CarlaScenes: A synthetic dataset for odometry in autonomous driving

Andreas Kloukiniotis, Andreas Papandreou
University of Patras
{kloukiniotisandreas, apapandreou}@ece.upatras.gr

Christos Anagnostopoulos, Aris Lalos
I.S.I. - Industrial Systems Institute of Patras
{anagnostopoulos,lalos}@isi.gr

Petros Kapsalas, D.-V. Nguyen
Panasonic Automotive, Langen, Germany
{petros.kapsalas,duongvan.nguyen}@eu.panasonic.com

Konstantinos Moustakas
University of Patras
moustakas@ece.upatras.gr

Abstract

Despite the great scientific effort to capture adequately the complex environments in which autonomous vehicles (AVs) operate there are still use-cases that even SoA methods fail to handle. Specifically in odometry problems, on the one hand, geometric solutions operate with certain assumptions that are often breached in AVs, and on the other hand, deep learning methods do not achieve high accuracy. To contribute to that we present CarlaScenes, a large-scale simulation dataset captured using the CARLA simulator. The dataset is oriented to address the challenging odometry scenarios that cause the current state of art odometers to deviate from their normal operations. Based on a case study of failures presented in experiments we distinguished 7 different sequences of data. CarlaScenes besides providing consistent reference poses, includes data with semantic annotation at the instance level for both image and lidar. The full dataset is available at https://github.com/CarlaScenes/CarlaSence.git .

1. Introduction

The research field of visual odometry (VO) and simultaneous localization and mapping (SLAM) has met immense evolutions [14, 21, 30, 36], especially in the context of autonomous driving (AD) [23, 24, 28, 32, 33]. A pivotal factor for this advancement has been the publication of large-scale datasets [5, 10, 16, 27] oriented to AD. Though despite the research activity and the effort dedicated to odometry algorithms they fail to cover the wide range of use cases being present in automotive scenarios and handle dynamically changing and challenging conditions. Extensive benchmarks [10] may present low errors in pose estimation though there are specific scenarios that cause even the existing leading methods to deviate from the ground truth pose. For instance, scenarios with featureless regions such as the sky or uniform ground, dynamic objects populating most of the region of expansion of the sensors, road surfaces with a big positive or negative slope, e.t.c, can cause major drifts and wrong poses. The extensive evaluation of the case study of failure, as presented in the experimental section and Table 2, was the guideline for separating the dedicated sequences to the categories that are presented in Table 1.

The main contributions of the paper are summarized epigrammatically as follows:

1. Generate a benchmark dataset dedicated for odometry methods in autonomous driving, including annotations for other perception tasks
2. Showcase vulnerabilities of state-of-the-art odometry algorithms in dedicated scenarios and present their behavior in the published dataset.
3. Discuss potential future directions to improve localization methods.

2. Related Datasets

The most popular benchmarks related to autonomous driving developed during the last decade [1–5, 9–11, 13, 15–17, 20, 22, 27, 34, 35] are summarized in Table 1 and an extensive description in chronological order is given below.

Kitti [10] was the pioneering dataset for applications related to autonomous driving systems by A. Geiger et al in

Table 1. Summary of various autonomous driving datasets.

Dataset	Sensors	Metadata	Weather	Hours	Environments
Kitti (2011) [10], **Semantic Kitti - (2019)** [1, 2]	1 LiDAR: 64 channels, 4 Perspective cameras, 4 Optics lenses, GPS, IMU	3D BBox, Semantic and Instance segmentation, Lane marking, Multi-object tracking, Visual Odometry/SLAM	sunny, cloudy	day	urban, highway, rural
Kitti-360 (2021) [17]	1 LiDAR: 64 channels, 1 SICK LMS 200, 2 Perspective cameras, 2 Fisheye cameras, GPS,IMU	3D BBox, 2D/3D Semantic annotations, 2D/3D Instance annotations	-	-	suburban
Cityscapes (2016) [5]	2 Cameras, GPS	Semantic segmentation, Outside temperature, Vehicle odometry		day	urban
nuScenes (2018) [3]	1 LiDAR: 32 channels ,5 RADAR, 6 Cameras, GPS, IMU	3D BBox, HD maps	sunny, cloudy, rainy	day, night	urban, residential, nature, industrial
WoodScape (2021) [34]	1 LiDAR: 64 channels, 4 Fisheye cameras, GNSS, IMU, GNSS Positioning with SPS	2D/3D BBox, Semantic and Instance segmentation, Motion segmentation, Soiling detection, Depth estimation, Odometry/SLAM, End-to-end driving	-	-	urban, highway, parking
A2D2 (2019) [11]	5 LiDAR: 16 channels, 6 Cameras, GPS, IMU, steering angle, brake, throttle, odometry, velocity, pitch, roll	Semantic segmentation, Point cloud segmentation, 3D BBox	sunny, cloudy, rainy	day	urban, highway, country road
Argoverse (2019) [4]	2 LiDAR: 32 channels, 7 ring Cameras, 2 stereo Cameras, GPS	3D track annotations, Motion forecasting, Stereo depth, HD maps	multiple conditions	day, night	urban
BDD100K (2018) [35]	1 Camera, GPS, IMU	2D BBox, Semantic and Instance segmentation, Lane marking, Multiple object tracking	sunny, cloudy, rainy	day, night	city, residential, highway, parking lot, tunnel
ApolloScape (2018) [15]	2 LiDAR: 64 channels, 6 Cameras, GNSS, IMU	3D BBox, Semantic segmentation, Lane marking	multiple, conditions	day	urban
Mapillary (2017) [22]	-	Semantic and Instance segmentation	multiple conditions	day, night	urban, countryside, off-road
Waymo (2019) [9, 27]	5 LiDAR: 64 channels, 5 Cameras	2D/3D Tracking IDs, 2D/3D BBox, HD Maps, rainy	sunny, cloudy	day, night	suburban, downtown
Lyft (2019) [13, 16]	3 LiDAR: 2x40, 1x64 channels, 7 Cameras, 5 Radars	3D BBox ,HD Maps, Trajectories	multiple conditions	day	urban
4Seasons (2020) [31]	1 stereo Camera, GNSS	Trajectories	multiple conditions	multiple conditions	garage, highway, urban, tunnels, countryside
ONCE (2021) [20]	1 LiDAR: 40 channels, 7 Cameras	2D/3D BBox	sunny, rainy	multiple conditions	downtown, highway, suburban, tunnel, bridge
CarlaScenes (2021)	2 LiDAR: 1x16, 1x64 channels, 1 Camera, GPS, IMU	Semantic segmentation, Point cloud segmentation, Depth Estimation, Odometry/SLAM, Lane marking	sunny, cloudy ,rainy, wet	day, noon, sunset, night	urban, tunnel, slopes, highway, complex scenes, infinite loop

2011. They provided valuable information for challenging computer vision tasks, such as object detection and tracking, semantic and instance segmentation, visual odometry, etc. To accomplish that, they utilized a Velodyne lidar scanner, a GPS/IMU, and two high-resolution color and grayscale video cameras. They collected data from rural areas and highways during the daytime. Even though Kitti was a very important tool for the research community, it could not capture the complexity of real-world scenes. Hence five years later, M. Cordts et al built the Cityscapes [5] dataset, which focuses on semantic understanding of urban areas. The dataset consists of 5000 annotated images with fine annotations and 20.000 annotated images with coarse annotations, capturing 50 different cities during the daytime. Aiming to capture the real-world complexity, the authors provided a huge variety of annotations, like road, person, car, building, ground, and some more. They provided some metadata for other tasks as well, such as ego-motion data from vehicle odometry. In 2017 G. Neuhold et al generated the Mapillary Vistas [22], which is five times larger than Cityscapes. The authors provided 25.000 high-resolution images annotated by 66 object classes, to be utilized for the tasks of semantic and instance segmentation. They have been captured using different weather and viewpoint conditions. In addition, some other datasets have been generated for other computer vision tasks [8, 18, 29]. Moreover, in 2018, H. Caesar et al aimed to create a dataset extracting information from a variety of sensors along with images. As a result, they provided the nuScenes [3] dataset combining lidar, cameras, and radars. It contains 3D bounding boxes for 23 classes and has seven times more annotations and one hundred times more images than the Kitti dataset. To tackle the issue of real-world complexity, Fisher Yu et al generated BDD100K [35] dataset. It is a diverse driving dataset for heterogeneous multitask learning with 100k images. It also provides data from multiple weather conditions, allowing deep learning models to be trained properly. X. Huang et al generated the ApolloScape [15] dataset which contains much richer information from the previous ones. It contains 100k images, 80k lidar samples, and 1000km trajectories from multiple cities, under various conditions. Furthermore, J. Behley et al noticed that there is a lack of a dataset aiming to provide 3D scene understanding from point clouds. Hence, they built the Semantic Kitti [1, 2] dataset, by annotating all the sequences of the Kitti benchmark. To support startups and academic researchers, J. Geyer et al generated the A2D2 [11] dataset providing 40.000 frames with semantic segmentation image and point cloud labels and 3D bounding boxes annotations. Overall, the majority of the previous authors did not take into consideration the influence of HD maps for tracking and motion prediction in applications related to AVs. However, M. Chang et al provided the Argoverse [4] dataset including HD maps with semantic metadata. They helped the community to provide more robust perception mechanisms. Motivated by the contribution of the large datasets on deep learning systems, J. Houston and R. Kesten et al provided the Lyft [13] dataset. It consists of a dataset for perception and prediction, including over 1000 hours of movement of traffic agents alongside their 3D bounding boxes. Following the previous structure, P. Sun and S. Ettinger et al generated the Waymo dataset [9, 27] providing multiple 2D/3D labels, object trajectories, and 3D maps. Another remarkable work is Kitti-360 [17] dataset by Y. Liao et al. It is a successor of the Kitti dataset and is comprised of 300k images and point clouds with 2D/3D semantics of a suburban area. Moreover, to adapt deep learning algorithms for the fisheye camera, S Yogamani et al generated WoodScape [34] dataset. As metadata, it provides 2D/3D bounding boxes, semantic segmentation, soiling detection, odometry data, and some more. Finally, J. Mao et al provided the ONCE [20] dataset which is twenty times longer than the nuScenes or Waymo. It is composed of 1 million point clouds and 7 million images, capturing data for 144 hours.

The work closer to ours is the one presented in [31] by Wenzel et al. and focuses on covering seasonal and demanding perceptual conditions for AD oriented for visual odometry tasks. They provide multiple traverses of the same path covering it that way a large variation caused by weather or the changes in the scene. They also provide nine different environments ranging from multi-level parking garages over urban (including tunnels) to countryside and highways. The main weak point is that GNSS-denied environments, e.g. garages, tunnels, or urban canyons, can not guarantee a high accuracy of the reference poses. The sensor system consists of a stereo image sensor GNSS receiver. GNSS-denied environments are absent in a simulated environment which is the case of CarlaScenes. Consequently having the data annotated without any errors added will allow us to know the exact uncertainty of the methods.

3. Overview of CarlaScenes Dataset

Even though there is a variety of datasets related to autonomous driving, not many of them focus on the odometry problem. Our goal is to provide a synthetic dataset extracted from the CARLA [6] simulator dedicated specifically to odometry, global place recognition, and relocalization tasks. Indicative plots of the trajectories are presented in Figure 1.

3.1. Scenarios Dedicated for Odometry

There is a huge variety of odometry and mapping techniques, aiming at achieving low-drift in motion estimation. However, the majority of them provide robust results only on a few use-cases with specific sensor configurations. Hence, our goal is to showcase the vulnerabilities of these

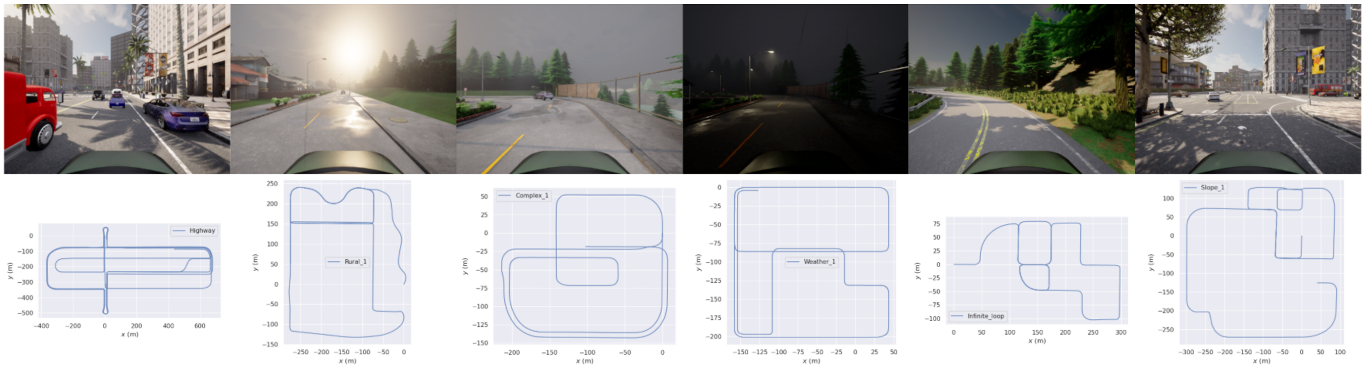

Figure 1. Indicative paths that are included in the dataset alongside images of the respective.

approaches and gather all the challenging scenarios in one dataset. A brief description of these scenarios is given in Table 2. More details are given below:

Roads with positive or negative slope The most complex town, with a 5-lane junction, unevenness, a tunnel, and more have been evaluated in this section. The SoA odometry approaches in extreme conditions such as tunnels or roads with a positive or negative slope will be examined here.

Rural environment: A rural environment with narrow roads, barns and hardly any traffic lights has been evaluated here. This scenario investigates how SoA approaches can handle an environment without many buildings or other static features like poles, traffic lights, etc. However, only a few clear objects are available for landmarks, especially in real cases. One of the challenges of the odometer here is to detect sufficiently discriminative features.

Sequence with an "infinite" loop: A dedicated scenario with an infinite loop, a looping trajectory that is traversed multiple times, has been generated here to examine loop closure, which is very important for global mapping. More specifically, the loop closure is related to the drift correction of ego-vehicle based on the recognition of a previously visited place.

Modify weather and lighting conditions: A basic town layout consisting of "T junctions" and multiple weather conditions has been evaluated here. The weather and lighting conditions can be chosen from a set of predefined settings, which are shown in Table 2. These weather conditions can model seasonal changes. Hence, this scenario investigates whether SoA odometry and slam methods are capable of handling dynamic environments and extracting robust features against those changes. Two cases with dynamic weather conditions have been generated here and they differ in the point cloud generation. The first one consists of original point clouds. In the second case, points in the cloud have been dropped off in order to simulate noise due to external perturbations. Abrupt changes in weather conditions are included for the methods to be tested in extreme scenarios.

Existence of moving objects on the 50% focus expansion of cameras: Data from a basic town with multiple vehicles that cover the 50% focus expansion of cameras have been generated for this scenario. The scope of this experiment is to examine the odometers in areas with many dynamic objects in the scene. For instance, some algorithms may fail to detect whether the front vehicle is moving or not.

Complex city environment: A city environment with different environments such as an avenue or promenade and more realistic textures have been evaluated here. The experiments, in this case, will examine whether odometry and slam approaches can extract robust landmarks in a map of superb visual quality, with detailed buildings and realistic roads. The map used for this scenario is shown in Figure 2.

Long highways: An environment with long highways with many entrances, exits and roundabouts has been assessed in this scenario. The extracted landmarks in this case have larger shifts due to the high speed of the vehicle and sometimes move out of the field of view. Other landmarks may exist at high distances and as a result, their shifts in the image plane are noisy. Hence, the SoA approaches will be examined whether they can track the extracted landmarks and provide robust results. The map used for this scenario is shown in Figure 2.

3.2. Sensor Setup

CarlaScenes consists of data coming from multiple environments with different conditions. In general, the Carla [6] simulator is initialized with several pre-defined settings. For instance, the environment of the map or the number of actors, the weather conditions should be defined before each simulation. Other options are available for the user as well. Detail on the parameters used for the dataset generation can be found in the released repository. The ego vehicle is set to travel around the city with some basic configuration, and

Table 2. Generated scenarios for odometry evaluation

Scenario: Case Study for Failures
Roads with positive or negative slope : Algorithms with planarity assumption may fail to detect the road surface.
Rural environment without many buildings and narrow roads: Check if robust features could be extracted, because rural regions especially in an image are featureless and thus there are far fewer feature points.
Sequence with an infinite loop: Check loop closure detection accuracy.
Modify weather and lighting conditions: Check whether multiple weather conditions in images or scattering in points clouds affect trajectory. The weather conditions that can be chosen are ClearNoon, CloudyNoon, WetNoon, WetCloudyNoon, MidRainyNoon, HardRainNoon, SoftRainNoon, ClearSunset, CloudySunset, WetSunset, WetCloudySunset, MidRainSunset, HardRainSunset and SoftRainSunset.
Existence of moving objects on the 50%, focus expansion of cameras: The odometry algorithm may fail to recognize whether the front vehicle is moving or not.
Complex environment in the city: Check trajectory error in a complex city environment with traffic elements such as multiple intersections, complex lane roundabouts, or tunnels.
Long highways: Check whether the high speed of the vehicle could affect the estimated odometry. Also, some 3d landmarks at long distances may be detected that have noisy shifts on the image plane and affect negatively the accuracy of the algorithm.

data from all sensors are gathered and stored in each fame. The sensors that their recordings are saved for this dataset are shown in Table 3. All of them use the Unreal Engine coordinate system (x-forward, y-right, z-up) and return coordinates in local space. Also, intrinsic and extrinsic matrices are provided as well as timestamp files to allow synchronization of the data.

Alongside the released Dataset, we provide in the source code XML files that are compatible with Carla ScenarioRunner [6]. ScenarioRunner is a module that allows traffic scenario definition and execution for the CARLA simulator. It gives the capability to run multiple times the same scenario in a CARLA environment but with different conditions about the weather, the actors (cars, pedestrians), and change multiple other parameters of the simulation. This is an important capability because it allows odometry methods to run multiple times the same path but with different illuminations or occlusion conditions.

Table 3. Sensors configuration

Type	**Dimensions**	**Description**
RGB camera	1280x960	Get images from the scene
Semantic segmentation camera	1280x960	Every object is classified in a different color according to its tags
Depth camera	1280x960	Get depth values
Lidar	Velodyne 16/64	Get the 3d coordinates and intensity values
Semantic Lidar	Velodyne 16/64	Get the index of the Carla object hit and its semantic tag
Imu	-	Provides measurements from accelerometer, gyroscope and compass
Gnss	-	Provides the current gnss position

3.2.1 Data Description

Overall we store data with frequencies of 30 fps. The specific values that are used to parametrize the sensor are provided in the published repository. Data are saved as raw files, using the format of .png for images and .ply for Lidar data, we also provide .bag files with the data. A bag is a file format in ROS for storing ROS message data. Consequently, they could be easily used for testing techniques that have been implemented using ROS [26] framework.

3.2.2 Data Annotations

A huge bottleneck in the generation process of a real-world dataset is the annotation of the captured data. Labeling the data demand a lot of manpower and even using advanced annotation tools still lack precision. Consequently, the uncertainty of the methods trained or tested on the data is not negligible. The advantage of using a simulator for data generation is that it can provide data annotation for every object in the scene. Consequently, landmark tracking and evaluation can be performed with greater accuracy. For instance, annotation of lane marks which are a significant feature is provided both in data from camera and lidar. In detail annotation is provided for lidar data (instance and class id for 23 classes), camera data (23 different class ids), depth annotation for every pixel in image data. Also, GNSS and IMU measurements noise parameters have been set to zero and can be used as ground truth..

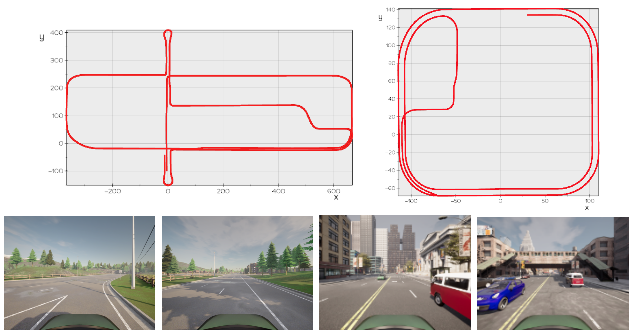

Figure 2. From left to right: maps generated in long highway and complex city environment.

4. Experiments

4.1. Monocular visual odometry

In this section, we will present the evaluation of different methodologies applied in automotive visual odometry and discuss the influence of various conditions. More specifically we will focus on evaluating monocular DSO [7], LEGO LOAM [25] and DVSO [33]. In short Direct Sparse Odometry (DSO) [7] is a visual odometry method based on sparse and direct structure and motion formulation. DSO tries to minimize the photometric error. LEGO LOAM [25] is a lidar odometry and mapping method using raw point clouds. It applies feature extraction to obtain distinctive planar and edge features to solve the 6doF transformation across successive point clouds. The last method that we included in our evaluation was the methods proposed in the paper of Yang et al. (DVSO [33]). DVSO incorporates deep depth predictions into the pipeline of DSO as direct virtual stereo measurements.

4.2. Evaluation Metrics

The absolute pose error (APE), also called absolute trajectory error (ATE), is a metric for analyzing the global consistency of SLAM systems. The APE metric calculates the difference between the ground truth poses and the estimated poses. This can be expressed as:

$$E = P_{est,i} \ominus P_{ref,i} = P_{ref,i}^{-1} P_{est,i}^{-1} \in SE(3) \qquad (1)$$

where $\ominus$ is the inverse compositional operator, which takes two poses and gives the relative pose [19]. $P_{ref,i}$ and $P_{est,i}$ is ground truth and estimated 6-DoF Pose respectively. Different pose relations can be used to calculate the APE. The RMSE value was used of the full relative pose of E_i and APE_i is calculated form $||E_i - I_{3\times3})||$, which is unitless. RMSE is calculated from:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} APE_i^2} \qquad (2)$$

The visualization tool that was used is the open source library EVO [12].

4.3. Results

DSO and LEGO when tested on normal conditions on the Carla simulator perform with relatively small deviations from the ground truth trajectory. Nevertheless, when tested on the published dataset the resulting RMSE errors were high and prohibitive for automotive applications. For a trajectory with uneven ground, LEGO and DSO had 15.13 and 35.24 RMSE values. The trajectories are shown in Figure 5. Most of the SLAM methods use the assumption that the ground is flat, so abrupt elevations in the road result in a decreased accuracy of the methods.

Additionally, tests on multi-weather scenarios showed that DSO could not provide correct outputs and the initialization of the method was failing. DSO relay of the photometric consistency assumption which is breached when abrupt weather changes occur. On the other hand, LEGO LOAM operates solely with raw point clouds which are not affected to the same extent by weather conditions as image-based methods. Consequently, as it is shown in Figure 3 we can see the output of the LEGO LOAM in the multi-weather scenario which is quite well.

Both LEGO and DSO fail in the scenarios with scenes from the highway and rural conditions where the featureless environment makes it difficult to calculate the correct trajectory. More specifically both of the methods as a first processing step find features on the image or the point cloud respectively. These features will be then used for calculating the displacement of the camera/ lidar by matching them between consecutive frames. When the geometry of the scene is simple and most of the reading of the sensors are areas with identical texture (e.g long roads in highways, rural environments) or shape finding unique and distinctive features to perform matching is rather difficult. For an accurate localization, estimated features should be uniformly distributed in the processed frames.

Following the same pipeline, in order not to introduce the uncertainty of a deep learning module, the ground truth depth image from the dataset was used to see the upper limiting of the accuracy of the proposed method in CarlaSense.

In the experiment with DVSO, we concluded that the DVSO was more robust in losing the scaling when compared with DSO. DSO failed in most of the scenarios to

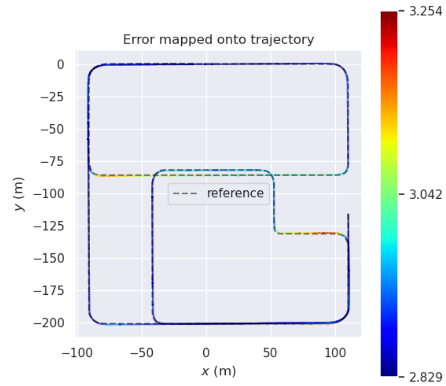

Figure 3. Error mapped onto trajectory. The tested method is LEGO LOAM and the trajectory is from a multi-weather scenario.

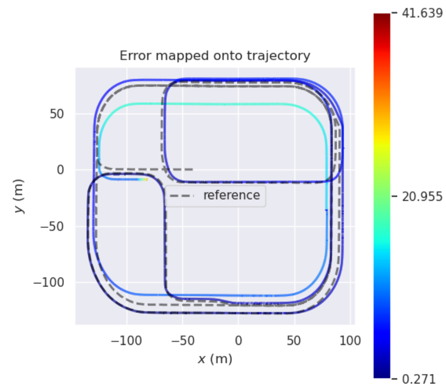

Figure 4. Error mapped onto trajectory. The tested method is DVSO and the trajectory is from a scenario captured within a complex city environment.

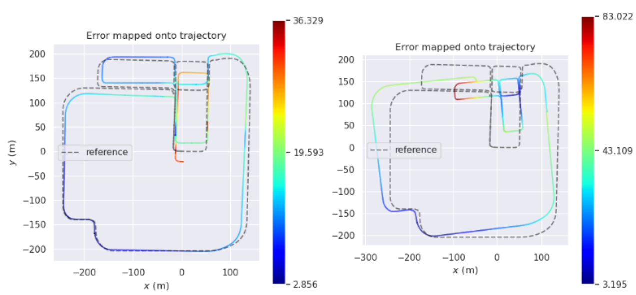

Figure 5. Error mapped onto trajectory. The tested methods are LEGO and DSO from left to right respectively and the trajectory is from a scenario captured within an environment with uneven ground.

compute the trajectory with a correct scale and when abrupt changes were happening due to the weather of moving objects it failed to initialize. DVSO was more robust though the drift was still high for automotive applications. Some indicative results are shown in Figure 4 and the APE error for DVSO was 8.26 while for DSO was 94.66.

4.4. Discussion

First, an important factor for a reliable odometry system is its generalization ability in unseen situations. The uncertainty of the deep learning-based methods can be decreased by training and validating in a multi-scene environment consequently the amounts of available datasets should be adequate. The generalization ability of a model is of crucial importance so that the behavior of the perception engine in an autonomous vehicle remains unaffected by the dynamic environment.

Second geometry based methods are, in contrast to deep learning which are used as a black box, straightforward and well understood. Though geometric methods usually fail to initialize and lose track since they are not robust to abrupt changes and dynamic scenes. So testing them in dynamic environments to adjust their performance and test their accuracy is mandatory. Consequently, the existence of datasets that violate classic assumptions, such that the photoconsistency of the scene or planarity of the ground, is important.

Third, a bottleneck for the development of deep learning slam methodologies is the lack of annotated data. A challenging part of releasing a real-world dataset is to find a way to get accurate annotations, which undoubtedly is a resource-demanding task. Though understanding the semantic information of a scene is the most meaningful step to obtaining a high level of perception. Employing semantic information and object detection as constraints to localization tasks could improve the accuracy and robustness and allow the AV to infer the surrounding environment. Adding to that landmarks used for visual odometry are something that can not be annotated manually. They can only be defined theoretically and during the execution of the odometry algorithm. Landmarks can not be annotated manually. So providing annotation for every pixel, or lidar point would help the investigation of the properties of selected landmarks if the class they belong to is known.

5. Conclusions

In this paper, we present a simulated dataset oriented to odometry tasks for automotive applications. By giving access to specific challenging scenarios for odometry we aim to enhance the effort of researchers in the field and help to handle open issues such as drift, overfitting to datasets, poor generalization in multiple conditions. As future work, we plan to integrate the whole dataset in XML files that operates with Carla scenario runner. So by running the Carla simulator data of predefined trajectories will be produced locally without the need for a dataset that requires huge storage resources.

Acknowledgment

This work was supported by two European Union's Horizon 2020 research and innovation programs. CARAMEL which is under grant agreement No.833611 and CPSoSaware which is under Grant Agreement No. 871738.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, C. Stachniss, and Juergen Gall. A dataset for semantic segmentation of point cloud sequences. *ArXiv*, abs/1904.01416, 2019. 1, 2, 3

[2] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019. 1, 2, 3

[3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2020. 1, 2, 3

[4] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Sławomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8740–8749, 2019. 1, 2, 3

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. 1, 2, 3

[6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 3, 4, 5

[7] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, mar 2018. 6

[8] Christian Ertler, Jerneja Mislej, Tobias Ollmann, Lorenzo Porzi, Gerhard Neuhold, and Yubin Kuang. The mapillary traffic sign dataset for detection and classification on a global scale. In *ECCV*, 2020. 3

[9] Scott M. Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Han Zhao, Sabeek Pradhan, Yuning Chai, Benjamin Sapp, C. Qi, Yin Zhou, Zoey Yang, Aurelien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Drago Anguelov. Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset. *ArXiv*, abs/2104.10133, 2021. 1, 2, 3

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 1, 2

[11] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Ganapati Mirashi, Chiragkumar Savani, M. Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2d2: Audi autonomous driving dataset. *ArXiv*, abs/2004.06320, 2020. 1, 2, 3

[12] Michael Grupp. evo: Python package for the evaluation of odometry and slam. `https://github.com/MichaelGrupp/evo`, 2017. 6

[13] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska. One thousand and one hours: Self-driving motion prediction dataset. `https://level-5.global/level5/data/`, 2020. 1, 2, 3

[14] Baichuan Huang, Jun Zhao, and Jingbin Liu. A survey of simultaneous localization and mapping. *CoRR*, abs/1909.05214, 2019. 1

[15] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. *arXiv: 1803.06184*, 2018. 1, 2, 3

[16] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Level 5 perception dataset 2020. `https://level-5.global/level5/data/`, 2019. 1, 2

[17] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *ArXiv*, abs/2109.13410, 2021. 1, 2, 3

[18] Manuel López-Antequera, Pau Gargallo, Markus Hofinger, Samuel Rota Bulò, Yubin Kuang, and Peter Kontschieder. Mapillary planet-scale depth dataset. In *ECCV*, 2020. 3

[19] Feng Lu and Evangelos E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997. 6

[20] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Xiaodan Liang, Yamin Li, Chao Ye, Wei Zhang, Zhenguo Li, Jie Yu, Hang Xu, and Chunjing Xu. One million scenes for autonomous driving: Once dataset. *ArXiv*, abs/2106.11037, 2021. 1, 2, 3

[21] Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *CoRR*, abs/1610.06475, 2016. 1

[22] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5000–5009, 2017. 1, 2, 3

[23] Nikos Piperigkos, Aris S. Lalos, and Kostas Berberidis. Graph laplacian diffusion localization of connected and automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–15, 2021. 1

[24] Nikos Piperigkos, Aris S. Lalos, and Kostas Berberidis. Multi-modal cooperative awareness of connected and automated vehicles in smart cities. In *2021 IEEE International*

Conference on Smart Internet of Things (SmartIoT), pages 377–382, 2021. 1

[25] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, 2018. 6

[26] Stanford Artificial Intelligence Laboratory et al. Robotic operating system. 5

[27] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott M. Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2443–2451, 2020. 1, 2, 3

[28] Ke Wang, Sai Ma, Junlan Chen, and Jianbo Lu. Approaches challenges and applications for deep visual odometry toward to complicated and emerging areas. *IEEE Transactions on Cognitive and Developmental Systems*, 14:1–15, 2020. 1

[29] Frederik Warburg, Søren Hauberg, Manuel López-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2623–2632, 2020. 3

[30] Wenle Wei, Linin Tan, Guodong Jin, Libin Lu, and Changjiang Sun. A survey of uav visual navigation based on monocular slam. In *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 1849–1853, 2018. 1

[31] Patrick Wenzel, Rui Wang, Nan Yang, Qing Cheng, Qadeer Khan, Lukas von Stumberg, Niclas Zeller, and Daniel Cremers. 4seasons: A cross-season dataset for multi-weather SLAM in autonomous driving. *CoRR*, abs/2009.06364, 2020. 2, 3

[32] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1278–1289, 3 2020. 1

[33] Nan Yang, Rui Wang, Jörg Stückler, and Daniel Cremers. Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11212 LNCS:835–852, jul 2018. 1, 6

[34] Senthil Kumar Yogamani, Ciarán Hughes, Jonathan Horgan, Ganesh Sistu, Padraig Varley, Derek O'Dea, Michal Uřiář, Stefan Milz, Martin Simon, Karl Amende, Christian Witt, Hazem Rashed, Sumanth Chennupati, Sanjaya Nayak, Saquib Mansoor, Xavier Perroton, and Patrick Pérez. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9307–9317, 2019. 1, 2, 3

[35] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2633–2642, 2020. 1, 2, 3

[36] Shishun Zhang, Longyu Zheng, and Wenbing Tao. Survey and evaluation of rgb-d slam. *IEEE Access*, 9:21367–21387, 2021. 1