2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

VIODE: A Simulated Dataset to Address the Challenges of Visual-Inertial Odometry in Dynamic Environments

Koji Minoda[1], Fabian Schilling[2], Valentin Wüest[2], Dario Floreano[2], and Takehisa Yairi[1]

***Abstract*—Dynamic environments such as urban areas are still challenging for popular visual-inertial odometry (VIO) algorithms. Existing datasets typically fail to capture the dynamic nature of these environments, therefore making it difficult to quantitatively evaluate the robustness of existing VIO methods. To address this issue, we propose three contributions: firstly, we provide the VIODE benchmark, a novel dataset recorded from a simulated UAV that navigates in challenging dynamic environments. The unique feature of the VIODE dataset is the systematic introduction of moving objects into the scenes. It includes three environments, each of which is available in four dynamic levels that progressively add moving objects. The dataset contains synchronized stereo images and IMU data, as well as ground-truth trajectories and instance segmentation masks. Secondly, we compare state-of-the-art VIO algorithms on the VIODE dataset and show that they display substantial performance degradation in highly dynamic scenes. Thirdly, we propose a simple extension for visual localization algorithms that relies on semantic information. Our results show that scene semantics are an effective way to mitigate the adverse effects of dynamic objects on VIO algorithms. Finally, we make the VIODE dataset publicly available at https://github.com/kminoda/VIODE.**

***Index Terms*—Data Sets for SLAM, Visual-Inertial SLAM, Aerial Systems: Perception and Autonomy**

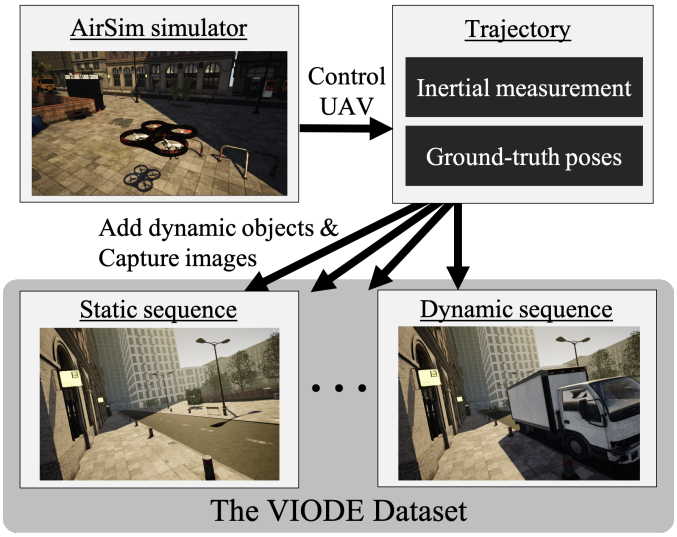

Fig. 1. The VIODE dataset is created using a photorealistic UAV simulator to which we systematically add moving objects while keeping trajectory, lighting conditions, and static objects the same – a setup which would be prohibitively difficult to achieve with real-world data. Each environment in the dataset contains four sequences with the same trajectory and IMU data, but with increasing levels of dynamic objects.

I. Introduction

VISION-BASED localization in dynamic environments is an important and challenging task in robot navigation. Camera images can provide a rich source of information to estimate the pose of a robot, especially in environments such as indoor and urban areas where GNSS information may be unreliable or entirely unavailable. In such environments, however, dynamic objects such as humans, vehicles, or other mobile robots often coexist in the same workspace. These dynamic objects can be detrimental to vision-based algorithms since they commonly use the assumption of a static world in which they self-localize. Breaking the static-world assumption may result in large estimation errors since the algorithms cannot distinguish between dynamic objects and static ones.

One possible way to remove these errors is to introduce additional proprioceptive information, such as data from inertial measurement units (IMUs), into the estimation pipeline. Unlike a camera that measures the environment, proprioceptive sensors measure the internal state of the robot, which is usually not affected by the environment. Thus, visual-inertial odometry (VIO) has the potential to perform more robustly compared to purely visual odometry (VO). However, as we show in this study, adding inertial information does not necessarily guarantee robustness, and further information is required to obtain a reliable state estimation algorithm.

Despite the importance of visual-inertial fusion for robust localization in dynamic scenarios, there are no publicly available datasets which could be used as a benchmark of VIO algorithms in dynamic environments. While a considerable amount of VIO datasets are proposed, most of them contain only a few dynamic objects. While there are VIO algorithms that aim to perform robustly in dynamic environments [1], [2], it is difficult to quantitatively compare them due to the lack of a specific benchmark.

This paper presents a novel dataset called VIODE (VIO dataset in Dynamic Environments) – a benchmark for assessing the performance of VO/VIO algorithms in dynamic scenes. The environments are simulated using AirSim [3], which is a photorealistic simulator geared towards the development of perception and control algorithms. The unique advantage of VIODE over existing datasets lies in the systematic introduction of dynamic objects at increasing numbers and in different environments (see Fig. 1). In each environment, we use the

Manuscript received: October, 15, 2020; Revised January, 10, 2021; Accepted January, 31, 2021.

This paper was recommended for publication by Editor Javier Civera upon evaluation of the Associate Editor and Reviewers' comments.

[1]Koji Minoda and Takehisa Yairi are with the Artificial Intelligence Laboratory, Department of Aeronautics and Astronautics, University of Tokyo, Tokyo, Japan. koji.m.minoda@gmail.com

[2]Fabian Schilling, Valentin Wüest, and Dario Floreano are with the Laboratory of Intelligent Systems, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

Digital Object Identifier (DOI): see top of this page.

TABLE I
COMPARISON OF EXISTING DATASETS AND OURS

Dataset name	Year	Carrier	Sensor setup	Ground-truth trajectory	Environment	Dynamic level
KITTI [5]	2012	Car	Stereo, IMU, LiDAR	Fused IMU, GNSS	Outdoors	Low
Malagan Urban [6]	2014	Car	Stereo, IMU	GNSS	Outdoors	Mid
UMich NCLT [7]	2016	Segway	Omni, IMU, LiDAR	Fused GNSS, IMU, LiDAR	In-/outdoors	Low
EuRoC MAV [8]	2016	UAV	Stereo, IMU	MoCap	Indoors	Low
Zurich Urban [9]	2017	UAV	Monocular, IMU	Fused camera, Google Street View	Outdoors	Mid
PennCOSYVIO [10]	2017	Handheld	Stereo, IMU	AprilTag markers	In-/outdoors	Low
TUM VI [11]	2018	UAV	Stereo, IMU	MoCap (partial)	in-/outdoors	Low
ADVIO [12]	2018	Handheld	Monocular, ToF, IMU	IMU with manual pose fix	In-/outdoors	High
Urban@CRAS [13]	2018	Car	Stereo, IMU, LiDAR	GNSS (partial)	Outdoors	Mid
Oxford Multimotion [14]	2019	Handheld	Stereo, RGBD, IMU	MoCap	Indoors	Mid
KAIST Urban [15]	2019	Car	Stereo, IMU	Fused GNSS, Fiber optic gyro, encoder, LiDAR	Outdoors	Mid
OIVIO [16]	2019	Handheld	Stereo, IMU	MoCap (partial), ORB-SLAM2 (partial)	In-/outdoors	Low
UZH-FPV Drone Racing [17]	2019	UAV	Stereo, IMU, event	Laser tracker	In-/outdoors	Low
UMA-VI [18]	2020	Handheld	Stereo, IMU	Camera (partial)	In-/outdoors	Low
Blackbird UAV [19]	2020	UAV	Stereo,* IMU, depth,* segmentation*	MoCap	In-/outdoors	Low
VIODE (ours)	2021	UAV (simulation)	Stereo,* IMU,* segmentation*	Simulation	In-/outdoors	High

* Simulated sensor

same trajectory of the aerial vehicle to create data sequences with an increasing number of dynamic objects. Therefore, VIODE users can isolate the effect of the dynamic level of the scene on the robustness of vision-based localization algorithms. Using VIODE, we show that the performance of state-of-the-art VIO algorithms degrades as the scene gets more dynamic.

Our dataset further contains ground-truth instance segmentation masks, since we believe that semantic information will be useful for researchers and engineers to develop robust VIO algorithms in dynamic scenes. In order to assess the effects of semantic information in algorithms operating on the VIODE dataset, we develop a method incorporating semantic segmentation into VINS-Mono [4]. We refer to this method as VINS-Mask in our research. VINS-Mask masks out the objects which are assumed to be dynamic. Using our dataset, we demonstrate that VINS-Mask outperforms the existing state-of-the-art algorithms. Though VINS-Mask uses ground-truth segmentation labels provided by AirSim, these results indicate the potential of the utilization of scene semantics in dynamic environments.

In summary, the main contributions of our work are:

- We propose a novel simulated visual-inertial dataset (VIODE) where we systematically add dynamic objects to allow benchmarking of the robustness of VO and VIO algorithms in dynamic scenes.
- Using VIODE, we experimentally show performance degradation of state-of-the-art VIO algorithms caused by dynamic objects.
- We show that incorporation of semantic information, implemented as a state-of-the-art VIO algorithm with semantic segmentation, can alleviate the errors caused by dynamic objects.

II. Related Work

In this section, we briefly outline related work in three areas. Firstly, we summarize existing VIO datasets and their shortcomings. Secondly, we discuss the state-of-the-art of VIO algorithms. Finally, we highlight methods for robust vision-based localization in dynamic environments.

A. VIO datasets

Existing VIO datasets are diverse, as they span various carriers, environments, and sensor setups. However, they are not sufficient to assess the robustness of VIO in dynamic environments on 6-DoF trajectories.

Most of the datasets that are recorded on UAV or handheld rigs contain only few moving objects. For instance, widely used VIO benchmarks such as the EuRoC MAV dataset [8] only contain a small number of people moving in indoor scenes and thus the scenes are mostly static. Thus they are not suitable for benchmarks for systematically studying performance in environments with moving objects.

Outdoor datasets tend to contain more dynamic objects. The TUM VI dataset [11] and the Zurich Urban dataset [9] contain outdoor sequences where sporadically moving vehicles and people appear. KITTI [5], Urban@CRAS [13], and KAIST Urban [15] are recorded from driving cars in urban areas. These datasets, especially the latter two, contain several moving vehicles. However, the field of view in these datasets is still mainly filled with static objects and are thus not challenging enough to benchmark robustness in dynamic environments. The Oxford Multimotion Dataset [14] is an indoor dataset which contains dynamic objects. Their dataset aims at providing a benchmark for motion estimation of moving objects as well as vehicle self-localization. However,

the scenes do not contain environments typically encountered by robots such as UAVs. Furthermore, these datasets do not cover the most challenging dynamic scenes which can occur in real-world applications of vision-based localization.

To the best of our knowledge, the ADVIO dataset [12] contains the most challenging scenes in terms of dynamic level among the existing VIO datasets. However, their work does not contain quantitative information on the dynamic level. Moreover, unlike our VIODE dataset, the effect of dynamic objects cannot be isolated on the robustness of VIO.

In conclusion, the existing VIO datasets are not suitable to systematically benchmark the robustness of VIO methods in the presence of dynamic objects. The VIODE dataset seeks to enhance the development of localization in dynamic scenes by providing a challenging, quantitative, and high-resolution benchmark. We achieve this goal with a systematic configuration of dynamic objects and an evaluation of dynamic levels in each scene. Tab. I provides a comparison of existing datasets and the VIODE dataset.

B. VIO algorithms

Here, we provide an overview of existing VIO algorithms, including VINS-Mono [4] and ROVIO [20] which we benchmark on the VIODE dataset.

Common VIO algorithms can be classified as either filter-based or optimization-based. MSCKF [21] and ROVIO [20] are both filter-based algorithms that fuse measurements from cameras and IMUs using an extended Kalman filter. On the contrary, optimization-based approaches utilize energy-function representations for estimating 6-DoF poses. OKVIS [22] utilizes non-linear optimization and corner detector. VINS-Mono [4] and VINS-Fusion [23] are optimization-based algorithms that contain a loop detection and closure module and, as a result, achieve state-of-the-art accuracy and robustness. Furthermore, the recently proposed ORB-SLAM3 [24] is an optimization-based visual-inertial SLAM algorithm which achieves high performance and versatility. In [25] the authors provide a comprehensive comparison among the existing open-sourced monocular VIO algorithms. Their survey highlights that VINS-Mono and ROVIO are the two best-performing methods among the existing monocular VIO algorithms in terms of accuracy, robustness, and computational efficiency. Thus, in this study, we compare these two algorithms on the VIODE dataset.

C. Vision-based localization in dynamic environments

In the context of vision-based localization algorithms such as visual odometry (VO) and visual simultaneous localization and mapping (vSLAM), the random sample consensus (RANSAC) [26] is the most commonly used algorithm. RANSAC is an iterative algorithm to estimate a model from multiple observations which include outliers. Thus, common vision-based methods often adopt this algorithm to enhance their performance in dynamic environments. One of the major drawbacks is that it is more likely to fail when the observation contains outliers that follow the same hypothesis. For example, RANSAC in visual localization is more vulnerable to one large rigid-body object than several small objects with different motions.

Some recent works report that the use of semantic segmentation or object detection can improve the robustness of algorithms in dynamic environments. Mask-SLAM [27] uses the ORB-SLAM architecture while masking out dynamic objects using a mask generated from semantic segmentation. DynaSLAM [28] is built on ORB-SLAM2 and combines a geometric approach with semantic segmentation to remove dynamic objects. The authors of Mask-SLAM and DynaSLAM evaluate proposal methods on their original dataset recorded in dynamic environments. Empty Cities [29] integrates dynamic object detection with a generative adversarial model to inpaint the dynamic objects and generate static scenes from images in dynamic environments.

There are several works that not only address robust localization in dynamic environments but also extend the capability to track surrounding objects [30], [31], [32], [33]. For example, the results of DynaSLAM II [30] show that the objects motion estimation can be beneficial for ego-motion estimation in dynamic environments.

Compared to VO/vSLAM, the performance of VIO algorithms in dynamic environments has not been investigated in depth. Several works, however, report that the utilization of semantic/instance segmentation improves the performance of VIO in dynamic environments [1], [2]. A limitation of these two studies is, however, that they reported performances only on limited data. For example, [1] uses only a short subsequence of the KITTI dataset. Our VIODE benchmark is well suited to systematically assess and compare these VIO algorithms.

III. THE VIODE DATASET

We created the VIODE dataset with AirSim [3], a photo-realistic simulator based on Unreal Engine 4, which features commonly used sensors such as RGB cameras, depth cameras, IMUs, barometers, and LiDARs. To simulate motions that are typically encountered in VIO applications, we used a quadrotor as the carrier vehicle as it is a platform where VIO algorithms are commonly employed.

The VIODE dataset is designed for benchmarking VIO algorithms in dynamic environments. This can be accredited to three unique features. Firstly, compared to other existing datasets, VIODE contains highly dynamic sequences. Secondly, we provide quantitative measures of the dynamic level of the sequences based on clearly defined metrics. Finally, and most importantly, VIODE allows us to isolate the effect of dynamic objects on the robustness of VIO. In general, the performance of VIO is influenced not only by dynamic objects but also by a variety of parameters such as the accuracy of the IMU, the type of movements (e.g. rotation or translation), the lighting condition, and the texture of surrounding objects. Therefore, a way to fairly assess algorithm robustness in dynamic scenarios is to compare multiple sequences where the only difference is the dynamic level. To achieve this, we create four sequences of data on the same trajectory and strategically add dynamic objects.

A. Dataset content

The dataset is recorded in three *environments*, one indoor and two outdoor environments. In each environment, we generate four *sequences* which are recorded while executing the same trajectory. The only difference between these four recorded sequences is the number of dynamic vehicles. To set up the situation as realistic as possible, we also placed static vehicles in all the sequences. The three environments in our dataset are as follows (see Fig. 2):

- `parking_lot` contains sequences in parking lot indoor environment.
- `city_day` contains sequences during daytime in modern city environment. This is a commonly encountered outdoor environment which is surrounded by tall buildings and trees.
- `city_night` contains sequences during night time in the same environment as `city_day`. Trajectories of ego-vehicle and dynamic objects are different from those of the `city_day` environment.

B. Dataset generation

The procedure for generating VIODE data involves two steps: 1. collect IMU and ground-truth 6-DoF poses, and 2. capture images and segmentation masks (see Fig. 1). These steps were executed on a desktop computer running Windows 10 with an Intel(R) Core i7-9700 CPU and a GeForce RTX 2700 SUPER GPU.

In the first step, we use the ROS wrapper provided by AirSim to generate synchronized IMU measurements and ground-truth odometry data at a rate of 200 Hz. While recording this sensory data, we control the UAV with the PythonAPI in AirSim. We generate one trajectory for each environment. Each of the trajectories contains linear accelerations along the x/y/z axes and rotation movements around roll/pitch/yaw. In order to obtain accurate timestamps, we slow down the simulation by a factor of 0.05 during this procedure.

In a second step, we add images to the above data. To do so, we undersample ground-truth poses from the odometry sequence to a rate of 20 Hz. In each of the undersampled poses, we capture synchronized RGB images and segmentation maps with stereo cameras . This procedure is done four times in each environment to generate four sequences with different dynamic levels: `none`, `low`, `mid`, and `high`.

C. Sensor setup and calibration

The UAV in AirSim is equipped with a camera and an IMU. The camera captures RGB images and AirSim provides segmentation maps.

- **Stereo camera** uses two equal cameras and captures time-synchronized RGB images at a rate of 20 Hz with WVGA resolution (752×480 px). We set up the stereo camera with a baseline of 5 cm. Both of these cameras are global shutter cameras and have a 90° field of view (FOV).
- **Ground-truth segmentation** images are computed for both cameras at a rate of 20 Hz, synchronized with the corresponding RGB images (see Fig. 2). The provided segmentation is an instance segmentation. As such, the vehicles and all the other objects in the scene are labeled as different objects.
- **IMU** data is also acquired from AirSim at a rate of 200 Hz. This uses the same parameters as MPU-6000 from TDK InvenSense. The noise follows the model defined in [34]. Since our IMU data does not take into account the vibration of rotors, the noise magnitude is lower than that of IMU data recorded on real-world UAV flights and similar to the one recorded with handheld carriers.
- **Ground-truth trajectory** is also included for all the sequences in our dataset. This consists of ground-truth 6-DoF poses provided by the simulator. This is recorded at a rate of 200 Hz.
- We also provide ground-truth **intrinsic and extrinsic parameters** for the stereo camera and camera-IMU extrinsics.

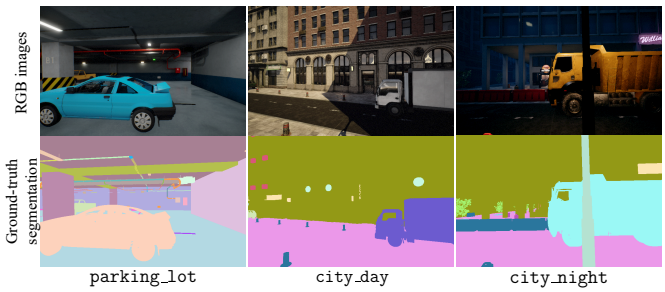

Fig. 2. The images in the first row are RGB image examples from each of the environments and the images in the second row are the corresponding ground-truth segmentation maps provided by AirSim. In the VIODE dataset, we included these two images in a time-synchronization fashion.

D. Dynamic objects & dynamic level evaluation

We use cars provided in Unreal Engine 4 as dynamic objects since cars are commonly encountered dynamic objects in real-world applications. We use six types of vehicles with varying textures and sizes. These dynamic objects are controlled by Unreal Engine with constant speeds along the designated trajectories. As mentioned previously, we generated four sequences in each environment: `none`, `low`, `mid`, and `high`. The `none` sequences do not contain any dynamic objects. Starting from `none`, we progressively add the vehicles to generate three different dynamic sequences: `low`, `mid`, and `high`. Furthermore, we allocate not only dynamic vehicles but also some static ones to make the scenes more realistic.

We also introduce metrics to evaluate how dynamic the sequences are. Simply counting the number of dynamic vehicles in the scene is not a suitable metric of the dynamic level. For example, even if the number of dynamic objects is the same, the scene is more dynamic when the objects are close to the camera. Thus, a better metric is necessary to evaluate the dynamic level. Possible information sources for defining a better metric are semantic information, optical flow, ground-truth ego-motion, or rigid-body-motion of vehicles. Here, we

TABLE II
BASIC PARAMETERS OF VIODE DATASET

		Indoor/ outdoor	Distance [m]	Duration [s]	# of static objects	# of dynamic objects	Average pixel-based dynamic rate [%]	Maximum pixel-based dynamic rate [%]	Average OF-based dynamic rate [px]	Maximum OF-based dynamic rate [px]
parking_lot	none	Indoor	75.8	59.6	2	0	0.0	0.0	0.0	0.0
	low					1	0.5	13.1	0.03	0.83
	mid					2	8.9	68.6	0.22	3.92
	high					6	10.5	66.9	0.31	3.75
city_day	none	Outdoor	157.7	66.4	2	0	0.0	0.0	0.0	0.0
	low					1	1.1	50.3	0.03	4.66
	mid					3	2.0	52.1	0.14	5.51
	high					11	8.4	98.8	0.45	8.72
city_night	none	Outdoor	165.7	61.6	1	0	0.0	0.0	0.0	0.0
	low					1	0.9	33.4	0.08	3.10
	mid					3	1.6	32.9	0.11	3.59
	high					11	3.5	37.6	0.21	3.31

assess the dynamic level by considering how much of the FOV is occupied by dynamic objects based on ground-truth instance segmentation. Pixel-based dynamic rate r_{pix} is defined as

$$r_{\text{pix}} = N_{\text{pix}}^{\text{dyn}} / N_{\text{pix}}^{\text{all}} \tag{1}$$

where $N_{\text{pix}}^{\text{dyn}}$ is a number of pixels occupied by dynamic vehicles and $N_{\text{pix}}^{\text{all}}$ is the total number of pixels.

However, pixel-based dynamic rate does not contain speed information of the surrounding objects. In order to obtain a metric for the magnitude of the motion of objects, we also define the optic-flow-based (OF-based) dynamic rate r_{of} as

$$r_{\text{of}} = \frac{1}{N} \sum_{(x,y)\in D} \sqrt{\|\nabla I(x,y) - \nabla I_{\text{none}}(x,y)\|^2} \tag{2}$$

where D is a set of pixel coordinates in the frame which belong to dynamic objects. We determine D from ground-truth instance segmentation. N is the total number of pixels, $\nabla I(x,y)$ is the optic flow at (x,y) in a frame, and $\nabla I_{\text{none}}(x,y)$ is the optic flow at (x,y) in a corresponding frame in `none` sequence from the same environment.

We summarize these parameters as well as the basic parameters such as the total distance and duration of each data sequence (see Tab. II). We further provide the two types of dynamic rate along the time axis (see Fig. 8).

IV. Evaluation of VIO Methods On The VIODE Dataset

A. Evaluation metrics

To analyze the performance of VIO algorithms, we use the absolute trajectory error (ATE) [35] which is defined as

$$\text{ATE}(\mathbf{P}_{1:n}, \mathbf{Q}_{1:n}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|\text{trans}(\mathbf{F}_i)\|^2} \tag{3}$$

where $\mathbf{P}_1, ..., \mathbf{P}_n \in \text{SE}(3)$ is the estimated trajectory, $\mathbf{Q}_1, ..., \mathbf{Q}_n \in \text{SE}(3)$ the ground-truth trajectory, $\mathbf{F}_i = \mathbf{Q}_i^{-1}\mathbf{S}\mathbf{P}_i$ is the absolute trajectory error at time step i, $\text{trans}(\mathbf{X})$ refers to the translational components of $\mathbf{X} \in \text{SE}(3)$, and $\mathbf{S}$ the rigid-body transformation between $\mathbf{P}_{1:n}$ and $\mathbf{Q}_{1:n}$ calculated by optimizing a least-square problem. Moreover, we use relative pose error (RPE) [35] for local accuracy of the sub-trajectory. RPE at i-th frame is defined as

$$\text{RPE}(\mathbf{P}_{1:n}, \mathbf{Q}_{1:n}, i) = \|\text{trans}(\mathbf{E}_i)\| \tag{4}$$

where $\mathbf{E}_i = (\mathbf{Q}_i^{-1}\mathbf{Q}_{i+\Delta})^{-1}(\mathbf{P}_i^{-1}\mathbf{P}_{i+\Delta})$ is relative pose error at time step i. Here, Δ is a fixed time interval for which we calculate the local accuracy.

We additionally introduce degradation rate r_d for the VIODE dataset. This is defined as $r_d = \text{ATE}_{\text{high}}/\text{ATE}_{\text{none}}$ (5), where ATE_{high} and ATE_{none} are the ATE of the algorithm in the `high` and `none` sequences respectively. This metric illustrates the robustness of the VIO algorithm in dynamic sequences compared to that in static sequences. We calculate this value for all evaluated algorithm in each environment.

TABLE III
DEGRADATION RATE r_d FOR EACH ALGORITHM/ENVIRONMENT

Environment\Algorithm	ROVIO	VINS-Mono	VINS-Mask (Ours)
parking_lot	5.27	14.7	**1.12**
city_day	17.1	16.2	**1.26**
city_night	5.19	1.91	**1.26**

B. Existing VIO algorithms

We apply ROVIO and VINS-Mono, two state-of-the-art VIO algorithms, on the VIODE dataset. Since the performance of these algorithms is not deterministic, each is evaluated ten times.

Our findings show that both ROVIO and VINS-Mono perform worse in the presence of dynamic objects. We observe that the ATE of both algorithms increases as the scene gets more dynamic (see Fig. 3). Furthermore, estimated trajectories illustrate the degradation of VINS-Mono in highly-dynamic sequences. For example, in `parking_lot/high`, there is a drift around $(x,y) = (0,8)$ while is not present in `parking_lot/none` (see Fig. 6). The degradation rate is mostly higher than 5.0 for ROVIO and VINS-Mono (see Tab. III). Comparing the degradation rate of ROVIO and

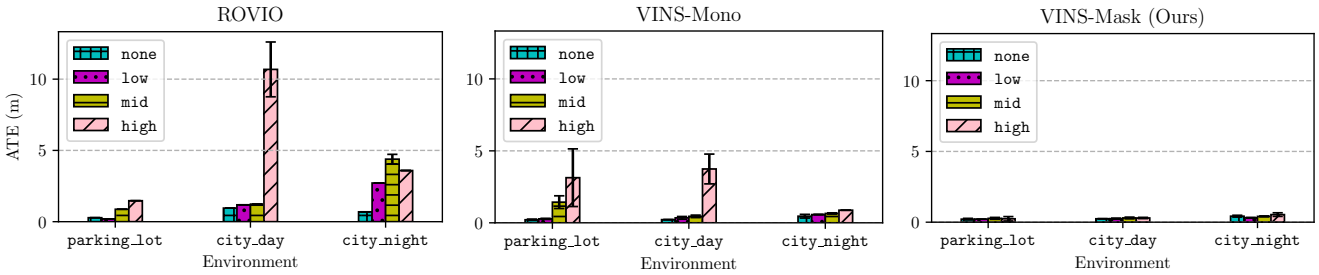

Fig. 3. Absolute trajectory error (ATE) [m] of ROVIO, VINS-Mono, and VINS-Mask. The performance of these methods is not deterministic. Thus, we run the simulation ten times for every sequence. We observe an increase in errors as the dynamic level increases for ROVIO and VINS-Mono. On top of that, the ATE of VINS-Mask remains considerably lower than the other two algorithms.

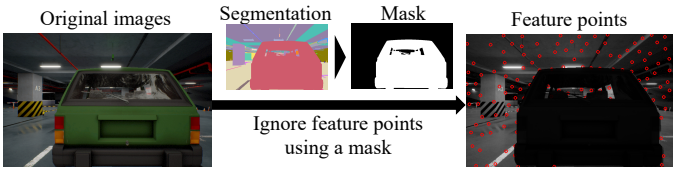

Fig. 4. In VINS-Mask, we create a binary mask from segmentation maps to ignore dynamic regions. Since this is based on VINS-Mono, these feature points are coupled with IMU information to estimate the 6-DoF poses.

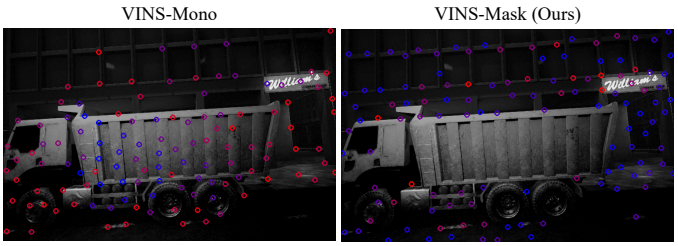

Fig. 5. We illustrate the extracted and tracked feature points in VINS-Mono and VINS-Mask as red and blue circles. The more red the feature point is colored, the longer the feature has been tracked and more likely to be used in VINS-Mono & VINS-Mask to estimate 6-DoF pose. VINS-Mono relies a lot on feature points on dynamic objects while VINS-Mask excludes those regions.

VINS-Mono among the three environments, `city_day` environment is the most challenging one among the three. The degradation rate of both ROVIO and VINS-Mono was the lowest in `city_night`, in which the average and maximum pixel-based dynamic rate were also the lowest (see Fig. 2).

We show that the wrong estimation often occurs when the dynamic objects are in FOV. One evidence is a correspondence between two types of dynamic rate (r_{pix} and r_{of}) and RPE estimated by VINS-Mono (see Fig. 8). For example, r_{pix}, r_{of}, and RPE mark the highest value during $40-50$ s in `parking_lot/high`. These correspondences between RPE and two types of dynamic rate support our hypothesis that the performance degradation of current VIO algorithms is caused by dynamic objects.

C. VINS-Mask

One of our goals is to show that scene semantics have the potential to improve the performance of VIO in challenging dynamic scenes. To this end, we develop the method VINS-Mask, which is based on VINS-Mono [4]. VINS-Mask uses the same algorithm as VINS-Mono, except that it avoids using feature points on dynamic objects by leveraging semantic information. In the feature points extraction phase of VINS-Mask, we perform semantic/instance segmentation for each image. By exploiting the segmentation and preliminary knowledge, we mask out the region of the dynamic objects right before the feature extraction phase of VINS-Mono (see Fig. 4). By applying the mask, VINS-Mask can estimate the robots' pose based on reliable feature points (see Fig. 5). Although this technique is not a novel idea as stated in Sec. III, we use this method to assess the impact of semantic information on the robustness of VIO in dynamic scenarios.

In our work, we use ground-truth segmentation labels provided in the VIODE dataset, instead of applying a semantic segmentation algorithm on camera images. The mask consists of the region which belongs to vehicles. We include both dynamic and static vehicles in the mask to make the setup more realistic, although it is possible to distinguish moving objects from static objects with ground-truth instance segmentation. This is on account of the fact that it is another challenging task to distinguish moving objects from static objects in real-world applications.

In addition to ROVIO and VINS-Mono, here we evaluate VINS-Mask on the VIODE dataset. VINS-Mask is also applied ten times for each sequence. We found that the performance of the VINS-Mask is almost independent of the dynamic level. The ATE of VINS-Mask is mostly lower than the results of ROVIO and VINS-Mono (see Fig. 3). The degradation rate of VINS-Mask is consistently lower than the other two algorithms (see Tab. III). The estimated trajectories in `high` sequences for each algorithm also illustrates the improvement by VINS-Mask (see Fig. 7). While the existing algorithms exhibit some drifts in highly dynamic sequences, VINS-Mask successfully suppresses these drifts. It is also worth noting that VINS-Mask performs similarly with VINS-Mono in static sequences, in which VINS-Mask masks out static objects while VINS-Mono does not. The improvement by VINS-Mask suggests the usefulness of this type of approach as a technique to run VIO robustly in dynamic environments. Besides that, the results further strengthen our confidence that the performance degradation of the existing algorithms is due to the dynamic objects in the scene.

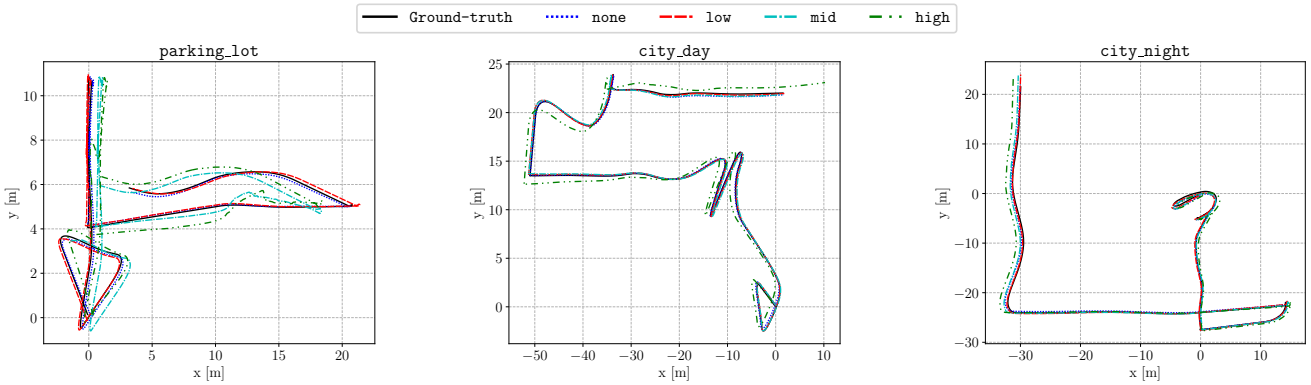

Fig. 6. Estimated trajectory of VINS-Mono in `none`, `low`, `mid`, and `high` sequence for each environment.

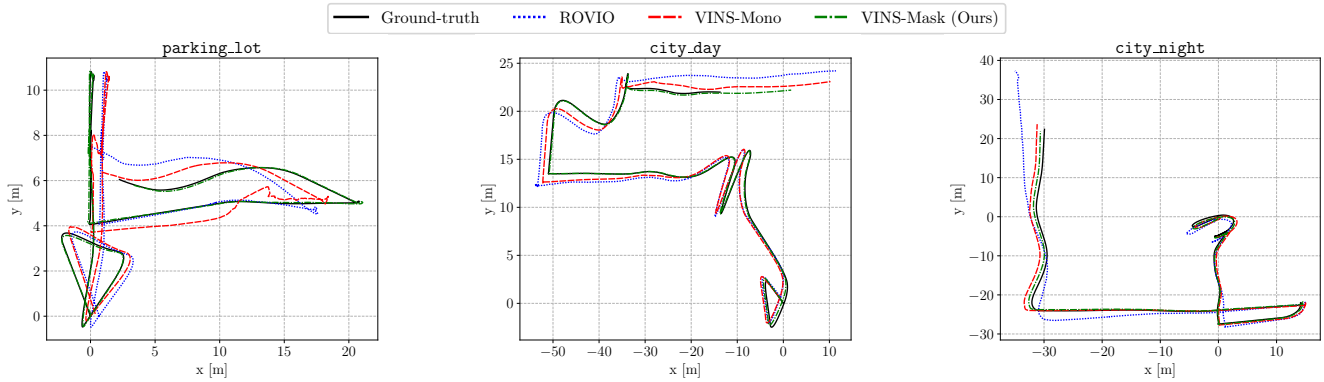

Fig. 7. Estimated trajectory of ROVIO, VINS-Mono, and VINS-Mask with ground-truth for `high` sequences.

V. Conclusion

In this paper, we proposed the VIODE dataset, a novel visual-inertial benchmark that contains variable and measurable dynamic events. Through the systematic introduction of dynamic objects, the users can isolate the effect of dynamic objects on the robustness of VIO. Using the VIODE dataset, we have shown that both VINS-Mono and ROVIO, the two state-of-the-art open-source VIO algorithms, perform worse as the scenes get more dynamic. We also demonstrated that the utilization of semantic information has the potential to overcome this degradation. By masking out the region of the objects, we could mitigate the impact of dynamic objects on the VIO algorithm.

As a future research direction, it is necessary to evaluate the VINS-Mask by using semantic segmentation instead of ground-truth segmentation. One of the challenges would be a real-time onboard deployment, as latency can be critical for the performance of VIO. It is also important to investigate the better utilization of semantic segmentation for VIO. One of the directions would be to distinguish static objects from dynamic ones. Since the current VINS-Mask can mask outs static objects such as parked vehicles, further research is still necessary for robust VIO in challenging dynamic scenes. As a future direction of the VIODE dataset, we are also interested in introducing other types of sensors in the dataset to enable the evaluation of localization algorithms with various sensor configurations. Another possible future work is, similarly to [19], to use real IMU and trajectory data recorded on a real UAV platform instead of simulated ones.

Acknowledgments

This research was partially supported by the Swiss National Science Foundation (SNF) with grant number 200021-155907 and the Swiss National Center of Competence in Research (NCCR).

References

[1] X. Mu, B. He, X. Zhang, T. Yan, X. Chen, and R. Dong, "Visual Navigation Features Selection Algorithm Based on Instance Segmentation in Dynamic Environment," *IEEE Access*, vol. 8, pp. 465–473, 2020.

[2] X. Bai, B. Zhang, W. Wen, L.-T. Hsu, and H. Li, "Perception-aided Visual-Inertial Integrated Positioning in Dynamic Urban Areas," in *IEEE/ION Pos. Loc. Nav. Symp. (PLANS)*, Apr. 2020, pp. 1563–1571.

[3] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," *Field and Service Robotics. Springer Proceedings in Advanced Robotics,*, vol. 5, July 2017.

[4] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Trans. Rob. (TRO)*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. Jour. of Rob. Res. (IJRR)*, vol. 32, no. 11, pp. 1231–1237, Sept. 2013.

[6] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. González-Jiménez, "The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario," *Int. Jour. of Rob. Res. (IJRR)*, vol. 33, no. 2, pp. 207–214, Feb. 2014.

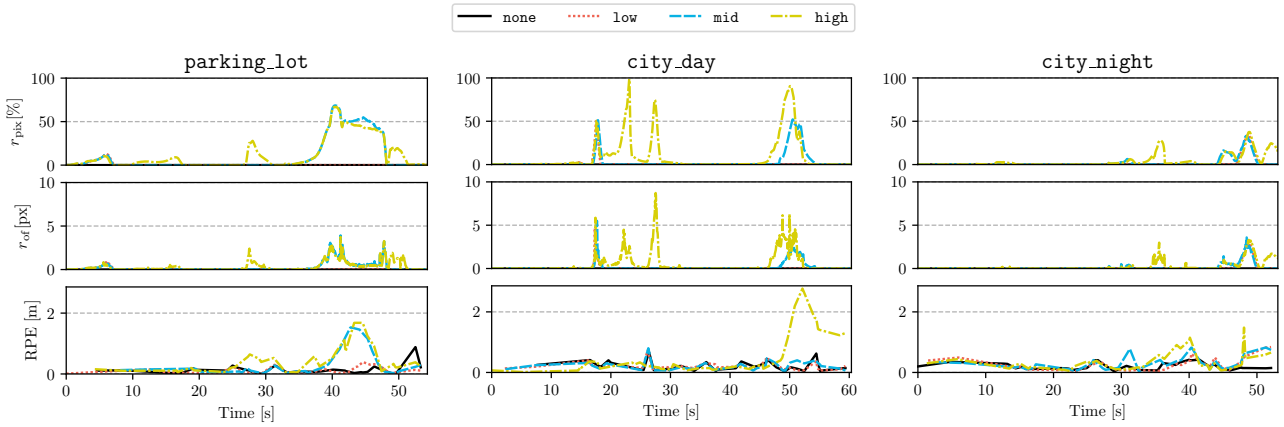

Fig. 8. Comparison of two types of dynamic rate and estimation performance by VINS-Mono along time axis for `none/low/mid/high` sequences in each environment. First row is the pixel-based dynamic rate r_{pix} (see Eq. 1), second row is the OF-based dynamic rate r_{of} (see Eq. 2), and the last row is the RPE (see Eq. 4).

[7] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and lidar dataset," *Int. Jour. of Rob. Res. (IJRR)*, vol. 35, no. 9, pp. 1023–1035, Aug. 2016.

[8] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. Jour. of Rob. Res. (IJRR)*, vol. 35, Jan. 2016.

[9] A. L. Majdik, C. Till, and D. Scaramuzza, "The Zurich urban micro aerial vehicle dataset," *Int. Jour. of Rob. Res. (IJRR)*, vol. 36, no. 3, pp. 269–273, Mar. 2017.

[10] B. Pfrommer, N. Sanket, K. Daniilidis, and J. Cleveland, "PennCOSYVIO: A challenging Visual Inertial Odometry benchmark," in *IEEE Int. Conf. Rob. Aut. (ICRA)*, May 2017, pp. 3847–3854.

[11] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry," in *IEEE Int. Conf. Int. Rob. Sys. (IROS)*, Oct. 2018.

[12] S. Cortés, A. Solin, E. Rahtu, and J. Kannala, "ADVIO: An authentic dataset for visual-inertial odometry," in *Eur. Conf. Comp. Vis. (ECCV)*, July 2018.

[13] A. R. Gaspar, A. Nunes, A. M. Pinto, and A. Matos, "Urban@CRAS dataset: Benchmarking of visual odometry and SLAM techniques," *Robotics and Autonomous Systems*, vol. 109, pp. 59–67, Nov. 2018.

[14] K. M. Judd and J. D. Gammell, "The Oxford Multimotion Dataset: Multiple SE(3) Motions with Ground Truth," *IEEE Rob. Aut. Let. (RA-L)*, vol. 4, no. 2, pp. 800–807, Apr. 2019.

[15] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments:," *Int. Jour. of Rob. Res. (IJRR)*, Apr. 2019.

[16] M. Kasper, S. McGuire, and C. Heckman, "A Benchmark for Visual-Inertial Odometry Systems Employing Onboard Illumination," in *IEEE Int. Conf. Int. Rob. Sys. (IROS)*, Nov. 2019, pp. 5256–5263.

[17] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are We Ready for Autonomous Drone Racing? The UZH-FPV Drone Racing Dataset," in *IEEE Int. Conf. Rob. Aut. (ICRA)*, May 2019, pp. 6713–6719.

[18] D. Zuñiga-Noël, A. Jaenal, R. Gomez-Ojeda, and J. Gonzalez-Jimenez, "The UMA-VI dataset: Visual–inertial odometry in low-textured and dynamic illumination environments," *Int. Jour. of Rob. Res. (IJRR)*, July 2020.

[19] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, and S. Karaman, "The Blackbird UAV dataset," *Int. Jour. of Rob. Res. (IJRR)*, vol. 39, no. 10-11, pp. 1346–1364, Sept. 2020.

[20] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE Int. Conf. Int. Rob. Sys. (IROS)*, Sept. 2015, pp. 298–304.

[21] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *IEEE Int. Conf. Rob. Aut. (ICRA)*, Apr. 2007, pp. 3565–3572.

[22] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. Jour. of Rob. Res. (IJRR)*, vol. 34, no. 3, pp. 314–334, Mar. 2015.

[23] T. Qin, S. Cao, J. Pan, and S. Shen, "A General Optimization-based Framework for Global Pose Estimation with Multiple Sensors," *arXiv:1901.03642 [cs]*, Jan. 2019.

[24] 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 Multi-Map SLAM," *arXiv:2007.11898 [cs]*, July 2020.

[25] J. Delmerico and D. Scaramuzza, "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots," in *IEEE Int. Conf. Rob. Aut. (ICRA)*, May 2018, pp. 2502–2509.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[27] M. Kaneko, K. Iwami, T. Ogawa, T. Yamasaki, and K. Aizawa, "Mask-SLAM: Robust Feature-Based Monocular SLAM by Masking Using Semantic Segmentation," in *IEEE Conf. Com. Vis. Pat. Recog. Workshops (CVPRW)*, June 2018, pp. 371–3718.

[28] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: Tracking, Mapping and Inpainting in Dynamic Scenes," *IEEE Rob. Aut. Let. (RA-L)*, vol. 3, no. 4, pp. 4076–4083, June 2018.

[29] B. Bescos, C. Cadena, and J. Neira, "Empty Cities: A Dynamic-Object-Invariant Space for Visual SLAM," *IEEE Trans. Rob. (TRO)*, Nov. 2020.

[30] B. Bescos, C. Campos, J. D. Tardós, and J. Neira, "DynaSLAM II: Tightly-Coupled Multi-Object Tracking and SLAM," *arXiv:2010.07820 [cs]*, Oct. 2020.

[31] K. Qiu, T. Qin, W. Gao, and S. Shen, "Tracking 3-D Motion of Dynamic Objects Using Monocular Visual-Inertial Sensing," *IEEE Trans. Rob. (TRO)*, vol. 35, no. 4, pp. 799–816, Aug. 2019.

[32] R. Sabzevari and D. Scaramuzza, "Multi-body Motion Estimation from Monocular Vehicle-Mounted Cameras," *IEEE Trans. Rob. (TRO)*, vol. 32, no. 3, pp. 638–651, June 2016.

[33] K. M. Judd, J. D. Gammell, and P. Newman, "Multimotion Visual Odometry (MVO): Simultaneous Estimation of Camera and Third-Party Motions," in *IEEE Int. Conf. Int. Rob. Sys. (IROS)*, Oct. 2018, pp. 3949–3956.

[34] O. J. Woodman, "An introduction to inertial navigation," in *University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR*, Aug. 2007, p. 37.

[35] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE Int. Conf. Int. Rob. Sys. (IROS)*, Oct. 2012, pp. 573–580.