SceNDD: A Scenario-based Naturalistic Driving Dataset

Avinash Prabu, Nitya Ranjan, Lingxi Li*, Renran Tian, Stanley Chien, Yaobin Chen, and Rini Sherony

Abstract—In this paper, we propose SceNDD: a scenario-based naturalistic driving dataset that is built upon data collected from an instrumented vehicle in downtown Indianapolis. The data collection was completed in 68 driving sessions with different drivers, where each session lasted about 20–40 minutes. The main goal of creating this dataset is to provide the research community with real driving scenarios that have diverse trajectories and driving behaviors. The dataset contains ego-vehicle’s waypoints, velocity, yaw angle, as well as non-ego actor’s waypoints, velocity, yaw angle, entry-time, and exit-time. Certain flexibility is provided to users so that actors, sensors, lanes, roads, and obstacles can be added to the existing scenarios. We used a Joint Probabilistic Data Association (JPDA) tracker to detect non-ego vehicles on the road. We present some preliminary results of the proposed dataset and a few applications associated with it. The complete dataset is expected to be released by early 2023.

I. INTRODUCTION

The development of connected and automated vehicles (CAVs) technology has been growing at a fast pace in the last decade [1], [2]. Motion planning is one of the most important aspects for CAVs. Thus, there is an urgent need for developing efficient algorithms as well as their verification and validation methods using benchmark datasets. High-quality datasets enable researchers to virtually access, visualize, simulate, and analyze real-life urban driving scenarios, which greatly save their time and energy. In this paper, we develop SceNDD: a scenario-based naturalistic driving dataset that is built upon data collected from an instrumented vehicle in downtown Indianapolis. The data collection was completed in 68 driving sessions with different drivers, where each session lasts 20–40 minutes. The main goal of creating this dataset is to provide the research community with real driving scenarios that have diverse driving trajectories and behaviors to develop efficient motion planning and path following algorithms.

A multi-model dataset typically needs a combination of sensors for capturing data. Cameras are used for recording images/videos and the features of interest can be extracted and analyzed using image processing or computer vision algorithms. LiDAR data are used for 3D localization. Inertial measurement unit (IMU) and GPS devices are also popular to get accurate position, trajectory, velocity, and acceleration of the ego-vehicle. In literature, datasets such as Waymo Open Dataset [3], nuScenes [4], LYFT [5], KITTI [6], Argoverse [7], and ApolloScape [8] have been created using a combination of sensors mentioned above. They are popular benchmark datasets for the research community to use and develop efficient and effective algorithms for CAVs.

The nuScenes [4] dataset is one of the most commonly used benchmark datasets for CAVs. It includes around 6 hours of driving data captured by front-facing stereo cameras, LiDAR, and GPS/IMU sensors. The LYFT [5] and KITTI [6] datasets are similar, with difference in the perception environments and data sizes. These datasets are excellent for developing perception algorithms. However, they do not focus on capturing motion features such as driving scenarios and driving behavior.

Recently, multiple motion datasets have been proposed, which include various driving scenes, tracked objects, moving trajectories, and time segments. A notable dataset is the Waymo Open dataset [3], which includes motion prediction, interaction prediction, and occupancy/flow prediction. It is composed of 574 hours of naturalistic driving data, collected in different major cities in the U.S. Argoverse [7] is also a dataset that focuses on certain driving behaviors, such as turning at an intersection and changing lanes. The Interaction dataset [9] is collected via drone footage and contains naturalistic motion of various traffic agents in multiple driving scenarios in various countries, which has diverse and complex behaviors of vehicles at intersections and roundabouts. However, this dataset does not have 3D state estimates that our dataset includes. The ApolloScape dataset [8] also includes trajectories of different traffic agents but does not have trajectory of the ego vehicle. Datasets such as the Stanford Drone Dataset [10], NGSIM [11], ETH [12], UCY [13], Town Center [14] are popular in motion forecasting but do not focus on the driving environment and are much smaller in size compared to our dataset.

Our dataset includes naturalistic driving data of 68 driving sessions with different drivers, where each session lasts 20–40 minutes. One unique feature is that it provides flexibility for users to add/remove targets in the simulated environment and customize the environment to their needs. The development is based on the Matlab, which allows the addition of different sensors (e.g., radar, LiDAR, ultrasonic sensors, etc.) to the ego vehicle, so that it is adaptive to different
TABLE I: Comparison of various autonomous driving datasets

| Name             | Task                  | Focus         | Environment                  |
|------------------|-----------------------|---------------|------------------------------|
| nuscenes         | Perception, Prediction| Varied        | Urban Streets                |
| Waymo Open Dataset| Perception, Prediction| Varied        | Varied                       |
| KITTI            | Perception            | Varied        | Sidewalk                     |
| Argoverse Forecasting | Prediction          | Varied        | Urban Streets                |
| AppolloScape Trajectory | Prediction     | Varied        | Urban Streets                |
| The Interaction Dataset | Prediction  | Pedestrians   | Pedestrian only streets, Pedestrian Crossing |
| Stanford Drone Dataset | Perception    | Pedestrians   | Highway                      |
| NGSIM            | Prediction            | Pedestrians   | Pedestrian Crossing, Sidewalk |
| ETH              | Prediction            | Pedestrians   | Pedestrian Crossing, Sidewalk |
| UCY              | Prediction            | Pedestrians   | Pedestrian only Streets      |
| Town Center      | Prediction            | Vehicles      | Varied                       |

configurations of the ego-vehicle. Another attribute of our dataset is the time horizon of prediction. Since all the data is broken down into scenarios, the data can be easily broken down into timestamps of different history and horizon time. The comparison of various autonomous driving dataset is shown in Table I.

To the best of our knowledge, only a few existing datasets provide scenarios with flexibility in diverse real-time driving. In this paper, we propose creating a dataset that involves a diverse mix of driving data collected using different drivers around downtown Indianapolis. The main contributions of this paper are summarized as follows:

- Create a comprehensive dataset that has 68 driving sessions around downtown Indianapolis, with each session being 20-40 minutes.
- Process and fuse data from different sensors. In this paper, since only GPS and the LiDAR data are used for processing, a simple time synchronization of the sensors is sufficient. In the future versions of the dataset, a camera-LiDAR sensor fusion will be adopted.
- Develop a pipeline to generate driving scenarios from recorded vehicle data.

II. DATA ACQUISITION SYSTEM

In this section, we briefly describe the development of vehicle-based data acquisition system (DAS), which includes six cameras to cover all 360-degree angles, one 64-beam 360-degree Ouster LiDAR, a Reach Emlid GPS, and a desktop computer. The instrumented vehicle is shown in Fig. 1.

A. Cameras

The main features considered while selecting a camera were resolution, frames per second (FPS), shutter type, connector type, color coding, external hardware synchronization, data transfer speed, compatibility with ROS operating system, and cost. FLIR Grasshopper-3 cameras are selected, which are capable of providing a resolution of 2048 × 2048, 1 to 90 FPS, and programmable exposure and shutter speed. The camera also has a global shutter, which reduces blur due to fast motion. The camera has a CMOS sensor to facilitate noise reduction and has a GPIO port capable of bi-directional synchronization with the other sensors. We used two types of lenses, 95° and 43°. The wider-angle lens was used to cover the sides of the vehicle, and the narrower-angle lens was used to cover the front and back. The camera used is shown in Fig. 2.

B. LiDAR

The Ouster OS-1 64-beam medium-range LiDAR was used in our data collection. The OS-1 can generate 1,310,720 points at a frequency of 10 Hz, has a horizontal field of 360 degrees, and a 45 degrees vertical field of view. The LiDAR has a maximum distance range of 120m with an accuracy of +/- 5cm for Lambertian targets. OS-1 LiDAR also comes with a sensor interface that allows for transmitting data over Ethernet using UDP. The sensor interface also has a multi-purpose I/O port, which can be programmed to synchronize signals with cameras. The LiDAR is shown in Fig. 3.

C. GPS

The Real-time kinematic (RTK) GPS module with base correction is utilized to detect and track the instrumented
vehicle’s position and motion profile. The system has three key components: 1) Emlid Reach M+ RTK GNSS Module as a rover, 2) Tallysman Global Navigation Satellite System (GNSS) Antenna, and 3) INDOT-INCORS, a network of GNSS base collection systems operated by the Indiana Department of Transportation. The base station receives a signal from GPS satellites, creates a correction factor, and then sends the correction signal via WI-FI to the rover. Our system uses a 4G hotspot as the WI-FI for the communication to the base station. The Reach M+ RTK GNSS Module has a 5GB internal storage and can provide precise navigation and mapping. The GPS module is shown in Fig. 4.

D. PC

The computer we use runs on Linux 4.15.0-123-generic (x86-64) kernel and is installed with Ubuntu 20.04 LTS operating system. The system hosts an 8GB solid-state drive memory. The processor used is an Intel® Xeon(R) W-2123 CPU @ 3.60GHz × 8. The system also contains AMD Radeon Pro WX2100 (POLARIS12/DRM 3.23.0 / 4.15.0-123-generic, LLVM 6.0.0) for graphics Open GL rendering. In addition, the system consists of two SSDs and one hard disk of 500GB each for data saving. The system uses ROS [15] Noetic for recording data from the sensors.

E. Sensor Placement

The main goal of the vehicle-based data acquisitions system is to cover 360 degrees around the vehicle and record accurate GPS information. Four cameras are placed at the corners on the top of the vehicle, and two cameras are placed at the center, one front-facing and one back facing. The cameras on corners are equipped with wide-angle lenses, and the cameras at the center are equipped with 40-degree lenses. The LiDAR is placed at the center of the vehicle’s roof to provide maximum coverage around the vehicle. The GPS is put at the back of the vehicle and on the roof. Fig. 5 illustrates the placement of sensors on the instrumented vehicle.

III. OVERVIEW OF THE DATASET

The dataset is developed predominantly using the data from GPS and LiDAR, with the camera images as a reference. Each scenario is sixty seconds in length and consists of information about the ego vehicle (waypoints, velocity, and yaw angle), non-ego actors (waypoints, velocity, yaw angle, entry time, and exit time), and road information (name, centers, and banking angle). The dataset provides driving scenarios [16] with a diversity in driving behaviors.

A. GPS Processing

The GPS data is recorded as bag files, then extracted to obtain latitude, longitude, elevation, yaw rate, velocity, and timestamp. The map area covered by the GPS is then downloaded from the OpenStreetMap [17]. The road network is then created using MATLAB scenario generator along with road attributes. In this paper, the lane information is not included. It will be incorporated in the future using high-definition map service. The GPS data is filtered using Savitzkey - Golay filter [18], [16] and is used to compute the local trajectory and velocity of the ego vehicle. Fig. 6 shows the example GPS data collected in a driving scenario.
B. LiDAR Processing

1) Data Collected: The LiDAR data is recorded as UDP packets for better transfer speed and lossless recording. The UDP packets are also stored in bag files. These bag files are then extracted and the UDP packets are converted to MATLAB point-cloud object with their respective timestamps. The relative time of each point-cloud frame with respect to the GPS timestamp is obtained. The LiDAR data is then used to detect and track the non-ego actors. Fig. 7 shows exemplary point-cloud data recorded from LiDAR.

2) Object Detection: A virtual driving scenario was created with the GPS, IMU, and LiDAR data. The GPS data was used to show the trajectory and movement of the targets on a high-definition map. These scenarios allow the simulation and testing of different motion planning algorithms.

The Non-Ego actors were detected using the Joint Probabilistic Data Association algorithm [19] implemented on MATLAB. It allows for the detection of multiple targets. This algorithm enables estimation of states of unknown and time-varying targets from noisy and uncertain environments, which measures and calculates the hypothesis pairs for every agent for every track. Each track measurement hypothesis has a weight, which is calculated by the normalized sum of the conditional probabilities of the agent track association [20]. The tracker employs joint probabilistic data association to assign detection to each track. Multiple detection can contribute to each track; hence the tracker uses a soft assignment. Tracks are created, confirmed, corrected, predicted, and deleted by the tracker. In MATLAB, Detection reports created by objectDetection object are used as inputs to the tracker [21].

Any new track begins with a tentative state. If a tentative track receives enough detection, its status is changed to confirmed. The related track is instantly confirmed if the detection has a known classification. The tracker considers a track as a physical object once it has been confirmed. The track is removed if no detection is assigned within a certain amount of updates. The tracker calculates the state vector and state estimate error covariance matrix for each track [21]. Each detection has at least one track associated with it. If the detection cannot be assigned to an existing track, the tracker creates a new one. If a tracker maintains multiple tracks, the data association process becomes more complicated because one detection can fall within the validation gates of multiple tracks. In our case, clustering is used since there are multiple actors and tracks in one scenario.

The position and velocity of the surrounding vehicles in relation to the ego vehicle are provided by LiDAR detection. We converted the target vehicle positions to global coordinates and approximated the orientation angle based on vehicle motion using Equation (1) below.

\[
(X_t, Y_t) = (X_{ego}, Y_{ego}) + R(\psi_{ego}) \cdot (x_t, y_t)^T,
\]

where \( R \) is the Rotation Matrix, \((x_t, y_t)\) is the position in the ego vehicle’s coordinate, and \((X_t, Y_t)\) is the position in the global coordinate.

In scenarios where there are multiple non-ego actors close to each other, an issue arises where the tracks for all vehicles are noisy, and a track for a vehicle changes from one lane to another. This issue is resolved by reducing the default track difference length in MATLAB, which allows separate tracks for the respective non-ego actors. However, when the track difference length is small, noisy data can be included in the scenario. These noisy tracks are manually deleted by verifying the non-ego actors with the camera.

Since the Joint Probabilistic Data Association algorithm performs a soft assignment, it occasionally allows two non-ego actors very close to each other to be considered as one non-ego actor. This is again fixed by manually verifying it with the camera and changing the track assignment size so that each non-ego actor gets its own track.

C. Camera Data

Even though the camera data is currently used as reference for the generated scenarios, it will be used to add much more information to the dataset in the future. An example camera view from all six cameras is shown in Fig. 8.

IV. Scenario Development from the Dataset

This section describes several scenario development as examples and with reference images. Scenario 1 describes a case where the ego vehicle is in interaction with two
other vehicles, one at the back and one on the right side in the blind-spot range. This is depicted in the camera images in Fig. 9. The equivalent scenario generated using Matlab is shown in Fig. 10. The ego-vehicle is depicted and circled in blue. Comparing the images in Fig. 9(e) and Fig. 9(f), it can be seen that it matches with the three vehicles in the generated scenario. It can also be seen that the vehicle captured by the front center camera, Fig. 9(b), is not generated in the scenario. This is because the distance between the ego vehicle and the vehicle in front is large. This issue can be addressed where cameras are also involved in actor detection.

![Fig. 9: Reference camera frames - Scenario 1.](image1)

Scenario 2 depicts the situation where the vehicle in front slows down, changes lane to the left, and turns left. This can be seen in the reference images in Fig. 11. In Fig. 11, the front camera, right-front camera, and back camera shows the cars in view of the ego vehicle. It can also be seen in the generated scenario, as illustrated in Fig. 12.

Scenario 3 demonstrates an interaction at a traffic light. The scenario involves a non-ego vehicle which is about to start after a green light. The non-ego vehicle is about to be passed on the left by the ego vehicle and another non-ego vehicle on the right. The camera view is depicted in Fig. 13. The generated scenario is not shown due to the page limitation.

Note that all scenarios are generated at a frequency of 10 Hz. The scenarios contain information about all ego and non-ego vehicles’ waypoints, yaw angle, and the speed at each timestamp (see Fig. 14 as an example). The information about the scenarios can be exported as a Matlab function to include or change the scenario variable. We can change the variables of either the ego or non-ego vehicle to simulate a potential crash scenario. We can also add a barrier or a guardrail or even another non-ego actor into the scenario to mimic the real driving situation.

V. CONCLUSION

In this paper, we developed a scenario-based dataset using naturalistic driving data collected around downtown Indianapolis. The dataset contains ego-vehicle waypoints, velocity, yaw angle, as well as non-ego actors’ waypoints, velocity, yaw angle, entry-time, and exit-time. The real driving scenarios can be generated using the driving scenario designer in Matlab. Certain flexibility is provided to users so that actors, sensors, lanes, roads, and obstacles can be added to the existing scenarios. This dataset can be exported and it
serves as a great resource for the algorithm development on motion planning and control.

One future research direction is to enhance the existing dataset with camera-Lidar fused data. It is also interesting to study driving behavior at different traffic conditions.

REFERENCES

[1] Y. Chen and L. Li, *Advances in Intelligent Vehicles*. Academic Press, 2013.
[2] D. Cao, X. Wang, L. Li, C. Lv, X. Na, Y. Xing, X. Li, Y. Li, Y. Chen, and F.-Y. Wang, “Future directions of intelligent vehicles: Potentials, possibilities, and perspectives,” *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 1, pp. 7–10, 2022.
[3] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9710–9719, October 2021.
[4] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “auscenes: A multimodal dataset for autonomous driving,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11618–11628, 2020.
[5] 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,” 2019.
[6] J. Fritsch, T. Kühnl, and A. Geiger, “A new performance measure and evaluation benchmark for road detection algorithms,” in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 1693–1700, 2013.
[7] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, “Argoverse: 3d tracking and forecasting with rich maps,” in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8740–8749, 2019.
[8] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, “Trafficpredict: Trajectory prediction for heterogeneous traffic-agents,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6120–6127, 2019.
[9] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kümmerle, H. Königshof, C. Stillier, A. de La Fortelle, and M. Tomizuka, “INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps,” *arXiv:1910.03088 [cs, eess]*, Sept. 2019.
[10] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, “Learning social etiquette: Human trajectory understanding in crowded scenes,” in *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 549–565, Springer International Publishing, 2016.
[11] A. Lerner, Y. Chrysanthou, and D. Lischinski, “Crowds by example,” *Computer Graphics Forum*, vol. 26, no. 3, pp. 655–664, 2007.
[12] B. Coifman and L. Li, “A critical evaluation of the next generation simulation (ngsim) vehicle trajectory dataset,” *Transportation Research Part B: Methodological*, vol. 105, pp. 362–377, 11 2017.
[13] A. Lerner, Y. Chrysanthou, and D. Lischinski. “Crowds by example,” *Computer Graphics Forum*, vol. 26, no. 3, pp. 655–664, 2007.
[14] B. Benfold and I. Reid, “Stable multi-target tracking in real-time surveillance video,” pp. 3457–3464, 2011.
[15] Stanford Artificial Intelligence Laboratory et al., *Robotic Operating System*. 

Fig. 12: Generated Scenario 2.

Fig. 13: Reference camera frames - Scenario 3.

Fig. 14: Waypoints of vehicles in a generated scenario.
[16] S.-W. Park, K. Patil, W. Wilson, M. W. Corless, G. H. Choi, and P. A.
Adam, Creating Driving Scenarios from Recorded Vehicle Data for
Validating Lane Centering System in Highway Traffic. 2020.

[17] OpenStreetMap contributors, “Planet dump retrieved from
https://planet.osm.org,” 2017.

[18] R. W. Schafer, “What is a savitzky-golay filter? [lecture notes],” IEEE
Signal Processing Magazine, vol. 28, no. 4, pp. 111–117, 2011.

[19] “Sonar tracking of multiple targets using joint probabilistic data
association,” IEEE Journal of Ocean Engineering, pp. 173–184, 1983.

[20] Y. Bar-Shalom, F. Daum, and J. Huang, “The probabilistic data
association filter,” IEEE Control Systems Magazine, vol. 29, no. 6,
pp. 82–100, 2009.

[21] MATLAB, trackerJPDA, 2019.