Driving into the Dataverse: Real and Synthetic Data for Autonomous Vehicles

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in ***Computer Science and Engineering*** *by Research*

by

Shubham Dokania
2020701016
shubham.dokania@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled **“Driving into the Dataverse: Real and Synthetic Data for Autonomous Vehicles”** by Shubham Dokania, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. C V Jawahar

To my family and friends.

Acknowledgments

As I near the completion of my master's degree at IIIT Hyderabad, I find myself reminiscing about the incredible journey of learning and enjoyment I have experienced over the past few years. I am deeply grateful to all my advisors, mentors, collaborators, and friends who have accompanied me throughout this journey.

Firstly, I wish to convey my heartfelt appreciation to Prof. C.V. Jawahar for his exceptional guidance and dedication to fostering my research abilities. His expertise has illuminated the path to becoming a more accomplished researcher by providing invaluable insights and advice. Secondly, I extend my sincerest gratitude to Prof. Manmohan Chandraker, my co-advisor, for his profound wisdom and thoughtful feedback in our discussions. Each interaction with him has been an enriching experience, broadening my perspective on problem-solving. I am also thankful to Dr. Anbumani Subramanian, whose mentorship and active support have been indispensable in shaping my approach to research and guiding me towards success. Finally I'd like to thank Dr. A.H. Abdul Hafez, who has been an integral part of the dataset work and provided active support with his ideas and discussions. I will forever cherish the teachings of my esteemed mentors.

I fondly recall my arrival on campus, one year into the MS program, after the lockdown restrictions were lifted. The warm welcome from the people in the lab and the friendships I forged with Avijeet, Ashish, Seshadri, and Kiran helped me acclimate to the new environment. I am grateful to Radha Krishna for his assistance in setting up the data-collection car's hardware stack and facilitating communication with the annotation team. Additionally, I cherish the support of my project partners and friends Sai Amrit and Shanthika, who accompanied me during the online classes and course projects.

My time at CVIT has been enriched by the presence of wonderful friends, each unique in their own way. I am grateful for their unwavering support and the cherished memories we have created together, be it through intellectual discussions or spontaneous adventures. Madhav is one of those friends who is always up for anything. We have planned entire activities in a matter of minutes and roped in as many people we could in our shenanigans. Shivanshu has been a great friend and balances so many things between all the crazy plans and the serious work. I fondly recall the ECCV trip with Siddhant, as well as the exhilarating experiences shared with George and Soumya in Italy and Spain. The camaraderie and encouragement of my CVIT friends, including Pranav, Zeeshan, Ravi, Rudrabha, Aditya, Bipasha, Rupak, Sashank, Prachi, Darshan, and Varun, have fueled my growth throughout the MS journey.

I extend my deepest thanks to the administrative team, whose efforts have been instrumental in shaping my academic experience. Rohitha's unwavering support in handling documentation, Mr. Mahender's assistance with annotations, Mr. Ram's help in setting up the car and data collection, and Mr. Varun's IT infrastructure expertise have been invaluable. I am also grateful to Aradhana, Cecilia, Sarith, and the LACES team for their continuous support.

Finally, I cannot express enough gratitude to my best friend, Piyushi, who has been a pillar of strength and support during my time at IIIT Hyderabad. From listening to all of my rants to sharing everything, and planning all my trips, she's always had my back. Her selfless dedication and unwavering friendship have made my journey possible. I am eternally grateful to my family for their unconditional love, support, and encouragement, which have been a constant source of motivation throughout my academic pursuits.

In conclusion, I am indebted to everyone who has contributed to my growth as a researcher, student, and individual during my time at IIIT Hyderabad. I will forever treasure the memories, relationships, and lessons I have gained from this incredible journey.

Abstract

The rapid advancement of autonomous driving systems is transforming the future of transportation and urban mobility. However, these systems face significant challenges when deployed in complex and unstructured traffic environments, such as those found in many cities in south-east Asian countries like India. This thesis aims to address the challenges related to data collection, management, and generation for autonomous driving systems operating in such scenarios. The primary contributions of this work include the development of a data collection toolkit, the creation of a comprehensive driving dataset for unstructured environments (IDD-3D), and the proposal of a synthetic data generation framework (TRoVE) based on real-world data.

The data collection toolkit presented in this thesis enables sensor fusion for driving scenarios by treating sensor APIs as separate entities from the data collection interface. This framework allows for the use of any sensor configuration and demonstrates the smooth creation of driving datasets using our framework. This toolkit is adaptable to different environments and can be easily scaled, making it a crucial step towards creating a large-scale dataset for Indian road scenarios.

The IDD-3D dataset provides a valuable resource for studying unstructured driving scenarios with complex road situations. We present a thorough statistical and experimental analysis of the dataset, which includes high-quality annotations for 3D object bounding boxes and instance IDs for tracking. We highlight the diverse object types, categories, and complex trajectories found in Indian road scenes, enabling the development of robust autonomous driving systems that can generalize across different geographical locations. In addition, we provide benchmarks for 3D object detection and tracking using state-of-the-art approaches.

The TRoVE synthetic data toolkit offers a framework for the automatic generation of synthetic data for visual perception, leveraging existing real-world data. By combining synthetic data with real data, we show the potential for improved performance in various computer vision tasks. The data generation process can be extended to different locations and scenarios, avoiding the limitations of bounded data volumes and variety found in manually designed virtual environments.

This thesis contributes to the development of systems in complex environments by addressing the challenges of data acquisition, management, and generation. By bridging the gap between the current state-of-the-art and the needs of unstructured traffic scenarios, this work paves the way for more robust and versatile intelligent transportation systems that can operate safely and efficiently in a wide range of situations.

Contents

Chapter Page

1 Datasets for Driving Scenarios: An Introduction 1
1.1 Motivation 2
1.2 Indian Driving Dataset (IDD) and its Limitations 2
1.3 Challenges & Contributions 3
1.3.1 Data Collection Framework: 3
1.3.2 IDD-3D Dataset: 4
1.3.3 TRoVE - Synthetic Data Toolkit: 4
1.4 Organization of the Thesis 4

2 Collecting data in the real world: The Data Collection Toolkit 6
2.1 Introduction 6
2.2 Related Work 7
2.3 Infrastructure & Setup 7
2.3.1 Sensors 8
2.3.2 Design 9
2.4 Data Collection 10
2.4.1 Data Collection Interface 11
2.4.2 Data Processing 12
2.4.3 Calibration & Annotation 13
2.5 Summarizing Remarks 14

3 IDD-3D: The 3D Indian Driving Dataset 16
3.1 Introduction 16
3.2 Related Work 17
3.3 Proposed Dataset 20
3.3.1 Data Acquisition 20
3.3.2 Dataset Analysis 23
3.4 Experiments and Benchmarks 25
3.5 Comparison with KITTI and nuScenes 35
3.6 Summarizing Remarks 36

4 Synthetics Data Generation: TRoVE Toolkit 37
4.1 Introduction 37
4.2 Related Work 39
4.3 Our Approach 41

4.3.1 Building the Virtual Environment 42
4.3.2 Object and Camera Placement 42
4.3.3 Textures, Lighting and Background 43
4.3.4 Data Processing and Training 45
4.4 Experiments and Results 45
4.4.1 Datasets and Experiments 45
4.4.2 Result Analysis 46
4.4.3 Dataset Statistics 49
4.4.4 Additional Qualitative Results 50
4.5 Code and Data 50
4.6 Conclusions 53

5 Conclusion and Future Work 54
5.1 Future Work & Directions 55

Bibliography 58

List of Figures

Figure Page

2.1 The vehicle with all available sensors mounted. The LiDAR is mounted at the top of the car, and the 6 cameras are arranged in the configuration as front, front-left, front-right, back, back-left, and back-right. The GPS sensor is mounted towards the rear screen. All sensors are connected to a USB3 hub inside the car which is then connected to the on-board computing machine and storage. 9

2.2 The data collection interface: Shows a continuous stream of RGB images from the camera sensor to visually validate the current surroundings and sensor status. Provides instructions for data collection in simple steps. Details about the interface are mentioned in the corresponding section. 11

2.3 Data processing pipeline shows the various stages for the data acquisition/procurement, transfer, curation and processing of the dataset. The elements C1, C2, C3 refer to the camera/gps sensors, L1 is shown separately as the LiDAR sensor. The core components are divided as the online unit which is on-board, and the offline unit which performs tasks on the server. 12

2.4 A scene sample where the LiDAR points are projected on the RGB image from the front camera. The points are colored in a ”hsv” color-space in terms of the distance to the sensor in world coordinates. 13

2.5 Annotation example from a sample frame of the processed data. The bounding boxes are tight around the 3D LiDAR point cloud and colored according to object categories. Each box contains location and pose information for the objects. 14

3.1 Some examples from the dataset showing different traffic scenarios, LiDAR data with annotations, and a sample of LiDAR point clouds projected on camera data. 17

3.2 Samples from the dataset highlighting different (a) RGB images and (b) LiDAR Bird-Eye-View (BEV) along with bounding box annotations. The samples visualized above are taken from different sequences of the dataset. 18

3.3 Distribution of class labels in the proposed dataset. (a) The primary 10 classes are shown here along with the 3 super-categories (Vehicle, Pedestrian, and Rider) which are considered to make the proposed dataset more consistent with labels from existing datasets. (b) The additional 7 classes annotated in the dataset are shown in log-scale separately since they are currently not used for training the models. *The Rider class covers both riders and non-riders on two-wheeler motor vehicles.* We do not consider the Miscellaneous classes for evaluation of the dataset currently. 20

3.4 Figure showing (a) sensors on the vehicle (cameras, LiDAR) and their respective orientations, (b) image of the vehicle used along with the sensor rig. *Please note that the real-world car image has been edited to preserve anonymity.* 21

3.5 Generated trajectory from the LiDAR point clouds for one of the sequences of the dataset. 21

3.6 Samples of scenes of interest in our dataset (LiDAR and RGB samples) which especially differentiate our proposed dataset from those available in literature. *(Clockwise from top-left)* (a) Complex traffic scenarios with vehicles orientations in a wide variety of directions, (b) Perspective view of a scene with ego-vehicle on elevated flyover with ground level visible and another highway over the vehicle path with pillars, (c) humans in the middle of traffic (shown in red boxes) and jaywalking near moving vehicles, resulting in a safety critical scenario, (d) An example with very high density traffic scenario. Such case are abundant in the proposed dataset (rather than special cases when compared to other popular datasets) and hence require special attention for such unstructured environments. 22

3.7 Class-wise distribution of some common prominent classes (Car, MotorcycleRider, Pedestrian, TourCar) with respect to number of frames is visualized to show traffic and crowd density in proposed dataset. 23

3.8 (a) Distribution showing distances of all bounding boxes from the ego-vehicle. The short distance of vehicles and pedestrians provides motivation for the proposed dataset to facilitate modeling of shorter reaction times. (b) Cumulative distribution of the distances further highlight the differences in distance distributions, showing that most of the objects in the proposed dataset are close to the ego-vehicle compared to existing popular datasets. 24

3.9 Distributions of number of bounding boxes per LiDAR frame. The number of objects in a scene is usually higher in the frames present in the proposed dataset. We filter the boxes specifically based on the distance of less than 30m based on data shown in fig 3.7. (a) Shows statistics with KITTI dataset, and (b) shows the same without KITTI dataset to highlight the sparsity in the KITTI dataset. We note the heavier tail of our distribution indicating a greater density of objects close to the ego-vehicle. 24

3.10 Distribution of distances of the annotated bounding boxes with respect to the fraction of frames in the dataset. The plots are category specific in the above figures. 26

3.11 Distribution of bounding boxes annotated for each category in the dataset and the densities for number of boxes with respect to the fraction of frames in the dataset. 27

3.12 Some examples from the dataset showing different traffic scenarios, LiDAR data with annotations, and a sample of LiDAR point clouds projected on camera data. 28

4.1 A sample of the scene generated with the proposed method. The first image shown is a real-world sample from nuScenes dataset. Using some existing annotation, we are able to generate extensions such as depth, instance segmentation, RGB images, semantic segmentation, and surface normals. These modalities represent a part of the capabilities of the proposed method. 38

4.2 An overview of the synthetic data generation pipeline. Given real-world input dataset with annotations for object locations, we follow the process depicted in the above figure. (A) Map data extraction from OSM for building and road extraction. (B) Retrieval of categorical information and 3D object placement in scene with sampled camera poses. (C) Extraction of background data such as vegetation mask in world coordinates and PBR texture preparation. (D) Fusion of artifacts in the 3D environment and initialization of rendering process. (E) Generated RGB data and corresponding annotations stored for further use. 41

4.3 Example of variations in synthetic data for the same scene configuration. (a) The real-dataset image depicting a scene from the nuScenes dataset. (b) Sample render with different buildings, vehicles, pedestrians and lighting (c) Rendered image from ”b” after applying color transformation for the cityscapes dataset. (d) Render for the same scene from a different camera perspective. (e, f) Additional renders from the scene with variations in vehicles, buildings, lighting etc. 43

4.4 Samples from different steps in the proposed approach. Step (A) corresponds to the Section 3.1 for Building and Road processing, (B) corresponds to object placeholder creation and placement, (C) shows vegetation texture density as image, and part of building/road texture in step (A) output. Finally, Stage (D) shows the rendered output for the given scene and the camera pose data. 44

4.5 Qualitative results for different training strategies using real and synthetic data across KITTI-STEP and Cityscapes datasets. The nomenclature is R: Real, S: Synthetic, P: Real (partial), [F]: Fine-tuned on real, [M]: Real mixed with synthetic, C: Color transformation 49

4.6 A side-by-side comparison of a scene with day time lighting in the real as well as synthetic data. We slightly adjust the real-world objects in the synthetic scene and place the assets at the respective locations to replicate real-world structure. 50

4.7 This sample shows the color adjustment example where the objects are placed in similar locations with changes in assets and color properties in the scene. 51

4.8 We take a real-world scene with only the meta-data and environment information and can replicate the scene in different variations. The current sample shows the synthetic data rendered in night scene. The lighting is different and the sky is illuminated with stars which provides ambient lighting in the virtual environment. 51

4.9 We highlight the same scenario as figure 4.8 but keeping the environmental lighting similar while changing some background properties. 51

4.10 Furthermore, we can generate scenarios with different shading and sun positinos, while also eliminating existing weather conditions. 52

4.11 This sample shows the day-to-dusk shift with illuminated building windows and sunset lighting. These effects can be achieved with ray-traced rendering as shown and are superior to generative methods. 52

List of Tables

Table Page

2.1 Available sensors on-board the vehicle used for data collection. The description of each sensor and its configuration is provided in the corresponding sensors section. The resolution is mentioned wherever applicable. 8

3.1 A comparison with existing popular 3D autonomous driving datasets. Our dataset showcases the highest diversity with the highest average number of bounding boxes per frame and a wide distribution. The statistical distribution is further studied in the following sections. (*) Number reported on train-val-test set, experiments/statistics reported on train-val set. (**) The 17 classes are total of the 10 primary and 7 additional classes. . 19

3.2 Results on IDD-3D with popular methods. We report AP scores across different categories on the validation set. This table shows the results on each training class. The scores are reported with different thresholds for each class (Vehicles @ 0.5, Rider @ 0.4, Pedestrian @ 0.3) and all objects are considered till 30m distance. 29

3.3 Experimental results on proposed dataset with different popular methods. We report AP scores across different categories on the validation set. This table shows the results on Vehicle and Rider categories from the proposed dataset. 30

3.4 Experimental results (continued) on proposed dataset with different popular methods. We report AP scores across different categories on the validation set. This table shows the results on Pedestrian category and the mAP score from the proposed dataset. 30

3.5 Experimental results for 3D object tracking for the 10 primary classes present in the proposed dataset. We use SimpleTrack [61] for the task of tracking using detections from CenterPoint [95] in the presented table. 31

3.6 Experimental results on proposed dataset with different popular methods. We report AP scores across different categories on the validation set. This table shows the results on all training categories with all distance buckets. The ’overall’ distance metric ranges from 0-30m and is considered based on the distance distribution of objects present in the scene. 32

3.7 Tracking (3D-MOT) results on the proposed dataset for Centerpoints method only trained on the proposed dataset. 33

3.8 Tracking (3D-MOT) results on the proposed dataset for SECOND without any pre-training. 33

3.9 Tracking (3D-MOT) results on the proposed dataset for SECOND method pre-trained on the KITTI dataset. 34

3.10 Tracking (3D-MOT) results on the proposed dataset for the Pointpillar method. 34

4.1 Quantitative results for training on real and synthetic data and validation on Cityscapes dataset. We report class-wise IoU, mIoU and global accuracy for the 12 classes (excluding void) . . . 47
4.2 Quantitative results for training on real and synthetic data and validation on KITTI-STEM dataset. We report class-wise IoU, mIoU and global accuracy for the 12 classes (excluding void) . . . 48

Chapter 1

Datasets for Driving Scenarios: An Introduction

The rapid advancements in the fields of artificial intelligence, machine learning, computer vision, and robotics have paved the way for the development of autonomous vehicles and advanced driver assistance systems (ADAS). These technologies have the potential to revolutionize the transportation industry, enhance road safety, reduce traffic congestion, and contribute to a more sustainable and efficient mobility ecosystem. As the world moves toward greater urbanization, the need for intelligent transportation solutions becomes increasingly urgent. The adoption of autonomous driving and ADAS technologies could play a significant role in addressing the challenges posed by growing urban populations and the subsequent demands on transportation infrastructure.

At the heart of autonomous driving and ADAS development lies the need for high-quality, diverse, and structured data that can be used to train, validate, and test the sophisticated algorithms that drive these systems. As the complexity and diversity of driving environments increase, so too does the importance of having access to robust and comprehensive datasets that encompass a wide range of scenarios, geographical locations, and traffic conditions. This is particularly true for regions with highly unstructured and congested traffic, such as Southeast Asian countries, where traditional driving datasets collected in well-regulated environments may not adequately represent the unique challenges faced by autonomous vehicles.

In this thesis, we delve into the creation, management, and utilization of driving datasets for autonomous driving and road scene understanding. We explore various aspects of data collection, annotation, and generation, with a focus on multi-modal data and unstructured environments. Through this investigation, we aim to contribute to the development of more generalizable and robust autonomous driving systems that can effectively navigate the complex and diverse road conditions encountered around the globe.

Our work is driven by the belief that advancements in data acquisition, processing, and generation methods can have a profound impact on the performance and generalizability of autonomous driving systems. By developing innovative techniques and tools for data collection, annotation, and synthesis, we hope to create a strong foundation upon which the next generation of intelligent vehicles and ADAS applications can be built.

1.1 Motivation

The motivation behind this thesis is rooted in the recognition that the performance of autonomous driving systems and ADAS is highly dependent on the quality and diversity of the data used for training and validation. As autonomous vehicles are expected to operate in a wide variety of environments, it is crucial to ensure that the datasets employed during their development accurately reflect the complexity and variability of real-world driving conditions. Traditional datasets, often collected in well-structured and regulated environments [23, 24, 31, 33, 51], may not be sufficient to capture the nuances and intricacies of unstructured traffic scenarios, especially those encountered in densely populated regions like Southeast Asia.

The lack of comprehensive datasets representing such unstructured environments poses a significant challenge for the development of autonomous driving systems that can effectively navigate complex traffic conditions. The need for data that encompasses a diverse range of road scenes, object categories, and traffic behaviors is becoming increasingly apparent as researchers and engineers strive to create more versatile and adaptive autonomous vehicles.

This thesis aims to address the limitations of existing driving datasets by exploring novel methods for data collection, annotation, and generation that take into account the unique characteristics of unstructured environments. By developing innovative tools and techniques for data acquisition and management, we hope to contribute to the creation of more representative and diverse datasets that can be used to improve the performance and generalizability of autonomous driving systems and ADAS applications.

Additionally, we are motivated by the potential benefits of synthetic data generation as a means to augment real-world data [11, 28, 73, 91]. Synthetic data offers several advantages, including the ability to create a virtually unlimited number of unique scenes, simulate various sensor configurations, and incorporate diverse environmental conditions. By leveraging synthetic data, we can potentially overcome the limitations of manual data collection and annotation efforts, while still preserving the real-world structural properties that are essential for training effective autonomous driving systems.

In summary, the motivation for this thesis lies in the recognition of the critical role that high-quality, diverse, and representative datasets play in the development of autonomous driving systems and ADAS applications. By exploring novel methods for data collection, annotation, and generation in unstructured environments, we aim to contribute to the ongoing efforts to create more robust, adaptive, and versatile intelligent transportation solutions.

1.2 Indian Driving Dataset (IDD) and its Limitations

Overview of IDD: The Indian Driving Dataset (IDD) [82] is a comprehensive dataset that captures the unique characteristics of Indian road scenes. It includes diverse and challenging scenarios that are often absent in other popular driving datasets. The IDD consists of over 10,000 images, with detailed annotations for various object categories, including vehicles, pedestrians, riders, and static objects. The dataset is particularly valuable for its representation of unstructured traffic scenarios, which are common in densely populated regions like India.

Limitations of IDD: Despite its contributions, the IDD has certain limitations that this thesis aims to address. The dataset primarily consists of 2D images, which may not be sufficient for developing and validating advanced driver assistance systems (ADAS) and autonomous vehicles that rely on 3D perception. The lack of depth information and 3D annotations in the IDD limits its applicability for tasks such as 3D object detection and tracking, which are crucial for understanding complex traffic scenarios.

Moreover, the IDD, like many other real-world datasets, is limited by the cost and effort associated with data collection and annotation. The dataset's coverage of diverse scenarios is inherently constrained by the locations and conditions under which the data was collected. This limitation is particularly significant for autonomous driving systems, which are expected to operate in a wide variety of environments and conditions.

Addressing the Limitations: In this thesis, we aim to address these limitations by introducing the IDD-3D dataset and the TRoVE synthetic data toolkit. The IDD-3D dataset extends the original IDD by incorporating high-quality LiDAR data and 3D annotations, thereby enabling research in 3D object detection and tracking in diverse environments. On the other hand, the TRoVE toolkit allows for the automatic generation of synthetic data, which can augment real-world data and overcome the limitations of manual data collection and annotation efforts. By leveraging synthetic data, we can create diverse and physically meaningful variations in scenes, thereby improving the performance and generalizability of autonomous driving systems.

In summary, while the IDD has significantly contributed to the understanding of unstructured traffic scenarios, this thesis aims to further enhance its utility by addressing its limitations through the introduction of 3D data and synthetic data generation techniques.

1.3 Challenges & Contributions

The development of autonomous driving systems and ADAS applications for complex and unstructured traffic scenarios presents several challenges, which this thesis aims to address. Our primary contributions can be summarized as follows:

1.3.1 Data Collection Framework:

Challenge: Creating an efficient and flexible data collection framework that allows for seamless integration of multi-modal sensors, easy calibration, and adaptability to different sensor configurations.

Contribution: We developed a data collection and management system that relies on the robotic operating system (ROS) architecture and uses a Qt-based GUI for ease of use. The framework enables sensor fusion, supports various sensor configurations, and facilitates data processing and curation, ultimately contributing to the creation of driving datasets for diverse environments.

1.3.2 IDD-3D Dataset:

Challenge: Addressing the lack of comprehensive datasets representing unstructured driving scenarios, particularly in Southeast Asian countries where traffic densities and inter-object behaviors are more complex.

Contribution: We introduced the IDD-3D dataset, which captures unstructured driving scenarios in Indian road scenes, with data collected using high-quality LiDAR sensors and cameras. The dataset includes detailed annotations for various object categories and offers unique insights into complex trajectories, enabling research in 3D object detection and tracking in diverse environments.

1.3.3 TRoVE - Synthetic Data Toolkit:

Challenge: Overcoming the limitations of manual data collection and annotation efforts while preserving the real-world structural properties essential for training effective autonomous driving systems.

Contribution: We proposed a framework for automatic generation of synthetic data for visual perception using existing real-world data. Our approach enables the creation of diverse and physically meaningful variations in scenes while minimizing the domain gap between synthetic and real data. By combining synthetic data with real data, we demonstrated improvements in performance across various computer vision tasks.

These contributions tackle the main challenges in developing autonomous driving systems for complex and unstructured environments. By offering novel tools and techniques for data acquisition, management, and generation, this thesis aims to support the ongoing efforts to create more robust and versatile intelligent transportation solutions.

1.4 Organization of the Thesis

The thesis is organized into the following chapters, which provide an in-depth exploration of the challenges and contributions in developing autonomous driving systems for complex and unstructured traffic scenarios:

Introduction (Chapter 1): This chapter sets the stage for the thesis, providing an overview of the motivation, challenges, and primary contributions. It also outlines the organization of the subsequent chapters.

Data Collection Toolkit (Chapter 2): In this chapter, we present the development of a data collection framework that enables sensor fusion for driving scenarios. We discuss the design choices, sensor details, and data processing steps, highlighting the framework's ease of use, adaptability, and scalability.

IDD-3D Dataset ((Chapter 3): This chapter introduces the IDD-3D dataset, a comprehensive dataset for unstructured driving scenarios in Indian road scenes. We provide a thorough statistical and experimental analysis of the dataset, as well as benchmarks for 3D object detection and tracking using state-of-the-art approaches.

TRoVE - Synthetic Data Toolkit ((Chapter 4): In this chapter, we propose a framework for automatic generation of synthetic data for visual perception using existing real-world data. We discuss the advantages of combining synthetic data with real data, the potential for improved performance in various computer vision tasks, and the possibilities for expanding the data generation process to different locations and scenarios.

Conclusions and Future Work (Chapter 5): This chapter synthesizes the main findings and contributions of the thesis, reflecting on the overall impact of the work and its implications for the development of autonomous driving systems in complex environments. We also discuss potential future research directions and extensions of the current work.

By addressing the challenges of data acquisition, management, and generation in unstructured traffic scenarios, this thesis aims to contribute to the ongoing efforts in developing robust and versatile intelligent transportation systems. Through the development of new tools and techniques, we seek to improve the generalizability and applicability of autonomous driving systems across diverse environments and conditions.

Chapter 2

Collecting data in the real world: The Data Collection Toolkit

2.1 Introduction

Intelligent vehicles and Autonomous driving systems have come a long way and keep becoming more sophisticated over time. Much of this advancement is owing to the rapid progress in the deep learning and computer vision community. However, the core component for all these increments is the availability of high quality and structured data which augments the strengths of these sophisticated models and brings out the best performance. Recently, there have been many works which focus on the process of data selection and quality improvement [20, 70, 97], building high quality and large scale datasets in the autonomous driving community, and approaches built using these resources which improve the state of autonomous driving [36, 98].

In this chapter, we explore the process of building a data collection and management system for autonomous driving systems with multi-modal sensors and discuss ways for data cataloguing and processing. The developed system can consolidate data streams from different types of sensors (Camera, LiDAR, GPS etc.) and store them in the form of rosbags for raw data storage. The collection framework relies on a robotic operating system (ROS) [67] architecture and uses a Qt based GUI [89] as a front-end to facilitate easy access to the data collection without any technical training. The core highlights of our proposed system are:

1. Ease of data collection using a GUI tool for calibration, sensor monitoring and recording.
2. Post-processing and analysis of multi-modal sensor data.
3. Data management and transformations for annotation, consolidation and visualization.

The rest of the chapter is organised in the following sections: In Related work, we talk about the literature in the autonomous driving community, specifically about popular datasets and data collection frameworks; In Infrastructure & Setup, we discuss the need for the data collection kit, design choices and sensor details; and in Data Collection, we outline the details of the architecture, properties of the data we are managing with the proposed system and the database design for storage and post-processing.

We conclude the discussions about the future directions and the immediate steps for which the proposed system will be used, i.e. the large-scale dataset collection task.

2.2 Related Work

The importance of multi-modal datasets and related benchmarks for autonomous driving systems is very well established. Availability of large scale public datasets and associated benchmarks/challenges accelerate the progress of autonomous driving system and enables interdisciplinary research collaborations [98]. The existing autonomous driving datasets such as KITTI [33], Oxford Robotcar [54], Apolloscape [38], nuScenes [9] and Argoverse [15] help in achieving this purpose.

All of these popular datasets are collected in a vast set of environments with variations in the sensor configurations. Only a small fraction of the recent datasets provide a model for sensor-fusion in the range of sensors available [35, 62, 79]. In recent works, the volume of data collected and annotated has been increasing [79], and this effort enhances the quality of autonomous driving benchmarks. Similarly, the availability of many annotation tools make it easier to develop datasets for driving scenarios [3, 49, 101]. However, there is a lack of software available for data collection and curation which create a gap in the process. Some works make their platforms available [38], while some provide detailed information about the data collection process [9], but most of these are targeted towards a very specific set of sensors that the respective datasets are using. It becomes very difficult to use the same data collection setup, and often requires creation of new frameworks from scratch for different purposes. Some effort has been observed in the community for building low-cost interface for data collection [42], but scaling up such software and using new technologies is not available in the set of limited functionalities offered in such suites. Towards this, we propose a new data collection framework which makes the sensor kit and the collection kit as disjoint entities, and also provide methods for post-processing and data storage for creating structured datasets around driving scenarios, as described in this work.

2.3 Infrastructure & Setup

While many datasets exist for autonomous driving tasks, the support for openly available and easily accessible tools for data collection and management remains scarce and often tedious to set up. One of the major problems with data collection lies in the fact that usually it is performed by individuals who may not have extensive technical knowledge about the sensors or the system internal functions. This often results in loss of data, misalignment or bad synchronization in data streams from the multi-modal sensor kit or corruptions in the metadata. Such situations render the data capturing efforts unusable and are usually discovered at a later stage when data consolidation or processing is happening. Furthermore, it is also possible that the changes in sensor setup etc may result in modifications of sensor parameters (say extrinsic and intrinsic parameters for the camera) and as a result the captured data shows erroneous output when processed. To avoid such situations, we ensure that multiple checks are added to the

Table 2.1 Available sensors on-board the vehicle used for data collection. The description of each sensor and its configuration is provided in the corresponding sensors section. The resolution is mentioned wherever applicable.

Sensor	Qty.	Resolution	Configuration	Manufacturer/Model
LiDAR	1	64 channel (vertical) 1024 channel (horizontal)	10 Hz capture. XYZ, Intensity, Reflectivity, Range	Ouster OS1 sensor
Camera	6	2048 x 1536	BayerRG8 format 10 Hz capture	FLIR Blackfly S, C-mount
Lens	6	-	UC Series Fixed focal length 12/25mm	Edmund optics
GPS	1	-	G-Star IV BU-353-S4 sensor ˜1Hz	GlobalSat

collection interface so that each step in the data collection pipeline is automatically followed and all quality requirements are met.

2.3.1 Sensors

The available sensors are shown in table 2.1 along with the frame rate and resolution (wherever applicable). We now discuss the sensors and the corresponding SDKs used for construction of the data collection kit, and then discuss the overall system design in the later sections. These SDKs provide the APIs which form the core of the proposed system.

- **LiDAR**: The LiDAR sensor we are using is an Ouster OS1 with 64 channel vertical and 1024 channel horizontal resolution. The available sensor has a 10Hz data capture rate and is able to acquire data in the form of point clouds for the data collection kit.
- **Camera**: For this work, we are using a total of six RGB cameras with a resolution of 2048x1536 pixels. We capture data at 10 frames per second from the camera to align well with the LiDAR sensor capture rate. The six cameras are triggered in a sequential way to align with the rotation of

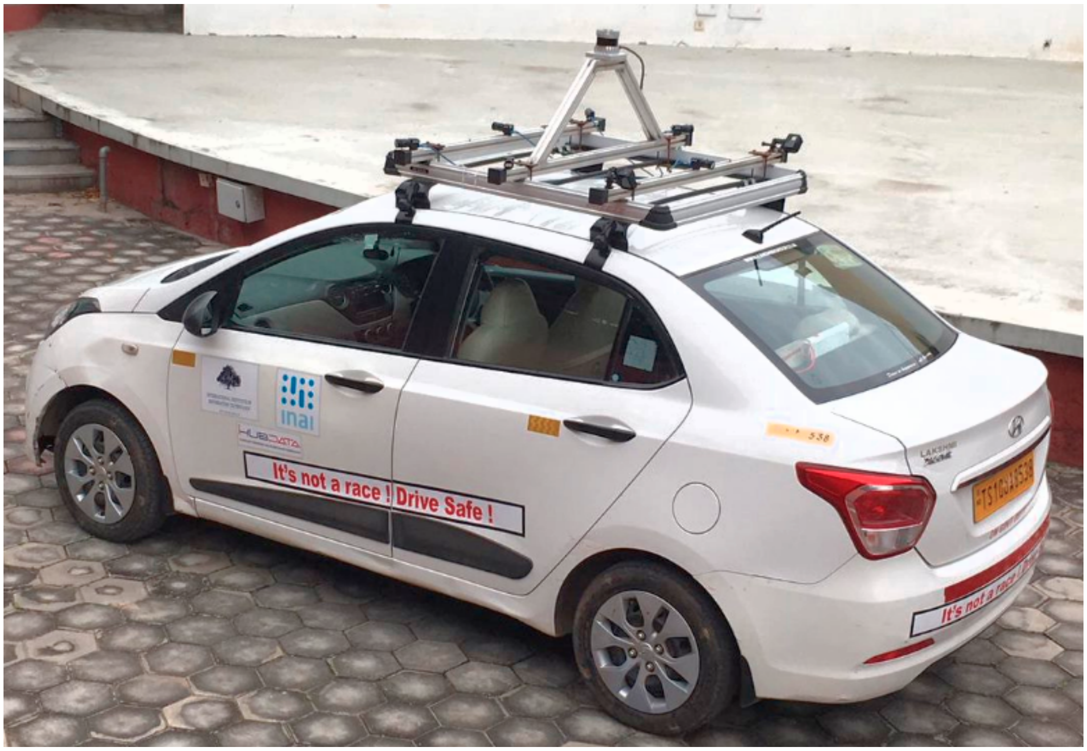


Figure 2.1 The vehicle with all available sensors mounted. The LiDAR is mounted at the top of the car, and the 6 cameras are arranged in the configuration as front, front-left, front-right, back, back-left, and back-right. The GPS sensor is mounted towards the rear screen. All sensors are connected to a USB3 hub inside the car which is then connected to the on-board computing machine and storage.

the LiDAR sensor, however, we also provide an option to perform a simultaneous trigger in the toolkit.

- **GPS**: We are using a globalsat G-Star BU-353-S4 global positioning system sensor and capturing the data at a rate of 1Hz as supported by the sensor. The GPS data is used to accurately keep track of the vehicle trajectory and make the geopositioning information available for further usage.

2.3.2 Design

Some of the challenges and the design decisions for the preparation of the data collection system are outlined in this section. We first discuss the essential aspects of the data storage, sensor calibration and on-board computing:

Data Storage: Now each of these sensors are generating at a high resolution and require very fast write rates on the disk. The approximate size of the data generated per minute is 15GB. This amount of data is enough to occupy a 1TB disk with roughly one hour of continuous data collection. For smooth write speeds to the disk, we are using SSDs with a very fast transfer rate and record the data directly to the external disk, avoiding any additional data transfer overhead on the system.

Sensor Calibration: The usability of the captured data, especially in a multimodal sensor scheme, depends on the availability of the sensor calibration information. For example, with the cameras and LiDAR, if we have the intrinsic and extrinsic calibration data, we can estimate the relative position of an object in the scene with respect to all the sensors and hence enable sensor fusion. It is crucial to know the relation and conversion between the data space for each of the sensors to enable adequate usage of the captured data. For our capture setup, we perform camera calibration as a necessary step before each capture and record the calibration video from the sensor streams so that it can be verified again offline. This allows us to also perform LiDAR-camera calibration in an offline manner without impacting the accuracy of the data captured. We show an example of the LiDAR-camera calibration done offline in Figure 2.4.

On-board Computing: For the on-board computing capability, we have a machine with Intel i7-11th Gen processor, 16GB RAM, and Ethernet port. For the data capture, we use a USB3 extension module and connect the 7 high-speed-transfer sensors (camera + LiDAR). Additionally, the machine also consists of an Nvidia GeForce RTX 3070 6GB GPU which can allow for deep learning applications to be run in parallel as well. This capability can be used in the future for smarter data collection processes as well.

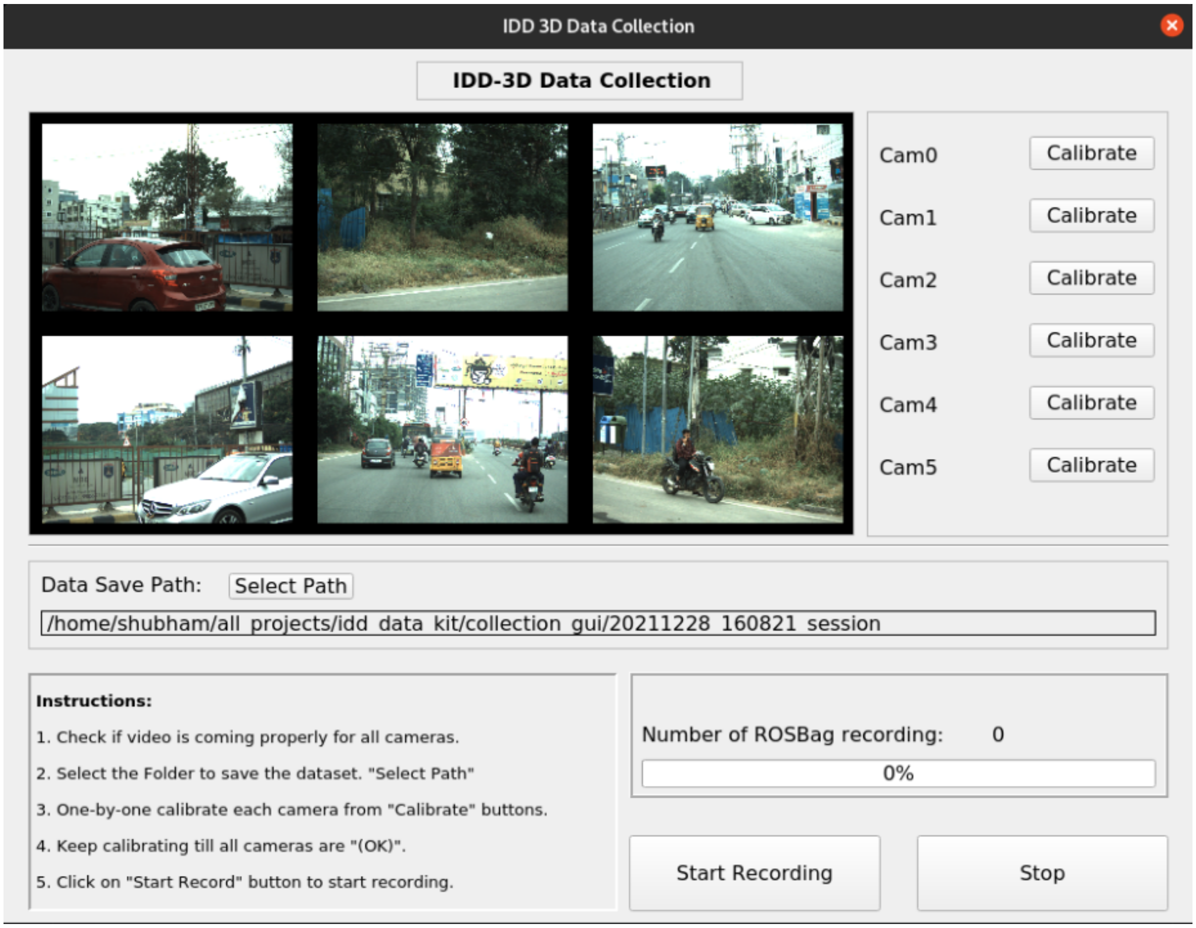
2.4 Data Collection

For the data collection kit, there are a some key points to keep in mind for efficient performance and design of the system:

- **Ease of use**: The system should be easy to use since after deployment in a real-world setting, a complete technical training of on-site individuals cannot be assumed. The system should be simple and cover all aspects of data collection including sensor checkups, data formatting and metadata collection.

- **Speed and Scalability**: It is important that the interface provided for data collection be fast to set up and be scalable to multiple systems (so that in the future a fleet of vehicles operating on the same technology can be used, for example).

Keeping in mind the key factors, we now describe the interface of the data collection tool, the data format and post processing steps for dataset curation, and show samples from annotation and sensor-fusion/calibration.



Data Save Path: /home/shubham/all projects/idd data kit/collection gui/20211228 160821 session

Instructions:

1. Check if video is coming properly for all cameras.
2. Select the Folder to save the dataset. "Select Path"
3. One-by-one calibrate each camera from "Calibrate" buttons.
4. Keep calibrating till all cameras are "(OK)".
5. Click on "Start Record" button to start recording.

Number of ROSBag recording: 0

0%

Figure 2.2 The data collection interface: Shows a continuous stream of RGB images from the camera sensor to visually validate the current surroundings and sensor status. Provides instructions for data collection in simple steps. Details about the interface are mentioned in the corresponding section.

2.4.1 Data Collection Interface

As shown in the figure 2.2, the graphical interface for the data collection tool comprises all the steps and provides instructions for the operation as well. The tool was developed with the Python Qt [89] with integration from ROS python. The first window shows the visual inputs from each of the cameras mounted on the vehicle. This enables us to visually verify the working condition of each of the camera sensors and the data quality. Furthermore, the instructions provided outline the steps to be followed for initiating the data collection. To ensure that the system is robust, there are multiple automated checks in place which do not allow a user to circumvent any of the steps mentioned in the instructions, i.e. an error will appear if a user tries to skip any of the steps.

Following the instructions, first a storage path needs to be selected from a popup window (which appears on clicking the button), which is usually on the storage SSD device. A directory is automatically

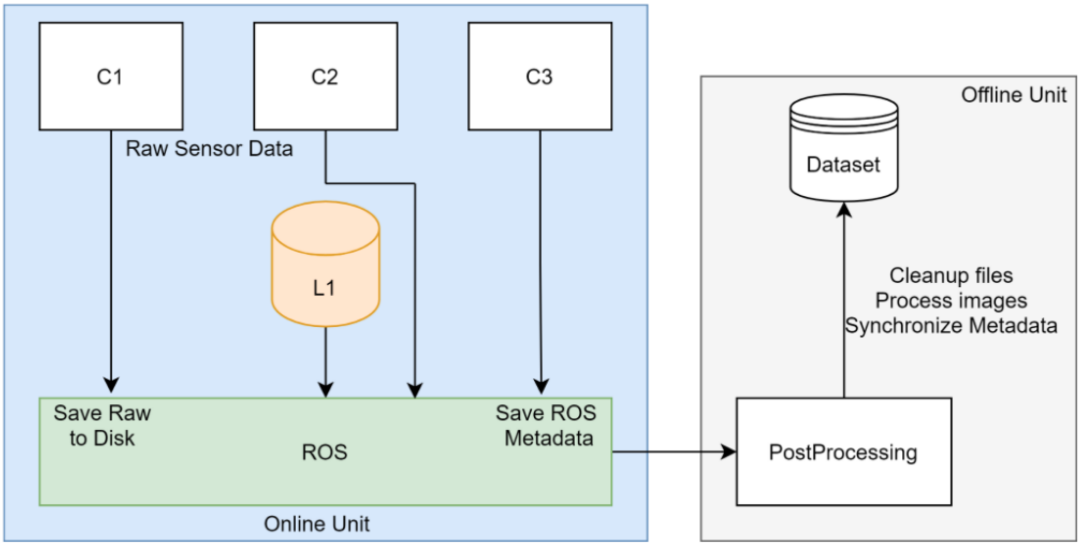


Figure 2.3 Data processing pipeline shows the various stages for the data acquisition/procurement, transfer, curation and processing of the dataset. The elements C1, C2, C3 refer to the camera/gps sensors, L1 is shown separately as the LiDAR sensor. The core components are divided as the online unit which is on-board, and the offline unit which performs tasks on the server.

created with the format YYYYMMDD_SessionID (as shown in the figure). This allows for uniquely identifying the raw data source, location and helps in future debugging. Then, each camera is calibrated one-by-one. To ensure proper calibration, an "(OK") symbol appears for each calibrated camera. Once the calibrations are completed, the user can click on the recording button and recording of the data in rosbag format is initiated in the background. To measure the progress of the recording, we display a progress bar which shows the number of rosbags created so far (each bag contains 5 minutes of data) and the progression of current bag recording. Once the user is satisfied with the recording, they can click on the "Stop" button and save the data recorded.

2.4.2 Data Processing

We follow a multi-stage approach for data collection and processing to ensure a robust flow in each stage of the dataset procurement, curation and production flow, as shown in figure 2.3, and explained in the below points:

- **Data procurement**: In the shown online unit (sensors, storage, and computing), we make use of the aforementioned data collection interface to procure data from each of the sensors (shown as C1, C2, C3, and L1 in the figure) as raw data along with the metadata (from camera, lidar, gps and ROS meta info). This collected raw data is stored in the form of rosbag files.

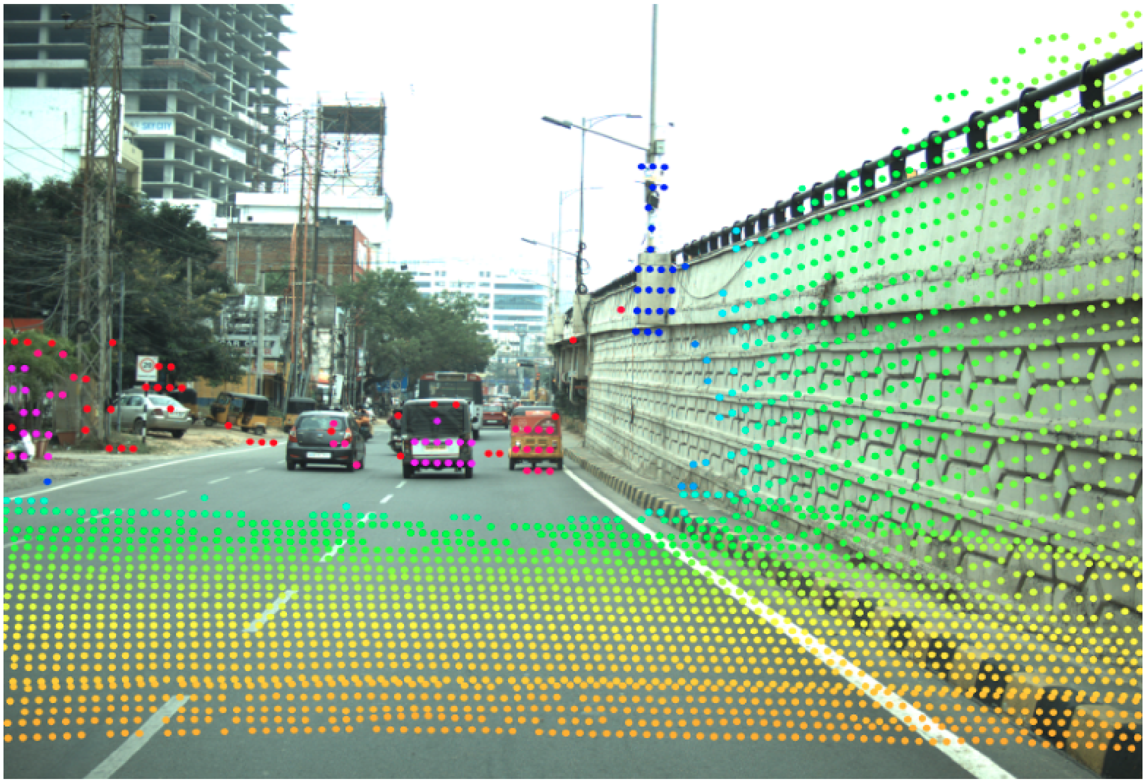


Figure 2.4 A scene sample where the LiDAR points are projected on the RGB image from the front camera. The points are colored in a ”hsv” color-space in terms of the distance to the sensor in world coordinates.

- **Data Curation**: After each data procurement session, the storage SSD devices are sent for curation to the data servers where all session recording are stored in the RAW format (along with necessary backups). The metadata is used to store information about the type and volume of collection and display on the metadata dashboard (an example was shown earlier in figure 2.2).

- **Post-processing**: To provide a standard dataset structure, we prepare a post-processing stage which crawls through the raw data and performs data cleanup, image processing, sensor synchronization and stores the corresponding multi-sensor information in different locations. A database is maintained with tokens from each session to ensure efficient lookup and availability of each item for different recordings, session parts, frames, locations etc.

2.4.3 Calibration & Annotation

As a final step before the data is made available in a structured manner, camera extrinsic and annotations are also required before the dataset can be used for analysis and model development purposes. While the details about the data statistics, annotation categories, and volumes are beyond the scope of

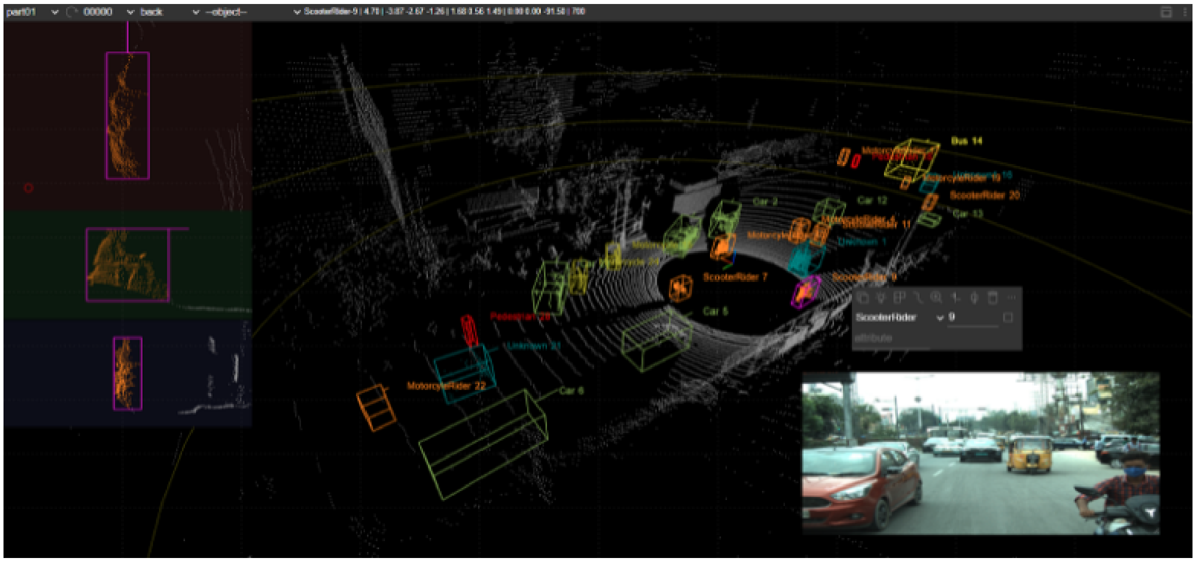


Figure 2.5 Annotation example from a sample frame of the processed data. The bounding boxes are tight around the 3D LiDAR point cloud and colored according to object categories. Each box contains location and pose information for the objects.

this manuscript, we show some samples of the annotation tool outputs and calibrated image visualizations in figures 2.4 and 2.5.

Figure 2.4 shows the projection of the LiDAR points from a sample frame onto the corresponding front camera image. The extrinsic calibration in the post-processing stage is performed in a semi-automated way. Since we already have the camera intrinsics available from the metadata available, we provide an initial estimate of the extrinsic parameters of the camera and then project the LiDAR points on the image. Then iteratively we align the projected points to the RGB image and measure the extrinsic data. This is performed in a similar way as shown in [41, 60, 80] for targetless extrinsic calibration. The annotation sample shown in figure 2.5 is taken from a similar scene where the back camera image is visualized along with the 3D bounding boxes for the objects present in the scene. The bounding boxes are colored based on different object categories. We are using a modified version of SUSTechPoints annotation tool [49] for this purpose.

2.5 Summarizing Remarks

We demonstrated the development of a data collection framework for driving scenarios which enables sensor-fusion. The key component of this framework is that the sensor APIs which rely on ROS (Robot operating system) are treated as a separate entity from the data collection interface. This disjoint system allows us to use any sensor configuration (from low- cost, low-resolution, high-availability to high-cost, high-precision sensors). We also outlined the data processing steps and show how the curation of data using our framework leads to smooth creation of driving datasets. This framework can be used in different environments and can be scaled easily. Although the current framework shows capability, we are always iteratively improving the performances and aiming for better systems. This collection kit is a step towards the bigger goal of creating a large-scale dataset for Indian road scenarios and benchmarks for autonomous driving, which we plan to release in the future as an extension of this work.

Chapter 3

IDD-3D: The 3D Indian Driving Dataset

3.1 Introduction

Existing datasets for road scenarios and autonomous driving use-cases are usually collected in well-structured environments with proper traffic regulations and relatively-evenly distributed traffic. In such situations, crowd behavior demonstrates low diversity and average densities. In south-east Asian countries, such as India, the traffic densities and inter-object behaviors are much more complex. Such complexities have been studied in the past [13,15,82], but extensive data coverage and multi-modal systems are still unavailable for such scenes. It hence may not be entirely applied to cases where the distribution of object categories and types varies greatly.

In this chapter, we propose a dataset on complex unstructured driving scenarios with multi-modal data, highlighting the capabilities of 3D sensors such as LiDAR for better scene perception in unstructured and sporadically chaotic traffic conditions. In the proposed dataset, we highlight a significantly different distribution of object types and categories compared to existing datasets collected in European or similar settings [31,51,79], due to the different nature of traffic scenes in Indian roads. Furthermore, the categories and annotations available in the proposed dataset vary greatly from existing datasets. Specifically, they cover objects in scenes that usually appear in still-developing cities, for example, Auto-rickshaws, hand carts, concrete mixer machines on roads, and animals on roads.

We provide data collected in Indian road scenes, from high-quality LiDAR sensors and six cameras that cover the surrounding area of the ego-vehicle to enable sensor-fusion-based applications. We provide annotations for 15.5k frames in the dataset, which spans 10 primary categories (and 7 additional miscellaneous categories), which we use for model training and evaluation. Along with the annotations, we also provide extra unlabelled raw data from the sensors to facilitate further research, especially into self- and unsupervised learning over such traffic scenes. A unique feature of the proposed dataset, which stems from the unstructured environment, is the availability of highly complex trajectories. We show samples from the dataset which emphasize such cases and display experiments on object detection and tracking, which is possible due to availability of instance specific labels for each object bounding box per sequence.

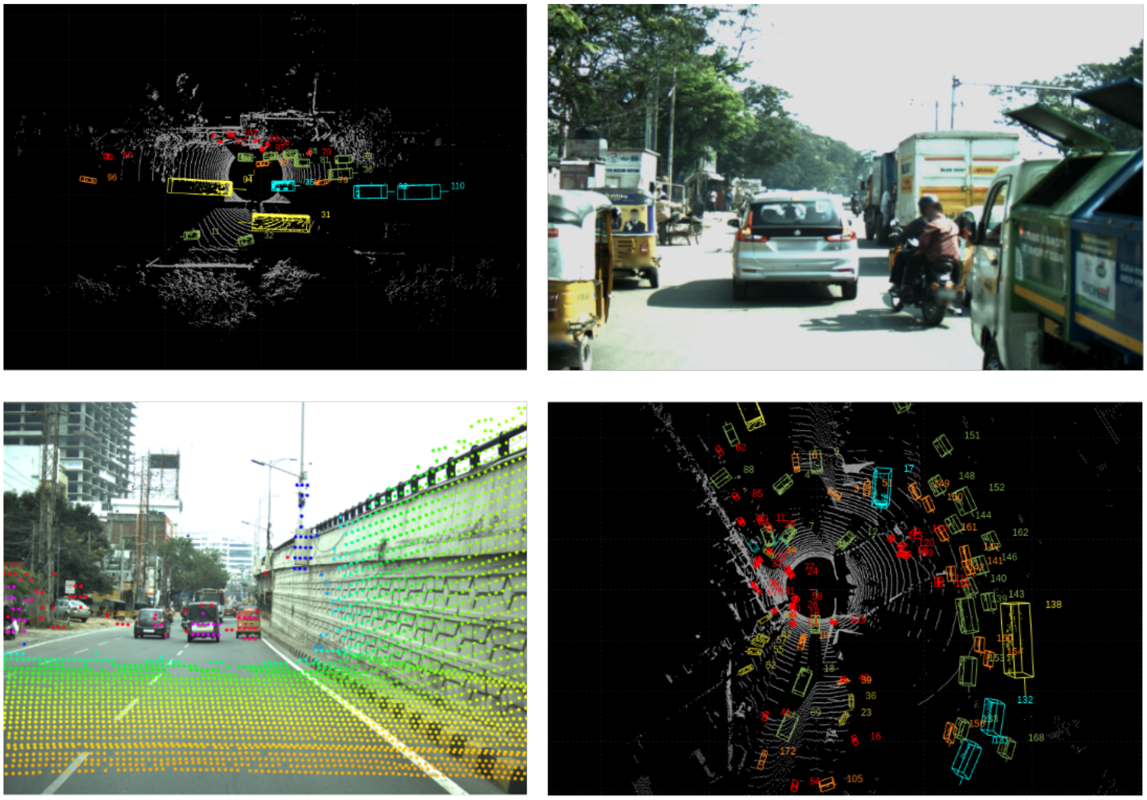


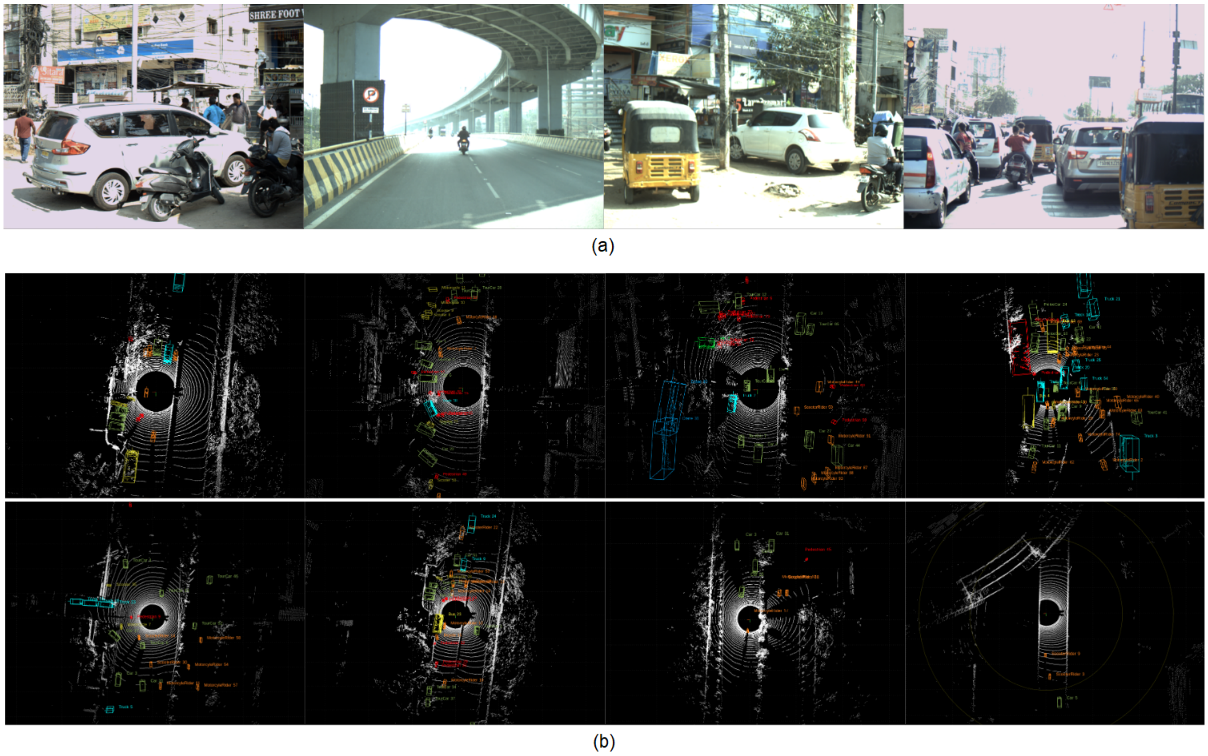
Figure 3.1 Some examples from the dataset showing different traffic scenarios, LiDAR data with annotations, and a sample of LiDAR point clouds projected on camera data.

Our main contributions can be summarised as follows: (i) We propose the IDD-3D dataset for driving in unstructured traffic scenarios for Indian roads with 3D information, (ii) high-quality annotations for 3D object bounding boxes with 9DoF data, and instance IDs to enable tracking, (iii) Analysis over highly unstructured and diverse environments to accentuate the usefulness of proposed dataset, and (iv) provide 3D object detection and tracking benchmarks across popular methods in literature.

3.2 Related Work

Data plays a huge role in machine learning systems, and in this context, for autonomous vehicles and scene perception. There have been several efforts over the years in this area to improve the state of datasets available and towards increasing the volumes of high-quality and well annotated datasets.

2D Driving: One of the early datasets towards visual perception and understanding driving has been the CamVid [7] and Cityscapes [23, 24] dataset, providing annotations for semantic segmentation and enabling research in deeper scene understanding at pixel-level. KITTI [33, 34] dataset provided 2D object annotations for detection and tracking along with segmentation data. However, fusion of multiple


(a)

(b)

Figure 3.2 Samples from the dataset highlighting different (a) RGB images and (b) LiDAR Bird-Eye-View (BEV) along with bounding box annotations. The samples visualized above are taken from different sequences of the dataset.

modalities such as 3D LiDAR data enhances the performance for scene understanding benchmarks as these provide a higher level of detail of a scene when combined with available 2D data. This multi-modal sensor-fusion based direction has been the motivation for the proposed dataset to alleviate the discrepancies in existing datasets for scene perception and autonomous driving.

Driving Datasets: Recent datasets such as nuScenes [10], Argoverse [15], Argoverse 2 [90] provide HD maps for road scenes. This allows for improved perception and planning capabilities and towards construction of better metrics for object detection such as in [79]. These large scale datasets cover a variety of scenes and traffic densities and have enabled systems with high safety regulations in the area of driver assistance and autonomous driving. However, the drawback for a majority of these datasets arises from the fact that the collection happens in well-developed cities with clear and structured traffic flows. The proposed dataset bridges the gaps of varying environments by introducing more complex environments and extending the diversity of driving datasets.

Complex environments: There have been multiple efforts to build datasets for difficult environments such as variations in extreme weather [63, 75], night-time driving conditions [25], and safety critical scenarios [5]. There have been recent works which make use of different sensors such as fisheye

Dataset	3D Scenes	Cameras	Lidar	Images	Classes	3D Boxes	Traffic Diversity
KITTI [34]	15k	2	yes	15k	3	80k	Low
nuScenes [10]	40k	6	yes	1.4M	23	1.4M	Mid
Apolloscape [38]	20k	6	yes	0	6	475k	Low
KAIST [19]	8.9k	2	yes	8.9k	3	0	Low
Waymo Open [79]	230k	5	yes	1M	4	12M	Mid
ONCE [55]	1M (16k)	7	yes	7M	5	417k	Mid
Cityscapes-3D [31]	20k	-	no	490k	8	-	Low
A* 3D [62]	39k	1	yes	39k	7	230k	Mid
Ours	20k*	6	yes	120k	10 (17**)	285k*	High

Table 3.1 A comparison with existing popular 3D autonomous driving datasets. Our dataset showcases the highest diversity with the highest average number of bounding boxes per frame and a wide distribution. The statistical distribution is further studied in the following sections. (*) Number reported on train-val-test set, experiments/statistics reported on train-val set. (**) The 17 classes are total of the 10 primary and 7 additional classes.

lenses to cover a larger area around the ego-vehicle [51, 96] and event camera [68] for training models with faster reaction times. However, most of these datasets have been collected in environments with little to no changes in the traffic patterns and consistency in the background objects. Some works in literature [43, 77, 82, 88] explore such situations where the label distributions can vary significantly, however these are either limited to mostly 2D modalities, or off-road environments. In this work, the proposed dataset enhances the availability of data for enabling research for autonomous driving in unconstrained traffic environments.

Object Detection and Tracking: Several popular methods have been explored in recent literature which handle the task of 3D object detection for the cases of driving scenarios [47, 93–95, 100]. In our work, we specifically talk about 3D object detection from point clouds, while we do note the effectiveness of multi-modal approaches as well [18, 66, 78]. We have used approaches such as SECOND [93] which voxelize the input point cloud and apply 3D convolution, which leads to discrete geometric representations of the data. CenterPoint [95] approach which assigns centers is known to perform well for smaller objects due to the fine level of details for each point feature. We also explore PointPillars [47] for an analysis of pillar based approaches where the data is projected to Bird-Eye-View mode and then treated as an image. We highlight the performance of each in the experiments section and draw our inferences specific to the proposed dataset.

Many methods have been proposed towards 3D Multi-Object Tracking (MOT) in literature which have been shown to perform well across a multitude of datasets in different scenarios. There are various ways to model the tracking task such as using the Bird-Eye View [52], approaches based on multi-sensor fusion [45], and simple tracking based on distance metrics and methods like Kalman filter [61]. In this work, we utilise the method presented in [61] using the detections from our trained models on IDD-3D and present the evaluations based on popular MOT metrics such as the ones presented in [87].

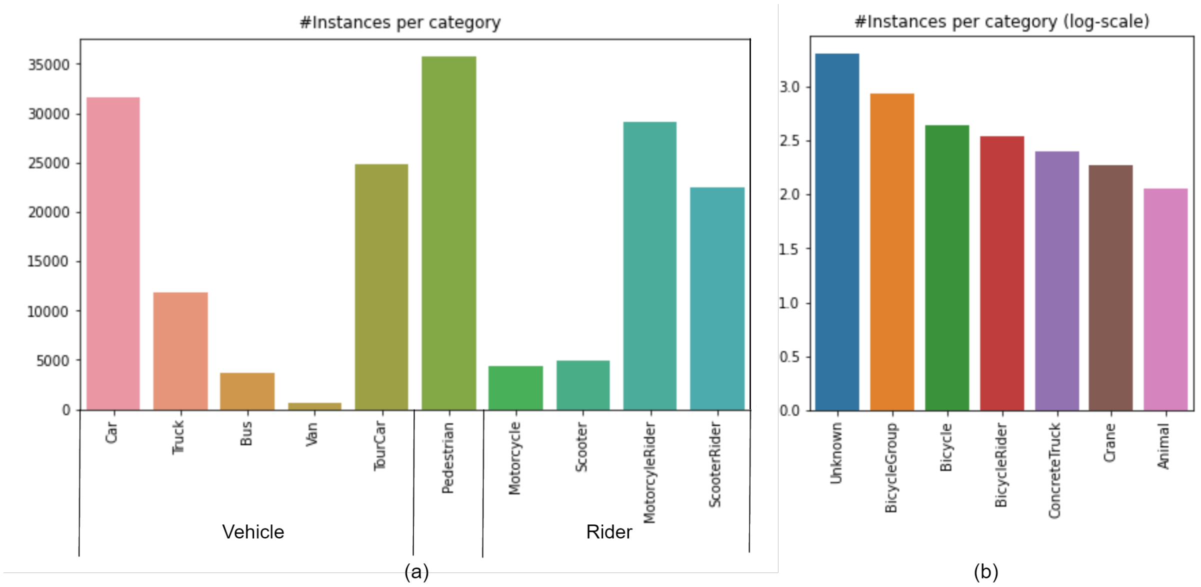


Figure 3.3 Distribution of class labels in the proposed dataset. (a) The primary 10 classes are shown here along with the 3 super-categories (Vehicle, Pedestrian, and Rider) which are considered to make the proposed dataset more consistent with labels from existing datasets. (b) The additional 7 classes annotated in the dataset are shown in log-scale separately since they are currently not used for training the models. *The Rider class covers both riders and non-riders on two-wheeler motor vehicles.* We do not consider the Miscellaneous classes for evaluation of the dataset currently.

3.3 Proposed Dataset

In the following sections we discuss and highlight the qualities of the proposed dataset, including the design choices and method for data collection, annotations and analysis of the dataset over interesting scenarios.

3.3.1 Data Acquisition

The data collection for the proposed dataset was covered in two driving sessions with over 5 hours of collected data during daytime. Afterwards, we manually sample scenes of interest in sequences of 100

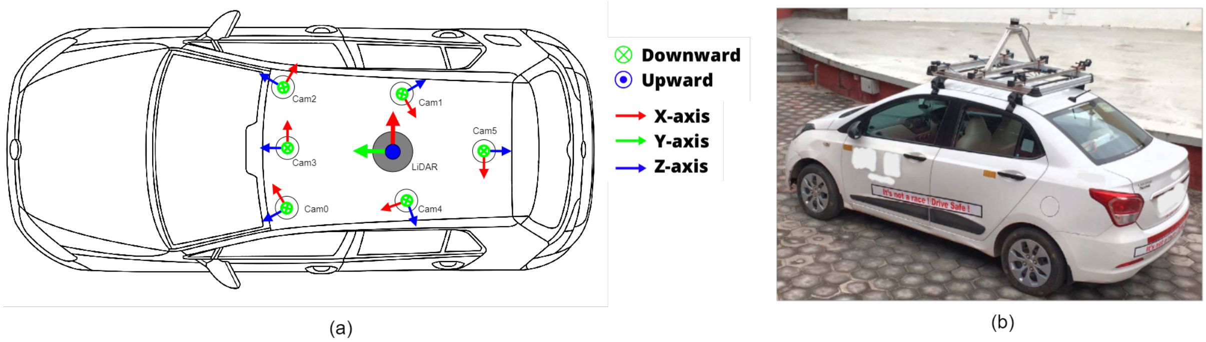


Figure 3.4 Figure showing (a) sensors on the vehicle (cameras, LiDAR) and their respective orientations, (b) image of the vehicle used along with the sensor rig. *Please note that the real-world car image has been edited to preserve anonymity.*

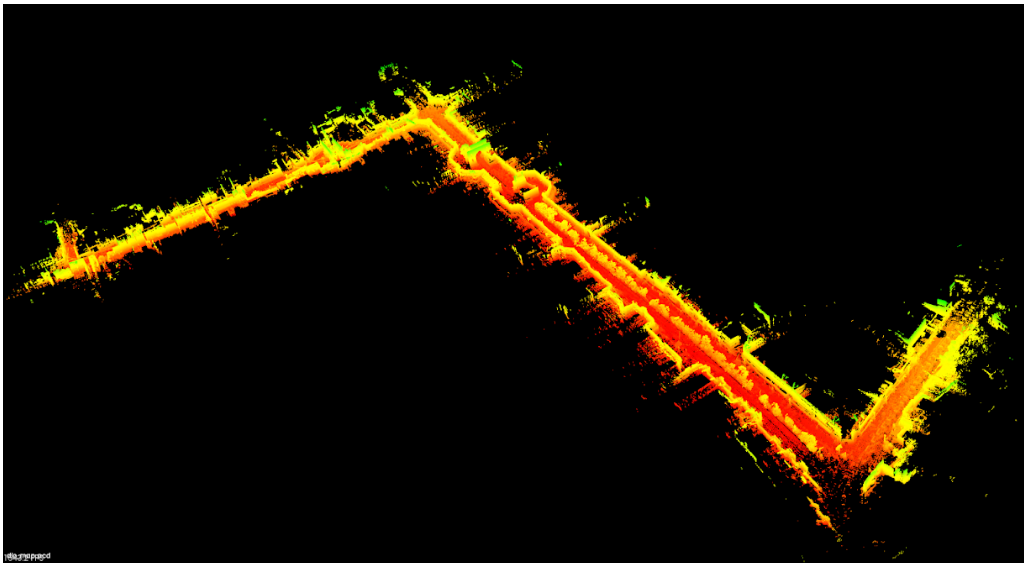
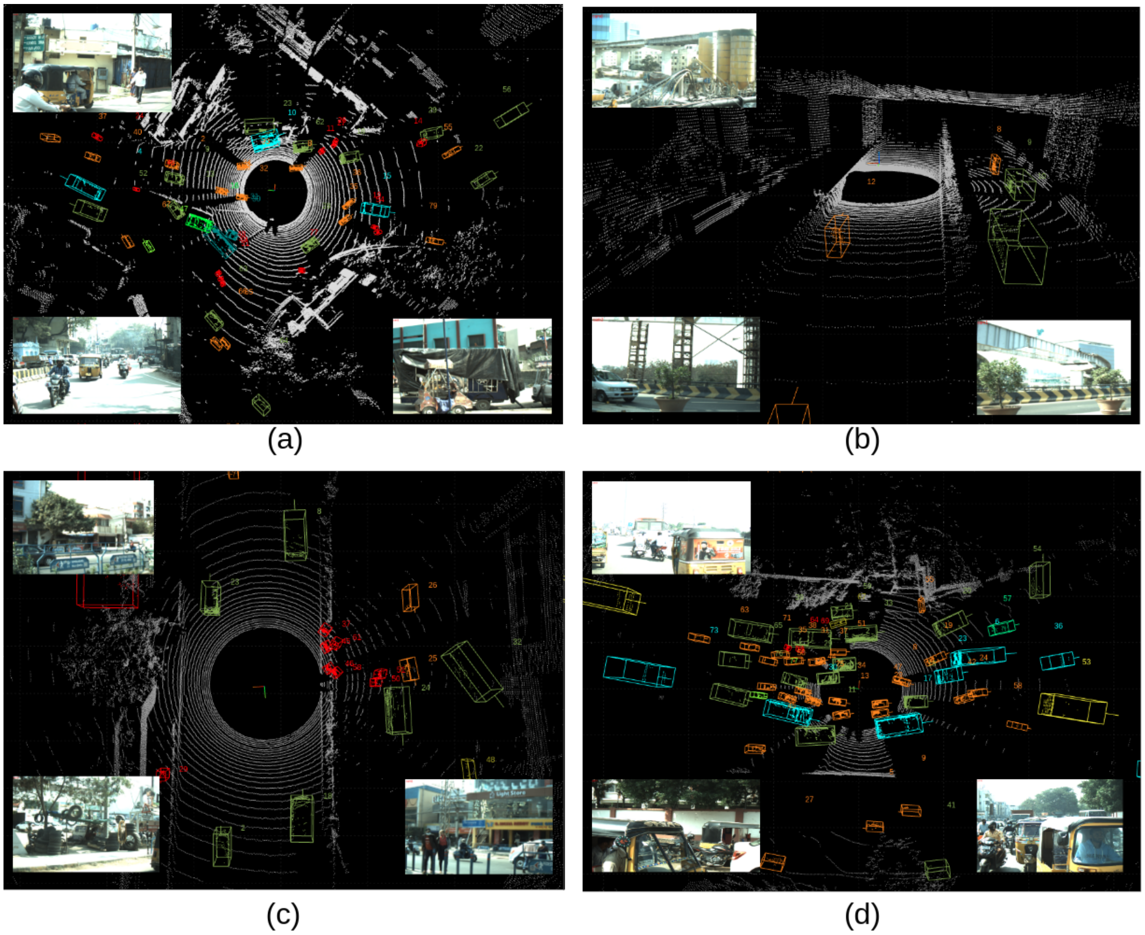


Figure 3.5 Generated trajectory from the LiDAR point clouds for one of the sequences of the dataset.

frames at 10fps making 150 sequences, each of 10s. The data collection has been performed in different regions of Hyderabad, India. We now provide details about the configuration and data preparation in the following. For visualization of the point clouds across a sample sequence of data collection, we perform point cloud registration and visual odometry as shown in fig. 3.5.

Sensors (Hardware configuration): The proposed dataset encompasses data from multiple sensors which include six RGB cameras and one LiDAR (Ouster OS1) sensor. The details about the sensors and data processing used have been highlighted in the previous chapter. The position and orientation of the sensors on the acquisition vehicle is shown in Fig. 3.4 along with the real-world image of the vehicle.



(a)

(b)

(c)

(d)

Figure 3.6 Samples of scenes of interest in our dataset (LiDAR and RGB samples) which especially differentiate our proposed dataset from those available in literature. *(Clockwise from top-left)* (a) Complex traffic scenarios with vehicles orientations in a wide variety of directions, (b) Perspective view of a scene with ego-vehicle on elevated flyover with ground level visible and another highway over the vehicle path with pillars, (c) humans in the middle of traffic (shown in red boxes) and jaywalking near moving vehicles, resulting in a safety critical scenario, (d) An example with very high density traffic scenario. Such case are abundant in the proposed dataset (rather than special cases when compared to other popular datasets) and hence require special attention for such unstructured environments.

Data Privacy: We ensure that all the faces and license plates in the dataset are blurred by first using automated approaches (such as [26,48]) and then performing a manual quality inspection. For the automated approaches, we run the object detection pipeline and then perform a NMS based matching to find any missing boxes in between frames. The missing boxes are interpolated, and finally, we blur the regions in the images for data protection.

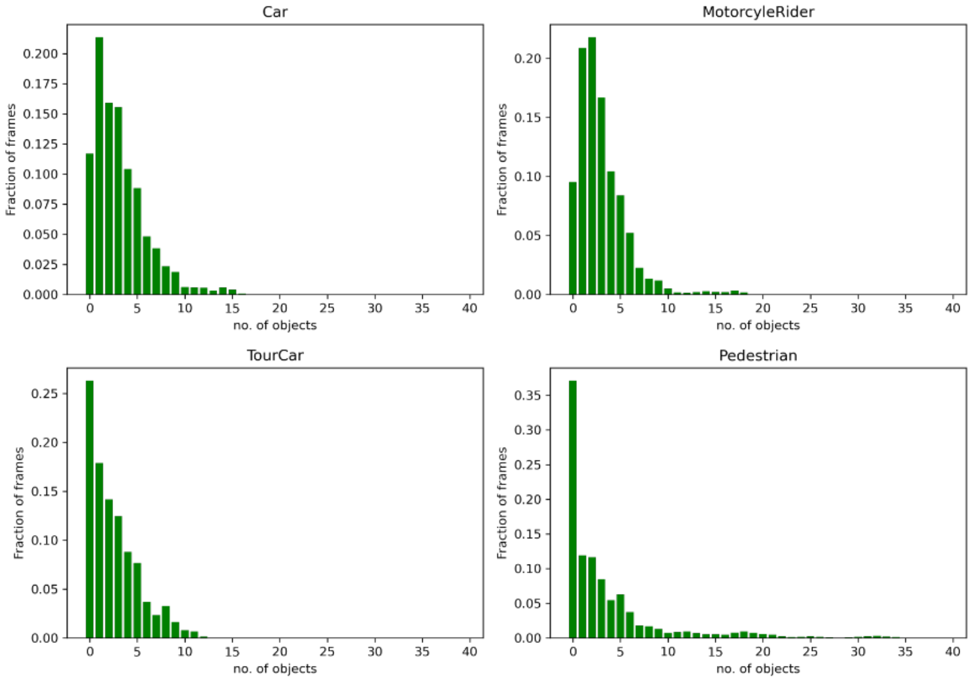
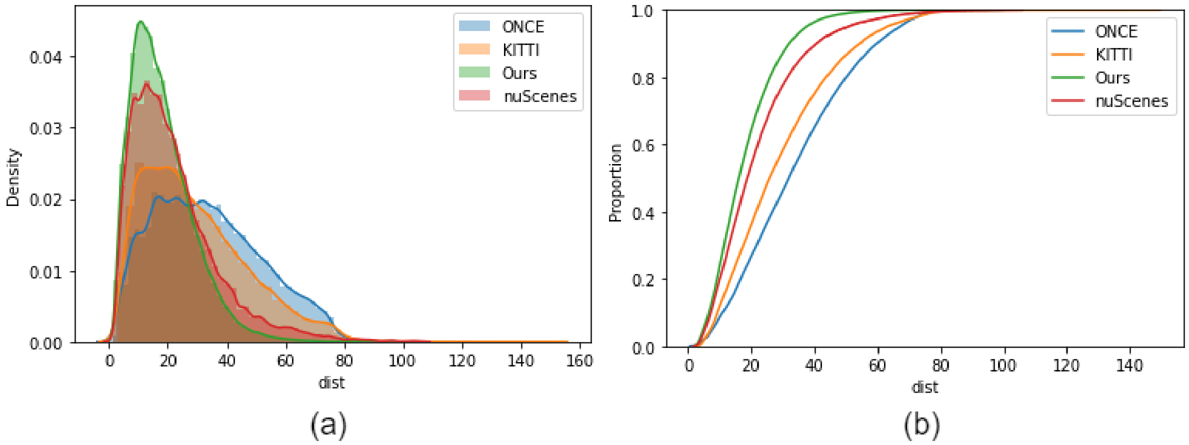


Figure 3.7 Class-wise distribution of some common prominent classes (Car, MotorcycleRider, Pedestrian, TourCar) with respect to number of frames is visualized to show traffic and crowd density in proposed dataset.

3.3.2 Dataset Analysis

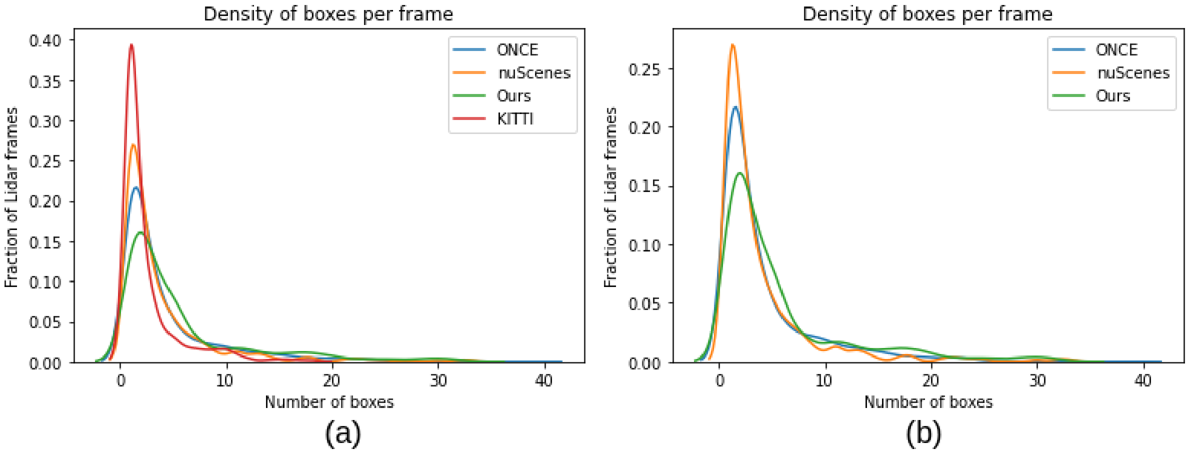
Labels and Annotations We provide 3D bounding box annotations for 15.5k (train-val-test) LiDAR frames with 223k 3D bounding boxes. We have used the annotation tool [49] for labeling data across 17 categories, shown in Fig. 3.3. Each object in a sequence contains a unique ID which enables tracking and re-identification. Furthermore, we provide class specific object distribution based on number of frames for some of the prominent categories in Fig. 3.7. We note that out of the 17 available classes (primary and additional), we are using 10 primary classes currently for training and validation is performed on 10 classes and 3 super-classes (Vehicle, Pedestrian, and Rider).

Data Statistics: We first highlight the bounding box distance distribution in IDD-3D and the comparison with existing popular datasets [10, 34, 55] in figure 3.8. In Fig. 3.8 (a) we show that IDD-3D consists of most of the annotations close to the ego-vehicle, caused by the low gaps between vehicles causing occlusion for LiDAR rays for longer distances. Nonetheless, it is crucial to highlight this feature of the proposed dataset because split-second decisions are important for safety, especially when other objects are close to the ego-vehicle. We also show better data density compared to KITTI, which is on a comparable scale to IDD-3D. Additionally, it can be seen that in the range of 0-25m (where most of the proposed dataset's annotations exist), we show higher densities than both ONCE and nuScenes as



(a) (b)

Figure 3.8 (a) Distribution showing distances of all bounding boxes from the ego-vehicle. The short distance of vehicles and pedestrians provides motivation for the proposed dataset to facilitate modeling of shorter reaction times. (b) Cumulative distribution of the distances further highlight the differences in distance distributions, showing that most of the objects in the proposed dataset are close to the ego-vehicle compared to existing popular datasets.



(a) (b)

Figure 3.9 Distributions of number of bounding boxes per LiDAR frame. The number of objects in a scene is usually higher in the frames present in the proposed dataset. We filter the boxes specifically based on the distance of less than 30m based on data shown in fig 3.7. (a) Shows statistics with KITTI dataset, and (b) shows the same without KITTI dataset to highlight the sparsity in the KITTI dataset. We note the heavier tail of our distribution indicating a greater density of objects close to the ego-vehicle.

shown in fig. 3.9(b). Additional statistic visualizations for number of bounding boxes per category and the distrbution of distances from ego-vehicle for each category are provided in fig. 3.11 and fig. 3.10, respectively.

Interesting cases: While existing datasets provide high diversity in type of traffic scenarios, these are usually restricted to controlled and well-structured environments with only a few anomalies. In IDD-3D, we show a large amount of diversity in the situations and also highlight some cases which could be of interest for progress in driving behaviour modeling such as the samples shown in Fig. 3.6. For example, we see safety critical cases where multiple pedestrians are seen jaywalking while vehicles are on the roads. Existing datasets claim high density traffic when there are 20-30 object bounding boxes in one frame, whereas in our samples we show 50-60 or more objects existing in the same frame, and in close proximity. Additional variations in the interesting scenes are provided in fig. 3.12. Considering the different variations of scenes in the proposed dataset, the applications for surveillance, road-safety, traffic quality, and crowd-behaviour are immense and show potential to be disparate from the data patterns from other datasets.

3.4 Experiments and Benchmarks

We present an extensive analysis of IDD-3D with existing methods to highlight the diversity and usefulness data. We first discuss the experimental setup and then based on the evaluations, report the understanding about the dataset properties and behaviour of different approaches.

Proposed Dataset: We use 10 primary categories which are highlighted in Fig. 3.3, however, since most datasets in literature ordinarily provide a few categories as common labels (For example, Car, truck, Van as Vehicle), we combine our class labels into three categories, namely Vehicle, Pedestrian, and Rider as super-categories. The network architectures are trained on 10 categories (Car, Bus, Truck, Scooter, Van, Motorcycle, Pedestrian, MotorcycleRider, ScooterRider, TourCar). We transform the annotations to a simpler format for the 3D object detection task a 7-dimensional vector as $(x, y, z, w, h, l, \alpha)$, where (x, y, z) represent the object location, (w, h, l) represent the dimensions of the bounding box and α represents the yaw angle.

3D Object detection: We discuss about some of the popular datasets which have been considered for comparison with the proposed dataset and highlight their strengths and weaknesses in the complex setting of the presented driving scenarios. For fair comparison, we train network architectures proposed in [47,93,95] for 3D object detection and show the results in Tables 3.2, 3.3 and 3.4. We report the mAP scores for the 3 combined categories (Vehicle, Pedestrian, and Rider) in Tables 3.3 and 3.4, and further report mAP scores in four sub-levels, i.e. overall AP score for each training class in Table 3.2. The scores reported in Table 3.2 are for a distance up to 30m in the dataset, and the distances in the super-classes are divided as upto 30m (denoted as Overall), 0-10m, 10-25m, and 25+m. The small distance buckets are considered due to the data distribution (as shown in Fig. 3.8) in the proposed dataset.

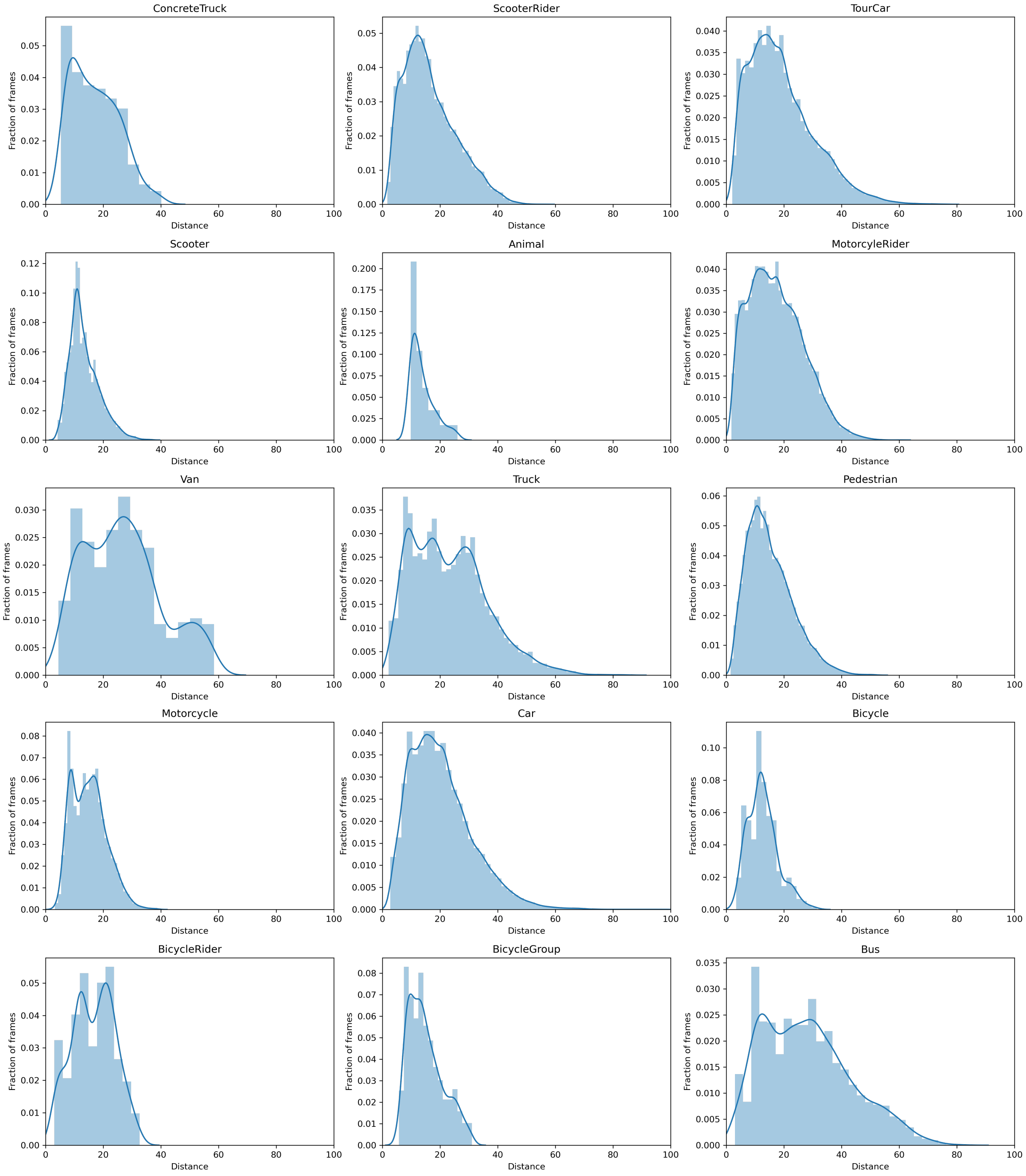


Figure 3.10 Distribution of distances of the annotated bounding boxes with respect to the fraction of frames in the dataset. The plots are category specific in the above figures.

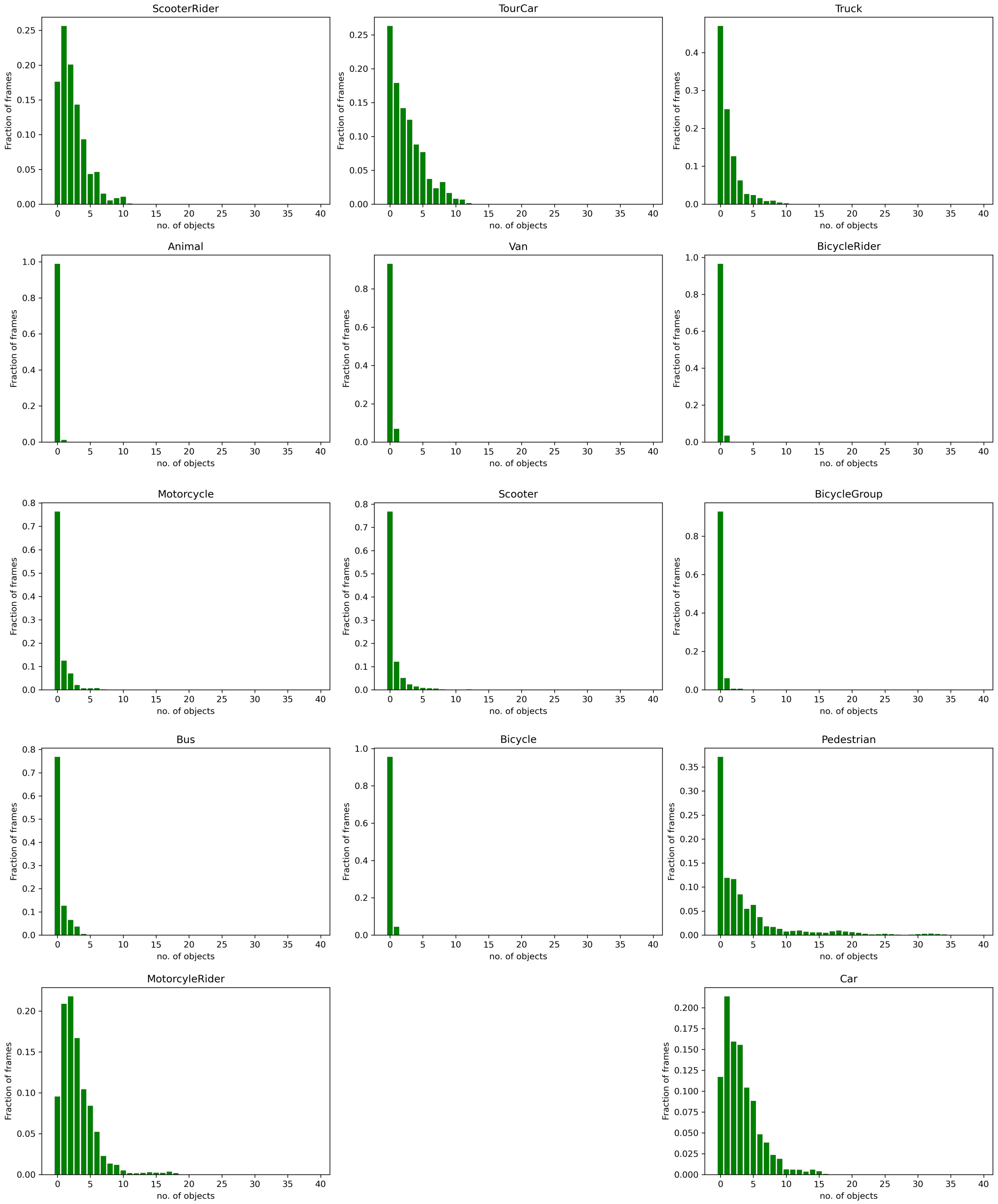


Figure 3.11 Distribution of bounding boxes annotated for each category in the dataset and the densities for number of boxes with respect to the fraction of frames in the dataset.

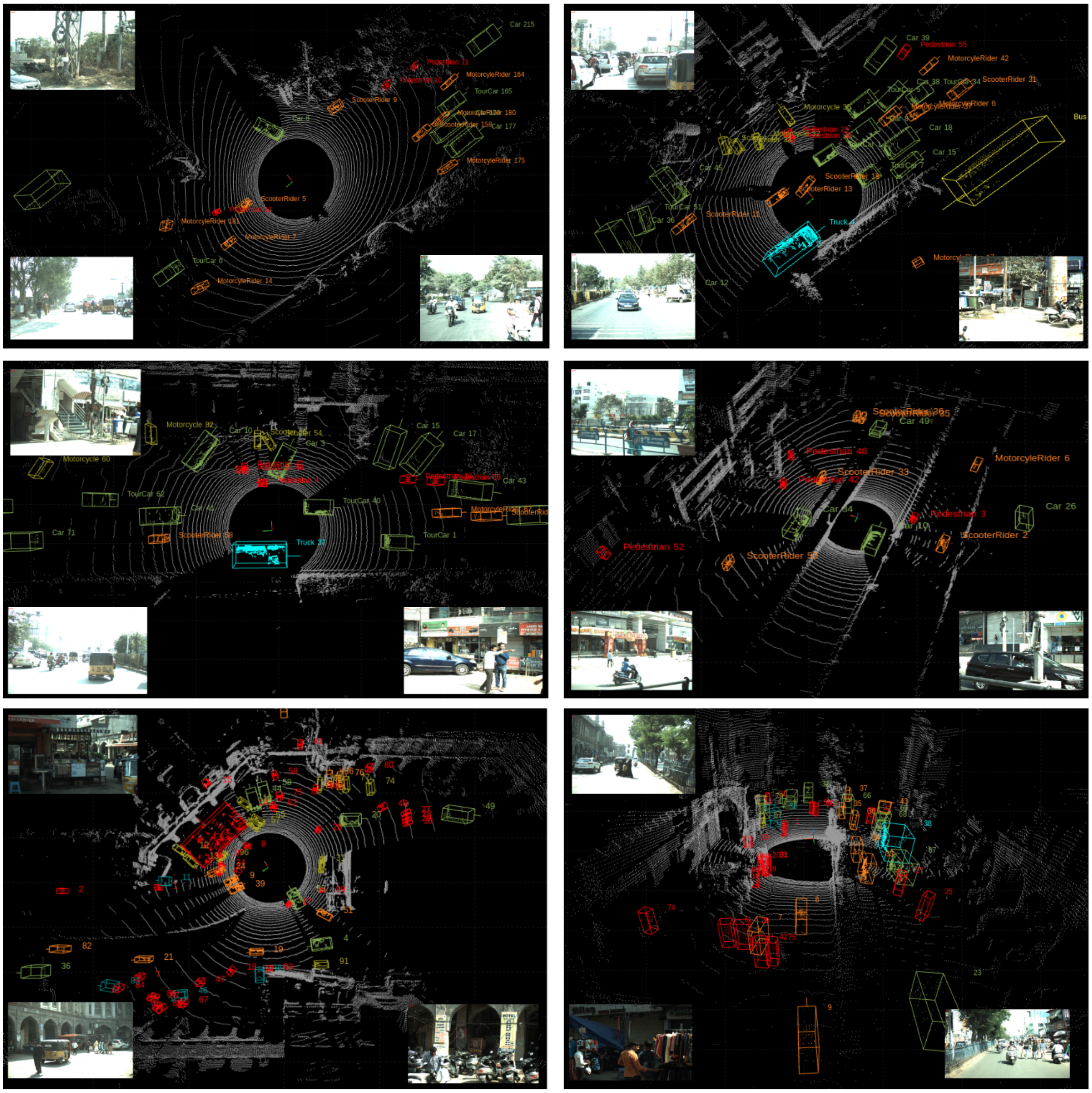


Figure 3.12 Some examples from the dataset showing different traffic scenarios, LiDAR data with annotations, and a sample of LiDAR point clouds projected on camera data.

3D Object Tracking: A notable property of the proposed dataset is the existence of the instance IDs for each 3D bounding box. In this work, we also show results on 3D object tracking and report important metrics such as AMOTA, AMOTP [87] in Table 3.5. We use SimpleTrack [61] for the task of object tracking and report the results based on the detections from Centerpoint [95] due to the highest mAP score on the detection task. The MOT scores are reported for all 10 primary classes and the overall categories.

SuperCategory	Categories/Methods	CenterPoint	CenterPoint (nuScenes)	SECOND	SECOND (KITTI)	PointPillar
Vehicle	Car	65.28	66.97	**69.89**	68.50	67.77
	Bus	59.09	**78.47**	59.12	49.69	43.70
	Truck	68.79	**72.18**	65.11	68.09	63.68
	Van	9.58	12.71	1.27	**15.77**	0.14
	TourCar	76.94	**77.40**	74.81	77.02	72.80
Pedestrian	Pedestrian	**28.60**	22.49	19.54	23.74	22.72
Rider	Motorcycle	23.65	**25.28**	21.69	22.79	16.97
	Scooter	**42.36**	38.05	26.98	23.73	16.81
	MotorcycleRider	59.29	**61.48**	53.39	48.90	46.52
	ScooterRider	**66.33**	64.65	52.27	50.62	41.60
	mAP	49.99	**51.97**	44.31	44.89	39.27

Table 3.2 Results on IDD-3D with popular methods. We report AP scores across different categories on the validation set. This table shows the results on each training class. The scores are reported with different thresholds for each class (Vehicles @ 0.5, Rider @ 0.4, Pedestrian @ 0.3) and all objects are considered till 30m distance.

Datasets: We use KITTI [33, 34] dataset and nuScenes [10] for pre-training of 3D object detection methods to further fine-tune on our proposed dataset. We note that cross-dataset training may not be fruitful in this scenario given the significantly different distribution of the categories and input data in the given datasets. The existing datasets usually utilise information such as LiDAR intensity, elongation, and timestamp information as input to the model, which is different from the proposed dataset. However, considering the wide research available based on these datasets, it is imperative that we highlight how using the existing models trained on these datasets as pre-training backbones usually enhances the performances. For this purpose, we consider using the models [93, 95] for pre-training by using the weights for the common layers and fine-tune for better performance.

Approach	Pre-Training	Vehicle				Rider			
		Overall	0-10m	10-25m	>25m	Overall	0-10m	10-25m	>25m
CenterPoint	nuScenes	**73.85**	87.57	**70.98**	**30.48**	71.03	84.24	69.54	23.42
CenterPoint	-	71.20	**88.84**	67.62	26.32	69.51	83.66	67.49	19.76
SECOND	KITTI	72.51	88.60	68.99	28.07	71.60	83.25	**70.98**	24.32
SECOND	-	73.01	88.71	67.82	29.46	**72.05**	**85.44**	70.89	**26.28**
PointPillar	-	68.61	87.64	64.59	26.30	69.66	82.56	68.60	25.64

Table 3.3 Experimental results on proposed dataset with different popular methods. We report AP scores across different categories on the validation set. This table shows the results on Vehicle and Rider categories from the proposed dataset.

Approach	Pre-Training	Pedestrian				mAP			
		Overall	0-10m	10-25m	>25m	Overall	0-10m	10-25m	>25m
CenterPoint	nuScenes	22.49	33.85	19.47	4.48	55.79	68.56	53.33	19.46
CenterPoint	-	**28.60**	**44.89**	**24.39**	3.48	**56.43**	**72.46**	53.17	16.52
SECOND	KITTI	23.74	33.67	21.05	5.58	55.95	68.51	**53.67**	19.32
SECOND	-	19.54	27.18	17.61	**6.44**	54.87	67.11	52.11	**20.73**
PointPillar	-	22.72	29.34	20.45	5.45	53.66	66.52	51.21	19.13

Table 3.4 Experimental results (continued) on proposed dataset with different popular methods. We report AP scores across different categories on the validation set. This table shows the results on Pedestrian category and the mAP score from the proposed dataset.

Result Analysis (3D Object Detection): We note that the performance of the architectures for both 10 categories and the 3 super-categories is consistent and aligns with our claims. It is clear that the number of annotated instances plays a major role for better mAP scores, for example, classes such as Car achieve a high mAP compared to classes such as Van or Scooter. Another major factor appears to be the object size, wherein larger and denser objects are easier to model and detect compared to smaller instances. An example of the variations in mAP scores based on sizes is the differences between the Pedestrian and Bus/Truck categories, even though Pedestrian category consists of the maximum bounding box instances. From Table 3.2, we see that CenterPoint approach generally performs better than SECOND or PointPillars for the proposed dataset, this could be due to the nature of the approach where it deals directly with point clouds to predict object centers instead of voxelizing the points (SECOND) or projecting the point to BEV (PointPillars).

Category	AMOTA	AMOTP	Recall	MOTAR	MOTP	MOTA	lgd	tid	faf
Bus	0.831	0.679	0.812	0.907	0.589	0.736	3.045	2.659	13.805
Car	0.641	0.726	0.667	0.787	0.518	0.521	3.422	2.035	44.806
Motorcycle	0.202	0.826	0.242	0.941	0.356	0.228	2.000	2.000	2.321
MotorcyleRider	0.507	0.735	0.496	0.801	0.320	0.390	5.027	2.585	36.410
Pedestrian	0.254	0.912	0.319	0.737	0.363	0.225	9.918	6.731	34.557
Scooter	0.250	0.494	0.323	1.000	0.092	0.323	0.000	0.000	0.000
ScooterRider	0.540	0.536	0.581	0.742	0.258	0.427	3.868	2.274	35.251
TourCar	0.796	0.433	0.848	0.821	0.351	0.692	2.877	1.034	48.866
Truck	0.701	0.635	0.675	0.903	0.403	0.607	5.108	2.676	17.796
Van	0.000	1.677	0.275	0.000	0.563	0.000	14.500	0.000	75.163
Overall	0.472	0.765	0.524	0.764	0.381	0.415	4.977	2.199	30.898

Table 3.5 Experimental results for 3D object tracking for the 10 primary classes present in the proposed dataset. We use SimpleTrack [61] for the task of tracking using detections from CenterPoint [95] in the presented table.

In continuation of the results reported in Tables 3.2, 3.3 and 3.4, we show the expansion of results across all categories on each distance bucket for the models prepared in Table 3.6. While we still arrive at the conclusion that CenterPoint provides better mAP scores on the maximum cases, we observe that CenterPoint approach performs better for objects which are closer to the ego-vehicle and usually perform worse than other methods for the distance buckets which are far. This could be attributed to the fact that point cloud density per object decreases as we move far and that affects CenterPoint approach since it follows prediction of centers for each point for object detection.

Another interesting observation is that in SECOND architecture, we see better performance on categories which won't get affected significantly when voxelized such as Cars and Buses. When the objects in Pedestrian category are voxelized, a significant amount of low-level information may be lost making the model prone to more errors. Hence, the performance gap in SECOND compared to both CenterPoint and PointPillars.

Result Analysis (3D Object Tracking): For the object tracking results presented in Table 3.5, we notice a correlation between the detection scores and tracking scores (AP and AMOTA/MOTA) for classes such as Pedestrian and Car. We highlight that the detection as well as tracking models perform adequately on the proposed dataset achieving an overall AMOTA score of 0.472 (higher better), while we also note that a similar configuration achieves an overall AMOTA of 0.668 on the nuscenes dataset

Category / Method	Distance	CenterPoint	CenterPoint (nuScenes)	SECOND	SECOND (KITTI)	PointPillar
Car	Overall	65.28	66.97	**68.89**	68.50	67.77
	0-10m	81.75	77.59	**84.79**	84.62	83.86
	10-25m	64.45	66.36	67.32	**67.94**	67.49
	>25m	18.14	23.15	25.07	23.94	**26.17**
Bus	Overall	59.09	**78.47**	59.12	49.69	43.70
	0-10m	76.55	**88.42**	82.43	67.41	54.83
	10-25m	60.09	**80.58**	56.22	47.84	43.10
	>25m	24.04	**32.24**	22.94	16.22	11.89
Truck	Overall	68.79	**72.18**	65.11	68.09	63.68
	0-10m	88.87	92.04	84.43	**93.44**	88.43
	10-25m	60.65	**66.22**	53.98	58.77	53.84
	>25m	23.43	23.63	**28.60**	24.16	24.96
Van	Overall	9.58	12.71	1.27	**15.77**	0.14
	0-10m	0.0	0.0	0.00	0.00	0.00
	10-25m	12.99	14.36	2.40	**19.85**	0.33
	>25m	0.0	0.0	0.00	0.00	0.00
TourCar	Overall	76.94	**77.40**	74.81	77.02	72.80
	0-10m	**88.94**	87.63	86.17	88.52	85.93
	10-25m	76.38	**77.17**	74.90	74.89	70.63
	>25m	33.85	40.44	**40.86**	42.69	39.37
Pedestrian	Overall	**28.60**	22.49	19.54	23.74	22.72
	0-10m	**44.89**	33.85	27.18	33.67	29.34
	10-25m	**24.39**	19.47	17.61	21.05	20.45
	>25m	3.48	4.48	**6.44**	5.58	5.45
Motorcycle	Overall	23.65	**25.28**	21.69	22.79	16.97
	0-10m	45.04	**47.28**	33.63	36.05	14.43
	10-25m	17.18	19.55	19.39	**20.19**	18.86
	>25m	4.72	**6.13**	3.54	3.48	3.56
Scooter	Overall	**42.36**	38.05	26.98	23.73	16.81
	0-10m	**40.39**	24.22	12.74	0.79	0.25
	10-25m	**42.75**	39.81	30.81	29.99	22.05
	>25m	**1.05**	0.51	1.00	0.00	0.00
MotorCycleRider	Overall	59.29	**61.48**	53.39	48.90	46.52
	0-10m	78.00	**79.40**	66.66	63.73	60.30
	10-25m	55.49	**57.88**	49.49	45.22	42.44
	>25m	12.66	**15.73**	13.11	11.26	12.09
ScooterRider	Overall	**66.33**	64.65	52.27	50.62	41.60
	0-10m	**76.36**	74.07	59.03	58.90	37.22
	10-25m	**68.18**	66.72	55.79	53.30	47.97
	>25m	14.62	**16.20**	8.88	7.39	10.22
mAP	Overall	49.99	**51.97**	44.31	44.89	39.27
	0-10m	**62.08**	60.45	53.71	52.71	45.46
	10-25m	48.26	**50.81**	42.79	43.90	38.72
	>25m	13.60	**16.25**	15.04	13.47	13.37

Table 3.6 Experimental results on proposed dataset with different popular methods. We report AP scores across different categories on the validation set. This table shows the results on all training categories with all distance buckets. The ’overall’ distance metric ranges from 0-30m and is considered based on the distance distribution of objects present in the scene.

(from the leaderboard). The complexity of the proposed dataset is especially highlighted in the results of the Pedestrian class, where the low scores prove complex motion present in the dataset.

Category	AMOTA	AMOTP	Recall	MOTAR	MOTA	MOTP	lgd	tid	faf
Bus	0.775	0.887	0.825	0.822	0.675	0.691	4.380	2.960	27.190
Car	0.641	0.775	0.691	0.766	0.525	0.558	3.373	2.115	51.056
Motorcycle	0.166	1.035	0.231	0.981	0.227	0.324	3.375	3.125	0.725
MotorcyleRider	0.480	0.730	0.520	0.759	0.383	0.337	4.781	2.204	45.556
Pedestrian	0.281	0.851	0.356	0.726	0.248	0.369	9.304	5.373	40.096
Scooter	0.383	0.447	0.361	1.000	0.361	0.122	2.750	2.750	0.000
ScooterRider	0.575	0.570	0.540	0.887	0.474	0.298	3.286	2.214	14.520
TourCar	0.780	0.443	0.808	0.840	0.673	0.350	3.848	1.201	43.006
Truck	0.671	0.634	0.730	0.760	0.553	0.451	4.628	2.395	42.224
Van	0.000	1.753	0.275	0.000	0.000	0.763	14.500	0.000	52.381
Overall	0.475	0.812	0.534	0.754	0.412	0.426	5.422	2.434	31.675

Table 3.7 Tracking (3D-MOT) results on the proposed dataset for Centerpoints method only trained on the proposed dataset.

Category	AMOTA	AMOTP	Recall	MOTAR	MOTA	MOTP	lgd	tid	faf
Bus	0.655	1.037	0.672	0.919	0.614	0.683	8.708	7.604	10.045
Car	0.607	0.891	0.668	0.778	0.515	0.564	3.738	2.040	45.625
Motorcycle	0.214	1.305	0.237	0.874	0.206	0.317	5.750	4.417	4.671
MotorcyleRider	0.429	0.992	0.420	0.825	0.339	0.323	6.199	3.029	26.982
Pedestrian	0.379	0.870	0.406	0.763	0.304	0.366	7.136	4.907	40.123
Scooter	0.285	0.991	0.323	1.000	0.323	0.110	0.000	0.000	0.000
ScooterRider	0.447	1.070	0.461	0.838	0.379	0.273	8.667	5.009	17.997
TourCar	0.725	0.619	0.714	0.887	0.628	0.333	6.294	2.825	27.187
Truck	0.633	0.758	0.670	0.822	0.550	0.427	4.368	2.763	31.275
Van	0.000	1.840	0.175	0.000	0.000	0.720	16.500	0.000	103.460
Overall	0.437	1.037	0.474	0.771	0.386	0.412	6.736	3.259	30.736

Table 3.8 Tracking (3D-MOT) results on the proposed dataset for SECOND without any pre-training.

The results for 3D-MOT (Multi-Object Tracking) for the other detectors are shown in Tables 3.7, 3.9, 3.8, and 3.10. We notice a lower performance in the Van category due to the low frequency of occurrence

Category	AMOTA	AMOTP	Recall	MOTAR	MOTA	MOTP	lgd	tid	faf
Bus	0.722	0.856	0.716	0.909	0.649	0.577	7.771	6.688	11.765
Car	0.597	0.917	0.668	0.750	0.495	0.567	4.023	2.004	51.263
Motorcycle	0.164	1.301	0.215	0.903	0.194	0.324	3.500	3.500	3.327
MotorcyleRider	0.385	1.037	0.457	0.697	0.305	0.338	6.493	3.421	49.066
Pedestrian	0.350	0.863	0.398	0.687	0.268	0.355	7.507	5.035	51.287
Scooter	0.250	1.212	0.323	1.000	0.323	0.100	0.000	0.000	0.000
ScooterRider	0.419	1.107	0.435	0.858	0.370	0.260	8.769	4.962	14.952
TourCar	0.751	0.560	0.733	0.898	0.655	0.307	5.441	2.495	24.896
Truck	0.630	0.766	0.644	0.829	0.531	0.414	5.263	2.776	29.618
Van	0.000	1.665	0.275	0.000	0.000	0.513	14.500	0.000	66.942
Overall	0.427	1.028	0.486	0.753	0.379	0.375	6.327	3.088	30.312

Table 3.9 Tracking (3D-MOT) results on the proposed dataset for SECOND method pre-trained on the KITTI dataset.

Category	AMOTA	AMOTP	Recall	MOTAR	MOTA	MOTP	lgd	tid	faf
Bus	0.663	0.884	0.677	0.948	0.640	0.582	8.854	7.229	6.729
Car	0.585	0.911	0.641	0.761	0.484	0.565	4.168	2.475	46.987
Motorcycle	0.108	1.307	0.152	0.986	0.149	0.285	2.000	0.500	0.362
MotorcyleRider	0.338	1.097	0.407	0.705	0.275	0.367	6.960	3.337	43.115
Pedestrian	0.326	0.896	0.320	0.811	0.256	0.311	7.305	4.229	25.784
Scooter	0.250	1.277	0.323	1.000	0.323	0.118	0.000	0.000	0.000
ScooterRider	0.356	1.154	0.341	0.845	0.285	0.245	10.757	6.486	12.747
TourCar	0.724	0.618	0.737	0.876	0.639	0.334	5.200	2.470	30.785
Truck	0.561	0.919	0.569	0.847	0.479	0.410	9.029	5.676	23.003
Van	-	-	-	-	-	-	-	-	-
Overall	0.391	1.106	0.417	0.778	0.353	0.522	7.427	5.240	68.951

Table 3.10 Tracking (3D-MOT) results on the proposed dataset for the Pointpillar method.

of the class in the dataset. We also observe the differences in the models based on the AMOTA and AMOTP scores. While the tracking method used for all the tables has been the same (SimpleTrack), we notice some differences in category specific performance in some of the models. For example, for the Pedestrian category, while the CenterPoint approach shows higher AP score compared to SECOND, we see that the SECOND approach reports better tracking results. This could be attributed to the fact that SECOND reports more false postive bounding boxes for the Pedestrian class, and due to the strict NMS (Non-maximal supression) threshold in the SimpleTrack, most of these are either removed of stabilized across frames, hence resulting in a minor improvement in performance. However, the overall AMOTA score for SECOND is still lower that CenterPoints due to the performance degradation in other categories. This is majorly due to the detection performance that objects with sparser points are not handled well with the SECOND approach.

We also note that the Van category in the PointPillars approach has been removed but still contributes to the result average. The category reports ”NaN” performances due to the lack of required number of predictions and hence did not get allocated to any predicted tracklets. Furthermore, we observe the number of false alarms per frame (Faf) is the lowest for centerpoints pre-trained with nuscenes dataset. The results from these popular approaches show that there is significant scope for improvement in the benchmarks present in the proposed dataset and that current approaches are not best suited for a general approach, especially in cases with variations in traffic density such as Indian road scenarios. Through this dataset, we hope to provide a step in the positive direction to bridge this gap.

Sure, here’s a draft for the new subsection:

3.5 Comparison with KITTI and nuScenes

In this section, we provide a comparative analysis between the proposed IDD-3D dataset and two of the most popular datasets in the field of autonomous driving, namely KITTI [33,34] and nuScenes [10].

(i) Contrasting IDD-3D: The KITTI and nuScenes datasets have been instrumental in the development of autonomous driving technologies, providing a diverse range of scenarios captured in Western countries. However, they lack the representation of the chaotic and unstructured driving scenarios often found in developing countries like India. This is where IDD-3D stands out. It introduces a new level of complexity with its high-density traffic scenarios, diverse vehicle categories, and unstructured road conditions. The dataset also includes a variety of safety-critical situations, such as jaywalking pedestrians and vehicles in close proximity, which are not commonly found in KITTI or nuScenes.

(ii) Areas of Improvement and Strengths: While IDD-3D provides a unique set of challenges, it also has areas where it can be improved. For instance, the dataset currently lacks experiments/benchmarks on sensor data like intensity, elongation, and timestamp information, which are present in KITTI and nuScenes. Although IDD-3D contains such information, incorporating such data in future versions of IDD-3D benchmarks could further enhance its utility. Despite these areas for improvement, the strengths of IDD-3D are undeniable. Its high-density, unstructured scenarios offer a unique testing ground for autonomous driving technologies. The dataset’s complexity and diversity can help in developing more robust and generalized models, capable of handling a wider range of real-world scenarios.

Through IDD-3D, we aim to bridge the gap between the structured driving scenarios of the West and the unstructured scenarios of developing countries. We believe that this dataset will play a crucial role in advancing the field of autonomous driving, particularly in improving the generalizability of models across different geographical locations and traffic conditions.

3.6 Summarizing Remarks

In this work, we presented IDD-3D, a dataset for unstructured driving scenarios with complex road situations is presented with thorough statistical and experimental analysis. Through this dataset, and the future release, we aim to solve the problem of generalizability across geographical locations and provide more diverse information in driving datasets and road scene analysis. We show interesting cases which cover a manifold of cases but also show some safety-critical situations which are frequent in several cities. We justify our claims for the proposed dataset through a set of experiments for 3D object detection and tracking using state-of-the-art approaches which were available as open-source implementations. The future works for the dataset shall extend these tasks to a vast number of applications, further enhancing the applicability of the proposed dataset to autonomous driving applications.

Chapter 4

Synthetics Data Generation: TRoVE Toolkit

4.1 Introduction

Computer vision applications, specifically autonomous driving systems, are constantly evolving owing to the rapid progress in the deep learning and machine vision community. At the core of such advancements lies the foundation created by high-quality, structured data, which augments the strengths of sophisticated architectures. While data acquisition and processing require expensive hardware setup and efforts to collect and process, there are several self-driving platforms which provide easier access to large volume of raw and processed data. However, data annotation still poses a huge challenge and is a resource extensive process. Even in the cases when it is feasible, the sheer number of possibilities in real-world diversity makes it unfathomable to observe all variations in object types, scenes, weathers, traffic densities, and sensor configurations. All these possibilities for variations create a near-insurmountable obstacle for dataset curation and annotations for self-driving and road scene scenarios. Use of synthetic data allows creation of such variations and extended diversity for a vast number of scenes, but requires expensive and expert manual efforts for simulations. In this chapter, we raise the question whether this abundant real-data can be used to automatically create synthetic datasets for training machine learning algorithms and be inclusive of large-variations while preserving real-world structural properties. Mimicking the physical properties of real-data in a synthetic pipeline helps towards minimizing domain gaps, while allowing to generate physically meaningful variations in scenes.

There have been several advancements in the preparation and utilization of synthetic datasets in recent times [28, 29, 32, 73, 76, 91]. While synthetically generated data may not be a complete substitute for real-world data yet, some works [32, 81, 83] discuss the usefulness of augmenting real-world data with synthetic datasets for improved performance . There have been significant improvements in approaches for domain adaptation [17, 37, 81], transfer of synthetic-to-real scenes and improvement in photo-realism [50, 71]. It becomes easier to add different environments, diversity in ambiance, lighting, weathers, and sensors using synthetic data. However, while the data synthesis through simulation is more straightforward than real-world dataset creation, it still requires a significant manual effort. Adding a new scene in such existing pipelines requires expertise to ensure a degree of photo-realism.

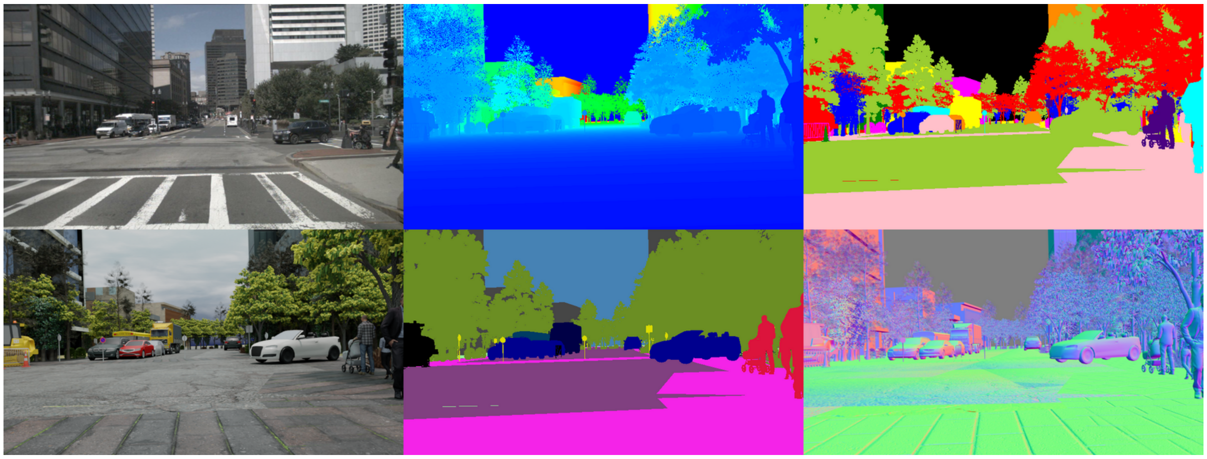


Figure 4.1 A sample of the scene generated with the proposed method. The first image shown is a real-world sample from nuScenes dataset. Using some existing annotation, we are able to generate extensions such as depth, instance segmentation, RGB images, semantic segmentation, and surface normals. These modalities represent a part of the capabilities of the proposed method.

Recent works highlight learning methods to generate novel trajectories and motion patterns [74,85,99]. Still, they may not necessarily delineate real-world behavior accurately, especially in complex scenes with varying crowd and traffic densities.

We propose an approach to model synthetic data based on real-world data distributions using available annotations and visual cues, mimicking real-world domain structure and enabling variations in a physically meaningful manner. Unlike most existing works, we show that using information from existing datasets for object placement and behavior can allow for fast construction of virtual environments while preserving the appeal of synthetic data generation systems for efficiency and diversity. We can use the same labels from a real scene to generate a diverse set of annotated data items from each scene (example shown in Figure 4.1 and 4.3) with diverse environmental conditions.

Our approach utilizes the location information from existing real scenes and visual cues available either as annotations or extracted from driving video sequences using currently available approaches in computer vision. We generate high-fidelity environment maps using geographic data available online and supplement this data with extracted cues from existing datasets and scenes. Our physically-based method of scene generation allows us to match different aspects of the scenes such as object positions, orientations, appearances, and ambient factors to recreate virtual environments that mimic real-world. The proposed approach is not restricted to manual design or hand-crafted environments, so it can be extended to virtually any location and complexity configuration for which visual cues can be automatically generated or already available.

A prime example of such a method is to extend existing datasets by utilizing the available annotation from the vast set of high-quality datasets [15,31,39,82] and generate multiple variations for each scene to increase the data volume and annotation modalities available. To support our claims, we outline our approach in the forthcoming sections and use the nuScenes dataset [10] as a base for the core set of experiments. An overview of our approach is shown in Figure 4.2. We use some parts of the already available annotations in nuScenes to generate new environments with traffic and pedestrian behavior similar to that available in the nuScenes dataset while varying the visual information to accommodate a diverse set of configurations. We show qualitative and quantitative analysis over segmentation tasks and compare validation metrics over popular datasets to outline the effectiveness of our data generation strategy for being physically consistent and adhering to real-world distributions. We highlight multiple modalities in our proposed dataset to enable various vision downstream tasks with different sensor configurations.

4.2 Related Work

Data acquisition and annotation for several downstream tasks can be challenging and resource-intensive. Especially for tasks like semantic segmentation, data preparation's cost and time estimate rise rapidly with data volume. There exist many real-world datasets in the community which targets specific problem statements such as vision tasks in indoor scenes [57], datasets with fine annotations for semantic segmentation and 3D object detection in outdoor scenes [2, 23, 24, 86], and 3D LiDAR point cloud segmentation in urban environments [6, 39, 51]. However, considering the cost of expensive annotations, it is often the case that some datasets only focus on specific modalities. For example, the nuScenes dataset [10] consists of high-quality annotations for object detection, tracking, trajectory prediction, LiDAR segmentation, and panoptic LiDAR segmentation but does not contain fine semantic segmentation annotations due to the sheer volume of available images.

Simulator-based methods: Synthetic datasets have been shown to improve performances across a variety of tasks including, but not limited to, object detection [44,58], trajectory prediction [11,99], depth estimation [4,56], semantic and instance segmentation [8,91], human pose estimation [83], object 6DoF pose estimation [44], 3D reconstruction [14], tracking and optical flow [92]. CARLA [29] is a popular simulator that relies on manually designed environment maps and places 3D object assets for vehicles, pedestrians, and dynamic entities in the environment. CARLA simulates different traffic conditions, variations in lighting, and some weather changes, which are rendered in a photo-realistic manner to provide significant overlap with real-world scenarios. The base version of the CARLA simulator provides a limited number of 3D city environments; different scenarios are simulated, and annotations are generated, which can be either exported to train deep learning models or utilized via their API to evaluate autonomous driving benchmark tasks. LGSVL Simulator [72] is a recent addition to the available simulation engines that delivers high-fidelity data for autonomous driving scenarios. LGSVL is built with an integration of Apollo Auto [1], which provides various features for interfacing with autonomous driving runtimes. The 3D environment is generated to mimic several real-world locations and integrate multiple sensor types, including RGB, Radar, LiDAR, which can be configured to behave like real-world sensors such as the Velodyne VLP-16 LiDAR. A recent simulation suite built on CARLA is the SUMMIT engine for urban traffic scenarios [11]. SUMMIT simulates complex and unregulated behavior in dense traffic environments and utilizes real-world maps to replicate difficult areas like roundabouts, highways, and intersection junctions. SUMMIT uses a context-aware behavior model, Context-GAMMA, an extension of GAMMA [53], to formulate agents' motion in complex environments for dynamic crowd behavior.

Non-simulation approaches: Several works in literature do not rely on simulation of the driving environment directly but provide structured fine annotations. A notable contribution to the community in this area is the Virtual KITTI dataset [32] which builds over the popular KITTI dataset [84] and extends the limited amount of annotated information by generating close-to-realistic images for digital-twins of sequences from the KITTI dataset. The Virtual KITTI dataset was recently extended in the Virtual KITTI 2 dataset [8], where the quality of images has been improved with a high definition render pipeline and the latest game engine (Unity 2018.4 LTS). The core approach in the Virtual KITTI dataset involves the acquisition of real-world data and measurements from the KITTI MOT benchmarks, then building a synthetic clone of the environments semi-automatically. Then, the objects of interest are placed in the scene manually, and lighting is adjusted to match the real-scene visually. There also exist datasets like SYNTHIA [73], that do not rely on any visual or geometric cues from real-world datasets and build novel virtual worlds to facilitate synthetic data generation. SYNTHIA provides generated annotations over 13 classes for pixel-level semantic segmentation and frames rendered from multiple view-points in the virtual environment. SYNTHIA dataset consists of large-scale annotations of up to 200k images across four-season settings in the form of video sequences and a random split of data with 13.4k images generated from randomly sampled camera locations across the synthetic map. While the volume and impact of the SYNTHIA dataset is prominent, the data generation process involves exorbitant manual effort. Furthermore, the dynamic entities in the scenes are also programmed manually to capture Spatio-temporal information between different vehicles and pedestrians.

Learning-based methods: Some recent approaches focus on imitation training [46] for iterative generation of more training data, especially for scenes where the model performs poorly. The work presented in [64] deals explicitly with the problem of domain gap in synthetic data by employing self-supervised learning over scene graphs to learn the scene layout and compare generated images with unlabelled images in the target domain. A recurring limitation observed in most datasets tends to be a lack of proper replication of real-world structures in an automated way without learning data or scene-specific layouts. The simulators can construct high-quality environments with variations in multiple factors. They, however, do not replicate the behavior of traffic entities from across a variety of locations around the world, and learned behaviors can only reproduce such motion and trajectory, which is reflective of the training data. In our proposed method for generating synthetic datasets, we leverage visual cues and pre-existing annotations from public datasets and enable the construction of large-scale scenes in virtual environments while mapping driver and pedestrian behavior from actual data into a virtual space.

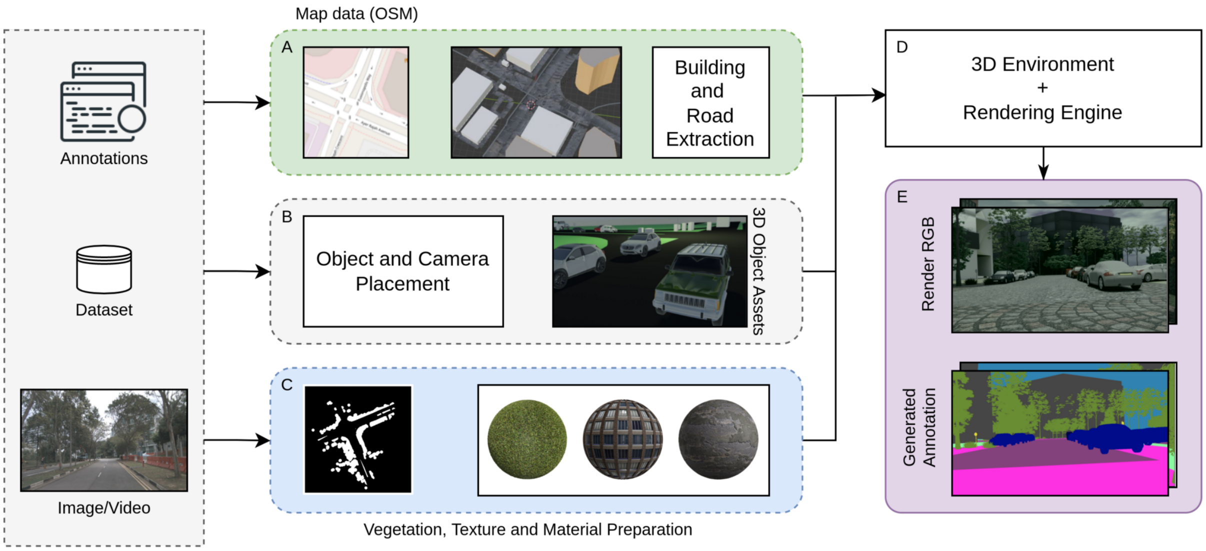

Map data (OSM)
A
Building and Road Extraction
D
3D Environment + Rendering Engine
Annotations
B
Object and Camera Placement
3D Object Assets
E
Render RGB
Dataset
C
Generated Annotation
Image/Video
Vegetation, Texture and Material Preparation

Figure 4.2 An overview of the synthetic data generation pipeline. Given real-world input dataset with annotations for object locations, we follow the process depicted in the above figure. (A) Map data extraction from OSM for building and road extraction. (B) Retrieval of categorical information and 3D object placement in scene with sampled camera poses. (C) Extraction of background data such as vegetation mask in world coordinates and PBR texture preparation. (D) Fusion of artifacts in the 3D environment and initialization of rendering process. (E) Generated RGB data and corresponding annotations stored for further use.

4.3 Our Approach

We describe the process for generating synthetic data automatically using either existing annotations from public datasets or visual cues from video sequences. We demonstrate the process with an example of the nuScenes dataset [10], but the same method can be applied to other datasets with available annotations [15, 51, 84]. The core components required to realize the data generation pipeline comprise of the geographic location and objects location/orientation in a given scene or video sequence. It is possible to construct a structured description of the scene and use it as configuration for the data generation process. To ensure diversity in object appearances, we use publicly available free 3D assets for different classes. The process of generating synthetic data using the proposed approach can be broken down into four major parts, which consist of (1) Building the world environment, (2) Placement of objects and camera in the scenes, (3) Applying textures and lighting information, and (4) Rendering and annotation processing. In the following subsections, we describe the steps in detail, referencing the overall process as shown in Figure 4.2, and samples from intermediate stages shown in Figure 4.4. All development of the 3D environment for our dataset is performed in Blender [21], an open-source 3D modeling and development software.

4.3.1 Building the Virtual Environment

Assuming the ego-vehicle to be the origin at each scene, we can estimate the geographic location using either GPS data (if available), or offset from scene geometry origin for which GPS information may be available.

Buildings: For generating the building placeholder, we refer to data from OpenStreetMaps (OSM) [59] through a blender add-on (blender-osm [65]). The meshes are generated without any texture initially. To extend variability in the scene, we choose buildings with an approximately rectangular layout and replace the mesh with a 3D building asset. The 3D assets used in this work were acquired from free-to-use resources on different forums such as [12, 40]. To check whether a building (say b) qualifies for replacement, we take the points (x, y) from the base plane of b as $\{(x, y)|b_z = 0\}$ and compute the edge length as well as orientations for the building base polygon. Let the edges be denoted by $e_{1,2} = d((x_1, y_1), (x_2, y_2)$ and the orientations be $\theta_{1,2} = \arctan((y_2 - y_1)/(x_2 - x_1))$, where d represents the euclidean distance function. We then compute a histogram of orientations, weighted by the respective edge lengths and select the pairs with close to $\pi/2$ difference. The selected pair with the highest edge weight (lengths) are then chosen to estimate the orientation of the rectangular building base. If no such edges and orientations are found, we assume a complex building outline and mark the building for applying facade texture in a later stage. Optionally, we can also compute the area of the polygon by projecting the base plane on a raster grid and taking ratio of area with the enclosing convex polygon. However, to avoid additional computations, we do not employ this approach in the current pipeline.

Roads: The road meshes are extracted from OSM map data as well via blender-osm. The road meshes are connected together to form a joint mesh object for the entire road network in the current context. The road width is adjusted to approximately match the road width available from real-world annotations (for nuScenes, extracted from the LiDAR point cloud).

4.3.2 Object and Camera Placement

We utilize the annotations available in the real-world datasets for extracting bounding boxes and camera poses for each scene and lay out the process to replicate the same structure in the virtual environment.

3D Object Placement: To assign a 3D asset to each bounding box, we estimate a *quality-of-fit* metric based on the Intersection over Union (IoU) for 3D bounding boxes. For a given object bounding box (say o_i) and a set of N 3D assets with corresponding bounding boxes (say $\{o_j|j \in 1, ..., N\}$) centered at the origin, we scale the asset box such that the largest dimension of the asset box o_j matches that of

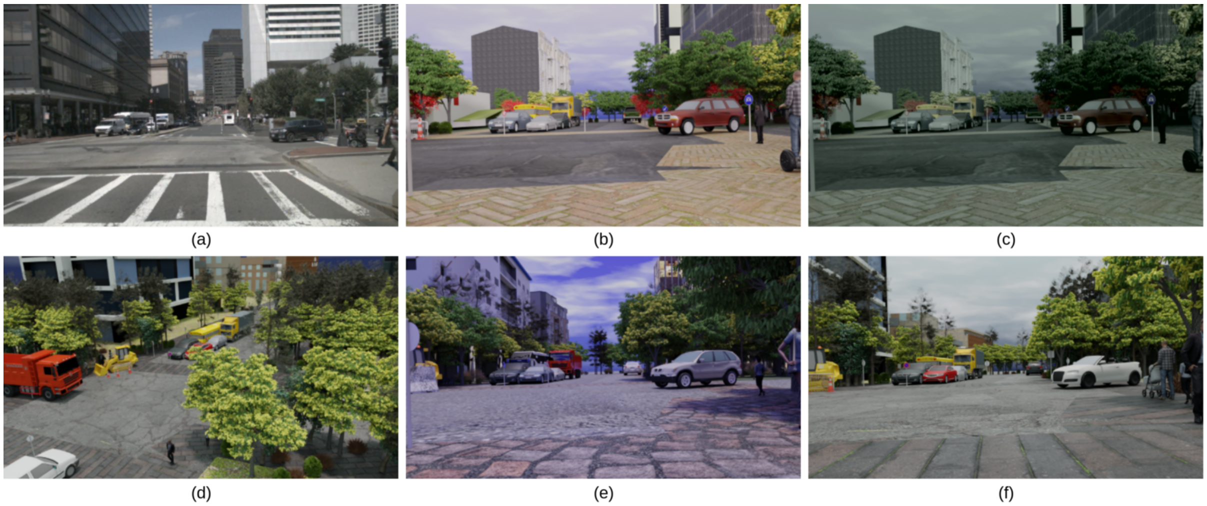


Figure 4.3 Example of variations in synthetic data for the same scene configuration. (a) The real-dataset image depicting a scene from the nuScenes dataset. (b) Sample render with different buildings, vehicles, pedestrians and lighting (c) Rendered image from ”b” after applying color transformation for the cityscapes dataset. (d) Render for the same scene from a different camera perspective. (e, f) Additional renders from the scene with variations in vehicles, buildings, lighting etc.

the query box o_i, while preserving aspect ratios. We then compute the 3D IoU metric as follows:

$$IoU_{3D}(o_i, o_j) = IoU_{xy}(o_i, o_j) * min(z_{o_i}, z_{o_j}) \tag{4.1}$$

Where, IoU_{xy} represents the 2D IoU metric for projection on the XY-plane, z_{o_i} is the length of the box along z-axis for object i. We are able to use a simplified implementation f the 3D IoU metric due to the known physical properties of both source and target object. We assign the asset with highest IoU_{3D} for a best match, or randomly sample from top-k to induce object diversity.

Camera Pose: We process the camera matrix and pose information from the source dataset and create virtual clones with similar configuration in the simulated environment. To extend the extent and visual coverage of generated images, we additionally sample camera poses from different vehicles in the scene (along with ego-vehicle), hence diversifying the view-points in a scene.

4.3.3 Textures, Lighting and Background

Texture and Lighting play a critical role towards achieving photo-realism in 3D virtual environment. Additionally, having dense background objects improves the content domain-gap and are a step forward towards realistic distribution of scene geometry.

Textures: We use high-quality 4K textures and PBR (Physically based rendering) materials from free-to-use forums [22]. High-resolution maps for color, displacement, roughness, normals, metallic,

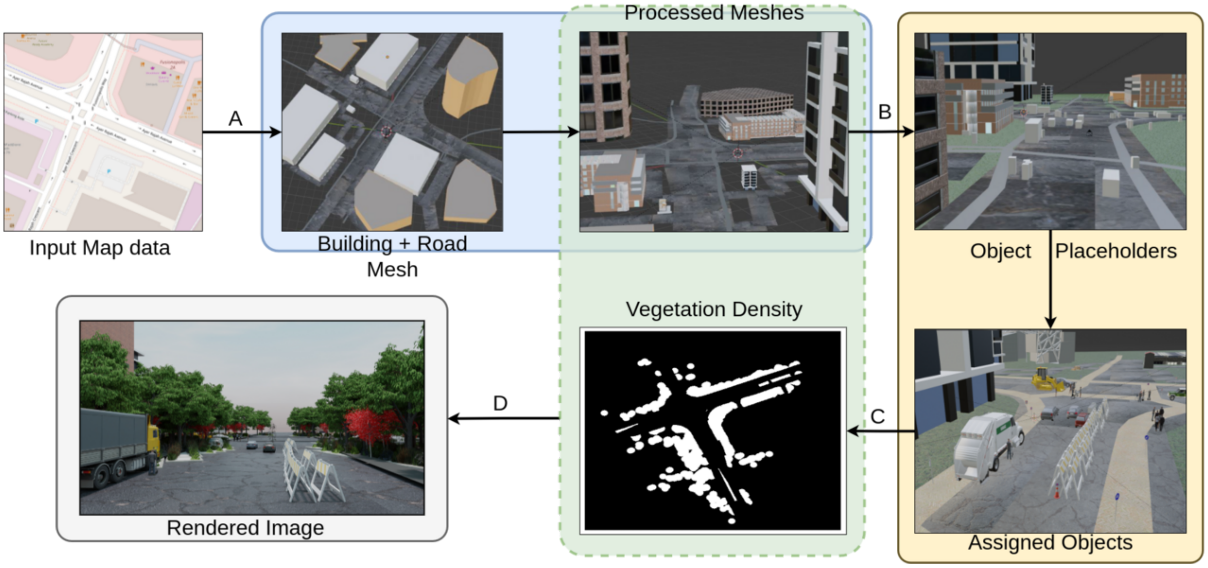


Figure 4.4 Samples from different steps in the proposed approach. Step (A) corresponds to the Section 3.1 for Building and Road processing, (B) corresponds to object placeholder creation and placement, (C) shows vegetation texture density as image, and part of building/road texture in step (A) output. Finally, Stage (D) shows the rendered output for the given scene and the camera pose data.

and emission are available, through which we create BSDF materials for building facades, roads, sidewalks, and ground/terrain.

Lighting: High-fidelity materials are important for photo-realism because the pixel-wise distribution of lighting, reflections and color impacts the level of visual perception. Lighting in virtual scenes is very important to accurately model scene dynamics from different view-points. We add different lighting environment models using High Dynamic Range Image (HDRI) to ensure a high range of illumination levels.

Background: We utilise LiDAR point cloud data from source and construct a 2D location density map for bushes and trees in the scene. To avoid unrealistic occurrences, we remove density data from road area by application of a binary mask. An example of the vegetation density in a scene is shown in Figure 4.2(C). In the virtual scene, we apply a probability distribution over the ground plane using the vegetation density map and instantiate trees/bushes. We sample different types and number of trees from the available assets with variations in sizes and sample locations. A similar approach is used for generating traffic signs and poles along the sidewalks in the scene.

4.3.4 Data Processing and Training

Once the preparation of the virtual environment is completed with all objects and entities populated in the scene, we proceed towards rendering and dataset curation stages for synthetic data availability in experiments.

Rendering and Annotations: We use the Cycles rendering engine in Blender to render the 3D scene and generate multi-modal annotations for semantic and instance segmentation, optical flow, depth estimation, 2D and 3D object detection. We employ the library provided in BlenderProc [27] for annotation extraction. Annotations are generated for 20 classes (including void) namely; sky, car, bus, jeep, truck, van, human, building, road, barrier, ground, cycle rider, construction (vehicle), bushes, trees, motorcycle rider, traffic cone, traffic sign, sidewalk, and void. However, for fair comparison with common benchmarks, we further process the available data into 13 classes as follows: void, car, bus, truck, person, rider, road, sidewalk, building, traffic poles, vegetation, terrain, and sky.

Training Data: For training and evaluation purpose, we sample a set of 5000 images from our synthetic dataset which have been selected based on the class distribution in annotation maps such that each sample contains a minimum of 6-8 different classes. Additional attention towards imbalance due to background classes is necessary to improve class-wise distribution.

Color Processing: Since we render images in different lighting conditions, the visual pixel-distribution may vary compared to a dataset we wish to benchmark against. Towards this, we optionally add a color processing stage similar to what was proposed in [69]. We transform the source and target images to L*a*b* color space and adjust the mean and variance of the source domain to a scaled metric between the two domains. A visual example is presented in Figure 4.3(b, c).

4.4 Experiments and Results

For a streamlined analysis and discussion about the experiments, we first outline details about the datasets used in our study and layout the experiment design. Then we analyse the results quantitatively and qualitatively to see how the proposed approach is useful to generate synthetic data that is useful for real-world model training and evaluation.

4.4.1 Datasets and Experiments

For our analysis, we use the Cityscapes [23] and KITTI-STEP [86] datasets. Both of these datasets have been selected for the fairly substantial amount of annotated data available in each.

Cityscapes: A widely used dataset for tasks related to visual odometry and perception for road scenes. The dataset provides 5000 finely annotated images for semantic and panoptic segmentation, and an additional 20000 images with coarse annotations with 30 class annotations. Collected over 50 cities in Europe, the dataset provides abundant diversity across scenes with different seasons, and some variations in weather. We use the full training set of the Cityscapes dataset (denoted as R) along with a random subset of 1000 images from the training set (denoted as P, partial) to show the impact of our synthetic dataset on evaluation of the validation set (denoted as V) from the real-world Cityscapes dataset. All images have been re-scaled to 512x256 without loss in aspect ratio for use in training semantic segmentation task. We only use the 5000 finely annotated (2975 for train and 500 for validation) for our experiments.

KITTI-STEP (Segmenting and Tracking Every Pixel) dataset, is an extension of the KITTI dataset with 21 training and 29 testing sequences from the raw KITTI dataset, based on the KITTI-MOTS [84] dataset and provides semantic as well as panoptic segmentation labels for each image in the sequence along with tracking IDs for non-background objects across frames in a scene. Since this dataset contains more samples compared to the KITTI semantic segmentation benchmark [2], we use the 5027 images in the train set and 2981 images in the validation set for our experiments. It is important to note that since the data is extracted from sequences, there is substantial overlap between many frames in the training set which will impact the results, as we shall see in the later analysis. We re-scale the images to a resolution of 620x188 without loss of aspect ratio for our experiments and perform semantic segmentation task. We follow the same notation where the full training data shall be denoted by R, partial data of 1000 images as P, and the validation set as V.

Training Details: We employ the Deeplab V2 architecture [16] with the resnet-50 backbone (pretrained with imagenet weights) without CRF. The architecture is kept consistent across all experiments to ensure fairness. For training, 5000 image samples from the generated synthetic dataset are considered, denoted as S. Each image is generated at a resolution of 1600x900, then appropriately downscaled and randomly cropped during training to adhere to aspect ratio in the image and match the real data for both training and validation, i.e., 512x256 for Cityscapes and 620x188 for KITTI-STEP. The color transformation scheme mentioned in Section 3.4 is optionally used and will be denoted as C wherever applicable in the results. For combined training of synthetic and real data, we use two methods; we can mix the real and synthetic images in the same batch while training the model (denoted by M) or train on synthetic data initially and then fine-tune with the real data (denoted by F). For an exhaustive comparison, we present results on all combinations of the settings and report the per-class IoU, mean IoU (mIoU), and global accuracy of each method. For training, we do not employ any additional augmentations apart from randomly cropping the synthetic image to adjust aspect ratio. The models are all trained for 30 epochs, with a batch size of 10 and initial learning rate $1e-04$. The models are trained on a Nvidia RTX 2080Ti GPU using Pytorch-lightning [30]. The quantitative results from experiments on Cityscapes and KITTI-STEP are presented in Tables 4.4.1 and 4.4.2, respectively.

4.4.2 Result Analysis

In Table 4.4.1 and Table 4.4.2, we present the **mIoU** and IoU per class for the 12 classes (excluding void), along with the global accuracy for each of the different training methods. Furthermore, we present qualitative results on both Cityscapes and KITTI-STEP datasets for some of the methods in Figure 4.5.

Training Method	Val. Data	Sky	Car	Bus	Truck	Person	Rider	Road	Sidewalk	Building	Traffic Poles	Veget.	Terrain	mIoU	Acc.
R	V	89.25	88.86	67.66	57.16	60.15	43.66	96.51	73.88	86.83	36.77	86.49	55.71	70.25	98.88
S	V	27.86	41.19	3.93	4.49	23.54	14.11	71.57	18.86	61.73	1.45	70.61	28.19	30.63	95.50
S + R [F]	V	89.03	88.86	68.27	51.16	60.53	43.21	96.67	74.88	86.89	**39.16**	86.51	57.62	70.23	98.89
S + R [M]	V	**90.06**	89.14	67.96	53.01	61.08	**44.73**	**96.84**	**75.92**	87.38	38.13	**87.04**	**58.49**	70.82	98.93
S + C	V	72.84	47.22	9.73	7.54	38.22	15.15	67.60	20.23	74.13	4.44	74.46	12.83	37.03	95.84
S + C + R [F]	V	88.95	88.83	69.07	57.38	60.33	43.99	96.66	74.83	86.97	38.20	86.49	57.07	70.73	98.89
S + C + R [M]	V	90.04	**89.51**	**72.09**	**65.48**	**61.28**	41.98	96.79	75.68	**87.46**	39.08	86.99	57.35	**71.98**	**98.94**
P	V	86.91	86.14	29.26	32.37	55.19	33.37	95.63	68.92	84.55	30.66	84.70	49.60	61.44	98.65
S + P [F]	V	88.00	86.71	46.71	35.02	55.22	28.40	96.03	70.62	84.99	32.80	85.03	53.16	63.56	98.70
S + P [M]	V	88.47	**87.25**	**58.44**	**44.99**	**57.92**	**40.24**	**96.39**	**73.02**	**85.86**	33.80	**85.78**	**54.34**	**67.21**	**98.80**
S + C + P [F]	V	87.19	86.71	49.94	36.39	55.05	26.36	95.61	68.59	85.13	33.20	85.10	48.05	63.11	98.69
S + C + P [M]	V	**89.05**	87.05	54.07	34.73	57.24	38.33	96.32	72.52	85.84	**34.01**	85.52	54.28	65.75	98.78

Table 4.1 Quantitative results for training on real and synthetic data and validation on Cityscapes dataset. We report class-wise IoU, mIoU and global accuracy for the 12 classes (excluding void)

Cityscapes (Full Real): We notice that training with a mix of real and synthetic data results in a boost of +1.73% mIoU when the synthetic data has gone through color adjustment. It is interesting to note that while synthetic data in itself may not be enough for achieving high performance (mIoU 30.63%), whenever combined with real data, helps improve accuracy compared to real data only. This is clear in the qualitative results as well where the model trained with a mix of modified synthetic data and real data is able to generate better segmentation mask for the classes person, sky, traffic pole/sign. It is worth highlighting that for the example in first example (left-sub column) of Cityscapes dataset, the model trained on real data is not able to detect some instance of the person class, while the model trained on the mixture, even with partial, is able to detect the same for this particular sample. When considering autonomous driving scenarios and real-world use cases, this is a crucial detail to consider towards enhancing the performance of deep learning architectures through synthetic data.

Cityscapes (Partial Real): We also highlight that for Cityscapes dataset, the model trained with synthetic and partial data achieves a significant improvement (+5.77% mIoU) over the model trained with just partial data. This result emphasizes on the practical implications of synthetic data where scarcity of real-world annotated data availability may be a bottleneck. Qualitatively as well, the model (S + P [M]) is able to predict sharp segmentation masks for pedestrians crossing the road, compared to the large masks predicted by model (P), trained only on partial real data. The visual results are coherent with the per-class IoU metrics highlighted in Table 4.4.1. The road, sidewalk, and traffic sign/pole

Training Method	Val. Data	Sky	Car	Bus	Truck	Person	Rider	Road	Sidewalk	Building	Traffic Poles	Veget.	Terrain	mIoU	Acc.
R	V	91.31	86.66	0.14	18.52	**60.99**	27.93	**85.51**	59.32	81.54	44.20	89.57	72.06	59.81	98.39
S	V	26.44	27.29	28.85	1.95	29.94	14.34	41.21	18.96	26.73	2.53	68.75	44.81	27.64	92.96
S + R [F]	V	91.54	**87.34**	1.68	13.38	60.57	27.80	84.95	59.20	82.06	**45.49**	**90.22**	73.20	59.79	98.41
S + R [M]	V	91.61	**87.34**	**53.55**	**28.56**	60.39	30.44	85.14	58.31	81.84	42.83	90.12	**74.26**	**65.37**	**98.42**
S + C	V	80.18	71.03	34.60	15.97	45.35	19.29	48.00	21.96	63.61	12.80	83.06	57.79	46.14	95.92
S + C + R [F]	V	91.41	86.99	2.23	19.98	59.58	**33.22**	85.40	**59.43**	81.63	45.36	90.01	73.24	60.71	98.41
S + C + R [M]	V	**91.66**	85.20	45.98	25.32	59.85	29.05	84.60	57.75	**82.24**	42.99	90.03	72.55	63.93	98.38
P	V	90.41	84.54	0	6.16	56.09	17.61	84.39	56.23	80.23	39.54	89.62	72.30	56.44	98.28
S + P [F]	V	90.80	**85.87**	12.91	8.17	56.91	16.01	83.33	54.39	80.14	42.21	89.66	72.09	57.71	98.25
S + P [M]	V	90.64	85.31	31.85	16.13	59.17	**29.60**	**84.88**	**57.21**	80.90	**41.77**	**89.88**	**73.25**	61.72	**98.34**
S + C + P [F]	V	90.28	85.44	2.09	12.63	54.91	20.54	84.26	56.38	79.74	40.60	89.21	71.64	57.31	98.27
S + C + P [M]	V	**91.35**	85.54	**48.59**	**22.65**	**59.25**	26.97	84.50	55.65	**81.95**	39.00	89.82	71.80	**63.09**	98.33

Table 4.2 Quantitative results for training on real and synthetic data and validation on KITTI-STEM dataset. We report class-wise IoU, mIoU and global accuracy for the 12 classes (excluding void)

segmentation metric for partial data shows a gap in the performances with and without synthetic data, and the same is observable in example from column 2 (right) in the Cityscapes qualitative sample. The traffic poles are missing almost entirely in the prediction from (P) and the sidewalk structure shows significant noise. Whereas for the same sample, (S + P [M]) shows a higher degree of accuracy and even generates predictions for the poles which have low pixel density.

KITTI-STEP: To strengthen our claims, we bring attention to the results presented in Table 4.4.2 where training with synthetic data, assuming full training dataset, shows an improvement of +5.56% mIoU and with the partial real dataset available only, an improvement of +6.65% mIoU. The seemingly high difference in the performance for these models can be attributed to the large performance gap in the ”bus” and ”truck” classes. The reason is that the number of annotated images with bus or truck appearing in the image is very low in the KITTI-STEP dataset. For qualitative confirmation of this case, we highlight column 2 (right) for KITTI-STEP dataset in Figure 4.5. The models trained using only real data misclassify the pixels of the truck object as car, while the model trained with a combination of real and synthetic data accurately segments the object as truck in (S + R[M]). A similar case can be observed for the bus visible in column 1 where only the model trained with synthetic and real combined are able to correctly segment some portion of the bus object. It is important to note that while synthetic data is useful for enhancement of deep learning models, generating such datasets usually requires manual efforts and careful design. However, in this work, we showed a method to generate synthetic dataset fully automatically, hence avoiding the dependency on manual design.

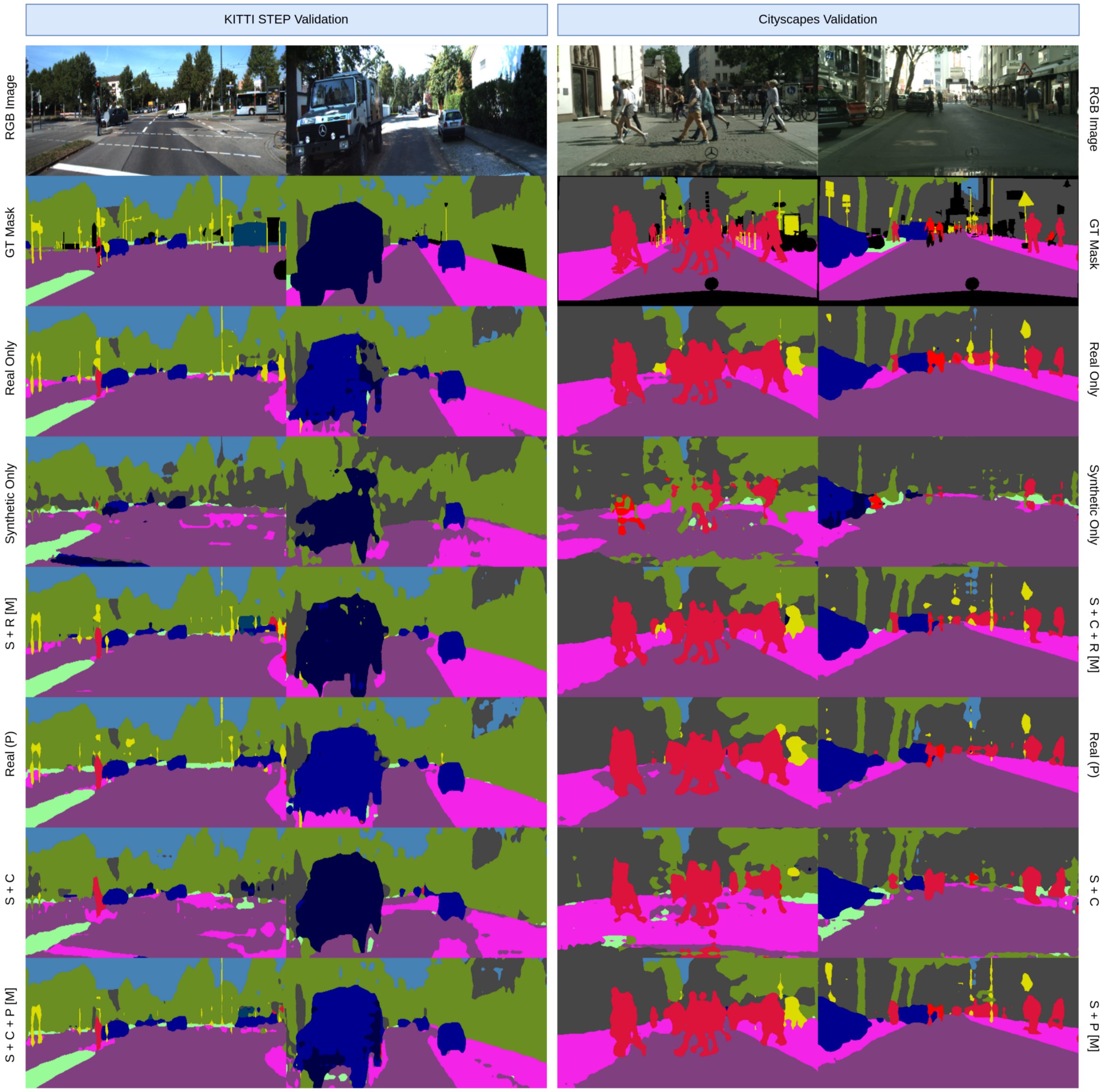


Figure 4.5 Qualitative results for different training strategies using real and synthetic data across KITTI-STEP and Cityscapes datasets. The nomenclature is R: Real, S: Synthetic, P: Real (partial), [F]: Fine-tuned on real, [M]: Real mixed with synthetic, C: Color transformation

4.4.3 Dataset Statistics

In the experiments presented, we use a sample of 5k images and semantic segmentation annotations, the 5k samples were selected after filtering out 2.8k samples with low variations in categorical labels per image. However, using the method described in this work, more data can be generated across different modalities including, but not limited to instance and panoptic segmentation, depth estimation, optical

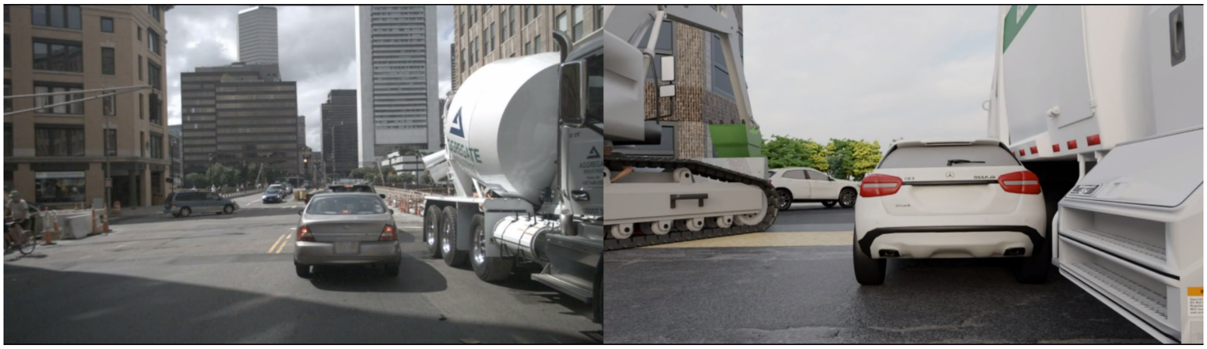


Figure 4.6 A side-by-side comparison of a scene with day time lighting in the real as well as synthetic data. We slightly adjust the real-world objects in the synthetic scene and place the assets at the respective locations to replicate real-world structure.

flow, surface normal estimation, 2D and 3D object detection, 6DoF object annotations, and 3D object reconstruction. We currently utilise 110 3D assets for different object categories and 40 PBR texture materials for roads, sidewalk, and building facades. In this work, we demonstrate the ability to generate synthetic data based on real-world annotations available in nuScenes dataset, however our approach can be extended to any public dataset to add more diversity in synthetic scenes. For each scene, we sample 20 images and corresponding annotations from different vehicles and camera poses. If OSM map data is available, it takes 25-30s to generate a virtual scene otherwise 30-40s considering an additional API call to OSM server to retrieve map data. Once a virtual scene is generated, it takes 65-85s to render each image of size 1600x900 and the corresponding annotations using a RTX 2080Ti GPU. Essentially, we can generate annotated data in orders of 100,000 within a span of 1 week with 12 GPUs in parallel.

4.4.4 Additional Qualitative Results

In this subsection, we highlight additional qualitative results from the synthetic data generation pipeline along with a side-by-side comparison with the real-world image counterparts. These results are highlighted in Figures 4.6, 4.7, 4.8, 4.9, 4.10, and 4.11. We also provide video demos for these scenes available at
https://iiitaphyd-my.sharepoint.com/:f:/g/personal/shubham_dokania_research_iiit_ac_in/EqxKJ9L1NaVMh4ooPO7hmPUB9kNKFyM7R5P7en175pVHqg?e=U4RIZR.

4.5 Code and Data

We release the code and data for the proposed framework at
https://github.com/shubham1810/trove_toolkit and
https://cvit.iiit.ac.in/images/Projects/trove_eccv22_data/raw_data_eccv.

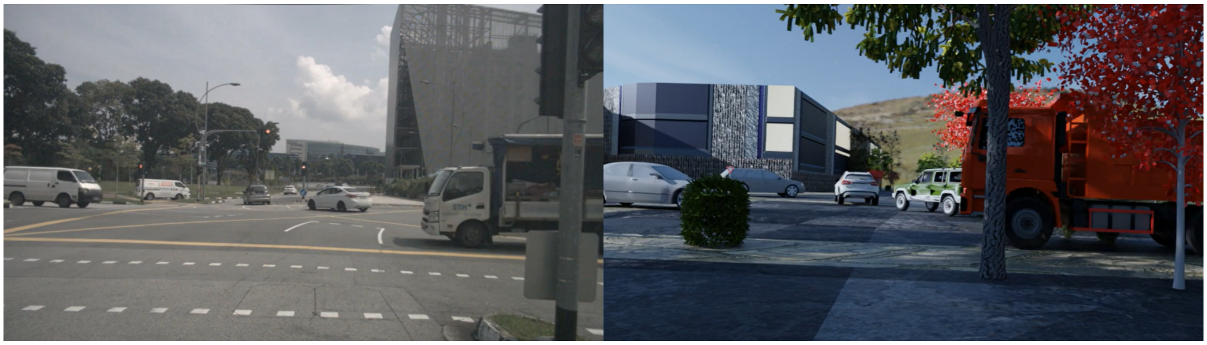


Figure 4.7 This sample shows the color adjustment example where the objects are placed in similar locations with changes in assets and color properties in the scene.

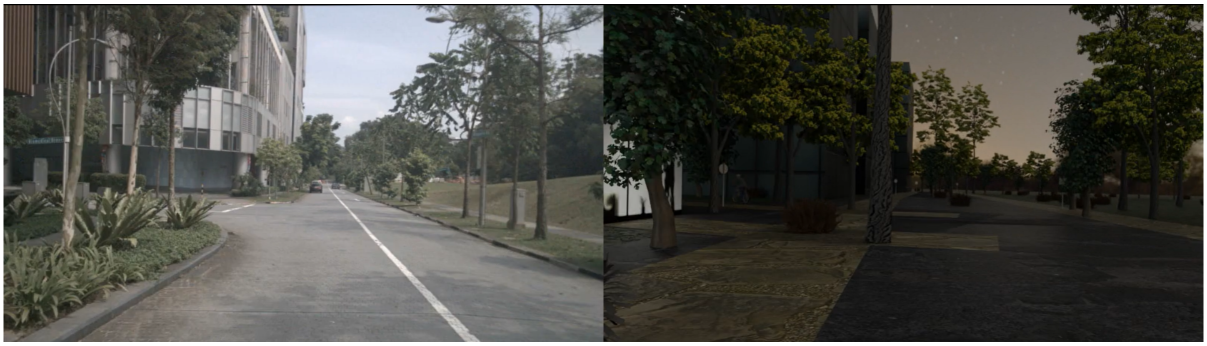


Figure 4.8 We take a real-world scene with only the meta-data and environment information and can replicate the scene in different variations. The current sample shows the synthetic data rendered in night scene. The lighting is different and the sky is illuminated with stars which provides ambient lighting in the virtual environment.

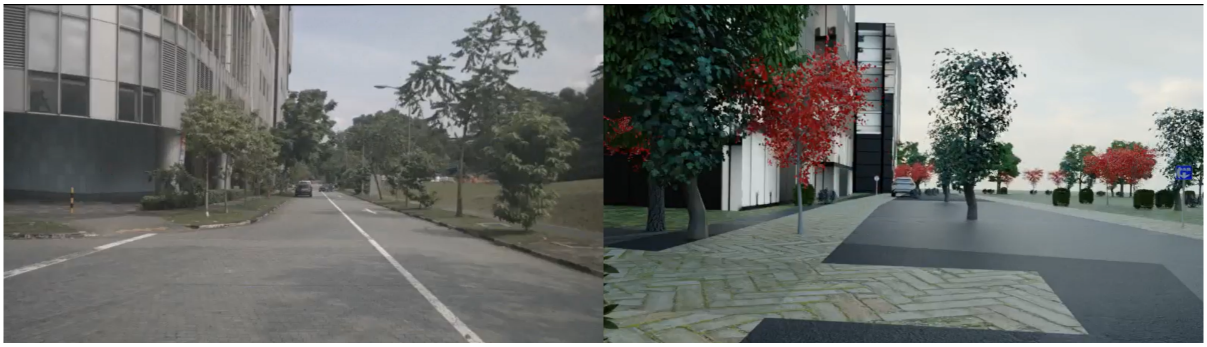


Figure 4.9 We highlight the same scenario as figure 4.8 but keeping the environmental lighting similar while changing some background properties.

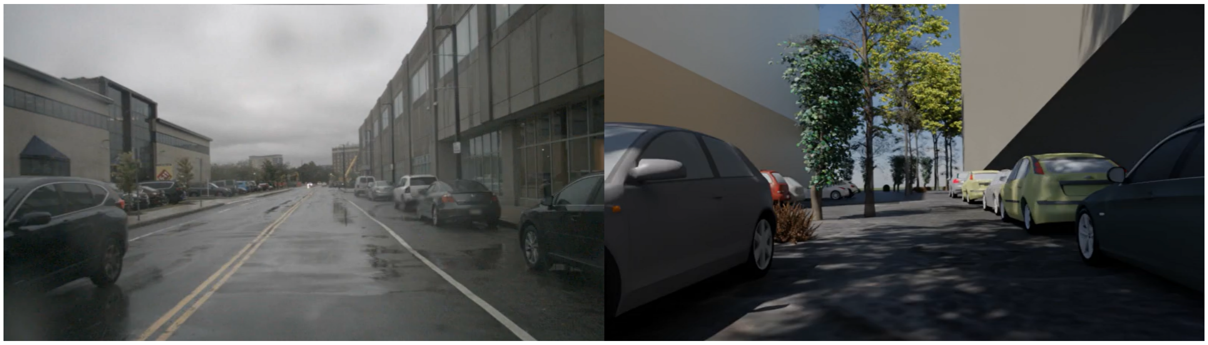


Figure 4.10 Furthermore, we can generate scenarios with different shading and sun positinos, while also eliminating existing weather conditions.

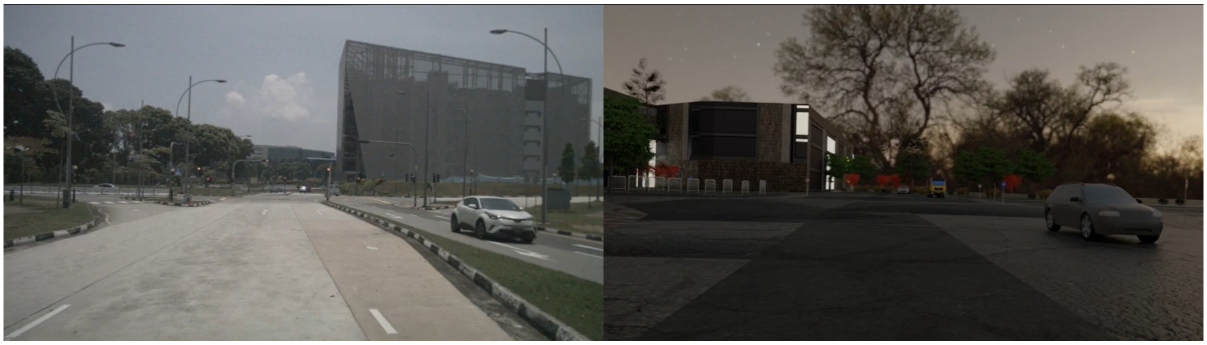


Figure 4.11 This sample shows the day-to-dusk shift with illuminated building windows and sunset lighting. These effects can be achieved with ray-traced rendering as shown and are superior to generative methods.

tar.gz. The processed files from the raw dataset are also available directly for the semantic segmentation experiments at : https://cvit.iiit.ac.in/images/Projects/trove_eccv22_data/eccv_train_data.tar.gz.

We provide the code toolkit for users to generate synthetic dataset using nuScenes annotations. However, as long as the configuration file can be generated in the same manner, we can produce data for any scenario.

4.6 Conclusions

We propose a framework for automatic generation of synthetic data for visual perception using existing real-world data. Using a set of 5k synthetically generate images and corresponding semantic segmentation annotations, we show the efficiency of combining synthetic data with real data towards improvements in performance. Given the potential scale of data generation capabilities, various types of data selection strategies can be applied without losing precious annotations. Our approach avoids the pitfalls and limitations of bounded data volumes and variety unlike manually designed virtual environments. Making use of geographical data, our data generation process can be extended to different locations across the globe. Further work in this direction could explore additional modalities and improved photo-realism with more complex scenes generated from a diverse set of datasets and benchmarks on a multitude of tasks. We also highlight another future direction towards better automation by using only visual inputs from a user side.

Chapter 5

Conclusion and Future Work

In conclusion, this thesis has successfully tackled key challenges in the realm of advanced driver assistance systems (ADAS) for autonomous vehicles, with a focus on the development of a data collection toolkit, the creation of a novel driving dataset, and the generation of synthetic data from real-world information.

A comprehensive data collection framework has been developed, which is both scalable and versatile. The use of sensor APIs based on the Robot Operating System (ROS) and a separate data collection interface has enabled the creation of driving datasets that can be easily adapted to different environments and sensor configurations. This framework lays the groundwork for future large-scale datasets, designed to encompass diverse driving scenarios and to serve as benchmarks for autonomous driving systems.

The IDD-3D dataset, which encompasses unstructured driving scenarios and complex road situations, has been meticulously analyzed both statistically and experimentally. This dataset aims to tackle the issue of generalizability across geographical locations and provide more diverse information for driving datasets and road scene analysis. The experiments conducted demonstrate the potential of the dataset in enhancing the performance of state-of-the-art approaches in 3D object detection and tracking. The insights gained from this dataset will pave the way for future research and development in autonomous driving applications.

The TRoVE synthetic data toolkit represents a significant step forward in overcoming the limitations associated with data collection and annotation. By generating synthetic data from existing real-world data, this framework combines the benefits of both real and synthetic data to improve the performance of visual perception tasks. The experiments conducted on the 5k synthetically generated images and their corresponding semantic segmentation annotations validate the effectiveness of this approach. The toolkit's potential for further scalability, automation, and diversity in data generation holds promise for advancements in a multitude of tasks and applications in the realm of autonomous driving.

Ultimately, this research has made substantial contributions to the field of autonomous driving, addressing key challenges in data collection, dataset creation, and synthetic data generation. These innovations not only serve to improve the performance and applicability of autonomous driving technologies but also pave the way for safer and more efficient autonomous vehicles capable of navigating complex and diverse road environments.

5.1 Future Work & Directions

Building upon the foundation laid by this thesis, several avenues for further research and development can be pursued to enhance the field of autonomous driving and advanced driver assistance systems (ADAS). Some potential directions for future work include:

- Expansion of the data collection framework: The current data collection toolkit can be refined and expanded to accommodate a wider range of sensors, environments, and driving scenarios. This could involve incorporating additional modalities, such as LiDAR and radar data, to improve the performance and robustness of ADAS algorithms.

- Enhancement of the IDD-3D dataset: To increase the diversity and applicability of the dataset, additional data from different geographical locations, traffic conditions, and weather scenarios can be included. Furthermore, the dataset can be extended to incorporate other tasks, such as lane detection, traffic sign recognition, and driver behavior analysis, providing a more comprehensive resource for researchers and developers in the field.

- Advancements in the TRoVE synthetic data toolkit: The toolkit's data generation capabilities can be improved by incorporating more complex scenes, higher levels of photorealism, and additional modalities. This would enable the creation of even more diverse and challenging synthetic datasets, further enhancing the performance of ADAS algorithms on real-world data.

- Exploration of active learning and data selection strategies: With the potential for generating vast amounts of synthetic data, it is crucial to develop efficient methods for selecting the most informative and relevant samples for training and validation. Investigating active learning and data selection strategies will help optimize the use of synthetic data in ADAS development.

- Evaluation of the impact of synthetic data on other ADAS tasks: In addition to semantic segmentation, the effectiveness of synthetic data can be assessed for other ADAS tasks, such as object detection, instance segmentation, depth estimation, and motion prediction. This would provide a broader understanding of the value of synthetic data in the development of autonomous driving systems.

- Development of end-to-end learning and transfer learning approaches: Leveraging the diverse data provided by real and synthetic datasets, it is possible to investigate end-to-end learning and transfer learning techniques to improve the generalizability of ADAS algorithms across different driving scenarios and environments.

By exploring these potential avenues for future work, the research community can continue to advance the field of autonomous driving, developing more efficient and robust ADAS applications that are capable of navigating complex and diverse road environments safely.

Related Publications

- **IDD-3D: Indian Driving Dataset for 3D Unstructured Road Scenes**, *Shubham Dokania*, A.H. Abdul Hafez, Anbumani Subramanian, Manmohan Chandraker, C.V. Jawahar. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2023*.

- **TRoVE: Transforming Road Scene Datasets into Photorealistic Virtual Environments**, *Shubham Dokania*, Anbumani Subramanian, Manmohan Chandraker, C.V. Jawahar. *In IEEE/CVF European Conference on Computer Vision (ECCV) 2022*

Unrelated Works:

- **MetaBoot: Self Supervised Set Representation learning to improve Few Shot Learning**, Shivanshu Sharma, *Shubham Dokania*, Anbumani Subramanian, Chetan Arora, Vineeth N. Balasubramanian, C.V. Jawahar. *Under Review*.

Bibliography

[1] Baidu Apollo team (2017), Apollo: Open Source Autonomous Driving, howpublished = https://github.com/apolloauto/apollo, note = Accessed: 2022-02-11.

[2] H. Alhaija, S. Mustikovela, L. Mescheder, A. Geiger, and C. Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision (IJCV)*, 2018.

[3] H. A. Arief, M. Arief, G. Zhang, Z. Liu, M. Bhat, U. G. Indahl, H. Tveite, and D. Zhao. Sane: Smart annotation and evaluation tools for point cloud data. *IEEE Access*, 8:131848–131858, 2020.

[4] A. Atapour-Abarghouei and T. P. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2800–2810, 2018.

[5] W. Bao, Q. Yu, and Y. Kong. Uncertainty-based traffic accident anticipation with spatio-temporal relational learning. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2682–2690, 2020.

[6] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019.

[7] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[8] Y. Cabon, N. Murray, and M. Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020.

[9] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[10] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[11] P. Cai, Y. Lee, Y. Luo, and D. Hsu. Summit: A simulator for urban driving in massive mixed traffic. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4023–4029, 2020.

[12] CGTrader. 3d model store. https://www.cgtrader.com/.

[13] R. Chandra, M. Mahajan, R. Kala, R. Palugulla, C. Naidu, A. Jain, and D. Manocha. Meteor: A massive dense & heterogeneous behavior dataset for autonomous driving. *arXiv preprint arXiv:2109.07648*, 2021.

[14] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[15] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[16] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[17] W. Chen, Z. Yu, S. D. Mello, S. Liu, J. M. Alvarez, Z. Wang, and A. Anandkumar. Contrastive syn-to-real generalization. In *International Conference on Learning Representations*, 2021.

[18] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[19] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon. Kaist multi-spectral day/night data set for autonomous and assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):934–948, 2018.

[20] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards scalable dataset construction: An active learning approach. In *European conference on computer vision*, pages 86–98. Springer, 2008.

[21] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[22] P. Community. Polyhaven 3d model and texture store. https://polyhaven.com/.

[23] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[24] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset. In *CVPR Workshop on The Future of Datasets in Vision*, 2015.

[25] D. Dai and L. Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3819–3824. IEEE, 2018.

[26] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5203–5212, 2020.

[27] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019.

[28] J. Devaranjan, A. Kar, and S. Fidler. Meta-sim2: Unsupervised learning of scene structure for synthetic data generation. In *European Conference on Computer Vision*, pages 715–733. Springer, 2020.

[29] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[30] W. Falcon et al. Pytorch lightning. *GitHub. Note: https://github. com/PyTorchLightning/pytorch-lightning*, 3:6, 2019.

[31] N. Gählert, N. Jourdan, M. Cordts, U. Franke, and J. Denzler. Cityscapes 3d: Dataset and benchmark for 9 dof vehicle detection. *arXiv preprint arXiv:2006.07864*, 2020.

[32] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016.

[33] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[34] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[35] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, et al. A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*, 2020.

[36] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[37] L. Hoyer, D. Dai, and L. Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. *arXiv preprint arXiv:2111.14887*, 2021.

[38] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang. The apolloscape dataset for autonomous driving. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1067–10676, 2018.

[39] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 954–960, 2018.

[40] T. Inc. 3d warehouse, trimble inc. https://3dwarehouse.sketchup.com/.

[41] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna. Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1110–1117. IEEE, 2018.

[42] J. Jacob and P. Rabha. Driving data collection framework using low cost hardware. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[43] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli. Rellis-3d dataset: Data, benchmarks and analysis. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 1110–1116. IEEE, 2021.

[44] J. Josifovski, M. Kerzel, C. Pregizer, L. Posniak, and S. Wermter. Object detection and pose estimation based on convolutional neural networks trained with synthetic data. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6269–6276. IEEE, 2018.

[45] A. Kim, A. Ošep, and L. Leal-Taixé. Eagermot: 3d multi-object tracking via sensor fusion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11315–11321. IEEE, 2021.

[46] A. Kishore, T. E. Choe, J. Kwon, M. Park, P. Hao, and A. Mittel. Synthetic data generation using imitation training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3078–3086, 2021.

[47] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019.

[48] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti. An efficient and layout-independent automatic license plate recognition system based on the yolo detector. *IET Intelligent Transport Systems*, 15(4):483–503, 2021.

[49] E. Li, S. Wang, C. Li, D. Li, X. Wu, and Q. Hao. Sustech points: A portable 3d point cloud interactive annotation platform system. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1108–1115. IEEE, 2020.

[50] Z. Li, T.-W. Yu, S. Sang, S. Wang, M. Song, Y. Liu, Y.-Y. Yeh, R. Zhu, N. Gundavarapu, J. Shi, et al. Openrooms: An end-to-end open framework for photorealistic indoor scene datasets. *arXiv preprint arXiv:2007.12868*, 2020.

[51] Y. Liao, J. Xie, and A. Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *arXiv preprint arXiv:2109.13410*, 2021.

[52] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022.

[53] Y. Luo, P. Cai, D. Hsu, and W. S. Lee. Gamma: A general agent motion prediction model for autonomous driving. *arXiv preprint arXiv:1906.01566*, 2019.

[54] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[55] J. Mao, M. Niu, C. Jiang, H. Liang, J. Chen, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li, et al. One million scenes for autonomous driving: Once dataset. *arXiv preprint arXiv:2106.11037*, 2021.

[56] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *International Journal of Computer Vision*, 126(9):942–960, 2018.

[57] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[58] F. E. Nowruzi, P. Kapoor, D. Kolhatkar, F. A. Hassanat, R. Laganiere, and J. Rebut. How much real data do we actually need: Analyzing object detection performance using synthetic and real data. *arXiv preprint arXiv:1907.07061*, 2019.

[59] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017.

[60] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice. Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[61] Z. Pang, Z. Li, and N. Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking. *arXiv preprint arXiv:2111.09621*, 2021.

[62] Q.-H. Pham, P. Sevestre, R. S. Pahwa, H. Zhan, C. H. Pang, Y. Chen, A. Mustafa, V. Chandrasekhar, and J. Lin. A∗3d dataset: Towards autonomous driving in challenging environments. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2267–2273. IEEE, 2020.

[63] M. Pitropov, D. E. Garcia, J. Rebello, M. Smart, C. Wang, K. Czarnecki, and S. Waslander. Canadian adverse driving conditions dataset. *The International Journal of Robotics Research*, 40(4-5):681–690, 2021.

[64] A. Prakash, S. Debnath, J.-F. Lafleche, E. Cameracci, S. Birchfield, M. T. Law, et al. Self-supervised real-to-sim scene generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16044–16054, 2021.

[65] prochitecture. blender-osm: Openstreetmap and terrain for blender. https://github.com/vvoovv/blender-osm, 2021.

[66] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018.

[67] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[68] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1964–1980, 2019.

[69] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[70] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9):1–40, 2021.

[71] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb, and J. M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10912–10922, 2021.

[72] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, et al. Lgsvl simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*, pages 1–6. IEEE, 2020.

[73] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[74] N. Ruiz, S. Schulter, and M. Chandraker. Learning to simulate. In *International Conference on Learning Representations*, 2019.

[75] C. Sakaridis, D. Dai, and L. Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10765–10775, 2021.

[76] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer, 2018.

[77] S. Sharma, L. Dabbiru, T. Hannis, G. Mason, D. W. Carruth, M. Doude, C. Goodin, C. Hudson, S. Ozier, J. E. Ball, et al. Cat: Cavs traversability dataset for off-road autonomous driving. *IEEE Access*, 10:24759–24768, 2022.

[78] V. A. Sindagi, Y. Zhou, and O. Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7276–7282. IEEE, 2019.

[79] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[80] L. Tamas and Z. Kato. Targetless calibration of a lidar-perspective camera pair. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 668–675, 2013.

[81] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018.

[82] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1743–1751, 2019.

[83] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 109–117, 2017.

[84] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe. Mots: Multi-object tracking and segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[85] J. Wang, A. Pun, J. Tu, S. Manivasagam, A. Sadat, S. Casas, M. Ren, and R. Urtasun. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9909–9918, 2021.

[86] M. Weber, J. Xie, M. Collins, Y. Zhu, P. Voigtlaender, H. Adam, B. Green, A. Geiger, B. Leibe, D. Cremers, A. Osep, L. Leal-Taixe, and L.-C. Chen. Step: Segmenting and tracking every pixel. In *Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2021.

[87] X. Weng, J. Wang, D. Held, and K. Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10359–10366. IEEE, 2020.

[88] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon. A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5000–5007. IEEE, 2019.

[89] J. Willman. Overview of pyqt5. In *Modern PyQt*, pages 1–42. Springer, 2021.

[90] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. 2021.

[91] M. Wrenninge and J. Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *arXiv preprint arXiv:1810.08705*, 2018.

[92] J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black. Lessons and insights from creating a synthetic optical flow benchmark. In *European Conference on Computer Vision*, pages 168–177. Springer, 2012.

[93] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[94] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[95] T. Yin, X. Zhou, and P. Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[96] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricár, S. Milz, M. Simon, K. Amende, et al. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9308–9318, 2019.

[97] D. Yoo and I. S. Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 93–102, 2019.

[98] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469, 2020.

[99] G. Zheng, H. Liu, K. Xu, and Z. Li. Learning to simulate vehicle trajectories from demonstrations. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1822–1825. IEEE, 2020.

[100] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[101] W. Zimmer, A. Rangesh, and M. Trivedi. 3d bat: A semi-automatic, web-based 3d annotation toolbox for full-surround, multi-modal data streams. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1816–1821. IEEE, 2019.