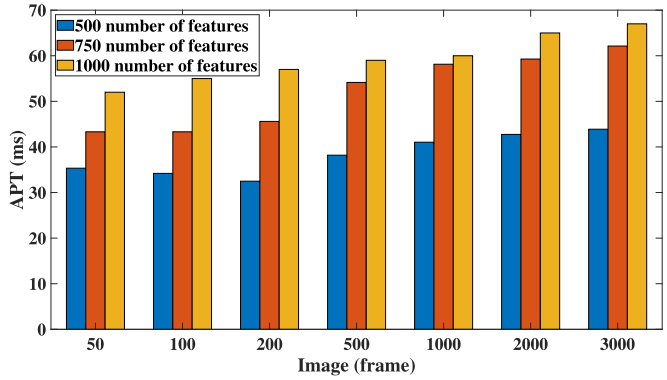

Fig. 7. The average processing time of VIS_{SLAM}-2 using different number of ORB features.

(3) Distance of Adjacent Semantic Objects. Distance of adjacent semantic objects is selected to evaluate the mapping accuracy of SLAM system. As can be seen from Table 9, we find that the average DAS of all sequences in BeVIS is 0.111 m, which demonstrates the effectiveness of our proposed VIS_{SLAM}-2. Besides, the DAS in each sequence decreases after the optimization with $\mathbf{E}_S$, demonstrating the effectiveness of the surround-view error term $\mathbf{E}_S$.

(4) Average Processing Time. We recorded the average processing time per frame of VIS_{SLAM}-2 when 1,000 ORB features are used. The result is presented in Table 10. It can be found that APT of VIS_{SLAM}-2 is 0.071 seconds, reaching 14 fps, which is qualified when the vehicle runs at a low speed in an indoor parking site. Additionally, APTs of VIS_{SLAM}-2 using different number of ORB features are also presented in Figure 7. In fact, the frame rate of VIS_{SLAM}-2 can be improved by reducing the number of extracted feature points. When the number of feature points is set as 500, the running speed undergoes a considerable improvement. Therefore, if there is a requirement for a higher frame rate, then we can reduce the number of extracted feature points.

5.4 Quantitative Comparison of VIS_{SLAM}-2 with Its Competitors

As seen in Section 2.2, among the existing SLAM studies for autonomous indoor parking, only Zhao et al.'s scheme [53] and Shao et al.'s scheme [41] make use of surround-view information. Therefore, in this quantitative comparison experiment, Zhao et al.'s work [53] and Shao et al.'s work [41] are chosen as the comparison targets. Apart from these two schemes, as a typical VI-SLAM system, Mur-Artal et al.'s scheme [27] is also included. The performance of these three competitors was evaluated in terms of RE, ATE, DAS, and APT, and the results are summarized in Table 11. It can be seen from Table 11 that Mur-Artal et al.'s work can reach satisfying performance with respect to three evaluation metrics of RE, ATE and APT. But it is not suitable for autonomous indoor parking due to the fact that it provides no semantic information during driving. From Table 11, we can see that VIS_{SLAM}-2 gains 88% and 78% of the favor compared with RE of 0.280 m in Zhao et al.'s work [53] and 0.157 m in Shao et al.'s work [41], respectively. Meanwhile, both ATE

Table 11. Quantitative Comparison of VIS$_{SLAM}$-2 with Its Competitors

Method \ Metric	RE	ATE	DAS	APT
Mur-Artal et al. [27]	0.239	0.501	—	**0.055**
Zhao et al. [53]	0.280	—	—	0.137
Shao et al. [41]	0.157	0.534	0.194	0.069
VIS$_{SLAM}$-2	**0.033**	**0.438**	**0.111**	0.071

The best result with respect to each performance metric is highlighted in bold.

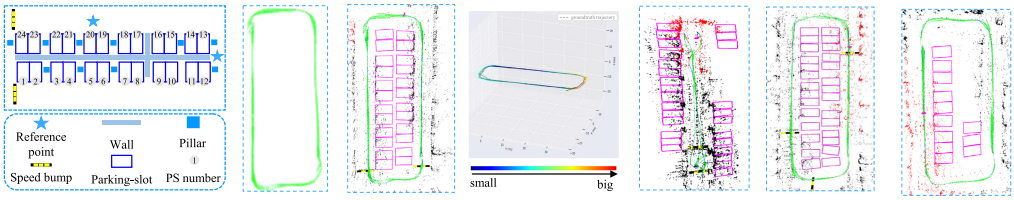

Fig. 8. Qualitative results of VIS$_{SLAM}$-2. (a) A sketch of the indoor parking site in SLAM-easy-01. (b) Mapping result using visual and inertial error terms during optimization. (c) Mapping result by VIS$_{SLAM}$-2 (parking-slot IDs are omitted here for display). (d) Difference between the estimated and the groundtruth trajectories. (e) Mapping result of SLAM-moderate-02. (f) Mapping result of SLAM-difficult-03. (g) Mapping result of SLAM-difficult-04.

and DAS of VIS$_{SLAM}$-2 enjoy a dramatic improvement by 0.096 m and 0.083 m compared with Shao et al.'s work [41], confirming the superiority of the localization and mapping accuracy of VIS$_{SLAM}$-2. Additionally, APT of VIS$_{SLAM}$-2 is significantly reduced compared with Zhao et al.'s work [53]. But compared with Shao et al.'s work [41], APT of VIS$_{SLAM}$-2 has a slight increase of 0.008s, which is due to the incorporation of more surround-view landmarks in its optimization model.

5.5 Qualitative Results of VIS$_{SLAM}$-2 in BeVIS

To qualitatively validate the effectiveness of the proposed VIS$_{SLAM}$-2, we evaluated it in terms of both the localization and mapping results in BeVIS. The mapping result of **SLAM-easy-01** is shown in Figure 8. Figure 8(a) depicts the sketch of the indoor parking site from a top-down viewpoint. Figure 8(b) illustrates the result incorporating both visual and IMU error terms during optimization. It records the driving path and maps the 3D landmarks in the indoor parking site (3D landmarks are omitted here for display). However, semantic objects on the ground that are essential for autonomous indoor parking are not incorporated in the map. Figure 8(c) demonstrates the result of VIS$_{SLAM}$-2, by which not only 3D landmarks but semantic objects detected in surround-view images are incorporated in the map. From Figure 8(c), we can find that the distances between each pair of adjacent parking-slots and the distances between the speed bumps and the parking-slots are in line with the spatial distribution of the real scene, which demonstrate the effectiveness of VIS$_{SLAM}$-2. Additionally, the difference between the trajectory estimated by our VIS$_{SLAM}$-2 and the groundtruth trajectory is illustrated in Figure 8(d). It can be seen from Figure 8(d) that the estimated and the groundtruth trajectories are roughly coincident, demonstrating the higher accuracy of the localization result of VIS$_{SLAM}$-2. Mapping results of other sequences in BeVIS are shown in Figure 8(e)–(g). Note that there are some parking-slots missing in the map. This is due to the fact that the entrance points of these parking-slots had been worn out or were occluded by parked cars.

Table 12. Optimization Results Using Various Error Terms

Configuration \ Metric	RE	ATE	DAS	APT
V-I_{SLAM}	0.239	0.501	—	**0.055**
VIS-T_{SLAM}	0.251	0.736	0.218	0.063
VIS_{SLAM}-2	**0.033**	**0.438**	**0.111**	0.071

The best result with respect to each performance metric is highlighted in bold.

5.6 Ablation Study of VIS_{SLAM}-2

We demonstrate how different error terms in our framework affect the optimization results by comparing VIS_{SLAM}-2 with two baselines using different optimization strategies. The two baselines are (1) V-I_{SLAM}: a visual-inertial error term based system without the incorporation of surround-view semantic features and (2) VIS-T_{SLAM}: a system that incorporates surround-view semantic features in optimization only during the tracking phase. The results are presented in Table 12. It can be seen from Table 12 that V-I_{SLAM} can reach satisfying performance with respect to three evaluation metrics of RE, ATE, and APT, which are 0.239 m, 0.501 m, and 0.055 seconds, respectively. But V-I_{SLAM} is not suitable for autonomous indoor parking due to the fact that it provides no semantic information during driving. As for the performance of VIS-T_{SLAM}, we can find that if we incorporate semantic features extracted from surround-views in optimization only during the tracking phase, then the optimization results are compromised and large RE and ATE errors occur. But if the surround-view semantic features are incorporated in optimization during all the phases of tracking, then local mapping and loop closing just as VIS_{SLAM}-2 does, three evaluation metrics of RE, ATE and DAS can be all considerably diminished, confirming the effectiveness of VIS_{SLAM}-2. In addition, APT of VIS_{SLAM}-2 is about 0.071 seconds (over 14 fps), which can be acceptable for an autonomous parking system running at a moderate speed.

6 CONCLUSION

In this article, we first establish a large-scale dataset called BeVIS, short for *Be*nchmark dataset with *V*isual (front-view), *I*nertial and *S*urround-view sensors. It contains synchronous multi-sensor data collected when driving a modified electric vehicle in four typical indoor parking sites. Notably, the groundtruth trajectories in BeVIS are obtained by tracking artificial landmarks scattered in these four indoor parking sites, whose coordinates are recorded in a surveying manner with a high-precision equipment ETS, enabling objective evaluation of different SLAM systems for autonomous indoor parking. The groundtruth trajectories are comprehensively evaluated in terms of two respects, the reprojection error and the pose volatility, respectively. To the best of our knowledge, as a benchmark dataset for evaluating the performance of SLAM systems developed for autonomous indoor parking, BeVIS is the first large-scale dataset where both the raw data and groundtruth trajectories are provided. Moreover, we propose a tightly coupled semantic SLAM framework, namely VIS_{SLAM}-2, leveraging *V*isual (front-view), *I*nertial, and *S*urround-view sensor modalities, especially for the task of autonomous indoor parking. It is the first work attempting to provide a general form to model the surround-view objects, and its effectiveness is verified by extensive experiments on BeVIS. In the future, we will continue enlarging BeVIS to make it a better benchmark in this field.

REFERENCES

[1] Youssef Ibrahim Abdel-Aziz and Hauck Michael Karara. 2015. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Photogram. Eng. Remote Sens.* 81, 2 (2015), 103–107.

[2] Joseph Awange, Erik Wilhelm Grafarend, Béla Paláncz, and Piroska Zaletnyik. 2010. *Positioning by Intersection Methods*. Algebraic Geodesy and Geoinformatics, Springer, Berlin, 249–263.
[3] Berta Bescos, José María Fácil, Javier Civera, and José Neira. 2018. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robot. Autom. Lett.* 3, 4 (2018), 4076–4083.
[4] Jose Luis Blanco-Claraco, Francisco Angel Moreno-Duenas, and Javier Gonzalez-Jimenez. 2014. The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario. *Int. J. Robot. Res.* 33, 2 (2014), 207–214.
[5] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. 2015. Robust visual inertial odometry using a direct EKF-based approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 298–304.
[6] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934. Retrieved from https://arxiv.org/abs/2004.10934.
[7] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W. Achtelik, and Roland Siegwart. 2016. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* 35, 10 (2016), 1157–1163.
[8] Carlos Campos, Richard Elvira, Juan José Gómez Rodríguez, José Manuel Montiel Montiel, and Juan Domingo Tardós. 2021. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Trans. Robotics* 37, 6 (2021), 1874–1890.
[9] Javier Civera, Dorian Galvez-Lopez, L. Riazuelo, Juan Domingo Tardos, and José Manuel Montiel. 2011. Towards semantic SLAM using a monocular camera. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1277–1284.
[10] Amaury Dame, Victor Adrian Prisacariu, Carl Yuheng Ren, and Ian Reid. 2013. Dense reconstruction using 3D object shape priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1288–1295.
[11] Jakob Engel, Vladyslav Usenko, and Daniel Cremers. 2016. A photometrically calibrated benchmark for monocular visual odometry. arXiv:1607.02555. Retrieved from https://arxiv.org/abs/1607.02555.
[12] Duncan Frost, Victor Prisacariu, and David Murray. 2018. Recovering stable scale in monocular SLAM using object-supplemented bundle adjustment. *IEEE Trans. Robot.* 34, 3 (2018), 736–747.
[13] Paul Furgale, Joern Rehder, and Roland Siegwart. 2013. Unified temporal and spatial calibration for multi-sensor systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1280–1286.
[14] Xiaoshan Gao, Xiaorong Hou, Jianliang Tang, and Hangfei Cheng. 2003. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 8 (2003), 930–943.
[15] Ana Rita Gaspar, Alexandra Nunes, Andry Maykol Pinto, and Aníbal Matos. 2018. Urban@CRAS dataset: Benchmarking of visual odometry and SLAM techniques. *Robot. Auton. Syst.* 109 (2018), 59–67.
[16] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* 32, 11 (2013), 1231–1237.
[17] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*. 1440–1448.
[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 580–587.
[19] Jinyong Jeong, Younggun Cho, Young Sik Shin, Hyunchul Roh, and Ayoung Kim. 2019. Complex urban dataset with multi-level sensors from highly diverse urban environments. *Int. J. Robot. Res.* 38, 6 (2019), 642–657.
[20] Kevin Michael Judd and Jonathan D. Gammell. 2019. The Oxford multimotion dataset: Multiple SE(3) motions with ground truth. *IEEE Robot. Autom. Lett.* 4, 2 (2019), 800–807.
[21] Masaya Kaneko, Kazuya Iwami, Torn Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Mask-SLAM: Robust feature-based monocular SLAM by masking using semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 371–3718.
[22] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. 2009. EPnP: An accurate O(n) solution to the PnP problem. *Int. J. Comput. Vis.* 81, 2 (2009), 155–166.
[23] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. 2015. Keyframe-based visual-inertial odometry using nonlinear optimization. *Int. J. Robot. Res.* 34, 3 (2015), 314–334.
[24] Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. 2007. Accurate non-iterative O(n) solution to the PnP problem. In *Proceedings of the IEEE International Conference on Computer Vision*. 2252–2259.
[25] Rodrigo Munguía, Emmanuel Nuno, Carlos I. Aldana, and Sarquis Urzua. 2016. A visual-aided inertial navigation and mapping system. *IEEE Int. J. Adv. Robot. Syst.* 13, 3 (2016), 94–112.
[26] Raúl Mur-Artal and Juan Domingo Tardós. 2017. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robot.* 33, 5 (2017), 1255–1262.
[27] Raúl Mur-Artal and Juan Domingo Tardós. 2017. Visual-inertial monocular SLAM with map reuse. *IEEE Robot. Autom. Lett.* 2, 2 (2017), 796–803.
[28] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. 2018. QuadricSLAM: Dual quadrics from object detections as landmarks in object-oriented SLAM. *IEEE Robot. Autom. Lett.* 4, 1 (2018), 1–8.

[29] Edwin Olson. 2011. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 3400–3407.
[30] Adrian Penate-Sanchez, Juan Andrade-Cetto, and Francesc Moreno-Noguer. 2013. Exhaustive linearization for robust camera pose and focal length estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 10 (2013), 2387–2400.
[31] Tong Qin, Shaozu Cao, Jie Pan, and Shaojie Shen. 2019. A general optimization-based framework for global oose estimation with multiple sensors. arXiv:1901.03642. Retrieved from https://arxiv.org/abs/1901.03642.
[32] Tong Qin, Peiliang Li, and Shaojie Shen. 2018. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* 34, 4 (2018), 1004–1020.
[33] Tong Qin and Shaojie Shen. 2018. Online temporal calibration for monocular visual-inertial systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 3662–3669.
[34] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6517–6525.
[35] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. arXiv:1804.02767. Retrieved from https://arxiv.org/abs/1804.02767.
[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 6 (2017), 1137–1149.
[37] Andrew Richardson, Johannes Strom, and Edwin Olson. 2013. AprilCal: Assisted and repeatable camera calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1814–1821.
[38] Renato F. Salas-Moreno, Richard Newcombe, Hauke Strasdat, Paul Kelly, and Andrew Davison. 2013. SLAM++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1352–1359.
[39] David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. 2018. The TUM VI benchmark for evaluating visual-inertial odometry. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1680–1687.
[40] Xuan Shao, Xiao Liu, Lin Zhang, Shengjie Zhao, Ying Shen, and Yukai Yang. 2019. Revisit surround-view camera system calibration. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. 1486–1491.
[41] Xuan Shao, Lin Zhang, Tianjun Zhang, Ying Shen, and Yicong Zhou. 2020. A tightly-coupled semantic SLAM system with visual, inertial and surround-view sensors for autonomous indoor parking. In *Proceedings of the ACM International Conference on Multimedia*. 2691–2699.
[42] Jrgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. 2012. A benchmark for the evaluation of RGB-D SLAM systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 573–580.
[43] Niko Sünderhauf, Trung Pham, Yasir Latif, Michael Milford, and Ian Reid. 2017. Meaningful maps with object-oriented semantic mapping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 5079–5085.
[44] S. Umeyama. 1991. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* 13, 4 (1991), 376–380.
[45] Chunxiang Wang, Hengrun Zhang, Ming Yang, Xudong Wang, Lei Ye, and Chunzhao Guo. 2014. Automatic parking based on a bird's eye view vision system. *Adv. Mechanical Engineering* 6 (2014), 847406.
[46] John Wang and Edwin Olson. 2016. AprilTag 2: Efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 193–4198.
[47] Stephan Weiss and Roland Siegwart. 2011. Real-time metric state estimation for modular vision-inertial systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 4531–4537.
[48] Oliver J. Woodman. 2017. An Introduction to Inertial Navigation. Retrieved from https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf.
[49] Shichao Yang and Sebastian Scherer. 2019. CubeSLAM: Monocular 3D object SLAM. *IEEE Trans. Robot.* 35, 4 (2019), 925–938.
[50] Shichao Yang, Yu Song, Michael Kaess, and Sebastian Scherer. 2016. Pop-up SLAM: Semantic monocular plane SLAM for low-texture environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1222–1229.
[51] Lin Zhang, Junhao Huang, Xiyuan Li, and Lu Xiong. 2018. Vision-based parking-Slot detection: A DCNN-based approach and a large-scale benchmark dataset. *IEEE Trans. Image Process.* 27, 11 (2018), 5350–5364.
[52] Zhengyou Zhang. 2000. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 11 (2000), 1330–1334.
[53] Junqiao Zhao, Yewei Huang, Xudong He, Shaoming Zhang, Chen Ye, Tiantian Feng, and Lu Xiong. 2019. Visual semantic landmark-based robust mapping and localization for autonomous indoor parking. *Sensors* 19, 1 (2019), 161–180.

Received 28 July 2021; revised 28 November 2021; accepted 9 January 2022