

DOCTORAL THESIS

Advancement in Perception Capabilities for Autonomous Vehicles: From Dataset Collection to Scene Interpretation

Junyi Gu

TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
60/2024

Advancement in Perception Capabilities for Autonomous Vehicles: From Dataset Collection to Scene Interpretation

JUNYI GU



TALLINN UNIVERSITY OF TECHNOLOGY
School of Engineering
Department of Mechanical and Industrial Engineering

**The dissertation was accepted for the defence of the degree of Doctor of Philosophy
on 02 October 2024**

Supervisor: Tenured Associate Professor Raivo Sell,
Department of Mechanical and Industrial Engineering,
Tallinn University of Technology,
Tallinn, Estonia

Co-supervisor: Adjunct Professor Mauro Bellone,
FinEst Centre for Smart Cities,
Tallinn University of Technology,
Tallinn, Estonia

Opponents: Professor Giulio Reina,
Department of Mechanics, Mathematics and Management,
Polytechnic University of Bari,
Bari, Italy

Dr. Dimitrios Giakoumis,
Information Technologies Institute,
Centre for Research and Technology Hellas,
Thessaloniki, Greece

Defence of the thesis: 06 November 2024, Tallinn

Declaration:

Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology, has not been submitted for any academic degree elsewhere.

Junyi Gu

signature

Copyright: Junyi Gu, 2024
ISSN 2585-6898 (publication)
ISBN 978-9916-80-210-6 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9916-80-211-3 (PDF)
DOI <https://doi.org/10.23658/taltech.60/2024>
Printed by Koopia Niini & Rauam

Gu, J. (2024). *Advancement in Perception Capabilities for Autonomous Vehicles: From Dataset Collection to Scene Interpretation* [TalTech Press]. <https://doi.org/10.23658/taltech.60/2024>

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
60/2024

Autonoomsete sõidukite tajuvõimekuse täiustamine: andmekogumisest stseeni tõlgendamiseni

JUNYI GU



Contents

List of Publications	7
Author's Contributions to the Publications	8
Abbreviations.....	9
1 Introduction	10
1.1 Background	10
1.1.1 Dataset and Dataset Collection for Autonomous Driving.....	11
1.1.2 Scene Interpretation for Autonomous Vehicles.....	13
1.2 Literature Review.....	14
1.2.1 Training Dataset for Autonomous Driving	14
1.2.2 Multi-Modal Sensor System for AVs	15
1.2.3 Deep Learning for Camera-LiDAR Fusion	16
1.2.4 Transformers for AV Perception	16
1.3 Motivation and Research Problems	17
1.4 Research Objectives and Hypotheses	18
1.5 Research Tasks and Contributions	18
2 Cyber-Physical Experimental Platforms	20
2.1 Range Sensor Deployment for Autonomous Shuttles	20
2.2 Multi-Sensor Perception and Collection Framework	21
2.2.1 Hardware for Framework Validation	23
2.2.2 Software System and Server	24
2.3 Training Datasets	24
2.3.1 Waymo Open Dataset	24
2.3.2 iseAuto Dataset.....	25
3 Methodologies	27
3.1 Sensor Calibration and Synchronization.....	27
3.1.1 Sensor Intrinsic and Extrinsic Calibration.....	27
3.1.2 Sensor Synchronization	28
3.2 Signal-level Camera-LiDAR-radar Sensor Fusion	30
3.2.1 Camera-LiDAR Fusion	30
3.2.2 LiDAR-radar Fusion	32
3.3 Neural Networks for Object Segmentation	33
3.3.1 Camera-LiDAR Fusion Convolutional Neural Network (CLFCN).....	34
3.3.2 Camera-LiDAR Fusion Transformer (CLFT).....	34
4 Experiments and Results	37
4.1 Performance Evaluation for Dataset Collection Framework	37
4.2 Domain Adaptation Analysis for CLFCN	38
4.3 Benchmark Comparison for CLFT.....	40
5 Conclusions and Future Work	44
References.....	46
Acknowledgements	57

Abstract.....	58
Kokkuvõte	59
Appendix 1.....	61
Appendix 2	69
Appendix 3	89
Appendix 4	117
Curriculum Vitae	131
Elulookirjeldus.....	133

List of Publications

The present Ph.D. thesis is based on the following publications that are referred to in the text by Roman numbers.

- Article I Junyi Gu** and Tek Raj Chhetri. Range sensor overview and blind-zone reduction of autonomous vehicle shuttles. *IOP Conference Series: Materials Science and Engineering*, 1140(1):012006, may 2021
- Article II Junyi Gu**, Mauro Bellone, Raivo Sell, and Artjom Lind. Object segmentation for autonomous driving using iseauto data. *Electronics*, 11(7), 2022
- Article III Junyi Gu**, Artjom Lind, Tek Raj Chhetri, Mauro Bellone, and Raivo Sell. End-to-end multimodal sensor dataset collection framework for autonomous vehicles. *Sensors*, 23(15), 2023
- Article IV Junyi Gu**, Mauro Bellone, Tomáš Pivoňka, and Raivo Sell. CLFT: Camera-LiDAR fusion transformer for semantic segmentation in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2024

Author's Contributions to the Publications

Article I The author reviewed the literature on range sensor deployment for modern autonomous vehicles and designed the experiments to validate the blind-zone coverage for different range sensor configurations on the iseAuto shuttle. The author contributed to the software development and paper writing.

Article II The author collected the iseAuto training dataset, designed and carried out the domain adaptation experiments of Waymo and iseAuto datasets. The author contributed to the software development, dataset publicity, paper writing, and revision.

Article III The author conceptualized the generic end-to-end dataset collection framework, developed the testing platform, and used it to validate the framework. The author contributed to the software and hardware development, visualization, paper writing, and revision.

Article IV The author investigated the proposal of a transformer-based network to fuse camera and LiDAR data for semantic segmentation. The author developed the networks and conducted the experiments to train, test, and evaluate the networks against other state-of-the-art techniques. The author contributed to the software programming, paper writing, and revision.

Abbreviations

AV	Autonomous Vehicle
SLAM	Simultaneous Localization and Mapping
CNN	Convolutional Neural Network
SAE	Society of Automotive Engineers
ADAS	Advanced Driver Assistance System
LiDAR	Light Detection and Ranging
IMU	Inertial Measurement Unit
GNSS	Global Navigation Satellite System
AI	Artificial Intelligence
FCN	Fully Convolutional Network
ViT	Vision Transformer
FoV	Field of View
GPS	Global Positioning System
ROS	Robot Operating System
BEV	Bird Eye View
RCU	Residual Convolution Unit
IoU	Intersection over Union
CLFCN	Camera-LiDAR Fusion Convolutional Neural Network
CLFT	Camera-LiDAR Fusion Transformer

1 Introduction

1.1 Background

Autonomous vehicles (AVs) shall be our ultimate form of ground transportation. Undoubtedly there is a long way ahead to achieve fully autonomous driving. However, the history of our explorations toward AVs is also long and dates back to the first thriving period of mass motorization in the 1920s [1]. Technically, instead of self-driving, the attempts in the early days were somewhat of the remote-controlling, which simply moved the driver outside the vehicles. At that time, this task required the integration of equally smart vehicles and roads [2]. One of the iconic pioneering experiments happened in the 1950s; General Motors embedded the electric circuits in a section of a public highway to demonstrate a self-guiding system [3]; although the self-driving system is not inside the car, it achieved the basic imagery of an AV. General Motors' experiments reflect the research concentrations at that time for AVs, which use the road to interfere with the vehicles' behavior to achieve autonomous driving and eliminate driver errors [4]. The rise of integrated circuits in the second half of the 20th century shifted the scope of AV research from building sophisticated roads to developing intelligent vehicles because computers and sensors are small enough to fit in ordinary production cars.

The advent of computer vision and machine learning marks the rapid progress of AVs, which are seen as independent transport able to perceive the environment and navigate through a multitude of sensor readings. The expectations toward AV are solving the traffic issues associated with ordinary vehicles, including pollution, congestion, and traffic accidents [5]. Alongside the development of AV's automation and intelligence, concerns of more than technical perspective were raised by the researchers [6]. Among all ethical and moral issues, the safety of AV draws the most attention [7]. The promises of safety require that AV technology is advanced and integrated with all functional perspectives, which are categorized as four blocks in work [8]: perception, planning and decision, motion and vehicle control, and system supervision.

The focus of this thesis is AV's perception capability. Although the history of AV is already decades long, the visions of the AV perception are ambiguous and have evolved with the emerging technologies. In the very beginning, perception plays an auxiliary role in developing the maps of surroundings through analyzing the distances of the AV and other objects [9]. For example, Simultaneous Localization and Mapping (SLAM) algorithms blur the boundary between perception and mapping. However, due to the rapid development of computer vision techniques, the definition of perception for AVs is evolving. The fast and precise object detection and classification capabilities lead to the proposal integrating the perception and planning/decision stages. Work [10] is a famous early attempt that adopted the integration idea, named the paradigm of direct perception. In contrast to mediated perception [11] and behavior reflex perception [12] referred in the paper, direct perception allocates more computation resources for environment perception, and aims to achieve autonomous driving with few classic mapping and localization stages. The essence of work [10] is a Convolutional Neural Network (CNN) based model that maps image input to several key prediction indicators, such as the vehicle's orientation to the road and distance to other road-related objects. Apparently, due to neural networks' limitations in the early times, this proposal is a trade-off between parsing entire scenes (mediated perception) and mapping images directly to driving actions (behavior reflex perception). Thus, global mapping and localization still exist in its process. Nonetheless, the direct perception method sparks the researcher's interest to exploit the potential of deep learning technology within the AV perception field [13].

The primary applications of neural networks for AV perception are traffic-related object detection and segmentation [14]. Due to the constant evolution of neural network technology, the recent trends of AV perception are multi-class classification and multi-modal sensor fusion. Moreover, the direct perception proposals for AVs aim to unify the mapping and decision-making processing into the same framework, which can further increase the complexity of perception tasks. Correspondingly, the learning process requires an exponentially increasing quantity of data. Therefore, datasets have become a pivotal issue for autonomous driving in recent years. To thoroughly analyze the advancement of AV's perception capabilities, besides the novel neural network proposals for scene interpretation, this thesis includes the dataset description and the corresponding acquisition system. The following subsections will introduce these two topics in detail.

1.1.1 Dataset and Dataset Collection for Autonomous Driving

The concept of the dataset is not only for automated vehicles but also closely related to traditional vehicles [15]. On the spectrum of the automation level, vehicular datasets are divided into two principal categories: naturalistic datasets for traditional driving and training datasets for autonomous driving.

The naturalistic datasets cover the insights related to traditional vehicles, which are fully under-controlled by human drivers with no automation. The system monitors the driver's behavior, the vehicle's status, and external environments such as temperature, precipitation, and illumination conditions. The naturalistic datasets provide the panorama of the transportation domains such as road safety [16], ecological effect [17], and traffic insurance [18]. However, this thesis focuses on the training datasets for high-automation-level vehicles. Unlike the naturalistic datasets that analyze the interaction between vehicle and human drivers from the statistical perspective, training datasets aim to the quantity of data covering as many vehicle and contextual scenarios as possible.

Referring to the Society of Automotive Engineers (SAE) Levels of Driving Automation [19] standard (six levels in total from level 0 to level 5), an AV technology benchmark broadly utilized by AV developers. The detection and tracking of traffic objects in the surrounding environments are critical for both level 4 and 5, which correspond to 'high' and 'full' automation, respectively. Currently, the achievements of scene-understanding tasks increasingly rely on sophisticated deep-learning technology, which further promotes the necessity for training datasets. Moreover, due to the strict requirements for AV safety, it is essential to ensure the AVs' robustness in challenging scenarios such as scenario diversity, adverse weather, and illumination conditions. Therefore, the scale of training datasets for environment perception has significantly enlarged in recent years to cover various driving conditions. As a result, no dataset currently individually fulfills all requirements. Intending for collaboration, innovation, and effort-sharing, both research communities and industrial groups endeavor to produce datasets and make them publicly available.

Nevertheless, the issues of training using large datasets are non-negligible. On the one hand, performing comprehensive dataset collection is limited to many research organizations due to its complexity and its high resource burden. On the other hand, compatibility and applicability are always researchers' concerns. Because of the rapid development of sensor technology and distinct learned scenarios of autonomous driving tasks, researchers encounter issues such as inconsistent hardware configurations, limited traffic and environment scenes, and nontransferable data formats among most training datasets. For instance, the multi-sensor KITTI dataset [20] was introduced in 2012 and has been famous for autonomous driving research for a long time. However, the KITTI dataset is outdated because it contains only clear weather scenarios, and the Light Detection and

Ranging (LiDAR) sensor it uses, Velodyne HDL-64E, is discontinued from Velodyne's product line [21]. Replication is another challenge for open training datasets because many of them are recorded by the customized sensor modules. For example, Waymo Open dataset [22] does not reveal any sensor model. The data acquisition module for ApolloScape [23] dataset consists of six video cameras, two laser scanners, and an integrated Inertial Measurement Unit (IMU) and Global Navigation Satellite System (GNSS) system, which is too complicated for most research groups to replicate.



Figure 1: Examples of semantic and instance segmentation. (a) is the semantic segmentation, where objects of the same class are highlighted by the same color. (b) is the instance segmentation, where each object from the same class is assigned by an individual color. The sample image is from the custom iseAuto dataset [24].

Data-driven end-to-end approach [25] has been investigated by many researchers as an alternative to the classic module-based counterpart to address problems of large-scale training datasets. The approach's essence is a unified system that directly takes raw sensor data as input and produces training datasets for ultimate autonomous driving tasks. The key advancements of end-to-end data collection methods lie in simplicity, efficiency, and generalizability. The holistic end-to-end approach integrates all raw-data-related processing (i.e., denoising, synchronization) and the intermediate representations (i.e., LiDAR point clouds filtering, data compressing) into a generic framework, practically improving the computational efficiency for dataset collection applications. Additionally, the merits of end-to-end methods open up the potential for researchers to establish the datasets based on their realistic scenarios. For example, one of the contributions of this thesis is a unique training dataset [24] recorded at the TalTech campus, which has lower illumination conditions than most other open datasets. Moreover, the iseAuto autonomous shuttle [26], which is the first level 4 self-driving shuttle in Estonia for research and educational purposes, was operated at the TalTech campus. Thus, a custom dataset provides the practical substance to improve the reliability and performance of many autonomous driving technologies. The accomplishment of this dataset relies on an end-to-end multi-sensor dataset collection framework [27], which is another objective of this thesis.

1.1.2 Scene Interpretation for Autonomous Vehicles

Scene interpretation for AVs is a concept built upon vehicle perception and requires comprehensive contextual information extraction of the surrounding environment. As a matter of fact, the concept of vehicle perception appears ahead of autonomous driving. One of the definitions of vehicle perception is the stage that directly receives data from sensors [28]. From this perspective, Advanced Driver Assistance System (ADAS), a developed and guaranteed technology for commercial vehicles throughout decades, is one of the most well-known examples that empower vehicles with perception capabilities. However, the most sophisticated ADAS can only be classified as level 2 'Partial Automation' standard among the SAE's six-level vehicle automation standard. Within SAE taxonomy, regardless of level 5, which expects vehicles to perform full automation under all conditions, both level 3 and 4 require vehicles to interpret the receiving data to generate a representation of the surrounding environment. This establishes the basic definition and requirement of AV perception.

The research on scene interpretation consists mostly of traffic elements, including pedestrians, vehicles, traffic signs, and lanes, among many others. The precise perception of these traffic objects, such as detection, classification, and tracking, is the essence of scene interpretation. Among all tasks related to object perception, segmentation is considered a challenging problem because it requires a particular class assignment for each pixel, thus attracting broad interest from the community [29]. Research directions for object segmentation span 2D semantic segmentation, 3D semantic segmentation, and instance segmentation. Semantic segmentation predicts per-pixel class labels, while instance segmentation provides individual instance information. The differences between these two segmentation methods are visualized in Figure 1. According to the survey work [30], semantic segmentation is the most broadly investigated method, which interprets the ongoing scene into different classes that are critical for autonomous driving.

This thesis focuses on semantic segmentation and aims to investigate cutting-edge deep learning technologies for camera and LiDAR fusion. For autonomous driving, the development of sensor technology in recent years provides the all-time perception regardless of the weather and illumination conditions. In principle, as a sensor with a long development history, the camera provides enough data to estimate the object's movement and interpret the driving scenarios. Therefore, there are already many popular studies that use CNNs to process camera images to detect [31, 32] and segment [33] 2D objects. However, camera sensors have the same limitations as human vision systems in darkness and low visibility, which are scenes that play crucial roles in traffic and road safety [34]. Moreover, one of the critical challenges for AVs is attaining an accurate real-time understanding of the 3D environment. To this point, scene interpretation based on range sensors become an emerging research topic. The radar sensor is the earliest range sensor installed on vehicles and is the primary perceptive sensor for ADAS. The mainstream radar sensor for ADAS is millimeter-wave radar, which cannot capture textual information and has limited range and resolution. Therefore, though ADAS technologies can perceive the existence of obstacles in the sensors' effective zones, they cannot classify the object types and, particularly, assign any semantic meaning. Recently, LiDAR sensors have attracted broad interest from research and industrial communities because of their reliability improvement and cost decrease. Despite LiDAR sensors' drawbacks, such as the absence of color and texture information, which cannot be ignored, LiDAR sensors compensate for the shortcomings of camera sensors concerning weather and illumination conditions. Therefore, modern AVs adopt LiDAR sensors for 3D spatial perception and sensor fusion, leveraging multiple sensors with different characteristics to achieve comprehensive perception. The

literature on camera and LiDAR fusion for perception is rich in survey [35–37]. The radar fusion research for AVs is relatively rare but has increasingly attracted attention in recent years [38]. In contrast to camera and LiDAR sensors, radar sensors have complementary advantages in speed estimation, moving object detection, and promising perception in environments such as dust and fog. Therefore, in addition to using deep learning technologies to fuse camera and LiDAR data, this thesis provides the signal-level fusion [30] algorithms to utilize the merits of the camera, LiDAR, and radar sensors in a manner to enhance the object perception and tracking.

1.2 Literature Review

This section extends the previous section’s discussion, providing deep insights into state-of-the-art literature regarding the thesis’ primary focuses related to the training dataset, multi-sensor system, and deep-learning technology for perceptive sensor fusion.

1.2.1 Training Dataset for Autonomous Driving

Recently, data is believed to be a valuable property for autonomous driving, especially training datasets. Compared with the naturalistic datasets mentioned in Section 1.1 that span fields such as transportation ecology, insurance, and driver behavior, the training datasets are primarily for autonomous driving tasks where deep learning technologies are broadly involved. Thus, the data quantity directly affects the performance. Moreover, training datasets are also repetitively used for benchmark comparison. Therefore, the research community and industry have allocated significant efforts to producing training datasets for autonomous driving research. Work [15, 39, 40] have surveyed publicly autonomous driving datasets in the last decade from different perspectives such as instrumentation information, acquisition time, and sequence length. Because this thesis’s topic is multi-sensor-based scene interpretation in various weather and illumination conditions, the open training datasets reviewed in this section all consider scenes and modality diversities. Table 1 introduces the details of the common datasets with corresponding literature references.

Table 1: The list of open training datasets with scene and modality diversity for autonomous driving.

Datasets	Sensors			Annotations		Scenes		
	camera	LiDAR	GNSS	bounding box	semantic mask	weather	illumination	season
KITTI [20]	✓	✓	✓	✓	✓		✓	
ApolloScape [23]	✓	✓	✓		✓	✓	✓	
Argoverse [41]	✓	✓	✓	✓		✓	✓	✓
Waymo Open[22]	✓	✓	✓	✓		✓	✓	
Berkeley DeepDrive [42]	✓		✓	✓	✓	✓	✓	
PadanSet [43]	✓	✓	✓	✓	✓		✓	
CityScapes [44]	✓		✓		✓			✓
IDD [45]	✓				✓	✓	✓	
KAIST Multi-Spectral [46]	✓	✓	✓	✓			✓	
lyft Motion Prediction [47]	✓	✓	✓	✓	✓			
NightOwls [48]	✓			✓		✓	✓	✓
NuScenes [49]	✓	✓	✓	✓	✓	✓	✓	✓
A2D2 [50]	✓	✓	✓	✓	✓	✓		

1.2.2 Multi-Modal Sensor System for AVs

The environment perception and data acquisition of the modern autonomous and assisted driving technology relies on the paradigm of multi-modal sensor systems [51]. Considering this thesis concerns the AV's perception of scene interpretation, the reviews of multi-modal sensor systems focus on exteroceptive sensors such as camera, LiDAR, and radar sensors for traffic object detection, tracking, and segmentation.

The research of the multi-modal sensor system can be divided into hardware and software two aspects. From the hardware perspective, the development of sensor manufacturing technology is outside the scope of autonomous driving research, so the reviews concentrate on vehicle sensor deployment. Practically, the basic requirement of sensor deployment for perception is covering as many blind zones as possible. For data acquisition purposes, all the sensors should have a clear view field and less interference.

In general, there are two strategies for deploying sensors on vehicles. The first strategy involves installing the sensors around the vehicle's body. The testing vehicle in [52] has 15 sensors integrated on different sides of the vehicle. The vehicle's appearance and performance are not much changed, hence, it does not need specific care and can conduct experiments in any situation. A similar sensor installation was adopted by BRAiVE [53] and VIAC [54]. All sensors and cables on BRAiVE's vehicle were hidden, and visual-based sensors were mounted together on top of the testing van in the VIAC project. However, this strategy is mainly used for fulfilling the legal requirements for real-traffic deployment. The second strategy, which integrates all sensors on a separate mount, is more suitable for experimental and testing cases, especially when multi-channel full Field of View (FoV) LiDAR sensors are used. The famous example in early times is the 2005 DARPA Grand Challenge winner Stenley [55], which has nearly all sensors held on a custom-made roof rack on top of a commercial vehicle. Other experimental platforms with sensors installed on detachable mounts for convenient accessibility and maintenance are [56, 57].

From the software perspective, the multi-modal sensor system for AVs primarily involves sensor calibration and fusion. For multi-sensor calibration, extrinsic and temporal calibrations are the main focus; extrinsic calibration calculates the transformation between sensors, and temporal calibration handles sensor synchronicity. The literature on extrinsic calibration is rich. Domhof et al. [58] proposed a thorough camera, LiDAR, and radar extrinsic calibration method that innovatively uses metallic trihedral corners to enhance the radar reflection. Work [59] relied on 2D planar objects for extrinsic calibration. Checkerboard and other auxiliary 2D objects were combined for estimating 3D-2D transformation. Calibrating sensors without a specific target is another strategy for multi-sensor calibration. Jeong et al. [60] estimated sensor motions by road markings and then determined the transformations between sensors. Schöller et al. [61] utilized a CNN network to calibrate camera and radar sensors. Compared with manually matching the radar point clouds and image features, neural networks have advantages in speed and efficiency.

The multi-sensor fusion has been one of the hottest topics in recent years. The corresponding works have been thoroughly reviewed by many researchers [62–64]. Remarkable work such as [65] employed low-level fusion for less computational consumption and low latency, also the aims of the multi-sensor fusion strategy proposed in this thesis. Another similar work is [66], in which authors combined the Fully Convolutional Neural Network (FCN) and Kalman Filter into a hybrid framework to fuse the camera, LiDAR, and radar data. Cost efficiency was deeply explored in work [67] that only relied on a Microsoft Kinect camera to produce color images and point clouds for road surface monitoring.

1.2.3 Deep Learning for Camera-LiDAR Fusion

Deep learning has been one of the hottest topics in recent years, and camera and LiDAR are the two most adopted sensors for AVs. Thus, combining deep learning and camera-LiDAR fusion stands out as one of the most intensively investigated research. The taxonomies to review the camera-LiDAR deep fusion algorithms are various. For instance, the approaches can be categorized based on applications such as depth completion, object detection, object tracking, instance segmentation, and semantic segmentation. However, one essential focus of this thesis, scene interpretation, was achieved by signal-level camera-LiDAR-radar fusion and multi-level fusion neural networks. Therefore, this section reviews the camera-LiDAR deep fusion algorithms in the taxonomy of signal-level, feature-level, result-level, and multi-level.

- **Signal-level.** The signal-level fusion mainly conducts the raw data integration, such as geometric coordinate matching or 3D-2D projection. Depth completion is the application that broadly adopts signal-level fusion. Ma et al. [68] proposed a supervised model that takes RGB and depth images as input and learns a direct mapping from sparse depth to dense depth prediction. Work [69] used camera and LiDAR data and adopted signal-level fusion as part of its image-guided framework for LiDAR completion. Other signal-level depth completion research are [70, 71]. Another application that can use signal-level fusion is road detection; the corresponding possibility and shortcomings were explored by [72–74] in detail. Work [75, 76] are two of the few research using a signal-level strategy for object detection because of the relatively heavy texture information loss in signal-level fusion.
- **Feature-level.** Feature-level fusion is broadly used for object detection and segmentation tasks. In general, the differences in feature-level approaches lie in LiDAR data processing. Work [77–79] used a strategy to project the LiDAR point clouds as 2D representations, and VoxelNet [80] represents another strategy to voxelize the LiDAR data for fusion with camera input.
- **Result-level.** Relatively few works adopt the result-level fusion. [81, 82] are two examples using the weight-based logical mechanisms to integrate the predictions from different modalities.
- **Multi-level.** Multi-level fusion is the trend nowadays for camera-LiDAR fusion because it combines all three other fusion strategies to mitigate their limitations. PointFusion [83] is an example of the result-level and feature-level fusion combination. The result-level lies in the LiDAR filtering, which is based on the 2D bounding boxes generated from images. The feature-level fusion uses ResNet [84] and PointNet [85] to integrate image and point cloud features for object prediction. Van Gansbeke et al. [69] proposed a depth completion network combining signal-level and feature-level fusion. Other multi-level fusion works are [86, 87].

1.2.4 Transformers for AV Perception

A key contribution of this thesis is a transformer-based neural network for camera-LiDAR fusion. Compared with other neural network proposals, transformer [88] has a relatively short history. In the vision field, the pioneering and iconic Vision Transformer [89] was first proposed in 2020. Therefore, the transformer-related works for AV perception were separately briefed in this section.

The deep-learning-based AV perception can be classified as 2D and 3D. For transformer-based works, the 2D perception application are [90–94]. Work [90–93] fo-

cused on road/lane segmentation. BEVSegFormer [90] proposed a multi-camera-based BEV network for road surface segmentation. Work [91] modeled lane marking as regressive polynomials and then used a transformer query algorithm to optimize the polynomial parameters. PersFormer [92] transformed the perspective view to the Bird Eye View (BEV) for precise lane detection. CurveFormer [93] used curve queries to transform the lane detection task to the curve propagation problem. Panoptic SegFormer [94] aimed to object semantic and instance segmentation by a supervised mask decoder and a query decoupling method.

For transformer-based 3D perception research, DETR3D [95] used multi-view images to computer 3D information and relied on backward geometric projection to combine 2D feature extraction and 3D prediction. FUTR3D [96] developed a modality-agnostic feature sampler to integrate multi-modal sensory input for 3D bounding box predictions. Other transformers for 3D object detection including PETR [97] relied on 3D position-aware embeddings and BEVFormer [98] employed spatial and temporal attention layers for BEV features. Work [99, 100] dedicated to 3D object segmentation. TPVFormer [99] transformed the volume to three BEV planes to reduce computation. VoxFormer [100] produced pseudo 3D voxels from 2D images, then performed cross and self-attention mechanisms to 3D voxel queries for object segmentation.

1.3 Motivation and Research Problems

The motivation behind this thesis is to enhance the perceptive capability of AVs by providing a comprehensive framework for deep-learning-oriented and multi-sensor-based segmentation tasks. This work not only focuses on state-of-the-art neural network architectures and deep learning techniques in the computer vision field but also dedicates significant efforts to datasets, which are valuable assets in the Artificial Intelligence (AI) era. Furthermore, there is a proposal for a generic dataset collection framework that aims to allow researchers to collect large-scale sensory datasets in end-to-end applications.

As indicated in Section 1.2, the research problems between the current literature and this work were summarized as follows:

Dataset Collection Framework Compared with using open datasets, it is more important to have the capability to produce custom datasets efficiently to fulfill individual needs. Existing dataset collection works have two problems: (i) lack of multi-modalities and corresponding post-processing. For instance, the fusion of synchronization of multi-sensors includes camera, LiDAR, and radar; and (ii) generic scalability and user-friendly end-to-end practical implementation.

Cross-datasets Domain Adaptation Neural networks, especially FCNs, have been broadly used for traffic object segmentation for many years. However, few works focus on the domain adaptation analysis of FCNs. Most models are presented based on a specific public dataset. Scenes in real traffic scenarios are various and changing rapidly; thus, it is critical for models to maintain high-level performance in different environments.

Attention Mechanism in Sensor Fusion for Scene Interpretation The models that based on the attention mechanism have been the ground-breaker for deep learning technology in recent years. The popular proposal of Vision Transformer (ViT) [89] brought the multi-head-attention mechanism [101] to the computer vision field.

Due to the novelty of transformer [88] networks, there is limited research exploring the potential of transformers in camera and LiDAR fusion for traffic object segmentation.

1.4 Research Objectives and Hypotheses

The primary objective of this research is to develop a thorough pipeline for using deep learning technologies to improve the perception of AVs. To address the research problems identified in Section 1.3, this work focuses on the following objectives:

- RO1** Developing an end-to-end generic multi-sensor dataset collection framework suitable for rapid and large-scale deployment. The framework should cover hardware solutions and post-processing algorithms related to data synchronization, fusion, and transfer.
- RO2** Collecting a custom training dataset for object detection and segmentation tasks. The dataset contains all-weather scenarios featuring the rainy and dark conditions that are common in the TalTech campus, where the iseAuto autonomous shuttle is operated. The organization and format of the dataset should follow the state-of-the-art to guarantee the consistency of future research.
- RO3** Developing an FCN-based network fusing camera and LiDAR data for object segmentation. The model's performance evaluation should focus on the domain adaptation between different datasets and traffic scenes.
- RO4** Adopting the popular ViT network for AV perception purposes. Developing a camera-LiDAR fusion transformer for semantic segmentation in autonomous driving. Conducting the controlled experiments to evaluate the models regarding the backbones and input modalities.

The research hypotheses of this thesis are:

- The autonomous shuttles should have the appreciate sensor configurations to ensure the safety and efficiency.
- The hardware and computational power of autonomous platforms should be maximally utilize to produce, process and share the data.
- The training datasets for AV perception should covers various weather and traffic scenarios.
- The traffic object segmentation tasks should make use of state-of-the-art deep learning technologies.

1.5 Research Tasks and Contributions

In general, the research tasks of this work can be divided into two sub-tasks in sequential order:

- (i) iseAuto Dataset Presentation** This task explores the hardware configurations, establishes the dataset collection standards and structures, and develops the toolbox for sensory data post-processing.

(ii) Camera-LiDAR Fusion FCN and Transformer Networks (CLFCN and CLFT) This task proposes two neural network architectures with different backbones for object segmentation and utilizes the iseAuto training dataset in experiments to evaluate the models.

The first task is practical and demonstration-oriented, aims to have the iseAuto dataset produced and the collection framework prototyped in real traffic scenario. The second task is theoretical and performance-pursued, explores various neural network backbones. Two network architectures based on FCN and transformer were proposed as the results of this stage to compete with other cutting-edge models. Table 2 indicates the contributions of articles included in this thesis correspond to each research objective mentioned in Section 1.4.

Table 2: Correlation between research objectives and the included articles.

Objectives	Article I	Article II	Article III	Article IV
1	✓		✓	
2		✓		
3		✓		
4				✓

2 Cyber-Physical Experimental Platforms

This section focuses on the hardware configurations for AV perception and presents the training datasets used for deep-learning-based object detection and segmentation.

Firstly, there is an analysis of range sensor deployment specifically for autonomous shuttles to reduce the blind zone. Secondly, this section presents a testing platform with a multi-sensor perceptive system installed. The platform was used for demonstrating and evaluating the end-end-end framework for data collection and processing. The details of hardware setup, operating system, and public access are included in this section. At last, this section introduces two training datasets for scene interpretation. Both datasets were thoroughly used in this work to train and test the two proposed neural networks for object segmentation (corresponding to the **RO3** and **RO4** in Section 1.4). Specifically, the production of the iseAuto dataset follows the regulations and toolboxes from the dataset collection framework, which is the **RO2** in Section 1.4.

2.1 Range Sensor Deployment for Autonomous Shuttles

The autonomous shuttle is an AV branch, which usually has a shuttle appearance and runs at a relatively low speed. Autonomous shuttle is an autonomous solution for the 'last-mile' mobility domain in specific urban transportation scenarios [102], for instance, the movement between the transportation hub to the final destination.

Currently, several autonomous shuttles have already been successfully demonstrated and validated in large-scale production. As shown in Figure 2, cubic design is common for autonomous shuttle appearance. Moreover, peculiar traffic scenes, such as children suddenly running across the streets, are highly likely to happen to autonomous shuttles due to where they are operated. Therefore, sensor deployment is critical for autonomous shuttles to improve their perceptive capability and reduce the blind zone.



Figure 2: Illustrations of several commercial autonomous shuttles. From left to right are Apollo Minibus [103], Navya Evo [104], Easymile EZ10 [105], and AuveTech MiCa [106].

Range sensors are broadly used on AVs for emergency detection and blind zone deduction. Specifically, LiDAR sensors attract the most interest because of their direct object detection and wide FoV. This section presents the LiDAR sensors deployment for the iseAuto shuttle, which plays the essential role in this thesis to validate the end-to-end dataset collection framework (**RO1**) and produce the iseAuto training dataset (**RO2**).

The iseAuto shuttle depends on the laser-based sensors to actively perceive the environment. Figure 3 illustrates the locations and orientations of all exterior laser-based range sensors. In total, there are five LiDAR sensors installed on the exterior of the iseAuto shuttle, including one Velodyne VLP-16 Puck, one Velodyne VLP-32C, two RoboSense RS-Bpearl, and one Benewake CE-30C. The choice of models and locations was based on practical tests, with the aim of utilizing all LiDAR sensors and reducing the expense efficiently. The primary range sensor is the front-top Velodyne VLP-32C, which has 32 channels in vertical. A Velodyne VLP-16P LiDAR sensor with 16 vertical channels was installed on the rear-top. Both Velodyne LiDAR sensors were tilted to the ground to reduce the

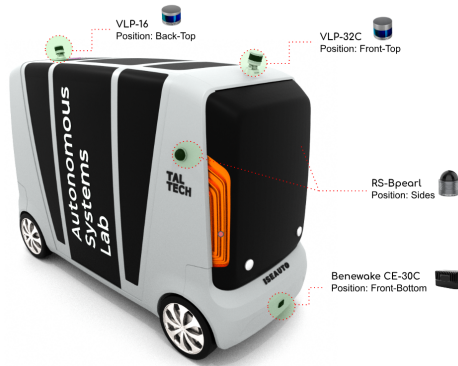


Figure 3: Exterior LiDAR sensors layout for the iseAuto shuttle. Adopted from the Article II.

interference patterns and shadowed azimuth ranges that might appear when using multiple Velodyne sensors on top of the vehicle. Two Robosense RS-Bpearl LiDAR sensors were installed on two sides of the shuttle to cover the blind zone that top LiDAR sensors cannot detect. The RoboSense RS-Bpearl LiDAR sensor has a unique 90° vertical FoV, which makes it suitable for installation on the shuttle's side. Figure 4 shows the point clouds produced by top Velodyne and side RS-Bpearl LiDAR sensors. The critical front-bottom blind zone for the iseAuto shuttle (shown in Figure 5(a)) was covered by the solid-state Benewake CE-30C LiDAR sensor, which has no internal rotational mechanism.

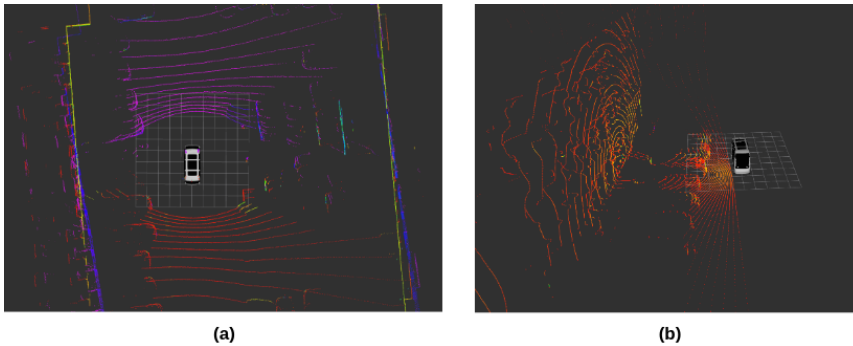


Figure 4: LiDAR point clouds in real outdoor environment. (a) is the combination of front-top and rear-top Velodyne LiDAR. (b) is from the right side RS-Bpearl LiDAR. Adopted from the Article I.

2.2 Multi-Sensor Perception and Collection Framework

Nowadays, a multi-sensor system is an essential requirement for AV to ensure reliability and safety. On the one hand, the involvement of sensors with different characteristics raises sensor calibration and synchronization issues. Moreover, sensor management and integration have become important aspects of AV's perception system. On the other hand, the advancement of the vehicle's onboard computational power allows for partially assigning data post-processing, such as decompressing, denoising, and fusion, to the vehicle's computer.

Considering the new requirements for AV perception systems, this thesis presents a

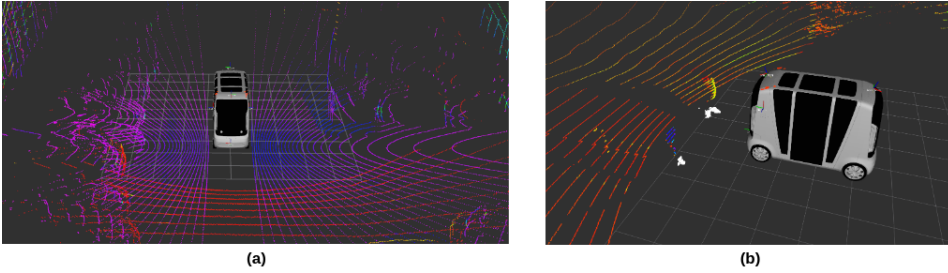


Figure 5: Front-bottom blind zone for iseAuto shuttle. (a) shows the point clouds from front-top and two sides LiDAR sensors. The empty space illustrates the blind zone. The white points in (b) are from the front-bottom solid state LiDAR sensor. Adopted from the Article I.

state-of-the-art multi-sensor framework that not only offers perception to AV but is also integrated with the algorithms for data fusion and collection. The framework contains proprioceptive sensors, such as Global Positioning System (GPS) sensors, to record vehicle positions and exteroceptive sensors, such as cameras and LiDAR sensors, to capture texture and distance information. Innovatively, this framework makes use of the advantages of radar sensors to reinforce the detection of moving objects in LiDAR point clouds. Furthermore, the framework includes the algorithms for point clouds projection to achieve the camera-LiDAR-radar fusion. Figure 6 presents the overview of framework architecture and data flow. It is important to note that this section focuses on introducing the hardware platform designed for testing and evaluating the framework. The details of methodologies related to sensor calibration, synchronization, and post-processing are available in Section 3.

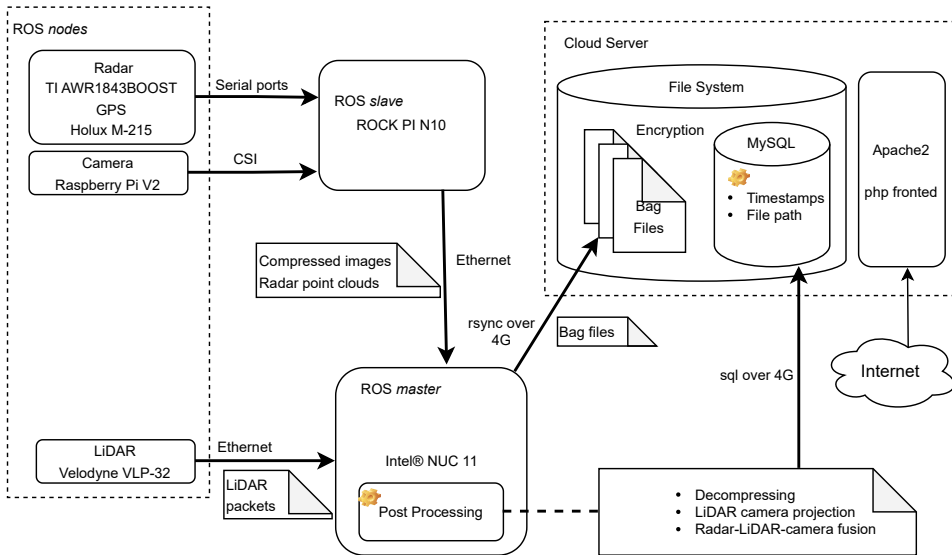


Figure 6: Overview of the framework architecture. The bold arrow-pointers represent the data flows and corresponding communication protocols. Adopted from the Article III.

2.2.1 Hardware for Framework Validation

The primary purpose of the framework is serving the iseAuto shuttle (as shown in Figure 3) to coordinate all onboard perceptive sensors. Moreover, the framework was designed as a generic end-to-end solution that is suitable for all kinds of urban autonomous platforms. In the development process, the testing and validation of the framework rely on a Mitsubishi i-MiEV car with a portable top mount that has all sensors mentioned in Figure 6 integrated.

There are five sensors and two processing units used on the testing platform to validate the framework. Five sensors include one LiDAR, one camera, one GPS, and two radars. Two processing units are one Intel® NUC 11 with a Core™ i7-1165G7 Processor as the main computer and one ROCK PI N10 with four cortex-A53 processors as the supporting computer. As indicated in Figure 6, the NUC 11 handles most of the operations, including raw sensory data subscription, data post-processing, and communication with a remote cloud server. The ROCK PI N10 stays outside the vehicle in a protective box (shown in Figure 7c), together with the camera, radar, and GPS sensors that lack water and dust prevention.

Figure 7 and Table 3 show the sensor layout and restricted specifications for the testing platform. The LiDAR sensor is the Velodyne VLP-32C, which has 32 laser beams and 40° FoV in vertical. The range of the LiDAR sensor was limited from 1.4 to 200 meters. Two Texas Instruments mmwave AWR1843BOOST radar sensors were used for the testing platform. The radar sensors were calibrated to be mainly reactive to dynamic objects, which can be simplified as 'vehicle' and 'human', two classes for urban transportation. Therefore, the specification of the first radar sensor is preferable for detecting the 'vehicle' class, and the second radar sensor is for the 'human' class. The camera sensor for the testing platform is the Raspberry Pi V2 camera with a wide 160° diagonal FoV. The camera was restricted to operate at 15 Hz in resolution 1920x1080.

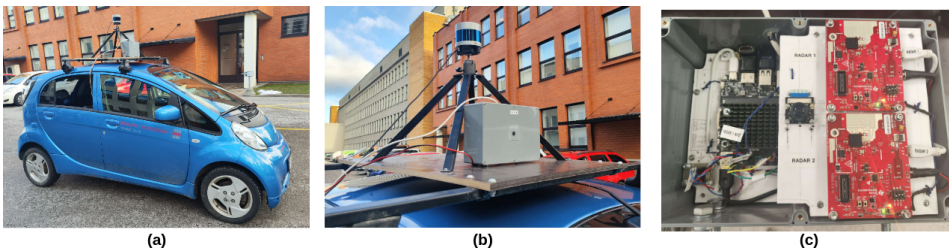


Figure 7: Sensor layout on testing platform. (a) is the overview of testing vehicle and top mount. (b) is the sensor mount. (c) shows the camera, radar, and GPS sensors inside the waterproof box. Adopted from the Article III.

Table 3: Restricted sensor specifications

Sensor Type	FoV	Range/Resolution	Frequency
LiDAR	40° (V)	1.4 - 200 m	10 Hz
radar 1	90° (H)	45 m	76 GHz
	15° (V)	15.73 m/s (radial)	
radar 2	90° (H)	30 m	76 GHz
	15° (V)	3.07 m/s (radial)	
camera	160° (D)	1920 x 1080	15 Hz

2.2.2 Software System and Server

The data capturing and communication of the dataset collection framework follow the Robot Operating System (ROS) regulations. The distributed computing environment of the ROS allows operating nodes to spread across multiple machines under the same master. Thus, the data in ROS format from different nodes is visible to the entire network. In the platform, the main computer, NUC 11, hosts the ROS master, which establishes communication between all other nodes. The supporting computer ROCK PI N10 acts as an ROS slave, and hosts the nodes to initiate camera, radar, and GPS sensors. To reduce the data transfer latency between main and supporting computers, the Gigabyte local area network was established across the whole testing platform. In practical tests, the average delay between the main and supporting computers when no data is transferred is 0.441 ms. In comparison, the average delay increases to 0.49 ms when the camera, radar, and GPS sensors fully operate and transfer the data to the main computer. In practice, such a minor latency caused by physical connection is neglected for sensor fusion. For instance, the camera and LiDAR sensor synchronization error are bounded from -6 to 8 ms in Waymo Open dataset [22].

As shown in Figure 6, another contribution of this framework is a cloud server that can adapt to other autonomous platforms. The connection between the vehicle and the cloud server relies on the mobile network. Raw and processed perceptive data collected by the testing platform were stored in a database. Moreover, the timestamp labels and file paths of data were simultaneously created in the database for public query tasks.

2.3 Training Datasets

In this work, the research objective related to scene interpretation for AV is achieved by exploring and analyzing cutting-edge neural networks for object detection and segmentation. A comprehensive dataset covering various autonomous driving scenarios is critical for all neural network procedures, from developing to training and then evaluating.

As discussed in Section 1.5, the thesis involves two models that segment objects by fusing camera and LiDAR data. The first model (CLFCN) is based on FCN, which has a relatively long history and has been widely explored. Therefore, the research of CLFCN focuses on the domain adaptation analysis between the datasets with different characteristics. Two datasets were used in domain adaptation experiments for CLFCN. The first one is the Waymo Open dataset, and the second is the iseAuto dataset (**RO2**) that was collected by the iseAuto shuttle under the end-to-end multi-modal dataset collection framework (**RO1**).

The second model (CLFT) is based on ViT, one of the most popular neural network proposals in recent years. Due to CLFT's novelty in terms of its LiDAR data processing strategy for object segmentation tasks, the experiments for CLFT aim to benchmark it with other models regarding object segmentation accuracy. The Waymo Open dataset was used in coherent controlled benchmark experiments.

2.3.1 Waymo Open Dataset

Waymo Open dataset is a multi-modal dataset recorded by industrial-strength sensors. It consists of 1150 sequences spanning various illuminations, and the LiDAR data is provided as the range images with vehicle pose integrated into each pixel. The annotations of the Waymo Open dataset are represented as 2D and 3D bounding boxes in camera and LiDAR data, respectively. For CLFCN domain adaptation and CLFT benchmark experiments in this work, 110 sequences were randomly selected. Since each Waymo sequence spans 20 seconds and records samples at 10 Hz, there are 2200 frames with manual-labeled

annotation. The process towards Waymo’s 3D ground-truth bounding boxes follows the algorithms presented in Section 3.2.1, the dense point clouds of objects are projected onto the camera plane as annotations (shown in Figure 8(c)) for training and testing.

The robustness and efficiency in challenging illumination and weather conditions are critical for AV scene interpretation. The evaluation of neural network models for traffic object segmentation should cover various real-world situations. Therefore, the Waymo Open dataset was partitioned into sub-categories based on the illumination and weather: day-dry, day-wet, night-dry, and night-wet. Table 4 presents the details of sub-categories for the Waymo Open dataset.

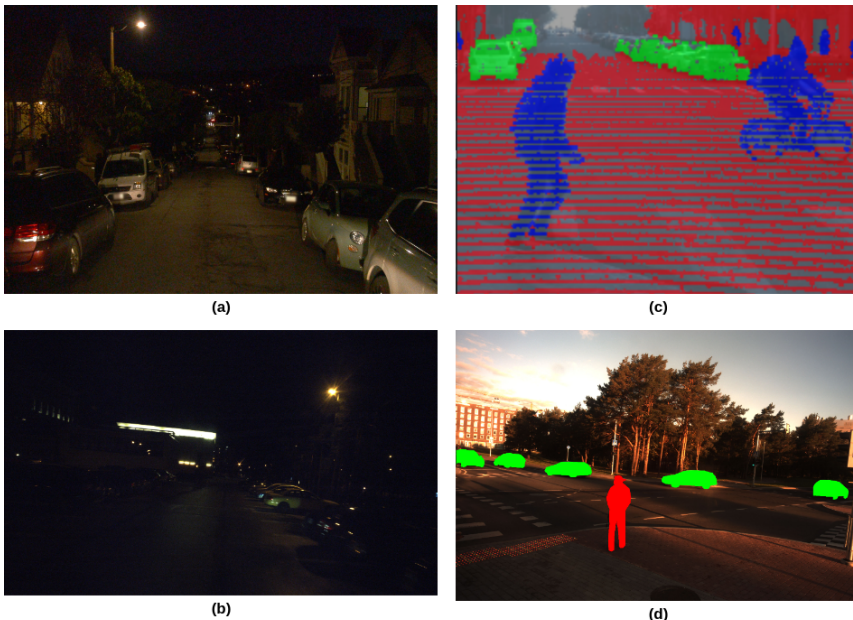


Figure 8: Examples of RGB and annotation images of the Waymo and iseAuto dataset. (a) and (c) are from the Waymo Open dataset. (b) and (d) are from the iseAuto dataset

Table 4: Amount of frames for each sub-category in the Waymo and iseAuto datasets

	Day-Dry	Day-Wet	Night-Dry	Night-Wet
Waymo Open Dataset	14940	4520	1640	900
iseAuto Dataset	2000	2000	2000	2000

2.3.2 iseAuto Dataset

The iseAuto Dataset contains the camera and LiDAR data collected by the iseAuto shuttle under the dataset collection framework proposed in this thesis. The specifications and deployment of sensors on the iseAuto shuttle were introduced in Section 2.1. The architecture of the dataset collection framework was presented in Section 2.2. In addition to testing the iseAuto shuttle’s range sensors and validating the dataset collection framework, another motivation to produce the iseAuto dataset is to make up for the shortcomings of the Waymo Open dataset to evaluate the models’ performance and domain adaptation capability comprehensively. The advantages of the iseAuto dataset compared with the

Waymo Open dataset are as follows:

- The locations to produce the Waymo Open dataset are Phoenix, Mountain View, and San Francisco, where the climate and urban transportation differ greatly from Estonia. The iseAuto dataset was collected at TalTech campus, which general illumination is lower than the Waymo Open dataset (Figure 8 (a) and (b) show the comparison). Therefore, the iseAuto dataset is more suitable for analyzing the model's performance in dark environments.
- Most of the sequences in the Waymo Open dataset are recorded in light and sunny conditions; thus, the unbalanced data allocation of sub-categories poses challenges for model training. For the iseAuto dataset, all four sub-categories (day-dry, day-wet, night-dry, and night-wet) have 2000 frames, which guarantees the models can learn the same amount of knowledge from different weather and illumination conditions.
- The object annotations in the Waymo Open dataset are based on the LiDAR point clouds projection onto images, which results in some pixels for the objects having no labels (shown in Figure 8 (c)). The ground-truth annotations in the iseAuto dataset were manually selected from the image. Therefore, the object masks are solid-filled and contain detailed contour information (shown in Figure 8 (d)).

3 Methodologies

This section presents the technical details of two aspects of this thesis: multi-modal dataset collection framework and camera-LiDAR fusion neural networks.

The first subsection introduces the protocols of multi-sensor calibration and synchronization for end-to-end dataset collection framework (**RO1**). The calibration and synchronization methods proposed in this work focus on the sensors operating in a discontinuous mode. For instance, radar sensors deployed on testing platform are only reactive to moving objects, resulting in a heterogeneous update rate. The second subsection concentrates on the signal-level fusion of camera, LiDAR, and radar sensors. A thorough camera-LiDAR-radar fusion algorithm was proposed as the backend of the end-to-end dataset collection framework (**RO1**). The last subsection provides the architecture details of CLFCN and CLFT (**RO3** and **RO4**), two camera-LiDAR fusion neural networks for object segmentation.

3.1 Sensor Calibration and Synchronization

Sensor calibration and synchronization are critical for any autonomous platform with a multi-sensor system. Specific to perceptive sensors, the calibration requires to acquire sensors' intrinsic and extrinsic information, and the synchronization aims to compute the data-pairs with the closest absolute timestamps from the sensors operating at different acquisition rates. As part of the toolbox of the dataset collection framework, the sensor calibration and synchronization methods proposed in this thesis focus on the camera, LiDAR, and radar sensors. The corresponding visualized results were produced by the sensor models mentioned in Section 2.2.1.

3.1.1 Sensor Intrinsic and Extrinsic Calibration

The essence of intrinsic calibration is retrieving the geometric match of features' position and orientation in the real world and the relative coordinates detected by the sensors. The intrinsic calibrations are conducted independently for each sensor. Due to the dataset collection framework focuses on the camera, LiDAR, and radar sensors, the intrinsic calibration processes for these three sensors in the framework's toolbox are following:

- **Camera:** The literature on the intrinsic calibration of the camera and LiDAR sensors is rich [107, 108]. The most common intrinsic calibration methods for camera sensors rely on photogrammetry [109], in which planner patterns with precise geometry are used during the calibration. The open-source ROS 'camera_calibration' package was integrated into the dataset collection framework's toolbox for calibrating pinhole and stereo cameras. The 'camera_calibration' package is built upon the OpenCV camera calibration modules but exclusively provides the interface for parameter tuning.
- **LiDAR:** The LiDAR sensors currently used on AV are highly integrated and industrialized. The intrinsic calibration of LiDAR sensors is usually conducted during manufacturing to improve accuracy. For instance, referring to the manual book, the range accuracy of the Velodyne VLP-32C LiDAR sensor used by the iseAuto shuttle and testing platform is no more than ± 3 cm. Therefore, no LiDAR sensor intrinsic calibration method was included in the framework.
- **Radar:** The existing literature on radar calibration concentrates on three aspects: i) coordinate matching of radar points and image objects [110], ii) radar points filtering to reduce the noise and faulty defections [111], iii) error corrections to compensate mathematical measurement errors [112]. The intrinsic calibration of radar sensors

follows the second strategy. Most of the points for static objects were filtered out. As shown in Figure 9, only the points representing dynamic objects (color dots) were kept in the frame.

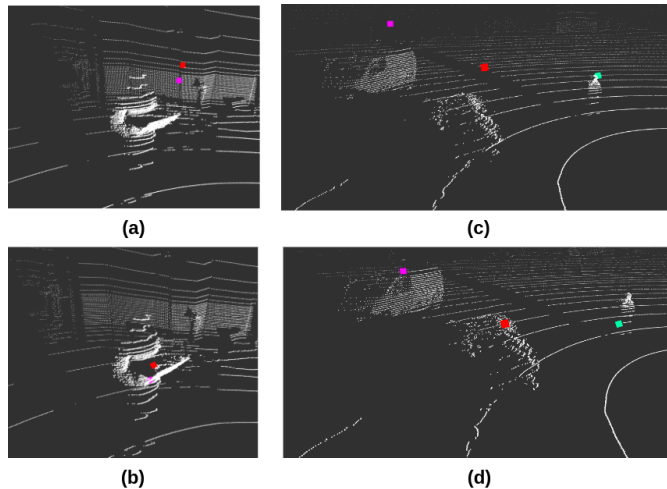


Figure 9: Comparison of Euclidean distance between LiDAR and radar points with and without the extrinsic calibration. Color dots are radar points; white dots are LiDAR points. (a) and (c) are without calibration, (b) and (d) have the radar points transformed based on the extrinsic calibration matrix. Adopted from the Article III.

The extrinsic calibration of a multi-modal sensor system estimates the transformation between the different sensor coordinates. The precise transformation matrix containing rotation and translation information of all sensors is critical to signal-level sensor fusion. The extrinsic calibration modules in the dataset collection framework provide the camera-LiDAR and LiDAR-radar extrinsic calibration. The camera-LiDAR calibration module was inspired by the work [113]. The planner patterns, such as checkerboard, are required in calibration. The LiDAR point and corresponding image pixel were manually paired in the process. The details and issues that needed to be noted are available in work [27] (**Article III** included in this thesis). The LiDAR-radar extrinsic calibration in the framework was conducted by a ROS-based tool that provides the Euclidean distance visualization of LiDAR and radar sensors' point clouds data. Manual tuning was required to ensure the point clouds clusters overlapped. To increase the calibration accuracy, it is recommended to carry out the calibration in the environment with the least interference and use the high reflective objects for radar sensors, such as metallic surfaces. Figure 9 illustrates the result of LiDAR-radar extrinsic calibration. The color and white points represent the radar and LiDAR point clouds, respectively. The pictures in the first row are without the extrinsic calibration, and the pictures in the second row show the result after implementing the transformation to radar points.

3.1.2 Sensor Synchronization

The perceptive sensors such as camera, LiDAR, and radar sensors operate at different frequencies. The camera sensors usually have high frequency, and the LiDAR sensors with internal rotating mechanisms scan at a rate of no more than 20 Hz. The frequency of radar sensors varies in different situations. For example, in the testing platform used to validate

the dataset collection framework, the radar sensors were configured to be only reactive to moving objects; thus, the acquisition rate is heterogeneous.

The existing sensor synchronization methods, such as [114], are designed to process the data streams with a constant update rate. Work [114] selects the latest message alongside the timeline as the reference-frame, then find the nearest message from another data stream to compose the synchronized message-pair. If there is no message in another data stream within the defined time threshold, it will discard the current reference-frame and move to the next message. However, such an algorithm only works for the sensor modalities with homogeneous update rates, such as camera and LiDAR sensors. This algorithm is unsuitable for the situation of sensor modalities with heterogeneous update rates, because it always picks the nearest message to the synchronized message-pair as the reference-frame. Figure 10(a) gives an example of how the algorithm in [114] synchronizes the multi-sensor modalities. The camera and LiDAR sensors operate at 15 and 10 Hz, respectively. The radar sensor works in heterogeneous mode. The reference-frame (red dot) was not fixed to the same sensor modality, resulting in significant sensor synchronization errors.

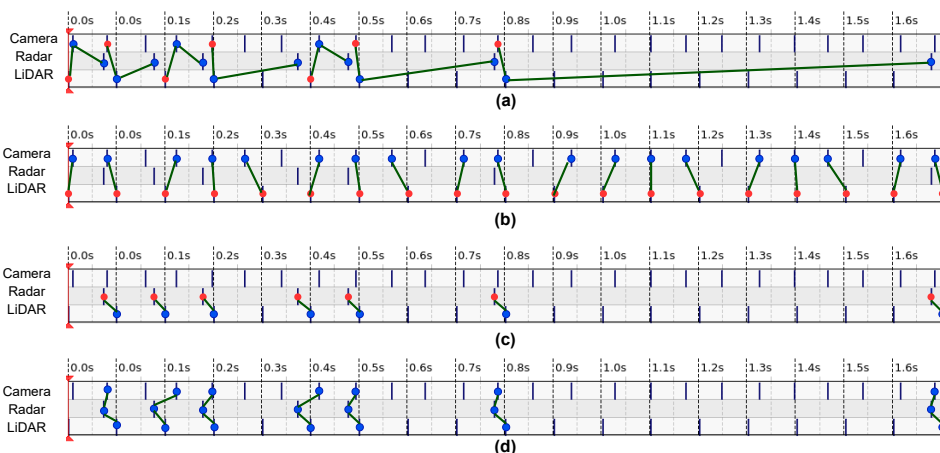


Figure 10: The blue and red dots represent the sensor messages, while the red dots are reference point picked by the algorithm for synchronization. Adopted from the Article III.

The gaps in existing techniques have been identified, and the multi-sensor synchronization approach proposed in the dataset collection framework aims to synchronize the sparse and unevenly scattered messages. The key point of the novel sensor synchronization method is dividing the synchronization process into three steps. The first step is the camera-LiDAR synchronization, and the second is the LiDAR-radar synchronization. The first two steps were illustrated in Figure 10(b) and (c), respectively. At last, a thorough camera-LiDAR-radar synchronization was conducted based on the results of camera-LiDAR and LiDAR-radar synchronization, as shown in Figure 10(d). The advantage of this algorithm is that the reference-frame in each synchronization step can be fixed to the sensor modality with lower frequency, which is the LiDAR sensor in camera-LiDAR synchronization, and the radar sensor in LiDAR-radar synchronization. The sensor synchronization method in this work keeps the advancements of all sensors such as the density and consistency of the camera and LiDAR data, while also provides the opportunity to synchronize the sparse and irregular radar data.

3.2 Signal-level Camera-LiDAR-radar Sensor Fusion

The sensor fusion backend is an essential composition of the multi-modal dataset collection framework proposed in this thesis. Due to the dataset collection framework's limited computational resources and application scenarios, the sensor fusion backend focuses on the signal-level fusion of camera, LiDAR, and radar data. The signal-level fusion is also expressed as low-level or early-stage fusion in literature [115] as it is the fusion of raw data. For instance, integrating 3D geometric coordinates and image pixel values for camera-LiDAR fusion. For the sensing modalities with the same work principles, such as LiDAR and radar sensors, the signal-level fusion usually targets spatial coordinate matching. The camera-LiDAR-radar fusion algorithms in this work follow the sensor synchronization strategy discussed in Section 3.1.2 and divide the process into three steps. The first step is the camera-LiDAR fusion, which aims to acquire the maximum amount of fusion results. The second step is the LiDAR-radar fusion, the point clouds clusters of moving objects in LiDAR data were highlighted and assigned with the velocity information. The last step combines the first two fusion stage results to achieve the thorough camera-LiDAR-radar signal-level fusion. The following subsections introduce the details of camera-LiDAR and LiDAR-radar fusions.

3.2.1 Camera-LiDAR Fusion

The essence of signal-level camera-LiDAR fusion is representing the 3D LiDAR point clouds as the 2D-based feature maps by 3D-2D projection. In general, there are three projection strategies:

- **Spherical Map.** The 3D LiDAR points are projected onto a front-view sphere with azimuth and zenith characters kept. The dense projection results have advantages in feature segmentation [116] but are unsuitable for deep-learning-based feature/multi-level fusion because of the different dimensions of camera images.
- **Camera-plane Map.** The perspective projections of LiDAR point clouds onto camera planes provide maps of the same size as camera images. Thus, the results can be fused in neural networks. However, there is a need to up-sample the sparse feature maps [117–119].
- **BEV Map** The BEV projections of LiDAR points clouds provide the objects' localization and dimension information. The BEV results are broadly used in 3D perception [120] but not applicable for 2D scene interpretation, which is the focus of this thesis.

The camera-LiDAR fusion in this work adopts the second strategy that projects the 3D LiDAR point clouds onto the camera plane in XY, YZ, and ZX channels, shown in Figure 11. In general, there are three steps in this process.

The first step is the transformation of LiDAR point clouds to the camera coordinate system based on the camera-LiDAR extrinsic calibration results. The process follows the equation:

$$[x_t, y_t, z_t]^T = (r \ p \ y) \left([x_i, y_i, z_i]^T - [x_c, y_c, z_c]^T \right) \quad (1)$$

where x_t , y_t , and z_t are the 3D point coordinates seen after transformation (seen from the camera frame); x_i , y_i , and z_i are the 3D point coordinates before transformation (seen from the LiDAR frame); x_c , y_c , and z_c denote the camera frame location coordinates. r , p , and y are the Euler rotation matrices to the camera frame, which are represented as the

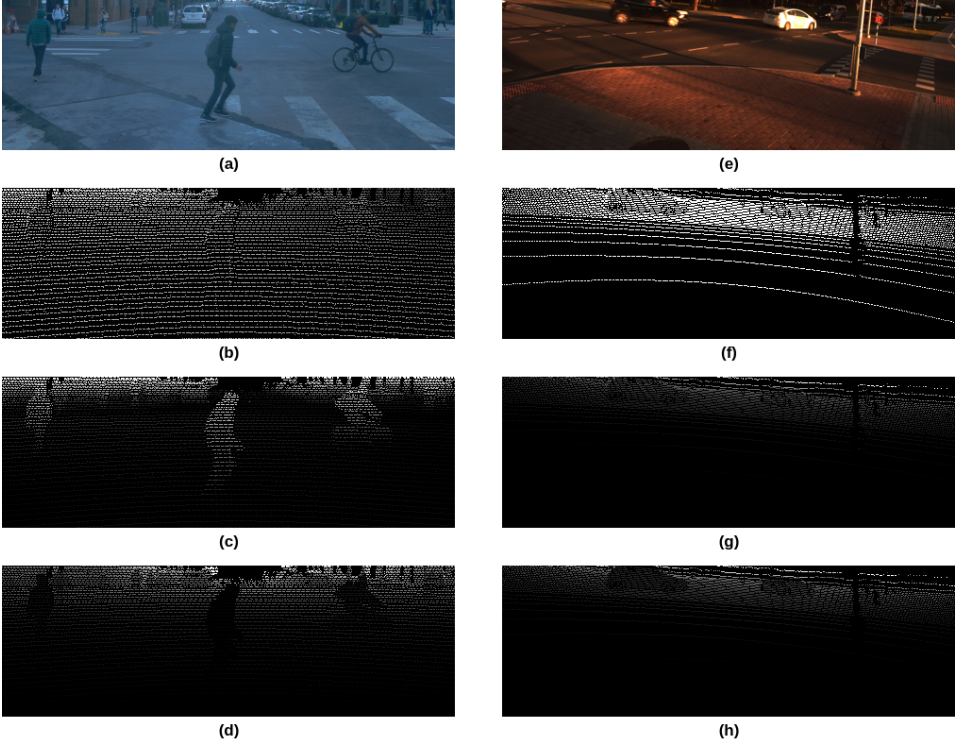


Figure 11: The LiDAR point clouds projection onto XY, YZ, and ZX camera planes. The first column (a-d) was extracted from the Waymo Open dataset. The second column (e-h) was extracted from the iseAuto dataset. In each column, from top to bottom, are RGB, XY, YZ, and ZX images, respectively. It should be noted that for visualization purposes, the grayscale intensity in all camera-plane images is proportionally scaled based on the numerical 3D coordinate values of the LiDAR point.

following equations:

$$r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\rho) & \sin(\rho) \\ 0 & -\sin(\rho) & \cos(\rho) \end{bmatrix} p = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} y = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where (ρ, θ, ϕ) represents the corresponding Euler angles.

The second step is projecting the transformed 3D LiDAR points as 2D image pixels onto camera plane, which follows the equation:

$$(u, v, 1)^T = \begin{bmatrix} f_x & 0 & \frac{w}{2} \\ 0 & f_y & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix} (x_t, y_t, z_t)^T \quad (3)$$

where u and v are column and row positions of the resulting 2D image pixels; f_x and f_y denote camera's horizontal and vertical focal length; w and h represent image resolution; x_t , y_t , and z_t are same as in Equation 1, which are transformed 3D point coordinates.

The last step is selecting the points that fall in the camera view and discarding the rest. Camera-plane maps denoted as XY, YZ, and ZX of LiDAR point clouds are generated in this step. The pixels of camera-plane maps with corresponding LiDAR points are assigned with x , y , and z coordinate values, while the rest are populated with zero. Algorithm 1 shows the detailed procedure of this step.

Algorithm 1 LiDAR points filtering and image pixel values population. Adopted from the Article IV.

Input: LiDAR point 3D coordinates L , projected LiDAR point coordinates P , image resolution w and h .

Output: LiDAR projection footprints XY , YZ , and ZX .

- 1: $idx = \text{argwhere}(P < \{w, h, +\infty\} \ \& \ P \geq \{0, 0, 0\})$
 - 2: $XY[w \times h] \leftarrow 0$
 - 3: $YZ[w \times h] \leftarrow 0$
 - 4: $XZ[w \times h] \leftarrow 0$
 - 5: $XY[idx] = L[idx, 0]$
 - 6: $YZ[idx] = L[idx, 1]$
 - 7: $XZ[idx] = L[idx, 2]$
-

3.2.2 LiDAR-radar Fusion

The goal of the signal-level LiDAR-radar fusion module in the dataset collection framework is to utilize the radar sensors' advantages in detecting moving objects and then integrating the moving object information into the LiDAR points cloud data. As a result, the LiDAR point clouds of moving objects were selected and assigned with velocity based on the radar detection results. Figure 12 illustrates the LiDAR-radar fusion process, which can be summarized as following four sequences:

1. Transforming the radar points from radar frame coordinate to LiDAR frame coordinate.
2. Applying the density-based spatial clustering of applications with noise (DBSCAN) algorithm [121] to LiDAR point clouds to filter the objects' point clusters.
3. Looking up the LiDAR point clusters with the nearest Euclidean distance to the transformed radar points.
4. Assigning the velocity readings from the radar sensor to the selected LiDAR point clusters that represent the moving objects.

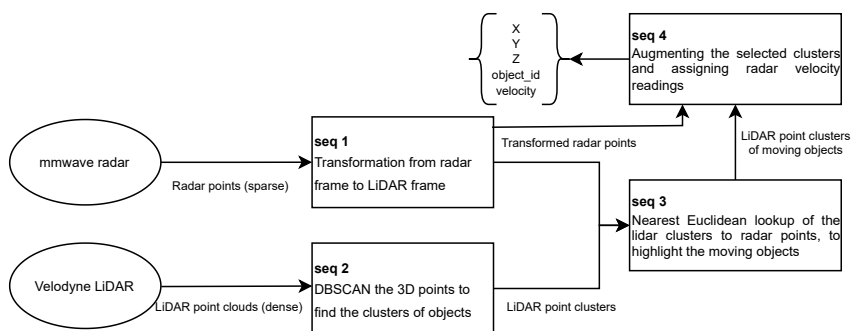


Figure 12: The workflow of signal-level LiDAR-radar fusion. Adopted from the Article III.

Following the algorithms described in Section 3.2.1, the resulting LiDAR clusters of moving objects were projected onto the camera plane to achieve the final camera-LiDAR-radar

fusion. Figure 13 visualizes some procedures of the camera-LiDAR-radar fusion module proposed by dataset collection in this thesis. Figure 13(a) illustrates the first two sequences of LiDAR-radar fusion. The green and red dots represent the radar points before and after the transformation, respectively. The blue dots are filtered LiDAR point clusters of moving objects. Figure 13(b) shows the projection of radar data onto the camera plane. In comparison, Figure 13(c) is the camera-plane projection of LiDAR point clusters concluded based on radar detection results.

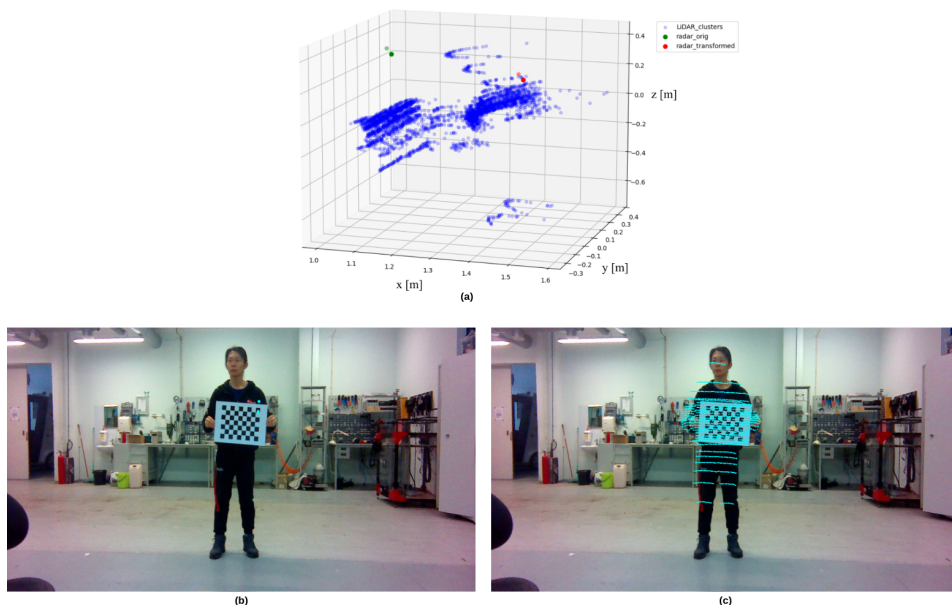


Figure 13: Illustrations of camera-LiDAR-radar fusion. (a) shows the relative locations of original radar points (green), transformed radar points (red), and LiDAR point clusters of the moving object (blue), (b) is the transformed radar points projection onto the camera plane, (c) visualizes the eventual outcome of camera-LiDAR-radar fusion, which is the object's LiDAR point clusters projection onto the camera plane. Adopted from the Article III.

3.3 Neural Networks for Object Segmentation

The perception of autonomous driving is a comprehensive domain. As a fundamental task of AV perception, scene interpretation aims to identify the objects and analyze their relationships with other scene contexts. This thesis focuses on using neural networks to improve the AVs' scene interpretation capability because the sensory dataset is the direct input of the neural networks. On the one hand, the corresponding neural network experiments are the ideal application scenarios for training datasets proposed in this work. On the other hand, the unexplored potential of AI for autonomous driving is a strong motivation for future research. This work involves two neural networks that fuse camera and LiDAR data for object segmentation. The first one is based on FCN and was initially developed by Caltagirone et al. [74] for road surface detection. The application of this network is mainly for domain adaptation analysis between the Waymo Open dataset and the iseAuto dataset. The second ViT-based neural network was first time proposed in [122] (include in this thesis as the **Article IV**) and was used to compete with other cutting-edge models for traffic object segmentation tasks.

3.3.1 Camera-LiDAR Fusion Convolutional Neural Network (CLFCN)

The CLFCN network is based on the popular ResNet50 [84] that consists of 21 layers in its encoder-decoder structure. Caltagirone et al. [74] proposed three fusion strategies, namely, early, late, and cross that based on the layer depth to concatenate camera and LiDAR representations. Figure 14 illustrates how the camera and LiDAR were fused in these three fusion strategies.

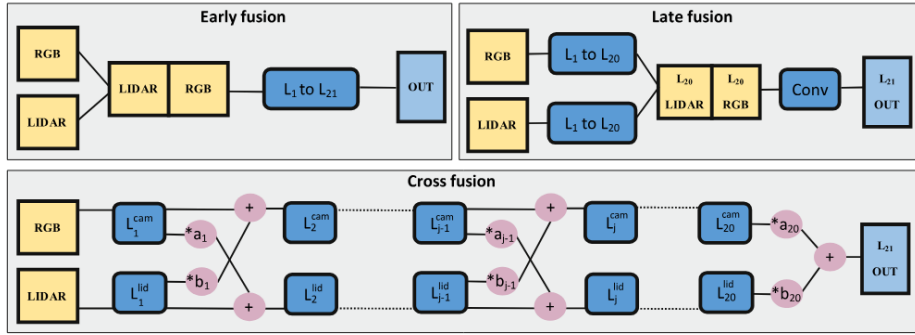


Figure 14: The graphic illustrations of three fusion strategies in [74].

Inspired by the late-fusion strategy, the CLFCN divides the ResNet50 layers into five stages and executes the concatenation of camera and LiDAR feature representations after stage 4. The concatenated and single modalities were forwarded to the last stage separately. Thus, there are three sub-models in the network to compute loss for the camera, LiDAR, and fusion modalities concerning mutual ground truth. Figure 15 shows the workflows of three sub-models of the CLFCN network.

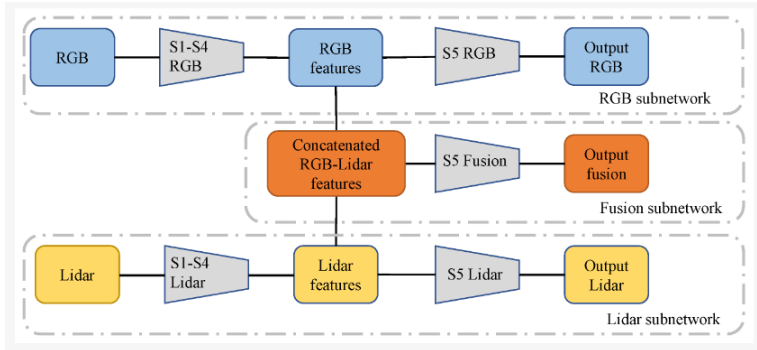


Figure 15: The illustrations of the CLFCN sub-models' workflow [77].

3.3.2 Camera-LiDAR Fusion Transformer (CLFT)

The CLFT network maintains the generic encoder-decoder structure of the transformer but invokes the progressive-assemble strategy from ViT on a double-direction network to process the camera and LiDAR data in parallel. The results of two directions for camera and LiDAR modalities are then integrated into the decoder layer following the cross-fusion strategy. As far as the latest literature reviews [123, 124], CLFT is the first open-source transformer-based network that adopts the camera-plane-projection strategy discussed

in Section 3.2.1 to process the LiDAR data for 2D object semantic segmentation. The projection of LiDAR point clouds in XY, YZ, and ZX camera-plane maps were concatenated as a three-channel representation and then amalgamated with RGB camera data into a unified data representation for subsequent processing.

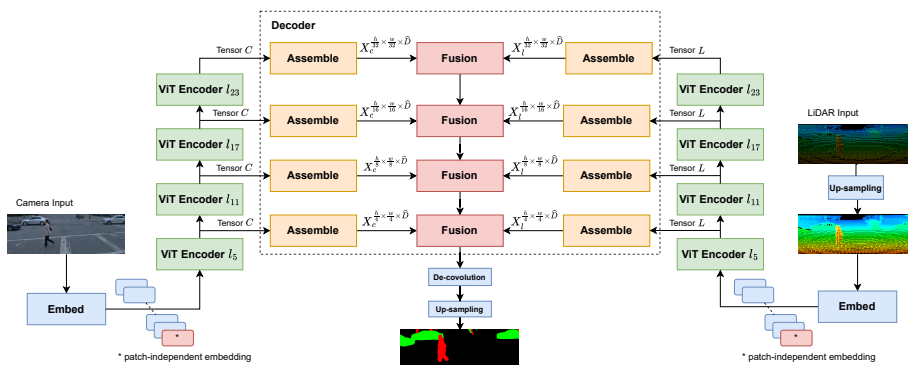


Figure 16: The overall architecture of the CLFT network. Adopted from the Article IV.

Figure 16 shows the overall architecture of the CLFT network. The double-direction for camera and LiDAR modalities was visualized from two ends of the diagram. From left to right is the camera data flow to the ViT encoder, while the LiDAR data flow is represented from right to left. The name of the CLFT encoder follows the ViT’s conventions, which are ‘CLFT-base,’ ‘CLFT-large,’ ‘CLFT-huge,’ and ‘CLFT-hybrid’. The ‘base,’ ‘large,’ and ‘huge’ variants use the patch-based embedding method, which divides the input image into fixed-size non-overlapping patches. The ‘hybrid’ variant adopts the strategy to extract feature patches from images’ CNN feature maps as input tokens for the transformer. The details of encoder variants’ parameters, such as layer amount, feature dimension, patch size, etc. are available in the work [122] (Article IV).

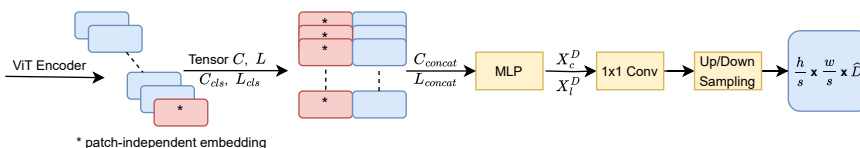


Figure 17: Assemble architecture for each transformer decoder block. Adopted from the Article IV.

The CLFT decoder was consisted of assemble and fusion two stages. Figure 17 illustrates the detailed assemble stage workflow. In general, the assemble stage can be divided into two steps. Algorithm 2 shows the detailed process of the first step. It first replicates and concatenates the patch-independent ‘classification token’ to all other tokens individually, then applies the GELU non-linear activation [125] to the concatenated representations. The ‘classification token’ is similar to the ‘class token’ concept in BERT [126]. The second step takes the concatenated results from the first step as input, and up or down-sample them to the same resolution based on the layer depth. The resolution was anchored to the input image size. Thus, the concatenated representations from the beginning layers were up-sampled to a resolution higher than themselves, and the representations from deep layers were down-sampled to a resolution lower than themselves.

The up and down-sample processes were achieved by two convolution operations and are illustrated in the following equation:

$$X_t^D \Rightarrow X_t^{\frac{h}{s} \times \frac{w}{s} \times \hat{D}} \quad (4)$$

$$X_t = \{X_c, X_l\} \quad s = \{4, 8, 16, 32\} \quad t = \{5, 11, 17, 23\}$$

Figure 18 illustrates the fusion stage of the CLFT decoder. The camera and LiDAR representations were forwarded through Residual Convolution Unit (RCU) and then summed with the results from the previous fusion operation. The output of the last fusion layer was passed to a deconvolutional and up-sampling module to compute the final predicted segmentation.

Algorithm 2 The projection of the 'classification token'. Adopted from the Article IV.

Input: Input tensor T , representing either the camera or LiDAR channels containing the 'classification token' and patch tokens.

Output: Concatenated tensor representations X_T

- 1: $T_{cls} = replicate\{T[:, 0]\}$
 - 2: $T_{concat} = T[:, i] \parallel T_{cls} \quad \forall i = 1, \dots, k$
 - 3: $X_T = GELU(W \cdot T_{concat} + b)$
-

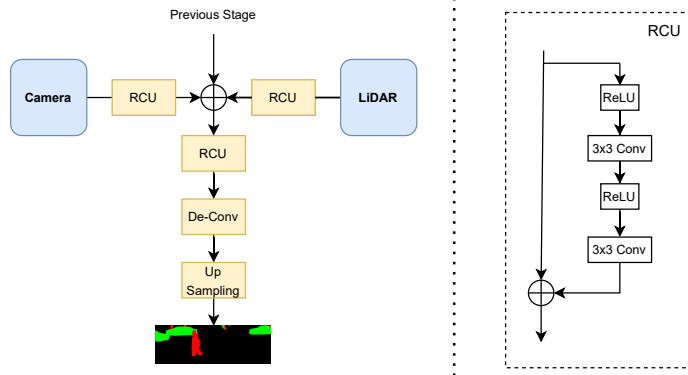


Figure 18: The progressive overview of fusion architecture. Adopted from the Article IV.

4 Experiments and Results

This section introduces the experiments carried out to evaluate this thesis’s research objectives and presents the numerical results attained from the corresponding experiments.

The first part of this section focuses on the efficiency of the end-to-end multi-modal dataset collection framework (RO1) regarding the computational consumption for post-processing algorithms and storage space requirement for the produced dataset. The second part analyzes the CLFCN (RO3) network’s capability to extract and inherit the knowledge from a public training dataset, then use the knowledge to achieve reasonable performance on a custom iseAuto dataset (RO2) with fewer annotations and more challenging scenarios. The last part provides a performance and inference time benchmark for the CLFT (RO4) network with respect to other state-of-the-art networks regarding the traffic object segmentation tasks.

4.1 Performance Evaluation for Dataset Collection Framework

The scope of the dataset collection framework is a generic practical solution for low-speed urban autonomous platforms such as autonomous shuttles and delivery robots to collect, process, and share perceptive data in their daily operations. Although the raw data and processed dataset were transferred to the remote server through the mobile network, there is a need to consider the on-board data storage due to the network bandwidth. Moreover, the framework’s post-processing, such as data decompression, sensor synchronization, and fusion, was carried out by vehicles’ built-in computers. Therefore, the evaluation of the dataset collection framework focuses on computational and storage efficiency. The inference time and storage occupation of each post-processing module are separately analyzed in performance evaluation experiments.

Table 5: Data size and time consumption of framework’s modules to process the whole data sequence. Data size in gigabyte (GB) and time in second (s). Adopted from the Article III.

	Sequence 1 City Urban	Sequence 2 Indoor Lab
Sequence Duration (s)	301	144
Raw Bag File Size (GB)	3.7	0.78
Synchronization (s)	4.28	1.24
Raw Data Decompressing (s)	0.36	0.09
Raw Data Writing (s)/(GB)	116.63/16.4	54.74/7.4
Camera-LiDAR Fusion (s)/(GB)	510.94/9.2	261.34/4.6
Camera-LiDAR-radar Fusion (s)/(GB)	61.97/5.8	39.38/3.3

Table 5 and Table 6 present the time consumption and data size of the framework’s different post-processing modules. Table 5 provides the framework’s insight to process the whole data sequence. Two example sequences are listed in the table: the first sequence was collected in the Tallinn urban area, and the second was recorded at the indoor laboratory. Both sequences were produced and processed by the multi-sensor perception hardware described in Section 2.2.1; thus, the computational time shown in this section is based on the specific and varies for different hardware setups. It is important to note that the ‘Camera-LiDAR-radar Fusion’ in Table 5 indicates two processing streams because two radar sensors are installed on the testing platform.

Table 6 shows the data size and average time consumption of the framework’s post-processing modules to process a single data frame. The output of each listed post pro-

cessing module is an RGB image in 1920x1080 resolution, and a binary pickle file contains the points' coordinates and velocity information.

Table 6: Data size and average time consumption of the framework's post-processing modules for single frame. Data size in megabyte (MB) and time in millisecond (ms). Adopted from the Article III.

	Raw Data Decompressing and Writing	LiDAR Projection	Radar-LiDAR Clustering
Size per frame			
RGB image in 1920 × 1080 + LiDAR points in binary	3 MB 1.2 MB	3 MB 0.9 MB	3MB <0.1 MB
Average time per frame			
(RGB image in 1920 × 1080 + LiDAR points in binary)	79.7 ms	647.7 ms	108.44 ms

4.2 Domain Adaptation Analysis for CLFCN

The experiments to analyze CLFCN's domain adaptation capability rely on transfer learning and semi-supervised learning techniques. The supervised baseline models of the Waymo Open dataset and the iseAuto dataset were trained first. Waymo's supervised models were continuously trained by the iseAuto dataset to conclude the transfer learning models. The best-performed transfer learning models were then used to predict the unlabeled iseAuto dataset. At last, the iseAuto dataset with manual-made ground-truths and transfer-learning-model-made predictions were mixed and used to train the transfer learning models. In comparison, the mixed iseAuto dataset was also used to train the iseAuto supervised baseline models. Figure 19 illustrates the training procedures for CLFCN's domain adaptation analysis.

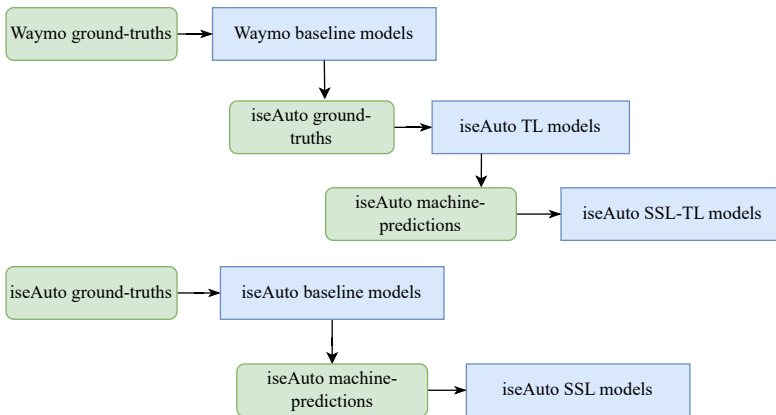


Figure 19: The training procedures of CLFCN's domain adaptation analysis experiments. 'TL' and 'SSL' stand for transfer learning and semi-supervised learning, respectively. Green blocks mean the dataset splits, and the blue blocks represent the models in different stages. Adopted from the Article II.

The hardware used to carry out the domain adaptation experiments is an NVIDIA RTX2070 Super GPU. The weighted cross-entropy loss function and Adam optimization

[127] were used in training. Moreover, data normalization, data augmentation, and early-stopping were employed to increase the dataset and save training time. The details of fine-tuning and data pre-processing are available in the work [24] (**Article II**).

Table 7: Performance comparison between iseAuto supervised and semi-supervised baseline models

		iseAuto baseline IoU(%)		SSL-iseAuto baseline IoU(%)	
		Vehicle	Human	Vehicle	Human
Day-Dry	camera	75.97	71.31	79.85	67.06
	LiDAR	71.19	56.87	73.69	58.05
	fusion	80.39	74.56	82.38	68.98
Day-Wet	camera	77.71	39.87	80.27	53.61
	LiDAR	76.00	42.10	80.58	44.09
	fusion	83.20	56.24	83.98	54.28
Night-Dry	camera	68.89	54.98	73.14	55.07
	LiDAR	74.25	47.19	75.75	49.59
	fusion	76.79	62.48	79.28	56.32
Night-Wet	camera	52.17	29.40	60.42	42.06
	LiDAR	59.49	36.76	64.89	41.32
	fusion	64.68	46.09	63.97	43.63

Table 8: Performance comparison between the transfer learning models with and without semi-supervised learning.

		waymo2iseAuto TL IoU(%)		SSL-waymo2iseAuto TL IoU(%)	
		Vehicle	Human	Vehicle	Human
Day-Dry	camera	77.10	75.87	80.32	69.25
	LiDAR	72.14	55.71	76.10	61.81
	fusion	83.27	74.24	82.85	71.09
Day-Wet	camera	80.26	48.11	82.49	57.12
	LiDAR	77.33	40.27	81.00	44.85
	fusion	84.92	57.61	85.04	54.84
Night-Dry	camera	66.07	52.38	75.97	55.46
	LiDAR	74.50	45.38	76.01	51.63
	fusion	80.43	64.03	79.82	60.21
Night-Wet	camera	51.70	41.39	60.79	48.30
	LiDAR	62.51	26.46	64.40	41.15
	fusion	67.89	45.68	66.92	48.36

Table 7 and Table 8 present the primary results of CLFCN domain adaptation experiments. A specific pixel-wise multi-class Intersection over Union (IoU) algorithm was developed to measure the models' performance on object segmentation. In the case of this work, two object classes, vehicle V and human H , were detected. The IoU of two classes is acquired by:

$$\text{IoU}_V = \frac{V_p V_g}{V_p V_g + V_p H_g + H_p V_g} \quad \text{IoU}_H = \frac{H_p H_g}{H_p H_g + H_p V_g + V_p H_g}. \quad (5)$$

where $V_p V_g$ denotes the number of pixels referred as vehicle class in both prediction and ground-truth. The same principle is applied to $H_p H_g$ for the human class. $V_p H_g$ represents the number of pixels indicated as a vehicle in prediction, but human in ground-truth. Similarly, $H_p V_g$ is the number of pixels labeled as human in prediction, but a vehicle in ground-truth.

Table 7 compares the iseAuto’s supervised and semi-supervised baseline models. Compared with supervised baseline models, the semi-supervised baseline models attain the apparent improvement in vehicle segmentation, which follows the general rule of machine learning that more data brings better performance. The semi-supervised baseline models behave weakly in some human segmentation scenes. This is because the human class is less represented in the iseAuto dataset; extra machine-labeled annotations in semi-supervised training increase the model’s uncertainty on the human class. Table 8 compares transfer learning models with and without the help of the semi-supervised learning technique. In some cases, the semi-supervised transfer learning models show a maximum of 10%

In summary, it is possible to say that the CLFCN network can adapt from one domain to another. Cross-comparing the results shown in Table 7 and Table 8, it could be concluded that the knowledge CLFCN network gains from one dataset is helpful in predicting another dataset. Work [24] (Article II) provides a more comprehensive evaluation of the CLFCN network, including other measuring metrics such as precision, recall, and auc-AP [128].

4.3 Benchmark Comparison for CLFT

The benchmark comparison for CLFT networks focuses on two critical aspects of neural networks: i) backbone architecture, and ii) input modality. The corresponding experiments for each aspect have another aspect identical to the environment. For instance, benchmark experiments to explore the effectiveness of different backbones take the same input data. Vice versa for experiments analyze the affection of input modality.

Table 9: Benchmark comparison of CLFT-hybrid variant, CLFCN and Panoptic SegFormer. Bold indicates the best values in each row per class. (in percentage unit)

	Day-Dry		Day-Wet		Night-Dry		Night-Wet	
	Vehicle	Human	Vehicle	Human	Vehicle	Human	Vehicle	Human
CLFT-Hybrid (C+L)	91.35	66.04	91.72	66.03	90.62	65.66	90.18	53.51
CLFCN (C)	88.08	55.57	88.54	52.13	81.16	42.87	74.49	43.14
CLFCN (L)	88.58	53.04	89.47	50.06	86.16	48.83	87.51	46.68
CLFCN (C+L)	91.07	62.50	92.77	64.66	89.41	60.33	89.90	56.70
Panoptic SegFormer (C)	85.89	61.02	83.58	49.70	81.45	44.67	70.50	14.68
Panoptic SegFormer (L)	66.41	40.78	63.07	29.87	70.25	38.69	54.40	39.00

In detail, the CLFCN networks were selected to explore the advantages of the transformer backbone. Because both CLFCN and CLFT networks rely on the camera-LiDAR fusion data for object segmentation and use the same LiDAR processing strategy, which is projecting the 3D point clouds on the camera planes (details are available in Section 3.2.1). The Panoptic SegFormer [94] networks were used to evaluate the differences between various input modalities. The Panoptic SegFormer networks are also based on the transformer but only take visual input. Following the procedures in Section 3.2.1, it is possible to produce the LiDAR point clouds projection images, which can be regarded as the LiDAR modality for the Panoptic SegFormer networks. Therefore, the Panoptic SegFormer networks in a singular camera or LiDAR modality were compared with CLFT networks to present the significance of sensor fusion in autonomous driving. Figure 20 demonstrates

the visualized examples of segmented images from all models used in benchmark experiments.

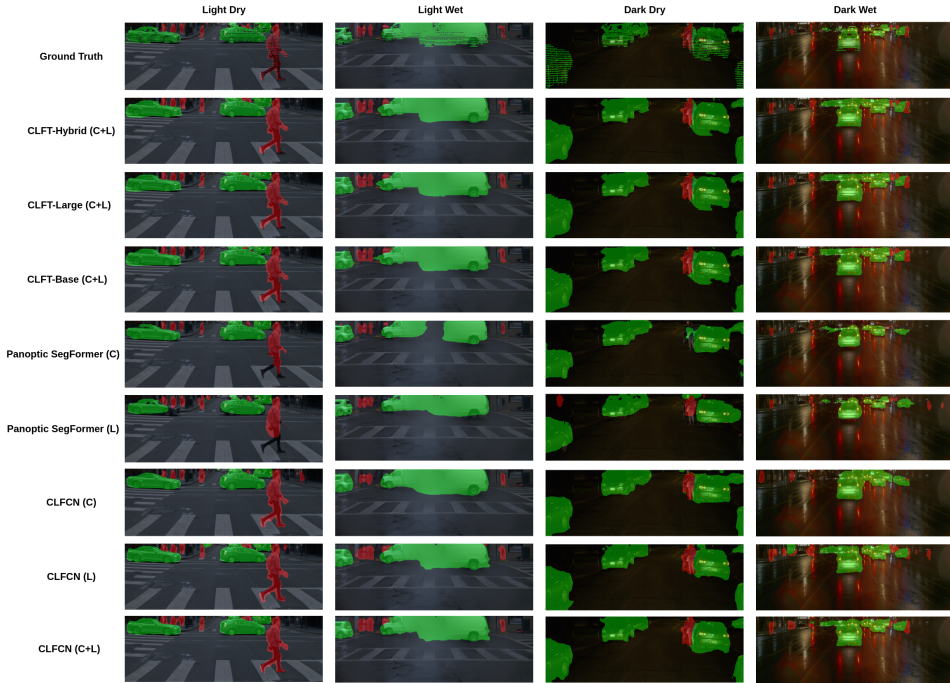


Figure 20: Qualitative comparison of segmentation results between different models. Adopted from the Article IV.

The dataset used for all benchmark experiments is the Waymo Open dataset. The sub-datasets based on weather and illumination conditions discussed in Section 2.3.1 were adopted in experiments. The CLFT networks' transformer encoders were initiated from ImageNet pre-train weights, and the transformer decoders were initiated randomly. The weighted cross-entropy loss function and Adam optimization were employed in training. The hardware for Panoptic SegFormer and CLFT-related training is an NVIDIA A100 80GB GPU due to the large memory requirements of transformer-based networks. The CLFCN training was executed on an NVIDIA RTX2070 Super GPU. Work [122] (**Article IV**) provides more details of other hyper-parameter settings.

The main results of the benchmark comparison work were reported in Table 9 and Table 10. The same IoU metric presented in Section 4.2 was adopted to evaluate networks' performance for two interest classes in different modalities and scenarios. The C, L, and C+L in two tables indicate the camera, LiDAR, and fusion modalities, respectively. In Table 9, the CLFT-hybrid variant reaches an average of 90% IoU for vehicle segmentation and outperforms the CLFCN and Panoptic SegFormer networks in most cases. There are two conclusions can be drawn from the Table 9:

- Transformers have a natural advantage regarding underrepresented samples. The Panoptic SegFormer networks achieve higher performance than CLFCN in camera modality (which is the only modality originally designed for Panoptic SegFormer) for the less-represented human class with fewer fine-tuning efforts.

- The combination of transformer and multi-modal sensor fusion has undoubted advancements and strengths because the CLFT-hybrid network leads performance in most cases.

Table 10: Ablation Study based on CLFT-Hybrid variant. Bold indicates the best values(in percentage unit)

C	L	IoU		Precision		Recall	
		Vehicle	Human	Vehicle	Human	Vehicle	Human
All weather							
✓		91.16	64.38	93.86	73.33	96.88	84.05
	✓	91.19	65.17	93.93	72.89	96.85	84.19
✓	✓	91.26	65.46	94.15	75.76	96.69	82.75
Light-Dry							
✓		91.23	64.87	93.83	72.63	97.05	85.86
	✓	91.32	64.92	93.96	72.68	97.02	85.88
✓	✓	91.35	66.04	94.14	75.31	96.86	84.29
Light-Wet							
✓		91.67	64.87	94.52	76.49	96.82	81.36
	✓	91.52	64.28	94.40	74.43	96.78	82.49
✓	✓	91.72	66.03	94.69	78.27	96.96	80.84
Dark-Dry							
✓		90.51	65.62	93.15	74.30	96.96	84.66
	✓	90.47	65.18	93.27	74.30	96.96	84.16
✓	✓	90.62	65.66	93.38	77.39	96.68	81.25
Dark-Wet							
✓		89.62	52.46	93.60	70.00	95.70	67.69
	✓	89.74	49.95	93.69	67.28	95.51	65.97
✓	✓	90.18	53.51	94.40	68.68	95.29	70.79

Table 10 presents the ablation study of the CLFT network with different modalities of camera (C), LiDAR (L), and fusion (C+L). The ablation study was based on the best-performed CLFT-hybrid variant. The CLFT-hybrid network shows a minor improvement in the all-weather category, which can be explained by heavily unbalanced data splits in different weather sub-categories. As discussed in Section 2.3.1 and values shown in Table 4, the number of light scenarios constitutes over 88% of the total number of frames in the Waymo Open dataset, affecting the overall results mainly. The weather-based splitting comparison in Table 10 offers a better view of the advancement of the fusion modalities. The CLFT-hybrid achieves a higher improvement (around 2-4%) in under-represented dark and wet scenarios.

Table 11: Inference time comparison of all CLFT variants, CLFCN and Panoptic SegFormer networks (in milliseconds unit).

NETWORK	MODALITY	TIME
CLFT-base	C+L	16.23
CLFT-Large		36.75
CLFT-Hybrid		25.69
CLFCN		15.94
Panoptic SegFormer	C	93.52
	L	93.45

Table 11 presents the study of inference time for CLFCN, Panoptic SegFormer, and all CLFT variants. All inference time experiments were carried out on the NVIDIA A100 GPU. The CPU and GPU were synchronized when calculating the CUDA event time. In general, the CLFCN networks have obvious advantages against the other transformer-based networks regarding computational efficiency. The evaluation results of other CLFT variants such as CLFT-base and CLFT-large were presented in work [122] (**Article IV**).

5 Conclusions and Future Work

The perception of AVs is a multidisciplinary field that plays an essential role in autonomous driving ecology. The requirements for AV perception evolve from the basic yes/no obstacle detection to intelligent environment analysis. Sensor fusion and artificial intelligence are regarded as two promising technologies to fulfill the new requirements. This thesis focuses on these two technologies and presents comprehensive research for advanced autonomous driving perception. The research outcomes cover all aspects of deep-learning-based AV perception, from hardware to software and from dataset production to model training. The details achievements were outlined as follows:

- A practical and real-traffic-oriented exploration of range sensor deployment for autonomous shuttles. Part of the **RO1** is exploring the hardware solutions for urban autonomous/robotic platforms to perceive the environment and collect data. Thus, the aspects such as sensor type, sensor model, and installation location were thoroughly analyzed to reduce the blind zones, which are critical for autonomous shuttles concerning their appearances and application scenarios.
- An end-to-end generic multi-sensor dataset collection framework includes signal-level camera-LiDAR-radar fusion as the backend, a universal toolbox for multi-sensor calibration and synchronization, and data transfer and sharing protocols. The average time consumption based on the testing hardware for critical post-processing, camera-LiDAR projection and radar-LiDAR clustering, are 647 and 108 milliseconds per frame, respectively. In real world tests, the time duration of these two processes for a 300 seconds city urban scenario sequence are 510 and 61 seconds. The evaluation results show the potential of framework for large-scale deployment on various urban robotics and autonomous platforms, which successfully addresses the primary focus of the **RO1**.
- The iseAuto dataset, a custom camera and LiDAR training dataset for object detection and segmentation. As declared in **RO2**, the dataset was collected by the iseAuto shuttle at TalTech campus under the multi-modal sensor collection framework (**RO1**). There are totally 8000 frames and equally distributed into four different weather subsets. All frames contain manual-made bounding box labels for multiple classes, and 30% of frames have manual-made mask labels for vehicle and human classes.
- Conducting a series of experiments to analyze the domain adaptation capability of a camera-LiDAR fusion FCN-based network (CLFCN), which address the primary objective in **RO3**. The iseAuto dataset (**RO2**) was used in the experiments. The purpose of this analysis is that as a custom training dataset produced with limited resources, the iseAuto dataset can not compete with the large-scale open datasets regarding the aspects that require heavy labor work, for instance, manual-labeled ground-truth. The experiments prove it is possible to transfer the knowledge from another dataset to the iseAuto dataset. Thus, there is no need to allocate significant labor resources to the data annotation work. In general, the domain adaptation and semi-supervised learning contribute an average increase to IoU between 2 to 5 percentages. Specially, in the average of all scenarios, the vehicle segmentation in fusion mode increase from 76% in iseAuto baseline model to 79% in semi-supervised transfer learning model.
- Developing a camera-LiDAR fusion transformer-based neural network (CLFT) for object segmentation. The network is the first transformer-based proposal to invoke

the strategy to project LiDAR point clouds as camera-plane maps for object segmentation tasks. The development of the CLFT network corresponds to the first part of the **RO4**. The second part of the **RO4** is conducting the evaluation experiments to prove the CLFT network is more efficient than the FCN-based network (**RO3**), and the significance of sensor fusion in scene interpretation. The quantitative assessments show the CLFT networks achieve an improvement of up to 10% in challenging dark-wet scenarios against to the FCN-based networks. Compared with other neural network models with transformer backbone, the all-around average improvement is 5-10%.

Overall, this thesis contributes to autonomous driving and intelligent transportation societies, and provides the vision and possibility to integrate sensor fusion and artificial intelligence into autonomous driving for precise and reliable perception.

Future Work

The future research follows the line this thesis for AV perception are suggested as:

- Object tracking with radar sensors is a challenging task for autonomous driving. The end-to-end dataset collection framework (**RO1**) has relatively weak performance regarding object identification and tracking, where the future works lie. Additionally, the framework should include more sensor types and models and develop the corresponding toolkit and sensor fusion algorithms.
- The iseAuto training dataset (**RO2**) was collected at the TalTech campus, with limited traffic volumes and identical road conditions. The extension work of the iseAuto training dataset should focus on the various traffic and road scenarios. Moreover, weather conditions such as snow and fog and extra label classes should be covered in the future.
- The characteristics of radar sensors, such as sparse point clouds and limited FoV, pose challenges to implementing the radar modality into neural networks for autonomous driving-related applications. However, the advantages of radar sensors in moving object detection and speed estimation are critical for autonomous driving. Both neural networks defined as research objectives in this thesis, CLFCN (**RO3**) and CLFT (**RO4**), have no radar input modality. Future development of the networks should include different modalities and scenarios.
- The CLFT (**RO4**) networks fill the research gap regarding the multi-modal fusion transformer that processes the LiDAR point clouds data as camera-plane-projection. However, the CLFT networks were only verified by the Waymo Open dataset with limited object classes. Future work should include testing the CLFT networks with more benchmarking autonomous driving datasets.

References

- [1] Fabian Kröger. Automated driving in its social, historical and cultural contexts. *Autonomous driving: Technical, legal and social aspects*, pages 41–68, 2016.
- [2] Jameson Wetmore. Driving the dream. the history and motivations behind 60 years of automated highway systems in america. *Automotive History Review*, 7:4–19, 2003.
- [3] Self-Drive Cars and You: A History Longer than You think. <https://velocetoday.com/self-drive-cars-and-you-a-history-longer-than-you-think/>, 2014. [Accessed 27-02-2024].
- [4] Erkki Huhtamo et al. The self-driving car: A media machine for posthumans? *Artnodes*, (26):1–14, 2020.
- [5] Morteza Taiebat, Austin L Brown, Hannah R Safford, Shen Qu, and Ming Xu. A review on energy, environmental, and sustainability implications of connected and automated vehicles. *Environmental science & technology*, 52(20):11449–11465, 2018.
- [6] Robert Martin. Av futures or futures with avs? bridging sociotechnical imaginaries and a multi-level perspective of autonomous vehicle visualisations in praxis. *Humanities and Social Sciences Communications*, 8(1):1–15, 2021.
- [7] Björn Lundgren. Safety requirements vs. crashing ethically: what matters most for policies on autonomous vehicles. *AI & SOCIETY*, 36(2):405–415, 2021.
- [8] De Jong Yeong, Gustavo Velasco-Hernandez, John Barry, and Joseph Walsh. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 21(6):2140, 2021.
- [9] Jessica Van Brummelen, Marie O’Brien, Dominique Gruyer, and Homayoun Najjaran. Autonomous vehicle perception: The technology of today and tomorrow. *Transportation Research Part C: Emerging Technologies*, 89:384–406, 2018.
- [10] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE international conference on computer vision*, pages 2722–2730, 2015.
- [11] Shimon Ullman. Against direct perception. *Behavioral and Brain Sciences*, 3(3):373–381, 1980.
- [12] Dean A Pomerleau. *Neural network perception for mobile robot guidance*, volume 239. Springer Science & Business Media, 2012.
- [13] Hrag-Harout Jebamikyous and Rasha Kashef. Autonomous vehicles perception (avp) using deep learning: Modeling, assessment, and challenges. *IEEE Access*, 10:10523–10535, 2022.
- [14] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

- [15] Leandro Masello, Barry Sheehan, Finbarr Murphy, German Castignani, Kevin McDonnell, and Cian Ryan. From traditional to autonomous vehicles: A systematic review of data availability. *Transportation research record*, 2676(4):161–193, 2022.
- [16] Feng Guo, Bruce G Simons-Morton, Sheila E Klauer, Marie Claude Ouimet, Thomas A Dingus, and Suzanne E Lee. Variability in crash and near-crash risk among novice teenage drivers: a naturalistic study. *The Journal of pediatrics*, 163(6):1670–1676, 2013.
- [17] Jack N Barkenbus. Eco-driving: An overlooked climate change initiative. *Energy policy*, 38(2):762–769, 2010.
- [18] Mercedes Ayuso, Montserrat Guillen, and Jens Perch Nielsen. Improving automobile insurance ratemaking using telematics: Incorporating mileage and driver behaviour data. *Transportation*, 46:735–752, 2019.
- [19] On-Road Automated Driving (ORAD) Committee. *Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles*. SAE international, 2021.
- [20] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.
- [21] Velodyne’s hdl-64e Lidar Sensor Looks Back on a Legendary Career. <https://velodynelidar.com/blog/hdl-64e-lidar-sensor-retires/>, 2021. [Accessed 22-04-2024].
- [22] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin 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.
- [23] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2702–2719, 2019.
- [24] Junyi Gu, Mauro Bellone, Raivo Sell, and Artjom Lind. Object segmentation for autonomous driving using iseauto data. *Electronics*, 11(7):1119, 2022.
- [25] Dean A Pomerleau. Alvin: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.
- [26] Estonian First Level 4 Self-driving Vehicle – ISEAUTO. <https://autolab.taltech.ee/portfolio/iseauto/>. [Accessed 22-04-2024].
- [27] Junyi Gu, Artjom Lind, Tek Raj Chhetri, Mauro Bellone, and Raivo Sell. End-to-end multimodal sensor dataset collection framework for autonomous vehicles. *Sensors*, 23(15):6783, 2023.

- [28] Gustavo Velasco-Hernandez, John Barry, Joseph Walsh, et al. Autonomous driving architectures, perception and data fusion: A review. In *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 315–321. IEEE, 2020.
- [29] Giulia Rizzoli, Francesco Barbato, and Pietro Zanuttigh. Multimodal semantic segmentation in autonomous driving: A review of current approaches and future perspectives. *Technologies*, 10(4):90, 2022.
- [30] Yaodong Cui, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):722–739, 2021.
- [31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.
- [32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.
- [33] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [34] Athanasios Theofilatos and George Yannis. A review of the effect of traffic and weather characteristics on road safety. *Accident Analysis Prevention*, 72:244–256, 2014.
- [35] Huazan Zhong, Hao Wang, Zhengrong Wu, Chen Zhang, Yongwei Zheng, and Tao Tang. A survey of lidar and camera fusion enhancement. *Procedia Computer Science*, 183:579–588, 2021.
- [36] Ratheesh Ravindran, Michael J Santora, and Mohsin M Jamali. Multi-object detection and tracking, based on dnn, for autonomous vehicles: A review. *IEEE Sensors Journal*, 21(5):5668–5677, 2020.
- [37] Xuan Wang, Kaiqiang Li, and Abdellah Chehri. Multi-sensor fusion technology for 3d object detection in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [38] Shanliang Yao, Runwei Guan, Xiaoyu Huang, Zhuoxiao Li, Xiangyu Sha, Yong Yue, Eng Gee Lim, Hyungjoon Seo, Ka Lok Man, Xiaohui Zhu, et al. Radar-camera fusion for object detection and semantic segmentation in autonomous driving: A comprehensive review. *IEEE Transactions on Intelligent Vehicles*, 2023.
- [39] Mauro Bellone, Azat Ismailogullari, Jaanus Müür, Oscar Nissin, Raivo Sell, and Ralf-Martin Soe. Autonomous driving in the real-world: The weather challenge in the sohjoa baltic project. In *Towards Connected and Autonomous Vehicle Highways: Technical, Security and Social Challenges*, pages 229–255. Springer, 2021.
- [40] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern recognition letters*, 30(2):88–97, 2009.

- [41] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva 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.
- [42] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, Trevor Darrell, et al. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018.
- [43] Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3095–3101. IEEE, 2021.
- [44] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- [45] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV 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. IEEE, 2019.
- [46] Yukyung Choi, Namil Kim, Soonmin Hwang, Kibaek Park, Jae Shin Yoon, Kyoungwan An, and In So Kweon. Kaist multi-spectral day/night data set for autonomous and assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):934–948, 2018.
- [47] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, pages 409–418. PMLR, 2021.
- [48] Lukáš Neumann, Michelle Karg, Shanshan Zhang, Christian Scharfenberger, Eric Piegert, Sarah Mistr, Olga Prokofyeva, Robert Thiel, Andrea Vedaldi, Andrew Zisserman, et al. Nightowls: A pedestrians at night dataset. In *Computer Vision—ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part I 14*, pages 691–705. Springer, 2019.
- [49] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.
- [50] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*, 2020.
- [51] Yi Xiao, Felipe Codevilla, Akhil Gurram, Onay Urfalioglu, and Antonio M López. Multimodal end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):537–547, 2020.

- [52] Junqing Wei, Jarrod M Snider, Junsung Kim, John M Dolan, Raj Rajkumar, and Bakhtiar Litkouhi. Towards a viable autonomous driving research platform. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 763–770. IEEE, 2013.
- [53] Alberto Broggi, Michele Buzzoni, Stefano Debattisti, Paolo Grisleri, Maria Chiara Laghi, Paolo Medici, and Pietro Versari. Extensive tests of autonomous driving technologies. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1403–1415, 2013.
- [54] Massimo Bertozzi, Luca Bombini, Alberto Broggi, Michele Buzzoni, Elena Cardarelli, Stefano Cattani, Pietro Cerri, Alessandro Coati, Stefano Debattisti, Andrea Falzoni, et al. Viac: An out of ordinary experiment. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 175–180. IEEE, 2011.
- [55] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006.
- [56] Shinpei Kato, Eijiro Takeuchi, Yoshio Ishiguro, Yoshiki Ninomiya, Kazuya Takeda, and Tsuyoshi Hamada. An open approach to autonomous vehicles. *IEEE Micro*, 35(6):60–68, 2015.
- [57] Ji Zhang and Sanjiv Singh. Laser-visual-inertial odometry and mapping with high robustness and low drift. *Journal of field robotics*, 35(8):1242–1264, 2018.
- [58] Joris Domhof, Julian FP Kooij, and Darius M Gavrila. An extrinsic calibration tool for radar, camera and lidar. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8107–8113. IEEE, 2019.
- [59] Pei An, Tao Ma, Kun Yu, Bin Fang, Jun Zhang, Wenxing Fu, and Jie Ma. Geometric calibration for lidar-camera system fusing 3d-2d and 3d-3d point correspondences. *Optics express*, 28(2):2122–2141, 2020.
- [60] Jinyong Jeong, Younghun Cho, and Ayoung Kim. The road is enough! extrinsic calibration of non-overlapping stereo camera and lidar using road information. *IEEE Robotics and Automation Letters*, 4(3):2831–2838, 2019.
- [61] Christoph Schöller, Maximilian Schnettler, Annkathrin Krämmer, Gereon Hinz, Maida Bakovic, Müge Güzet, and Alois Knoll. Targetless rotational auto-calibration of radar and camera for intelligent transportation systems. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3934–3941. IEEE, 2019.
- [62] Keli Huang, Botian Shi, Xiang Li, Xin Li, Siyuan Huang, and Yikang Li. Multi-modal sensor fusion for auto driving perception: A survey. *arXiv preprint arXiv:2202.02703*, 2022.
- [63] Zhangjing Wang, Yu Wu, and Qingqing Niu. Multi-sensor fusion in automated driving: A survey. *Ieee Access*, 8:2847–2868, 2019.
- [64] Chao Xiang, Chen Feng, Xiaopo Xie, Botian Shi, Hao Lu, Yisheng Lv, Mingchuan Yang, and Zhendong Niu. Multi-sensor fusion and cooperative perception for autonomous driving: A review. *IEEE Intelligent Transportation Systems Magazine*, 2023.

- [65] Matthias Pollach, Felix Schiegg, and Alois Knoll. Low latency and low-level sensor fusion for automotive use-cases. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6780–6786. IEEE, 2020.
- [66] Babak Shahian Jahromi, Theja Tulabandhula, and Sabri Cetin. Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles. *Sensors*, 19(20):4357, 2019.
- [67] Yulu Luke Chen, Mohammad R Jahanshahi, Preetham Manjunatha, WeiPhang Gan, Mohamed Abdelbarr, Sami F Masri, Burcin Becerik-Gerber, and John P Caffrey. Inexpensive multimodal sensor fusion system for autonomous data acquisition of road surface conditions. *IEEE Sensors Journal*, 16(21):7731–7743, 2016.
- [68] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3288–3295. IEEE, 2019.
- [69] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *2019 16th international conference on machine vision applications (MVA)*, pages 1–6. IEEE, 2019.
- [70] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 103–119, 2018.
- [71] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10615–10622, 2020.
- [72] Jae-Seol Lee and Tae-Hyoung Park. Fast road detection by cnn-based camera–lidar fusion and spherical coordinate transformation. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):5802–5810, 2020.
- [73] Florian Wulff, Bernd Schäufole, Oliver Sawade, Daniel Becker, Birgit Henke, and Ilja Radosch. Early fusion of camera and lidar for robust road detection based on u-net fcn. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1426–1431. IEEE, 2018.
- [74] Luca Caltagirone, Mauro Bellone, Lennart Svensson, and Mattias Wahde. Lidar-camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125–131, 2019.
- [75] Jian Dou, Jianru Xue, and Jianwu Fang. Seg-voxelnet for 3d vehicle detection from rgb and lidar data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4362–4368. IEEE, 2019.
- [76] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020.
- [77] Luca Caltagirone, Mauro Bellone, Lennart Svensson, Mattias Wahde, and Raivo Sell. Lidar-camera semi-supervised learning for semantic segmentation. *Sensors*, 21(14):4813, 2021.

- [78] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems*, 35:10421–10434, 2022.
- [79] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *International Journal of Computer Vision*, 128(5):1239–1285, 2020.
- [80] Yin Zhou and Oncel 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.
- [81] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12605–12614, 2020.
- [82] Shuo Gu, Tao Lu, Yigong Zhang, Jose M Alvarez, Jian Yang, and Hui Kong. 3-d lidar+ monocular camera: An inverse-depth-induced fusion framework for urban road detection. *IEEE Transactions on Intelligent Vehicles*, 3(3):351–360, 2018.
- [83] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 244–253, 2018.
- [84] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [85] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
- [86] Khaled El Madawi, Hazem Rashed, Ahmad El Sallab, Omar Nasr, Hanan Kamel, and Senthil Yogamani. Rgb and lidar fusion based 3d semantic segmentation for autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 7–12. IEEE, 2019.
- [87] Xin Zhao, Zhe Liu, Ruolan Hu, and Kaiqi Huang. 3d object detection using scale invariant and feature reweighting networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9267–9274, 2019.
- [88] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [89] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

- [90] Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird’s eye view semantic segmentation from arbitrary camera rigs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5935–5943, 2023.
- [91] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3694–3702, 2021.
- [92] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, et al. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *European Conference on Computer Vision*, pages 550–567. Springer, 2022.
- [93] Yifeng Bai, Zhirong Chen, Zhangjie Fu, Lang Peng, Pengpeng Liang, and Erkang Cheng. Curveformer: 3d lane detection by curve propagation with curve queries and attention. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7062–7068. IEEE, 2023.
- [94] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, Ping Luo, and Tong Lu. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1280–1289, 2022.
- [95] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022.
- [96] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 172–181, 2023.
- [97] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision*, pages 531–548. Springer, 2022.
- [98] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022.
- [99] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9223–9232, 2023.
- [100] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9087–9098, 2023.
- [101] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

- [102] Antonio Bucchiarone, Sandro Battisti, Annapaola Marconi, Roberto Maldacea, and Diego Cardona Ponce. Autonomous shuttle-as-a-service (asaas): Challenges, opportunities, and social implications. *IEEE Transactions on Intelligent Transportation Systems*, 22(6):3790–3799, 2020.
- [103] Apollo Minibus II. <https://en.apollo.auto/minibus>. [Accessed 09-05-2024].
- [104] Self-Driving Shuttle for Passenger Transportation. <https://www.navya.tech/en/solutions/moving-people/self-driving-shuttle-for-passenger-transportation/autonomous>. [Accessed 09-05-2024].
- [105] EZ10 Passenger Shuttle. <https://easymile.com/vehicle-solutions/ez10-passenger-shuttle>. [Accessed 09-05-2024].
- [106] auvetech MiCa. <https://auve.tech/products/>. [Accessed 09-05-2024].
- [107] Zhen Liu, Qun Wu, Suining Wu, and Xiao Pan. Flexible and accurate camera calibration using grid spherical images. *Optics express*, 25(13):15269–15285, 2017.
- [108] Martin Vel’as, Michal Španěl, Zdeněk Materna, and Adam Herout. Calibration of rgb camera with velodyne lidar. 2014.
- [109] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.
- [110] Stefan Milch and Marc Behrens. Pedestrian detection with radar and computer vision. *PROCEEDINGS OF PAL 2001-PROGRESS IN AUTOMOBILE LIGHTING, HELD LABORATORY OF LIGHTING TECHNOLOGY, SEPTEMBER 2001. VOL 9*, 2001.
- [111] Wei Huang, Zhen Zhang, Wentao Li, Jiandong Tian, et al. Moving object tracking based on millimeter-wave radar and vision sensor. *Journal of Applied Science and Engineering*, 21(4):609–614, 2018.
- [112] Feng Liu, Jan Sparbert, and Christoph Stiller. Immpda vehicle tracking system using asynchronous sensor fusion of radar and vision. In *2008 IEEE Intelligent Vehicles Symposium*, pages 168–173. IEEE, 2008.
- [113] Lu Yin, Bin Luo, Wei Wang, Huan Yu, Chenjie Wang, and Chengyuan Li. Comask: Corresponding mask-based end-to-end extrinsic calibration of the camera and lidar. *Remote Sensing*, 12(12):1925, 2020.
- [114] Message_filters—ros wiki. https://wiki.ros.org/message_filters. [Last accessed on 19-05-2024].
- [115] Koyel Banerjee, Dominik Notz, Johannes Windelen, Sumanth Gavarraju, and Mingkang He. Online camera lidar fusion and object detection on hybrid data for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1632–1638. IEEE, 2018.
- [116] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1887–1893. IEEE, 2018.

- [117] Andreas Pfeuffer and Klaus Dietmayer. Optimal sensor data fusion architecture for object detection in adverse weather conditions. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2018.
- [118] Alireza Asvadi, Luis Garrote, Cristiano Premebida, Paulo Peixoto, and Urbano J Nunes. Depthcn: Vehicle detection using 3d-lidar and convnet. In *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, pages 1–6. IEEE, 2017.
- [119] Cristiano Premebida, Luis Garrote, Alireza Asvadi, A Pedro Ribeiro, and Urbano Nunes. High-resolution lidar-based depth mapping using bilateral filter. In *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, pages 2469–2474. IEEE, 2016.
- [120] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian 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.
- [121] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.
- [122] Junyi Gu, Mauro Bellone, Tomáš Pivoňka, and Raivo Sell. Clft: Camera-lidar fusion transformer for semantic segmentation in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, pages 1–12, 2024.
- [123] Juan Zhong, Zheng Liu, and Xi Chen. Transformer-based models and hardware acceleration analysis in autonomous driving: A survey. *arXiv preprint arXiv:2304.10891*, 2023.
- [124] Hans Thisanke, Chamli Deshan, Kavindu Chamith, Sachith Seneviratne, Rajith Vidanaarachchi, and Damayanthi Herath. Semantic segmentation using vision transformers: A survey. *Engineering Applications of Artificial Intelligence*, 126:106669, 2023.
- [125] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [126] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [127] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [128] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. <http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html>.

Acknowledgements

This research was supported by the European Union's Horizon 2020 Research and Innovation Programme, under grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No 2014-2020.4.01.20-0289.

When I registered for the Ph.D. position four years ago, I never expected to make this accomplishment and conclude the Ph.D. in four years. I did not have any clue of what I should do for my doctoral research back then until I met Prof. Dr. Mauro Bellone for the first time in March 2021. Prof. Dr. Mauro Bellone played the role of the mentor in my doctoral period, provided me the guidance more than in academy and research, and also shaped my values towards life and career.

The success of my Ph.D. cannot be achieved without the unconditional support and understanding of my supervisor, Prof. Dr. Raivo Sell. He showed exceptional tolerance, patience, and kindness in the four years of my doctoral journey. I cannot imagine any environment and opportunity better than what he created for me to focus on my Ph.D. His supervision wisdom and style will be the model of my future academic career.

I am lucky to have the friend Artjom Lind in this country. Our supervision and personal relationship started during my Master's period in 2019. The computer science knowledge he passed to me is the key to opening the door to the wonderful scientific world. He dedicated tremendous help and advice to my doctoral research works. Moreover, the joys and happiness we had together will be my valuable memories forever.

At last, I would like to express my heartfelt gratitude to all my colleagues, friends, and co-authors who participated and contributed to my doctoral journey.

GU JUN-YI
SEPT. 2024
TALLINN, ESTONIA

Abstract

Advancement in Perception Capabilities for Autonomous Vehicles: From Dataset Collection to Scene Interpretation

Autonomous Vehicles rely on various sensors to perceive the environment. The precise and reliable perception guarantees the safety and performance of autonomous vehicles. This thesis focuses on the advanced perception capabilities of autonomous vehicles. It is based on four research articles dedicated to sensor hardware, dataset collection, sensor fusion, and AI-based scene interpretation.

The research begins with analyzing multiple range sensor deployment for autonomous shuttles. The analysis is based on the real-traffic-deployed iseAuto shuttle operating on the TalTech campus. Considering the appearance of shuttle buses and the LiDAR sensor characteristics such as full horizontal but limited vertical views, the sensor models and installation location choices are critical for autonomous shuttles to ensure the least sensor interference and cover the most blind zones.

The thesis then presents an end-to-end generic dataset collection framework that includes hardware deployment, multi-sensor calibration and synchronization solutions, dataset transferring and sharing protocols, and signal-level sensor fusion algorithms. The framework generalizes the implementation of the multi-modal perceptive system on various robotics and autonomous platforms. The camera, LiDAR, radar, and GNSS sensors were included in the framework. The merits of all sensors are fused in a manner useful for object detection and tracking.

The dataset collection framework was deployed on different autonomous platforms. The initial validation was carried out on a car roof rack with all integrated sensors. The validation tests cover various transportation scenes such as highway, urban, and neighborhood. The practical implementation of the framework is on the iseAuto shuttle. Relying on the tools and algorithms proposed in the framework, the iseAuto dataset contains camera and LiDAR data produced for object detection and segmentation tasks. The dataset features the fierce weather and illumination conditions in Estonia.

The iseAuto dataset was used by a fully convolutional neural network (FCN) for deep learning experiments. The experiment results prove two things: i) with the help of camera-LiDAR fusion, it is possible to achieve robust multi-class segmentation on a dataset with only a few annotations; ii) the proposed FCN-based network performs reasonably in poor weather and illumination scenarios.

The thesis concludes by proposing a novel vision-transformer-based network to carry out camera-LiDAR fusion for semantic segmentation. The network invokes the progressive-assemble strategy on a double-direction network to process the camera and LiDAR data in parallel. Moreover, the network is the first transformer-based proposal that uses the strategy to project LiDAR point clouds as camera-plane maps for semantic segmentation. The evaluation experiments report robust performance in all scenarios and prove the significance of combining attention-mechanism and multi-sensor fusion.

In summary, this thesis constitutes a comprehensive research journey through all aspects of deep-learning-based AV perception, from sensor deployment to multi-modal perceptive system, then to real-world dataset collection, and last to deep model training for scene interpretation. This research facilitates advanced perception capabilities for a safe and reliable autonomous transportation system.

Kokkuvõte

Autonoomsete sõidukite tajuvõimekuse täiustamine: andmekogumisest stseeni tõlgendamiseni

Autonoomsed sõidukid tuginevad oma juhtimisotsuste tegemisel mitmesugustele anduritele, et tajuda ümbritsevat keskkonda. Täpne ja usaldusväärne taju on autonoomsete sõidukite kriitiline funktsionaalsus. Käesolev uurimistöö keskendub autonoomsete sõidukite täiustatud tajumisvõimekusele ja põhineb neljal teadusartiklil, mis käsitlevad andurite riistvara, andmekogumite loomist, andurite kombineerimist ja tehisintellektil põhinevat olukorra tõlgendamist.

Uurimus algab mitme kaugusanduri paigutuse analüüsiga autonoomsete minibusside jaoks. Analüüs põhineb reaalses liikluses kasutataval iseAuto minibussil, mis opereerib TalTechi ülikoolilinnakus. Arvestades minibusside füüsilist kuju ja LiDARite omadusi, mis kattavad küll täielikult horisontaalvaate, kuid on piiratud nägemisulatusega vertikaalses sihis, on andurite mudelid ja paigalduskohtade valik autonoomsete minibusside jaoks kriitilise tähtsusega, et vähendada andurite häireid ja katta võimalikult palju pimenurki.

Uurimistöö kajastab üldist andmekogumise raamistikku, mis hõlmab riistvara paigutust, mitme anduri kalibreerimist ja sünkroonimist, andmete edastamise ja jagamise protokolle ning signaalitasemel andurite kombineerimise algoritme. Raamistik üldistab mitmeliigilise tajusüsteemi rakendamist erinevatel robot- ja autonoomsetel platvormidel. Raamistikus kasutati kaamera, LiDARi, radari ja GNSS-i andureid. Kõigi andurite eeliseid kombineeritakse viisil, mis on kasulik objektide tuvastamiseks ja jälgimiseks.

Andmekogumisraamistik juurutati erinevatel autonoomsete sõidukite platvormidel. Esialgne valideerimine toimus testsõidukiga, kus olid integreeritud kõik andurid. Valideerimistestid hõlmasid mitmesuguseid liiklussituatsioone, nagu kiirtee, linn ja lähilinn. Raamistiku praktiline rakendamine toimus iseAuto minibussil. Tuginedes raamistikus pakutud tööriistadele ja algoritmidele, sisaldab iseAuto andmekogum kaamera- ja LiDARi andmeid, mis on mõeldud objektide tuvastamise ja segmenteerimise ülesannete jaoks. Andmekogum kajastab Eesti spetsiifiliste ilmastiku- ja valgustingimuste mõju.

TalTech iseAuto andmekogu peal rakendati täielikult konvolutsiooniline närvivõrgu (FCN) süvaõpet. Katsete tulemused tõestavad kahte asja: i) kaamera ja LiDARi kombineerimise abil on võimalik saavutada töökindel mitmeklassiline segmenteerimine andmekogul, millel on vaid mõned annotatsioonid; ii) pakutud FCN-põhine närvivõrk toimib mõistlikult halva ilmastiku ja valgustuse stsenaariumides.

Uurimistöö pakub välja uue nägemistransformaatori-põhise võrgu kasutamiseks kaamera ja LiDARi andmete kombineerimise semantilise segmenteerimise jaoks. Võrk kasutab progresseeruva komplekteerimise strateegiat kahesuunalises võrgus, et töödelda kaamera ja LiDARi andmeid paralleelselt. Lisaks on see esimene transformaatoripõhine lahendus, mis kasutab strateegiat LiDARi punktipilvede projektsiooniks kaameraplaane semantilise segmenteerimise jaoks. Tulemused näitavad head jõudlust kõigis stsenaariumides ja tõestavad tähelepanumehhanismi ning mitme anduri kombineerimise olulisust.

Kokkuvõttes kujutab uurimistöö endast põhjalikku teadustööd, mis hõlmab kõiki süvaõppepõhise autonoomse sõiduki taju aspekte, alates andurite paigutamisest kuni mitmeliigilise taju süsteemini, reaalse maailma andmete kogumiseni ja lõpuks süvavõrgu treenimiseni stseeni tõlgendamiseks. See uurimus hõlbustab luua täiustatud taju võimekusega ohutuid ja usaldusväärseid autonoomseid transpordilahendusi.

Appendix 1

Article I

Junyi Gu and Tek Raj Chhetri. Range sensor overview and blind-zone reduction of autonomous vehicle shuttles. *IOP Conference Series: Materials Science and Engineering*, 1140(1):012006, may 2021

Range Sensor Overview and Blind-Zone Reduction of Autonomous Vehicle Shuttles

Junyi Gu¹, Tek Raj Chhetri²

¹Department of Mechanical and Industrial Engineering, Tallinn University of Technology

²Department of Computer Science, University of Innsbruck

E-mail: junygu@taltech.ee

Abstract. In recent years, with the advancement in sensor technologies, computing technologies and artificial intelligence, the long-sought autonomous vehicles (AVs) have become a reality. Many AVs today are already driving on the roads. Still, we have not reached full autonomy. Sensors which allow AVs to perceive the surroundings are keys to the success of AVs to reach full autonomy. However, this requires an understanding of sensor configurations, performance and sensor placements. In this paper, we present our experience on sensors obtained from AV shuttle iseAuto. An AV shuttle iseAuto designed and developed in Tallinn University of Technology is used as an experimental platform for sensor configuration and set-up.

1. Introduction

Recently, there has been growing interest in autonomous vehicles (AVs), which are regarded as a potential trend of transportation in the future. A reliable AV perceives the environment consistently by different sensors, then transfers the sensory data to a computer for post-processing. Sensors in AVs produce information with different characteristics, Global Navigation Satellite System (GNSS) provides the approximated location of vehicles with a general reference; Inertial Measurement Unit (IMU) measures angular rates, linear velocities and orientation of the vehicles base body; range sensors include cameras, LiDARs, radars and sonars detect the objects that are around vehicles in different scales and properties. Sensor fusion algorithms combine sensory data to create more coherent and certain results than using the data individually. Path planning module uses real-time perception of the surrounding environment to update paths of the vehicle in short and long ranges. Fully AVs are supposed to be able to control the self-motions, as well as auxiliary functions in practical situations, for example, the closing and opening of the door. The motion control module, at last, controls the movement of the vehicle to follow the paths and execute the motion commands that are computed by path and motion planners. Security measurements like emergency braking and obstacle avoidance are invoked to the control system directly to improve the safety and reduce the accidents. Fig.1 summarizes the general workflow of AV modules.

Range sensors provide 3D geometry information of vehicles surrounding environment and reflect the properties (speed and acceleration) of objects that are expensive to compute from vision-based perception systems. Examples of range sensors for AVs include radar, LiDAR, sonar and infrared sensors. In particular, radar sensors make a crucial contribution to Advanced Driver Assistance System (ADAS) in the aspects of emergency braking/brake assist, collision



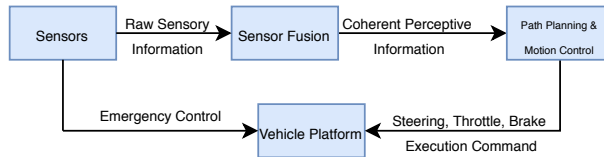


Figure 1. Modules workflow of autonomous vehicles

warning/avoidance, park assist, distance control and so on. LiDAR sensors are a relatively new technology in the AVs field and have attracted much attention in recent years. LiDARs are widely used to measure the distance and describe the environment in three dimensions. However, because of the natural characteristics of laser-based systems, LiDARs are limited by the low level of target reflectivity, resolution and refresh rate. Contrarily, the energy source of sonar sensors is acoustic/ultrasonic waves in a specific frequency, which is less affected by the reflectivity level of targets. Compared with LiDARs, sonar sensors have the advantage of low-cost and are widely used in underwater applications. Typical infrared (IR) sensors are a relatively well-developed technology, which has the advantages of cost, size and reliability. Active IR sensors share the similar principle of sonar but rely on infrared waves (wavelength usually bigger than 780nm that above the visible red light). Passive IR sensors only have receivers to detect infrared radiation and are irreplaceable in the scenarios of human/human-motion detection. This paper analyzes range sensor configurations and blind zones reductions for particular AV shuttle iseAuto.

2. Primary Perceptive Range Sensor Set-ups

In recent years, autonomous vehicles commonly use laser-based range sensors as the primary approach to perceive the environment and measure distances. One of the most popular laser-based sensors is LiDAR, a light-based detection and ranging remote sensing tool that has contributed significantly to AV technology due to the high accuracy and precision.

Currently, decreasing cost and power consumption of LiDARs promote their usage in applications that are sensitive to the vehicle's size and weight, such as Unmanned Aerial Vehicles for mapping and navigation purposes. Other related research and experimental platforms are:

- On the roof of Stanley, the vehicle that won the 2005 DARPA Grand Challenge, there are five lasers measuring cross-sections of the approaching front terrain in different distances out to 25 meters [1];
- VIAC vehicles were equipped with four laser scanners (two lateral laser scanners, one off-road laser scanner and one central laser scanner) which have different characteristics [2];
- More recently, Gao *et al.* [3] set four laser sensors (two single-line lasers, one four-line laser and one 64-line laser) in their Mengshi autonomous vehicle;
- Other experimental vehicles that have laser-sensors installed [4] [5] [6];

AV shuttles are the most common low-speed vehicles using LiDARs, and not only for ranging but also for localization and object classification. AV shuttles that were deployed on the real traffic pilot cases around the world are in limited numbers. Most known brands are Navya and Easymile, followed by iseAuto and GACHA. All these vehicles are relying mostly on LiDARs as the main localization and ranging sensor. Fig.2 presents the main sensor locations on these vehicles.

In this paper we are focusing on the TalTech iseAuto that was designed and developed in the Autonomous Vehicles lab in TalTech, Estonia [7] [8]. The initial design and sensor configuration



Figure 2. Real traffic AV shuttles: Navya Evo, Easymile EZ10, TalTech iseAuto and GACHA

development of iseAuto was supported by mechatronic modeling methodology [9] [10], which emphasizes the importance of early design stage. The output of the initial conceptual design stage proposed to use two Velodyne VLP-16 Puck LiDARs on the front top corners, as shown in Fig.3(a). To detect as many blind zones that are in front of the shuttle as possible, the sensor plane inclined toward the front and side (in practical, 8.3° toward the front and 6.9° toward the side), as shown in Fig.3(b) and 3(c).

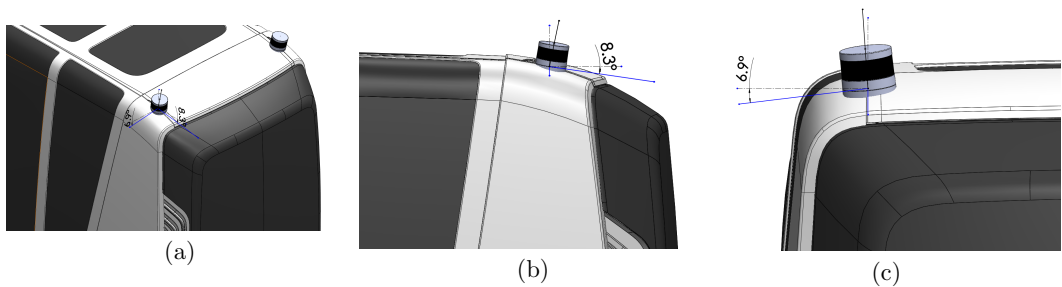


Figure 3. (a) Initial design of LiDARs location, (b) Front-tilted angle, (c) Side-tilted angle

However, in practical situations, our initial Velodyne VLP-16 sensor location (Fig.3(a)) has no vision of the shuttle's backside because laser beams shooting toward the back were heavily blocked by the shuttle itself. The points cloud of initial location configuration were showed in Fig.4(c). Additionally, the vision of left and right sides is too limited to cover the blind zones of the automatic door, which directly affects the safety of the shuttle. Therefore, our latest configuration of two VLP-16 sensors is locating them in the middle of the front and back sides with some inclines, as shown in Fig.4(a). Moreover, an adjustable mount base allows us to change the front-tilted-angle of the VLP-16 sensor, and a bigger angle helps to detect more blind zones in the front/back of the shuttle but reduce the maximum detection range correspondingly. The points cloud based on the latest configuration were presented in Fig.4(b). Compared with the previous configuration (Fig.4(c)), the coverage of the left and right sides is reduced, but the full view of the backside is available. On the other hand, current configuration helps to reduce the occasional interference patterns and shadowed azimuth ranges that may appear in data when using multiple Velodyne sensors close to one another (especially on top of the vehicle).

3. Blind Zones Reduction

Blind zones detection is an essential task for AVs because it has straight affections to safety. LiDAR-based sensors generally are installed on top of the vehicles to have wider horizontal Field of View (FoV) and further detection range. However, a top-placement configuration of LiDARs results in bigger blind zones around the vehicles, which raise problems in many post-practical

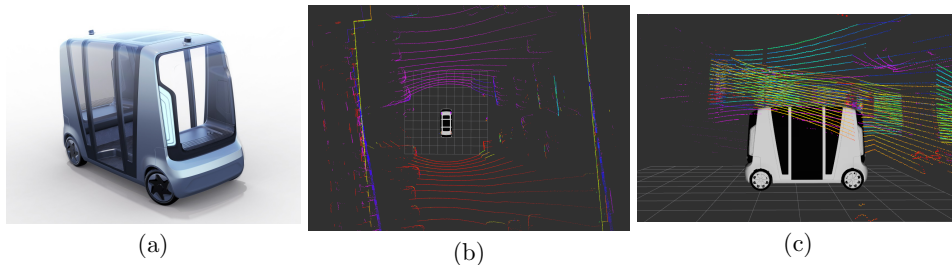


Figure 4. (a) Latest design of LiDARs location, (b) Points cloud of latest design, (c) Points cloud of initial design

processes such as motion planning in multi-interaction environments and lane change alert in ADAS.

The common solution to reduce the blind zones is installing specific sensors in corresponding positions. Variety types of sensors can be used to detect the objects in blind zones.

- Jamaluddin *et al.* [11] installed an ultrasonic sensor above the rear tire to measure the distance of approaching vehicles. The selection of the ultrasonic sensor maximally prioritizes the cost of the total sensor setup but compromises the performance and accuracy in some real-life scenarios [12].
- Using LiDARs to cover the blind zones is a popular topic in recent because they create detailed 3D points cloud. Researchers can carry out complex post-processes that are based on points cloud data to pursue the best performance. The work in [13] formulated the blind zone problems by occupancy grid and proposed a generic algorithm to optimize the configuration of LiDAR placements. Meadows *et al.* [14] introduced a system that has three LiDARs and used neural work to evaluate the effectiveness of various LiDAR poses.
- Other sensor choices include cameras and radars. Image-based information is usually processed alongside other sensory data. Rangesh *et al.* [15] described a multi-object tracking approach which is capable of working with varying camera FoVs and LiDARs. Dey *et al.* [16] put the camera and radar together and proposed a framework that can optimize the location and orientation for a heterogeneous set of sensors on a given target vehicle.

Installing sensors in corresponding areas provides direct sensory information of the objects in blind zones. However, in the scenarios that the objects' detailed detections are not vital, mathematical processes can be used to calculate the states of the objects when they are in blind zones. Zhou *et al.* [17] proposed to use Kalman Filter to estimate the movement of the approaching vehicles in blind zones for traffic intersection motion planning. Correspondingly, substitute sensors with mathematical algorithms help to reduce power consumption and hardware maintenance work.

In our case, because of the structure of the iseAuto shuttle, the front-top and back-top Velodyne LiDARs cannot detect the blind zones on two sides. Accurate and detailed detection of the objects in the right blind zone, especially the area that is close to the shuttle, is vital for us because the control of the automatic door should be strictly based on it. Our solution is installing two RS-Bpearl LiDARs on the left and right sides of the shuttle.

RS-Bpearl is a short-range LiDAR specifically designed for the detection of the blind zones. Compared with VLP-16 Puck, RS-Bpearl has a shorter range of detection but a much wider 90° vertical FoV and 32 channels. For iseAuto shuttle, the unique FoV design of RS-Bpearl helps to cover more areas on two sides, and the dense points cloud data provides more details of the objects in blind zones. Fig.5 presents the points cloud data that was produced by an

RS-Bpearl LiDAR that was installed on the right side of the shuttle. The scenario in Fig.5(a) is the outdoor environment that has buildings and parking cars. Fig.5(b) shows the ability of the right RS-Bpearl LiDAR to detect the object details (human and ladder) that are close to the automatic door.

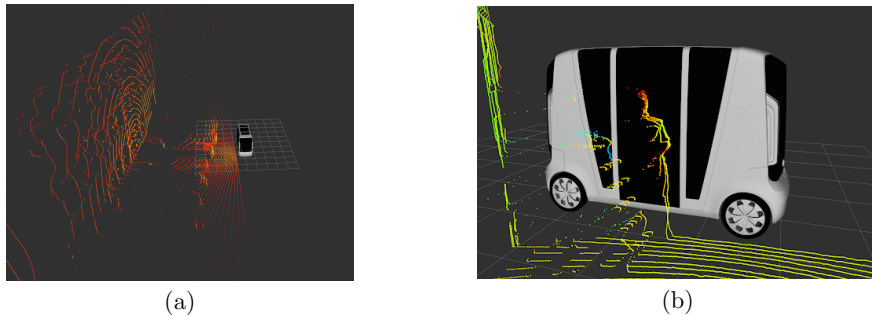


Figure 5. (a) Points cloud in outdoor environment, (b) Details around automatic door area

Another key blind zone for iseAuto shuttle is the close front area, which is not able to be detected by either front-top VLP-16 or side RS-Bpearl LiDARs, as shown in Fig.6(a). The perception of small objects (kids, pets, etc.) in this area is important for the shuttle's safety system. Available sensor choices to detect this blind zone such as IR and ultrasonic sensors have the economic advantages but compromise in the accuracy. In terms of cost and performance, solid-state LiDARs are believed to be more suitable for large-scale deployment, because solid-state LiDARs are relatively cheaper and do not have inside complex mechanical mirror systems.

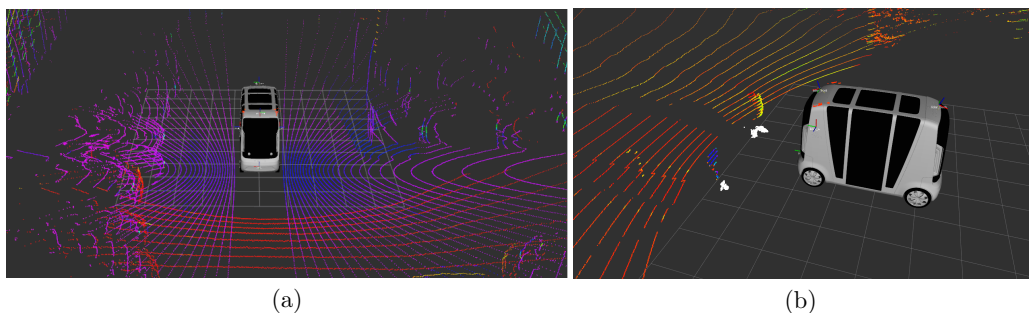


Figure 6. (a) Front blind zone, (b) Points cloud from bottom-front and top-front LiDARs

We deployed a Benewake CE30-C LiDAR on the front-bottom of the shuttle to detect the blind zones. Benewake CE30-C is a typical solid-state LiDAR that is based on the Time of Flight (ToF) ranging principle. The measurement is performed based on the received emitted modulated near-infrared light, which is reflected by the objects. Fig.6(b) shows the merged points cloud data from front-top Velodyne and front-bottom Benewake LiDARs. Benewake solid-state LiDAR can detect the down part of traffic signs and human legs (white points) that cannot be seen by the front-top Velodyne LiDAR.

4. Summary and Future Work

This paper provided an overview of the most common range sensors that are used for AVs and more specifically for AV shuttle, iseAuto. We evaluated the configuration and location of all

range sensors that were deployed on iseAuto shuttle for primary perception and blind zones detection. As a result of the analysis, we managed to get a full view of the shuttle surroundings and cover most of the vital blind zones by five LiDARs that have different characteristics.

The future work will focus on the sensor fusion and integration of the long and short range radars into the range-sensor set of the iseAuto as well as implementing AI-based situation awareness defined in the research [18]. The second target is to create a digital twin, which is compliant to our other research results [19] of the vehicle in order to simulate all critical traffic situations and increase the total safety of the deployed system.

Acknowledgments

The research is supported by the EU H2020 project Finest Twins (grant No. 856602).

References

- [1] Thrun S *et al.* 2007 Stanley: The Robot That Won the DARPA Grand Challenge *Springer Tracts in Advanced Robotics* vol 36 (Springer, Berlin, Heidelberg)
- [2] Bertozzi M *et al.* 2011 VIAC: An out of ordinary experiment *IEEE Intelligent Vehicles Symposium (IV)* pp 175-180
- [3] Gao H, Cheng B, Wang J, Li K, Zhao J and Li D 2018 Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment *IEEE Transactions on Industrial Informatics* vol 14 no 9 pp 4224-31
- [4] Wei J, Snider J M, Kim J, Dolan J M, Rajkumar R and Litkouhi B 2013 Towards a viable autonomous driving research platform *IEEE Intelligent Vehicles Symposium (IV)* pp 763-770
- [5] Broggi A *et al.* 2013 Extensive Tests of Autonomous Driving Technologies *IEEE Transactions on Intelligent Transportation Systems* vol 14 no 3 pp 1403-15
- [6] Zhang J and Singh S 2018 Laser-visual-inertial odometry and mapping with high robustness and low drift *J. Field Robotics* pp 1242-64.
- [7] Sell R, Leier M, Rassölkin A, and Ernits J 2018 Self-driving car ISEAUTO for research and education *Proc. of the 19th International Conference on Research and Education* (Mechatronics, Delft, Netherlands)
- [8] Rassölkin A, Sell R and Leier M 2018 Development case study of the first estonian self-driving car, iseauto. *Electrical, Control and Communication Engineering* 14 pp 81–88
- [9] Sell R, Coatanea E and Christophe F 2008 Important Aspects of Early Design in Mechatronic *Proc. of the 6th international conference of DAAAM Baltic industrial engineering* (Tallinn) pp 177-182
- [10] Christophe F, Sell R, Bernard A and Coatanéa E 2009 OPAS: Ontology processing for assisted synthesis of conceptual design solutions *Proc. of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* pp 249-260
- [11] Jamaluddin M H, Shukor A, Miskon M F and Redzuan M 2016 An Analysis of Sensor Placement for Vehicle's Blind Spot Detection and Warning System *J. of Telecommunication, Electronic and Computer Engineering* 8 pp 101-106
- [12] Lim B S, Keoh S L and Thing V L L 2018 Autonomous vehicle ultrasonic sensor vulnerability and impact assessment *IEEE 4th World Forum on Internet of Things (WF-IoT)* (Singapore) pp 231-236
- [13] Kim T and Park T 2020 Placement Optimization of Multiple Lidar Sensors for Autonomous Vehicles *IEEE Transactions on Intelligent Transportation Systems* vol 21 no 5 pp 2139-45
- [14] Meadows W *et al.* 2019 Multi-LiDAR placement, calibration, co-registration, and processing on a Subaru Forester for off-road autonomous vehicles operations *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2019*.
- [15] Rangesh A and Trivedi M M 2019 No Blind Spots: Full-Surround Multi-Object Tracking for Autonomous Vehicles Using Cameras and LiDARs *IEEE Transactions on Intelligent Vehicles* 4 pp 588–599
- [16] Dey J, Taylor W and Pasricha S 2020 VESPA: A Framework for Optimizing Heterogeneous Sensor Placement and Orientation for Autonomous Vehicles *IEEE Consumer Electronics Magazine* p 1
- [17] Zhou D, Ma Z and Sun J 2020 Autonomous Vehicles' Turning Motion Planning for Conflict Areas at Mixed-Flow Intersections *IEEE Transactions on Intelligent Vehicles* vol 5 no 2 pp 204-216
- [18] Wang R, Sell R, Rassölkin A, Otto T and Malayjerdi E 2020 Intelligent functions development on autonomous electric vehicle platform *J. of Machine Engineering* vol 20 no 2 pp 114-125
- [19] Kuts V, Otto T, Bondarenko Y, and Yu F 2020 Digital Twin: Collaborative Virtual Reality Environment for Multi-Purpose Industrial Applications. *ASME 2020 International Mechanical Engineering Congress and Exposition*, (ASME, November 16-19, 2020, Portland, OR, USA.)

Appendix 2

Article II

Junyi Gu, Mauro Bellone, Raivo Sell, and Artjom Lind. Object segmentation for autonomous driving using iseauto data. *Electronics*, 11(7), 2022

Article

Object Segmentation for Autonomous Driving Using iseAuto Data

Junyi Gu ^{1,*}, Mauro Bellone ², Raivo Sell ¹ and Artjom Lind ³

¹ Department of Mechanical and Industrial Engineering, Tallinn University of Technology, 12616 Tallinn, Estonia; raivo.sell@ttu.ee

² Smart City Center of Excellence, Tallinn University of Technology, 12616 Tallinn, Estonia; mauro.bellone@ttu.ee

³ ITS Lab, Institute of Computer Science, University of Tartu, 51009 Tartu, Estonia; artjom.lind@ut.ee

* Correspondence: junygu@ttu.ee

Abstract: Object segmentation is still considered a challenging problem in autonomous driving, particularly in consideration of real-world conditions. Following this line of research, this paper approaches the problem of object segmentation using LiDAR–camera fusion and semi-supervised learning implemented in a fully convolutional neural network. Our method was tested on real-world data acquired using our custom vehicle iseAuto shuttle. The data include all weather scenarios, featuring night and rainy weather. In this work, it is shown that with LiDAR–camera fusion, with only a few annotated scenarios and semi-supervised learning, it is possible to achieve robust performance on real-world data in a multi-class object segmentation problem. The performance of our algorithm was measured in terms of intersection over union, precision, recall, and area-under-the-curve average precision. Our network achieves 82% IoU in vehicle detection in day fair scenarios and 64% IoU in vehicle segmentation in night rain scenarios.

Keywords: object segmentation; LiDAR–camera fusion; autonomous driving; artificial intelligence; semi-supervised learning; iseAuto



Citation: Gu, J.; Bellone, M.; Sell, R.; Lind, A. Object Segmentation for Autonomous Driving Using iseAuto Data. *Electronics* **2022**, *11*, 1119. <https://doi.org/10.3390/electronics11071119>

Academic Editor: Stefanos Kollias

Received: 28 February 2022

Accepted: 25 March 2022

Published: 1 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The ability to detect objects in different visibility conditions has caused wide interest in computer vision techniques, which are comprehensively integrated with modern autonomous vehicles. Being aware of any obstacles around the vehicle is a critical prerequisite to achieve effective autonomous driving to ensure safe and accurate motion planning. As a matter of fact, fully autonomous driving requires a detailed classification and segmentation of objects in different illumination and weather conditions. Currently, advanced driver assistance systems (ADASs) in many cars provide reliable collision warnings with the help of radar and sonar sensors. However, ADASs can only detect the presence of obstacles in the limited premises of the vehicle; they cannot recognize the types of objects and, particularly, assign a semantic meaning.

Many state-of-the-art methods to classify objects use convolutional neural networks (CNNs) to detect 2D objects [1,2], and semantic segmentation [3] of image data created by cameras. As a passive sensor with a long history of development, cameras have advantages such as reliability and texture-density under fair illumination. However, cameras are noticeably susceptible to changes in lighting conditions. To address the problem of cameras, light detection and ranging (LiDAR) sensors have attracted broad interest from researchers. Due to the development of LiDAR sensor manufacturing, the affordability and accuracy of LiDAR sensors have been improved significantly. Therefore, more LiDAR-data-based research [4,5] for object detection and segmentation has appeared in recent years. Unfortunately, LiDAR data is sparse and non-uniformly distributed. Furthermore, it lacks texture

and color information compared to camera data. The drawbacks of LiDAR sensors make LiDAR-only-based object detection and segmentation tasks more challenging to carry out.

Considering all the benefits and downsides of camera and LiDAR sensors, the straightforward solution is to combine the information from both LiDAR point clouds and camera images. We choose to use a fully convolutional neural network (FCN), which was proposed by Caltagirone et al. [6], to perform 3D semantic segmentation. The integration of point clouds and images information was conducted at the last layers of the network, and this is can be described as a late-fusion strategy [7,8]. This choice is due to late-fusion strategies having a predefined depth level and thus being easier to build. More importantly, late-fusion systems incorporate single-modality detectors. Therefore, our method projects point clouds into the camera plane to create a three-channel tensor with the same width and height of the image, of which each channel encodes one of the 3D spatial coordinates [9].

An additional focus of this work is the domain adaptation analysis of the network from the public dataset to our custom dataset recorded during an extensive experimental campaign in the campus of TalTech. Today, open datasets available for autonomous driving have gained massive attention. For example, KITTI [10] is one of the most popular datasets that was used in deep learning research for real traffic semantic segmentation. Though successful for a very long period, KITTI is now outdated, and it no longer fulfills research needs as it includes only clear weather scenes. The latest open datasets, such as Waymo [11], Argoverse [12], and nuScenes [13], adopt state-of-the-art sensors and contain various weather scenarios.

A comprehensive dataset for fully autonomous driving tests covers most traffic cases, and different illumination and weather conditions. Collecting enough data requires a considerable amount of expense. As a result, most deep learning studies use a public dataset as the benchmark. Very little research focuses on evaluating the network for custom data. To fill this gap, this work analyzes an FCN comparing performance between the Waymo dataset and our custom dataset recorded by the iseAuto shuttle on the university campus. iseAuto is an autonomous vehicle (AV) shuttle that was designed and developed in the Autonomous Vehicles Lab in TalTech, Estonia [14–16]. This paper extends a work submitted to the IEEE International Conference on Intelligent Transportation Systems. In comparison, this paper exclusively reviews the relevant literature in the perspectives of open datasets, semi-supervised learning proposals, and deep-learning-based LiDAR-camera fusion algorithms. This version contains additional results and figures to describe many technical details, such as the sensor specifications of the iseAuto shuttle, the workflow of training procedures, and the description of data augmentation processes carried out in the data loader; furthermore, the metrics used in this paper to evaluate the models' performance are described in detail in the methodology section. Therefore, the section containing results and discussion was presented from a different perspective.

The contributions of this work are summarized as follows:

- The development of a ResNet50 [17]-based FCN to carry out a late fusion of LiDAR point clouds and camera images for semantic segmentation.
- A custom dataset (<https://autolab.taltech.ee/data/>) (accessed on 27 February 2022) that was generated by the real-traffic-deployed iseAuto shuttle in different illumination and weather scenes. The dataset contains high-resolution RGB images and point clouds information that was projected into the camera plane. Furthermore, the dataset contains manual annotations for two classes: humans and vehicles.
- The performance evaluation for the domain adaptation of the neural network from the Waymo Open dataset to custom iseAuto dataset.
- The evaluation of the contribution of pseudo-annotated data to the performance on the iseAuto dataset.

The structure of the remainder of this paper is as follows: Section 2 reviews the open datasets, semi-supervised learning proposals, and deep-learning-based LiDAR-camera fusion algorithms for autonomous driving. Section 3 introduces the splits of the Waymo and iseAuto datasets that were used in this work. Specifically, there is also a brief introduction

of the sensor configurations used to produce the iseAuto dataset. Methodologies including network structure, LiDAR projection, object segmentation, and metrics for model evaluation were described in Section 4. Section 5 reports the experimental results and discussion. Finally, a summary and conclusions were provided in Section 6.

2. Related Work

This section revisits literature on three aspects of LiDAR–camera fusion-based machine learning for object segmentation. The first part is the existing datasets specifically for autonomous driving research. The second part is the usage of semi-supervised learning to improve the overall performance of the models. The last aspect is the popular deep learning fusion algorithms to leverage the benefits of both camera and LiDAR sensors in autonomous driving.

In recent times, data is believed to be a valuable asset. Focusing on autonomous driving specifically, many research groups have dedicated themselves to producing datasets recorded by mainstream perceptive sensors and covering various scenarios. In [18,19], autonomous-driving-related datasets over the last 20 years were categorized by time of acquisition, sensor configuration, illumination, and weather conditions. As it happens, some datasets only contain sunny (including cloudy) and daytime scenes [20–22]. The datasets that possess illumination and weather diversity, such as Nuscenes [13], Waymo [11], and Argoverse [12], soon became the preferable option for training models. However, there is no consistency of the sensor configuration in all these datasets, which means it is difficult to merge the knowledge from different datasets together to improve the efficiency of the learning process. In addition, some experiment-oriented datasets were recorded by highly customized sensor modules on commercial cars. For example, in the ApolloScape [23] dataset, a particular acquisition system consisting of two laser scanners, up to six video cameras, and a combined IMU/GNSS system was mounted on top of a Toyota SUV. A platform like this requires intensive maintenance routines and is unsustainable for large-scale deployment. Very few works focus on the actual traffic pilot case considering finance and reliability. For most open datasets, enormous human effort was applied to data synchronizing, labeling, and denoising, which is not suitable for evaluating the models' performance in extreme practical situations.

To reduce the amount of human work on the data processing task, several machine learning techniques have been conceived. Semi-supervised learning is a machine learning technique involving a small amount of labeled data and much unlabeled data. It provides the benefits of supervised learning while avoiding the slow process requiring humans to review samples one by one and give them the correct label. Recent survey papers [24,25] summarize both previous and new research on semi-supervised learning, presenting a full picture of the topic according to various taxonomies. The literature of semi-supervised learning can be explored in different ways, referring to the availability of labels and their relationship to the supervised learning algorithms. The classic methods include generative models [26,27], semi-supervised support vector machines [28], and graph-based methods [29]; all have a long research history. The method related to our work is pseudo-labeling, which relies on high-confidence pseudo-labels added to the training split as labeled data. There are two main patterns of the pseudo-labeling methods. The first one is based on using disagreeing views from multiple networks to improve performance. A typical example is co-training [30], a method to train two different models by using different data splits. It is an iterated process that passes the prediction from one model to the other; thus, each model is retrained with the additional unlabeled samples given by the other model. The pattern used in our work is self-training, which is one of the earliest semi-supervised learning ideas and can be dated back to the 1970s [31]. It starts by training on the labeled data first. Then, part of the unlabeled data is predicted according to the current decision function. The most confident prediction will be added to the training set for the supervised learning algorithm. This procedure is repeated in self-training methods until all the unlabeled examples have been predicted. The latest self-training

research, for example [32], first trains a teacher model with a labeled dataset, then uses the teacher model to generate labels for an unlabeled dataset that is re-used to train a student model. The authors prove that the student model outperforms the teacher model with the Cityscapes [33], CamVid [34], and KITTI [10] datasets. Xie et al. [35] propose utilizing various techniques such as data augmentation, dropout, and stochastic depth to train the student models. Similar approaches can be found in [36]; in addition to data augmentations, extra cropping, rotating, horizontal mirroring, and color randomization were also used to improve the model performance.

Recent breakthroughs in deep learning have significantly improved the capability of LiDAR–camera fusion algorithms. The main applications that benefit from deep-learning-based LiDAR–camera fusion methods include depth computation, object detection (bounding box), and semantic segmentation. Although it is possible to extract the 3D geometry from vision-based systems, LiDAR sensors naturally have the accuracy advantage in long-range, textureless scenarios (such as nighttime scenes). The purpose of LiDAR–camera fusion for depth computation is to combine the two sensors’ merits to acquire a dense and accurate depth map. Ma et al. [37] propose a self-supervised learning model that requires only sequences of RGB and sparse depth images for training. The deep regression model learns a direct mapping from sparse depth input to dense depth prediction. In [8], early- and late-stage fusions were combined in an image-guided framework that consisted of a global and local network to process RGB data and depth information in parallel. The stereo-camera system is also widely used for depth completion because of the rich 3D geometry in its disparity map. An example work is [38], which shows a two-stage CNN design that first produces fused disparity by LiDAR and stereo disparity, then computes the final disparity by fusing the fused disparity and left RGB image at the feature level. Three-dimensional object detection aims to recover the pose and the bounding box dimensions for all objects of interest in the scene. An example of early-stage fusion, [39] uses a ResNet [17] and a PointNet [40]-based network to process cropped image and raw point cloud data. Afterward, two fusion networks were used to regress the box corner locations and predict the spatial offset of each corner relative to an input point, respectively. Liang et al. [41] present a multi-task multi-sensor 3D object detection network. The authors exploit the fact that multiple complementary tasks such as 2D object detection, ground estimation, and depth completion are helpful for the network to learn better feature representations. In contrast to 3D object detection that classifies the bounding boxes of objects, semantic segmentation aims to predict per-pixel and per-class labels. Su et al. [42] employ bilateral convolution layers in their network to compute spatially aware features of point clouds data. Features from images and point clouds were fused to predict per-point labels. Another common semantic segmentation application for autonomous driving is road surface detection; related research includes [6,43,44].

3. Dataset

As mentioned in the introduction, the primary purpose of this work is to evaluate the model’s performance when adapting it from a public dataset to a realistic and coarser environment. To train the supervised learning baseline models, we use the Waymo Open dataset [11]. The dataset for semi-supervised learning and domain adaptation analysis is collected using our custom vehicle, the iseAuto shuttle.

3.1. Waymo Open Dataset

Waymo Open dataset is an open-source autonomous driving dataset captured by a high-quality camera and LiDAR sensors. The diversity across different weather and illumination conditions of Waymo Open dataset offers opportunities in the research for domain adaptation, which is one of the primary purposes in our work. Therefore, we manually partition a total of 22,000 frames of data into four sub-categories based on weather (fair or rain) and illumination (day or night) conditions. This includes a total of 14,940 frames in the day-fair subset, 4520 frames in the day-rain subset, 1640 frames in

the night-fair subset, and 900 frames in the night-rain subset. Note that the proportions of different weather conditions are highly unbalanced in the Waymo dataset since more data was recorded under sunny daytime. Correspondingly, more frames (80%) were used for training in the day-fair subset, while 60% of total frames were used in training for the other three subsets.

Theoretically, K-fold cross-validation should be applied to reduce the performance-dependence from the specific data split. However, the focus of this paper is transferring the best knowledge gained in supervised learning to the new domain. Therefore, combining the holdout method and early-stopping validation is more suitable in our case and saves a large amount of training time. All frames were randomly shuffled before splitting them into the training, validation, and testing datasets. The proportion for early-stop validation is 10% of all four subsets, and the remaining data were used for testing.

3.2. iseAuto Dataset

3.2.1. iseAuto Sensor Configuration

The sensor configurations of the iseAuto shuttle evolved along with the comprehensive practical tests [45]. The location of primary range sensors changed from two front-top corners to the middle of the front-top and back-top. Two 90° vertical field-of-view (FoV) LiDARs on two sides of the shuttle were installed to cover the essential side-blind zones, especially in the proximity of door areas. Our latest upgrade is installing two solid-state LiDARs on the inside-door-top and outside-front-bottom for the door-movements safety and emergency brake, respectively. The main camera was installed inside the cabin, located in the front and behind the windshield. Figure 1 illustrates the positions and orientations of all perceptive sensors for the iseAuto shuttle.

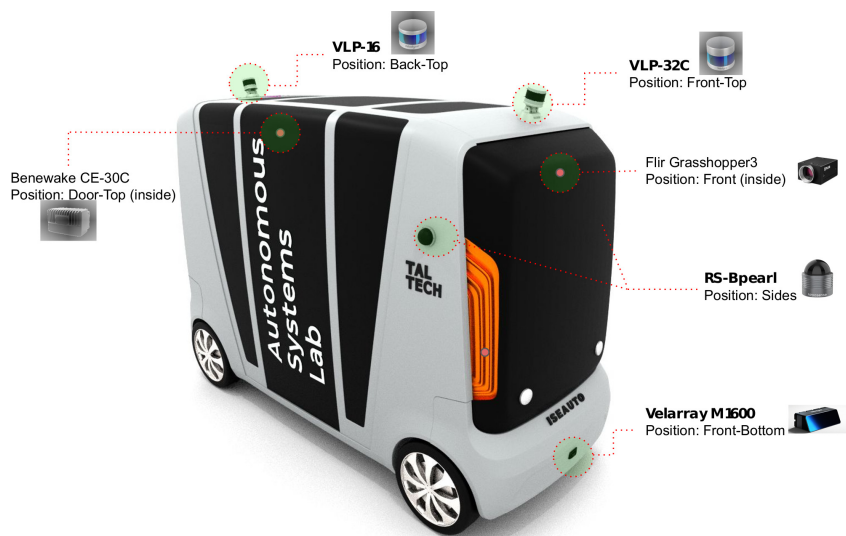


Figure 1. Perceptive sensors layout. The Benewake CE-30C LiDAR is located inside the shuttle and attached on top of the door.

In this work, the data collection was conducted by using the front-top Velodyne VLP-32C and front-inside FLIR Grasshopper3. As the two primary perceptive sensors for the iseAuto shuttle, the Velodyne VLP-32C has 32 channels that provide dense points clouds. The resolution of FLIR Grasshopper3 is up to 4240×2824 to guarantee a sharp vision of small objects such as traffic signs in the distance. Table 1 contains detailed specifications of the camera and LiDAR sensors.

Table 1. Specifications of the primary camera and LiDAR sensors of the iseAuto shuttle.

	FoV (°)	Range (m)/Resolution	Update Rate (Hz)
Velodyne VLP-32C	40 (vertical)	200	20
Grasshopper3	89.3 (D) 77.3 (H) 61.7 (V)	4240 × 2824	7

3.2.2. iseAuto Dataset Split

The environment of the iseAuto dataset is the TalTech campus, where the experimental campaign was conducted. Compared to the Waymo dataset used in the supervised learning baseline model, the night subset of the iseAuto data has a lower illumination condition. The ambient light of the campus is typically darker than the urban area where Waymo records their night data (shown in the first column of Figure 2). The partition of the iseAuto dataset follows the same principle as the Waymo dataset partitions, with four categories: day-fair, day-rain, night-fair, and night-rain. Both LiDAR and camera sensors were set to work at 7 Hz, which is the maximum frequency that the camera can shoot at 4k resolution. Correspondingly, one out of every seven frames of all point clouds and images were selected. To avoid the unbalanced data allocation that exists in the Waymo dataset (more data in day and fair conditions, less in night and rain), the total number of frames in each subset of the iseAuto dataset is 2000, to make sure that the same amount of knowledge can be gained from different weather and illumination conditions. For each subset, 600 frames were manually annotated with vehicle and human pixel-level classes. The data splits for training, early-stopping validation, and testing of all subsets have 300, 100, and 200 frames, respectively.

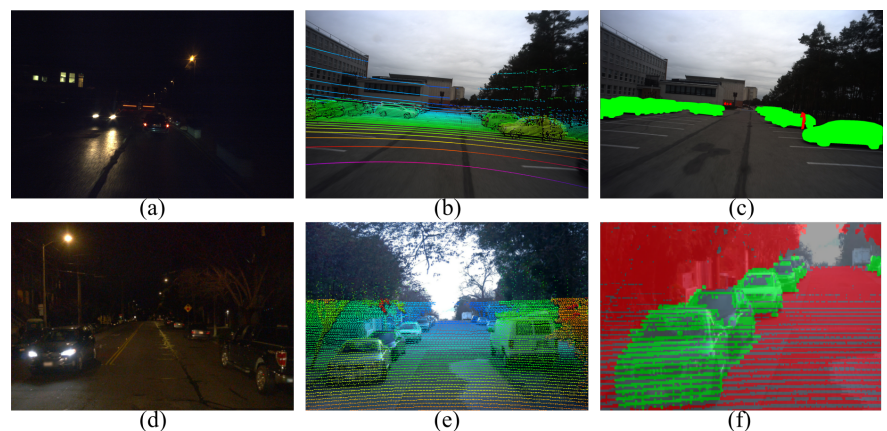


Figure 2. Extract from the iseAuto dataset (a–c) on the first row; the second row represents the Waymo dataset (d–f). For additional details, please refer to Section 4.

4. Methodology

Our work is an extension of the research presented in [9], where the authors differentiate the Waymo dataset by illumination (day/night) and weather (fair/rain) conditions to perform LiDAR–camera fusion and semi-supervised learning for semantic segmentation. We first follow the same principle to partition the Waymo dataset into four subsets (day-fair, day-rain, night-fair, and night-rain), then use them for the baseline model training. Secondly, we transfer the knowledge gained in the baseline supervised learning to the iseAuto dataset, in order to evaluate the network’s performance in new domains. At last, we conduct semi-supervised learning, which was expected to further improve the performance

and domain adaptation. This section describes the data processing, model construction, and training procedures specifically designed for our work.

4.1. LiDAR Point Cloud Projection

One of the most common methods to process LiDAR data is converting 3D point clouds to 2D occupancy grids to efficiently exploit existing 2D convolutional networks. The late-fusion FCN used in this work requires both LiDAR and camera input as 2D images. Therefore, we apply perspective projection to project the point clouds into the camera plane for the Waymo and iseAuto datasets. This means that both LiDAR data and camera information need to be transformed into tensors. The usual procedure is to build n -2D tensors of a specific size (for instance, the input of the CNN); thus, the 3D LiDAR information should be projected into 2D tensors. For our case, the camera image tensor is $C_i \in \mathbb{R}^{h,w,3}$, where h is the height of the camera image, w is the width, and 3 is for the RGB color channels. Analogously, the LiDAR tensor is $L_i \in \mathbb{R}^{h,w,3}$, where h and w are the same height and width, and we use the three channels to represent the LiDAR data projection into the XY-YZ-ZX planes. The projection is carried out in a typical reference frame. For our case, the camera reference frame was chosen. Let $p_i^L = [x_i, y_i, z_i]^T$ be the i -th point of the point cloud obtained as a LiDAR reading, in its own reference frame. Please observe that the reflective intensity value is ignored. The first step is to transform the point cloud from the LiDAR to the camera reference frame using a homogeneous transformation matrix, $p^C = T_C^L p^L$, where $p^C = [x, y, z, 1]$ is a point represented in the camera reference frame, $p^L = [x, y, z]$ is a point represented in the LiDAR reference frame, and $T_C^L \in \mathbb{R}^{4 \times 4}$ is the LiDAR–camera transformation matrix.

Now, it is possible to simply project each point in a 2D image, and thus each pixel value (u, v) of a generic point p_i , where $u = 1, \dots, h$ is the row pixel coordinate, $v = 1, \dots, w$ is the column coordinate in pixels, and h, w are the height and width of the camera image. Let R be the rectification matrix, and P the projection matrix; then, $[u, v, 1]^T = PRp^C$.

The procedure above is applied to all points in the point clouds data. To ensure that the projected LiDAR plane has the same dimensions as the camera image, only the points within the field of view will be selected.

The Waymo dataset encodes the LiDAR data as range images with the same camera images format. Each pixel in the range image corresponds to a laser point reading. All the point information, such as range, vehicle pose, and camera projection, are included in the range image pixel. With the assistance of the toolkit, developers can directly extract point clouds images and well-overlaid camera projection from the Waymo dataset; thus, there no need to deal with the raw data.

For the iseAuto dataset, since the shuttle was operated upon by the robot operating system (ROS), LiDAR and camera data were captured as corresponding ROS formats and stored as the bag files. Pre-processes are needed to handle the point clouds and images. Extrinsic calibration of the LiDAR and camera first must be executed to compute the camera projection matrix and the LiDAR–camera transformation matrix. The rectification matrix was set to identify when projecting point clouds to images for the iseAuto dataset because rectification has been done internally by the camera. An example output of point clouds projection is shown in the second column of Figure 2. Note that the alignment of LiDAR points and image pixels is not ideal without extra operations to optimize the calibration and synchronization of LiDAR and camera sensors. Nevertheless, errors and interference always exist in the real world, which are the factors that we want to consider in this work through the iseAuto dataset.

4.2. Object Segmentation

Ground truth annotation is essential in machine learning and requires many labor costs. In the Waymo dataset, annotations were created separately for LiDAR and camera data. There are four kinds of objects (vehicles, pedestrians, signs, and cyclists) labeled in LiDAR sensor readings and three kinds of objects (vehicles, pedestrians, and cyclists)

labeled in camera images. Both 3D and 2D annotations were represented by bounding boxes. Our process for the Waymo dataset is based on their LiDAR annotations. We select all LiDAR points within the 3D bounding box and project them into the image plane. Each projected point corresponds to a pixel in the image; the set of all projected points creates the semantic mask of the objects. Compared to the 2D annotations, the most important advantage of 3D annotations is that they provide a contour of the objects at a close distance. Correspondingly, the drawback of using 3D annotation is that some pixels in the semantic mask do not have labels because no LiDAR points fall into this area (see the third column of Figure 2).

The annotations of the iseAuto dataset were created based on camera images, as our high-resolution images contain more details of small objects (or objects that are far away). We develop a labeling tool that allows human annotators to draw objects' contours in images and save the segmented area with its corresponding label. Semantic masks in the iseAuto dataset are flood-filled, which means all pixels in the mask have a unique label. Moreover, our annotations have an awareness of the contour of objects. It is a fact that human error is inevitable in manual labeling work. For scenarios with poor illumination conditions, point cloud projection was also used to identify possible objects that are not clearly visible in the camera image. For the scope of this work, the resolution of annotation images in the iseAuto dataset is 1920×1280 ; only vehicles and humans were masked out. Further work includes labeling higher-resolution images and more object classes. More objects and label verification are also needed.

Figure 2 shows some extracts of both datasets. The first row corresponds to the iseAuto dataset, while the second row shows the Waymo dataset. The comparison of the illumination condition in night scenarios is shown in the first column. Please note that, in similar scenes, the iseAuto dataset is typically darker than the Waymo dataset due to the external illumination and light source from the vehicle itself. The second column contains an example of the point clouds projection. The camera coordinate of points was used to pick out the corresponding pixels in the image. The colors of the pixels were assigned using the HSV palette, based on the depth information of the point. An upsampling process was used to make the iseAuto projection, as the point projection into a 4k resolution image is visually sparse. The third column illustrates the annotations of two datasets. As discussed above, no-label-zones exist in the Waymo dataset annotations because of the nature of LiDAR sensors. These areas must be excluded from the metrics calculation. The annotation of the iseAuto dataset is based on the camera image, in which the object masks are solid-filled and contain contour information.

4.3. Model

The model implemented in this work can be considered the composition of three different submodels. All of them are based on a well-known pre-trained ResNet50 [17] model. The first model works only on camera images with their respective labels. The second model has LiDAR data input instead. Lastly, the fusion model works as a joint combination of feature maps coming from the camera images and the LiDAR point clouds, which can be considered a late-fusion strategy. For each step, the loss function can be calculated at the output of each submodel. This strategy is further described in [9].

4.4. Training

There are three training procedures for transfer learning experiments. The first step is training supervised learning baseline models with only the Waymo dataset. The saved models in this step were tested separately by the Waymo and iseAuto datasets. The second procedure is the transfer learning experiment. Supervised learning baseline models of the Waymo were continuously trained by the iseAuto dataset. To assess the contribution of the knowledge attained from the Waymo dataset in the transfer learning process, there is also a training process to get iseAuto baseline models (trained by only the iseAuto dataset from scratch) in this step. The last procedure is semi-supervised learning (SSL). The literature is

rich for SSL methods, such as the teacher-student model, co-training, or pseudo-labeling. Machine-made annotations of the unlabeled dataset were made by the transfer learning models from the Waymo to iseAuto dataset. Then unlabeled and labeled iseAuto data were mixed to continuously train the transfer learning models, and to train the iseAuto baseline model from scratch. Figure 3 summarily illustrates the training procedures that were mentioned above.

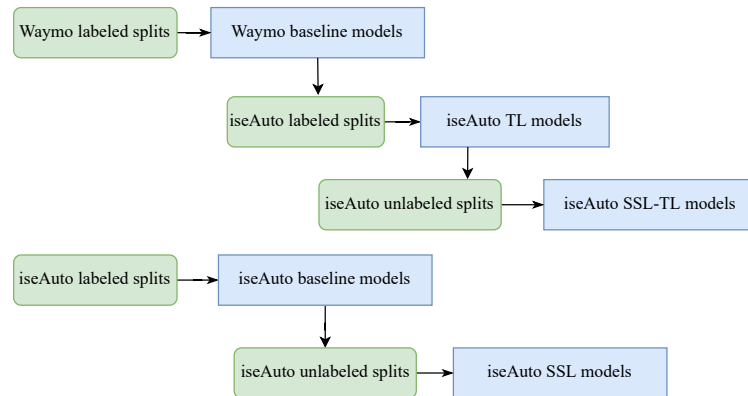


Figure 3. The workflow of the training procedures. ‘TL’ and ‘SSL’ stand for transfer learning and semi-supervised learning, respectively. ‘Waymo labeled splits’ and ‘iseAuto labeled splits’ represent the Waymo and iseAuto manual-labeled data. ‘iseAuto unlabeled splits’ means the iseAuto machine-labeled data produced by the iseAuto transfer learning fusion model. ‘Waymo baseline models’ and ‘iseAuto baseline models’ stand for supervised learning baseline models of the the Waymo and iseAuto dataset. ‘iseAuto TL models’ means the Waymo-to-iseAuto transfer learning models. ‘iseAuto SSL-TL models’ and ‘iseAuto SSL models’ are iseAuto semi-supervised learning models with and without knowledge adapted from the Waymo dataset, respectively. Please refer to Section 4.4 for further details.

Cross-entropy loss fusion and Adam optimization [46] were used in this work. The hardware used for training is an Nvidia RTX2070 Super GPU. The batch size is 16. An early stopping mechanism was applied to all training processes. The learning rate decay follows the equation:

$$n(i) = n_0 \left(1 - \frac{i}{N}\right)^a \quad (1)$$

where the n_0 is the starting learning rate, a is 0.9, and N was denoted as the total iterations. Data augmentation was composed of random crop, random rotate, color jitter, and random horizontal and vertical flip. Figure 4 shows an example of the data augmentation. The output size of the random crop is 128×128 . The random rotate range is $(-20^\circ, 20^\circ)$, referring to the center of the images. The probability of executing the random vertical and horizontal is 50%. To maximize the diversity of the data augmentation, the order of the five augmentation processes was shuffled in every iteration. Remarkably, data normalization plays a vital role in this work, especially for the LiDAR data. Given the significant differences in specifications of the LiDAR sensors used in the Waymo and iseAuto datasets, the normalization of point clouds data of the two datasets has different factors. The x, y, z coordinates of all points were appended together to compute mean and standard deviation values. Further fine-tuning to the normalization parameters was conducted to ensure the best performance.

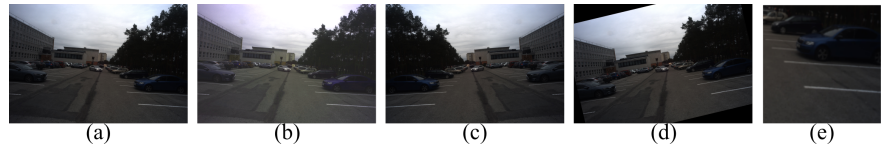


Figure 4. Data augmentation processes: (a) is the original image, (b) is color jitter, (c) is horizontal flip, (d) is random rotate in the range $(-20^\circ, 20^\circ)$, (e) is the random crop of dimension 128×128 .

4.5. Metrics

The measurements that were used to evaluate the performance of models include intersection over union (IoU), precision, recall, and area-under-curve average precision (auc-AP) [47].

IoU, also known as Jaccard index, is a measure to compare the similarity between two sample sets. The practical applications of IoU are mainly related to object detection, especially in the field of machine learning, to train a model to output boxes that fit around the objects. In this case, the ground-truth boxes (the hand-labeled bounding boxes that specify the location of the objects in the image) are needed to compute the IoU. The calculation is as follows:

$$\text{IoU} = \frac{A_I}{A_U} \quad (2)$$

where A_I is the overlap area, and A_U is the union area of predicted and ground-truth boxes. The overlap and union area calculation is based on the image coordinates of the bounding box corners. However, our model produces instance segmentation of objects (per-pixel labeling), instead of bounding boxes. Therefore, we adopt a pixel-wise multi-class IoU algorithm to evaluate the model. Two object classes (vehicles and humans) were detected in this work. It was assumed that V represents the vehicle class and H represents the human class. The total number of pixels inferred as vehicle class (or human class) in both prediction and ground-truth was denoted as $V_p V_g$ (or $H_p H_g$). $V_p H_g$ represents the number of pixels indicated as a vehicle in prediction, but human in ground-truth. Similarly, $H_p V_g$ is the number of pixels labeled as human in prediction, but a vehicle in ground-truth. The IoU of two classes is attained by:

$$\text{IoU}_V = \frac{V_p V_g}{V_p V_g + V_p H_g + H_p V_g} \quad (3)$$

$$\text{IoU}_H = \frac{H_p H_g}{H_p H_g + H_p V_g + V_p H_g} \quad (4)$$

Precision is a metric to reflect the model's reliability in classifying samples as positive. It is defined as the ratio between the number of positive samples correctly classified and the total number of samples classified as positive (either correctly or incorrectly). In our case, the total number of pixels detected as vehicle or human by the model is the denominator of the precision calculation. Therefore, the precision of the two classes is given by the following equations:

$$\text{Precision of vehicle} = \frac{V_p V_g}{V_p V_g + V_p H_g} \quad (5)$$

$$\text{Precision of human} = \frac{H_p H_g}{H_p H_g + H_p V_g} \quad (6)$$

Recall indicates the capability of the model to detect the positive result. It is the ratio between the number of positive samples correctly classified as positive and the total number of positive samples. In our work, the recall of two classes is calculated by:

$$\text{Recall of vehicle} = \frac{V_p V_g}{V_p V_g + H_p V_g} \quad (7)$$

$$\text{Recall of human} = \frac{H_p H_g}{H_p H_g + V_p H_g} \quad (8)$$

In general, precision measures the capability of the model to classify the positive samples, but it does not consider correctly classifying all positive samples. On the contrary, recall measures the number of positive samples that the model correctly classified, but it neglects if the negative samples were classified as positive. High-precision-and-low-recall means that the model is reliable if it classifies a sample as positive, but only a few positive samples were classified. By contrast, low-precision-and-high-recall means most positive samples were correctly classified, but there are also many negative samples classified as positive by the model.

Plotting precision (y-axis) against recall (x-axis), named the precision-recall curve, is an efficient way to analyze the tradeoff between precision and recall at various thresholds. Average precision (AP) summarizes the information of a precision-recall curve into a single value. Typically, AP is defined as the area under the precision-recall curve between 0 and 1. In practice, the integral is simplified as the sum over the precision of different thresholds multiplied by the corresponding change in the recall. The auc-AP that was used in this work was proposed by PASCAL VOC 2010 [47], and it computes the AP as a numerical integration, with precision monotonically decreasing, by setting the precision for recall r to the maximum precision obtained for any recall $r' \geq r$. The equation for computing auc-AP is:

$$\text{auc_AP} = \sum_{k=1}^N \Delta r(k) \max_{\tilde{k} \geq k} p(\tilde{k}) \quad (9)$$

5. Results and Discussion

As mentioned in Section 4.4, we conduct three training procedures in this work, which are described in this section and structured in the following way. We first evaluate the supervised learning baseline model of the Waymo dataset. Next, we analyze the transfer learning from the Waymo dataset to the iseAuto dataset. Finally, semi-supervised learning was applied for both iseAuto baseline and transfer learning models to assess its performance.

5.1. Waymo Supervised Learning Baseline

The Waymo supervised learning baseline models were trained by using all weather and illumination sequences of the Waymo dataset. As described in Section 3.1, the holdout method was used in the Waymo dataset to create the testing splits. Table 2 refers to the result of the models trained and tested using the Waymo dataset only, corresponding to RGB, LiDAR, and fusion modes. This first test shows that our network compares well with other state-of-the-art works in terms of instance segmentation for the Waymo dataset [48,49]. Please note that this paper does not aim to outperform the Waymo benchmarks, but rather to analyze how much knowledge gained from the Waymo dataset can be transferred to a custom dataset to achieve good performance with only a limited amount of labeling work. Therefore, the same model trained by the Waymo dataset was tested by the iseAuto data without any additional training; the result is shown in Table 3.

As the most represented class in the two datasets, the IoU and auc-AP of vehicles detection in the Waymo dataset reaches 93% and 96%, respectively. In the iseAuto dataset, the fusion model's performance in vehicles detection is acceptable, ranging from 45% to 56% for IoU, and from 52% to 68% for auc-AP in challenging nighttime scenarios. By contrast, humans, which are smaller than vehicles in size and less represented in both datasets, show a lower segmentation accuracy than vehicles in the iseAuto dataset. Particularly for the LiDAR model, the performance degrades, as shown in Table 3. The knowledge gained from the Waymo LiDAR data seems to be less effective in detecting humans in the iseAuto dataset. This was expected due to the different LiDAR sensors used to capture the two datasets. It is not easy to compare data from various sensor technologies.

Table 2. Performance of supervised learning Waymo baseline models tested by the Waymo dataset.

		IoU (%)		Precision (%)		Recall (%)		auc-AP (%)	
		Vehicle	Human	Vehicle	Human	Vehicle	Human	Vehicle	Human
Day-Fair	camera	88.08	55.57	91.21	59.77	96.25	88.77	92.60	71.13
	LiDAR	88.58	53.04	91.23	55.94	96.82	91.08	93.23	69.89
	fusion	91.07	62.50	93.05	65.16	97.72	93.87	94.35	76.05
Day-Rain	camera	88.54	52.13	91.14	57.43	96.88	84.97	94.04	76.12
	LiDAR	89.47	50.06	91.38	53.04	97.73	89.92	94.83	73.63
	fusion	92.77	64.66	94.35	68.53	98.23	91.97	95.80	84.55
Night-Fair	camera	81.16	42.87	86.77	49.33	92.62	76.60	86.74	61.10
	LiDAR	86.16	48.83	89.35	52.51	96.02	87.46	92.38	68.98
	fusion	89.41	60.33	91.96	65.08	97.00	89.22	92.18	73.02
Night-Rain	camera	74.49	43.14	83.39	51.91	87.47	71.87	85.83	53.04
	LiDAR	87.51	46.68	90.72	48.44	96.11	92.77	92.90	53.87
	fusion	89.90	56.70	92.86	60.84	96.58	89.28	94.52	66.81

Table 3. Performance of supervised learning Waymo baseline models tested by the iseAuto dataset.

		IoU (%)		Precision (%)		Recall (%)		auc-AP (%)	
		Vehicle	Human	Vehicle	Human	Vehicle	Human	Vehicle	Human
Day-Fair	camera	63.64	64.39	84.15	66.74	72.07	94.81	83.04	71.30
	LiDAR	40.56	0.06	63.19	57.87	53.12	0.06	49.68	0.35
	fusion	60.07	12.68	81.48	76.20	69.57	13.20	72.45	24.98
Day-Rain	camera	51.51	13.66	54.28	16.60	91.00	43.58	68.11	27.39
	LiDAR	43.56	2.86	67.96	11.96	54.81	3.62	51.98	7.62
	fusion	69.19	14.75	81.86	40.28	81.73	18.89	75.84	35.33
Night-Fair	camera	45.06	29.42	73.21	63.85	53.96	35.30	62.84	55.86
	LiDAR	41.75	0.54	56.92	20.72	61.04	0.55	48.20	1.61
	fusion	55.68	5.07	75.33	69.82	68.09	5.18	68.26	13.36
Night-Rain	camera	17.34	5.64	19.34	13.71	62.72	8.74	24.43	20.33
	LiDAR	33.55	0.01	48.09	0.33	52.59	0.01	41.26	0.08
	fusion	44.90	7.72	59.60	55.61	64.53	8.23	51.83	56.41

Specifically, in Table 2, precision values are lower than recall values (more significant difference for human class), which means that most of the objects were correctly classified. However, models also recognize some pixels belonging to other classes (e.g., background) as humans. In Table 3, it is the opposite: recall values are typically lower than precision values, which means that models cannot classify most of the objects correctly, but detection is relatively reliable. This shows that models trained by only the Waymo dataset realize the locations of the objects in the iseAuto dataset, but cannot draw out the whole object's contour.

5.2. Transfer Learning to iseAuto

In the transfer learning experiment, supervised learning baseline models of Waymo were continuously trained using 1200 frames of iseAuto data that included different illumination and weather scenarios. Table 4 provides the metric results of the Waymo-to-iseAuto transfer learning models. For comparison, the same amount of iseAuto data was also used to train the iseAuto baseline models. The models' performances are shown in Table 5.

By comparing Tables 4 and 5, one can note that, with the exception of human segmentation in the LiDAR model, all other metric results increase with transfer learning, as expected, even in challenging conditions such as night and rain. However, compared to the iseAuto baseline model, the transfer learning camera model significantly improves

human segmentation, which results in the fusion model in the transfer learning process also generally performing better than the baseline. Please note that the amount of iseAuto data for training, 1200 frames in total, is much smaller than the Waymo data (16,188 frames) used for transfer learning and domain adaptation. The transfer learning fusion model has the best accuracy at this stage and is used to generate the machine-made labels for the unlabeled iseAuto data.

Table 4. Performance of the iseAuto transfer learning models from the Waymo dataset.

		IoU (%)		Precision (%)		Recall (%)		auc-AP (%)	
		Vehicle	Human	Vehicle	Human	Vehicle	Human	Vehicle	Human
Day-Fair	camera	77.10	75.87	85.02	79.40	89.22	94.46	84.59	81.67
	LiDAR	72.14	55.71	81.07	57.48	86.75	94.75	80.65	61.08
	fusion	83.27	74.24	89.41	76.46	92.38	96.24	88.34	82.91
Day-Rain	camera	80.26	48.11	85.82	67.13	92.53	62.93	84.15	72.22
	LiDAR	77.33	40.27	82.35	45.06	92.70	79.11	81.48	63.36
	fusion	84.92	57.61	88.75	65.08	95.16	83.37	87.99	73.99
Night-Fair	camera	66.07	52.38	75.02	61.38	84.71	78.13	77.61	74.58
	LiDAR	74.50	45.38	80.58	47.78	90.79	90.04	82.93	60.75
	fusion	80.43	64.03	86.55	73.18	91.92	83.67	87.66	76.88
Night-Rain	camera	51.70	41.39	63.11	47.21	74.09	77.06	61.50	63.29
	LiDAR	62.51	26.46	68.24	27.05	88.15	92.38	73.02	50.79
	fusion	67.89	45.68	75.26	49.48	87.40	85.61	79.46	74.34

Table 5. Performance of the supervised learning iseAuto baseline models.

		IoU (%)		Precision (%)		Recall (%)		auc-AP (%)	
		Vehicle	Human	Vehicle	Human	Vehicle	Human	Vehicle	Human
Day-Fair	camera	75.97	71.31	86.43	74.10	86.26	94.99	84.26	79.01
	LiDAR	71.19	56.87	78.74	59.01	88.14	94.03	78.00	66.99
	fusion	80.39	74.56	87.26	77.63	91.08	94.97	86.40	83.10
Day-Rain	camera	77.71	39.87	81.15	51.82	94.82	63.35	82.28	66.49
	LiDAR	76.00	42.10	81.44	46.52	91.93	81.58	80.43	59.12
	fusion	83.20	56.24	87.37	65.16	94.58	80.43	87.52	75.12
Night-Fair	camera	68.89	54.98	76.04	62.79	87.99	81.54	79.27	73.55
	LiDAR	74.25	47.19	80.03	50.52	91.13	87.75	82.96	54.16
	fusion	76.79	62.48	85.75	75.66	88.02	78.19	87.11	77.40
Night-Rain	camera	52.17	29.40	60.88	32.26	78.49	76.81	66.67	54.27
	LiDAR	59.49	36.76	64.82	37.91	87.85	92.33	82.30	62.08
	fusion	64.68	46.09	74.42	50.30	83.17	84.64	78.96	76.59

5.3. Semi-Supervised Learning with Pseudo-Labeled Data

Semi-supervised learning uses the unlabeled iseAuto dataset, applied to the iseAuto baseline models and Waymo-to-iseAuto transfer learning models. For each subset, there are 1400 frames of data labeled by the Waymo-to-iseAuto transfer learning fusion model (the best-performing model in earlier experiments). The machine-labeled data was mixed with human-labeled frames to perform the semi-supervised training. The same iseAuto data was used in all testing processes to ensure a parallel comparison. The evaluation of semi-supervised learning models is illustrated in Tables 6 and 7.

Referring to Table 6, the semi-supervised learning iseAuto baseline models show 84% IoU and 89% auc-AP for vehicle segmentation in fair illumination and weather conditions. By comparing Tables 5 and 6, it is possible to see that the vehicle segmentation shows robust performance improvement even in more challenging scenarios with the help of the semi-supervised learning. The human segmentation is a weak point in this stage, which

can be explained by the fact that the humans class is less represented in the dataset; too few human samples are being recorded in the iseAuto dataset. Please note that recall shows an effective increase in semi-supervised learning baseline models, which means that there is an improvement of the models' capability to detect the positive human samples. This corresponds to the general principle in machine learning that more data can bring higher performance. While there is a minor decline in precision, which means models detect more negative samples as the human class, it proves the extra unlabeled data increases the model's uncertainty about the human class in semi-supervised learning.

Table 6. Performance of the semi-supervised learning iseAuto baseline models.

		IoU (%)		Precision (%)		Recall (%)		auc-AP (%)	
		Vehicle	Human	Vehicle	Human	Vehicle	Human	Vehicle	Human
Day-Fair	camera	79.85	67.06	85.27	68.18	92.63	97.62	85.86	75.88
	LiDAR	73.69	58.05	81.61	59.37	88.37	96.32	80.55	64.68
	fusion	82.38	68.98	87.24	69.98	93.67	97.96	87.72	76.36
Day-Rain	camera	80.27	53.61	82.57	56.91	96.64	90.24	84.41	67.23
	LiDAR	80.58	44.09	84.84	48.41	94.13	83.14	84.25	59.23
	fusion	83.98	54.28	87.13	56.95	95.87	92.06	88.63	66.87
Night-Fair	camera	73.14	55.07	78.67	61.71	91.23	83.66	81.74	69.23
	LiDAR	75.75	49.59	79.99	52.96	93.46	88.63	84.24	60.42
	fusion	79.28	56.32	82.34	59.81	95.52	90.61	86.68	71.81
Night-Rain	camera	60.42	42.06	66.33	43.80	87.16	91.37	69.26	68.97
	LiDAR	64.89	41.32	70.75	42.21	88.69	95.15	75.68	67.30
	fusion	63.97	43.63	69.38	44.74	89.13	94.59	75.67	67.84

Table 7. Performance of the semi-supervised transfer learning iseAuto models.

		IoU (%)		Precision (%)		Recall (%)		auc-AP (%)	
		Vehicle	Human	Vehicle	Human	Vehicle	Human	Vehicle	Human
Day-Fair	camera	80.32	69.25	87.41	70.53	90.83	97.45	85.04	76.99
	LiDAR	76.10	61.81	83.28	63.30	89.83	96.34	81.32	71.22
	fusion	82.85	71.09	87.93	72.55	93.49	97.24	87.91	78.48
Day-Rain	camera	82.49	57.12	86.33	60.85	94.87	90.31	87.75	69.98
	LiDAR	81.00	44.85	85.03	49.57	94.48	82.50	85.05	60.68
	fusion	85.04	54.84	88.16	61.61	96.00	83.32	88.36	70.4
Night-Fair	camera	75.97	55.46	83.13	65.45	89.81	78.41	84.64	71.00
	LiDAR	76.01	51.63	80.16	55.07	93.63	89.20	84.01	64.51
	fusion	79.82	60.21	83.88	67.71	94.28	84.46	88.20	73.43
Night-Rain	camera	60.79	48.30	69.38	51.45	83.07	88.76	71.65	72.03
	LiDAR	64.40	41.15	69.95	42.17	89.04	94.45	73.63	64.49
	fusion	66.92	48.36	73.19	50.64	88.65	91.49	77.76	72.81

Table 7 evaluates the semi-supervised learning iseAuto models with the transfer learning knowledge from the Waymo dataset. The best-performing Waymo supervised learning baseline models were continuously trained by full-annotated iseAuto dataset. Comparing the results to Table 6, major improvement can be seen in all modalities and domains, which means the knowledge gained from the Waymo dataset is still valuable for the semi-supervised learning stage. Compared to the transfer learning iseAuto models without semi-supervised learning (Table 4), the individual RGB and LiDAR networks have a maximum of 10% increase in some cases. At the same time, the fusion model does not show further improvement with additional machine-annotated data in training. This effect is more evident in challenging scenarios with the human class. The reason might be

attributed to a lack of accuracy in the labels, particularly in the semi-supervised learning mode, and a scarcity of data points for smaller objects, such as a human.

In summary, through all the above scenarios, it is possible to say that domain adaptation and semi-supervised learning can lead to an average increase between 2 to 5 percentage points in vehicle segmentation. Specifically, in the average of all above scenarios, vehicle segmentation in fusion mode improves from 76% in the iseAuto baseline to 79% in the semi-supervised transfer learning mode, an increase of three percentage points. However, accurately segmenting less-represented classes with fewer points in the scenario, such as the human class, remains a challenge due to the scarcity of data and inaccurate machine labeling.

6. Conclusions

In this paper, the results of our machine learning algorithm involving LiDAR–camera fusion, transfer learning, and semi-supervised learning on our custom dataset are shown. The data used in this work are acquired using our custom autonomous shuttle, iseAuto. This work extends the results presented in a previous conference paper by giving a deep insight and analysis of the performance of our machine learning algorithm. Our algorithm’s performance is first shown on a publicly available dataset, the Waymo data, used as a benchmark to show that this algorithm is aligned with the state of the art. As the main focus of this paper is to show that it is possible to achieve reasonable performance on a custom dataset with only a limited amount of annotation, we have trained the network with little data (only 10% of Waymo), showing an already reasonable performance. The baseline was compared against a network trained on Waymo and combining iseAuto data in transfer learning, providing over 80% performance in IoU in day-fair conditions and using the fusion algorithm. In the future, this work can be extended by adding more labeled and unlabeled data to the iseAuto dataset with more diversity for different weather conditions and traffic scenarios, and including more classes. The performance of the fusion model has enormous potential to be further improved. A different line of work could be adaptation research of our algorithms for different dataset sources to improve the networks’ capability in domain adaptation and detecting more challenging object classes.

Author Contributions: Conceptualization, J.G. and M.B.; methodology, J.G. and M.B.; software, J.G., A.L. and M.B.; validation, J.G. and M.B.; formal analysis, J.G.; investigation, J.G.; resources, R.S.; data curation, J.G.; writing—original draft preparation, J.G.; writing—review and editing, J.G., M.B. and A.L.; visualization, J.G.; supervision, R.S.; project administration, R.S.; funding acquisition, R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported via funding by two grants: the European Union’s Horizon 2020 Research and Innovation Programme grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, grant No. 2014-2020.4.01.20-0289.

Data Availability Statement: The dataset acquired using our vehicle iseAuto used to generate the analysis is publicly available at <https://autolab.taltech.ee/data/> (accessed on 27 February 2022).

Acknowledgments: The financial support from the Estonian Ministry of Education and Research and the Horizon 2020 Research and Innovation Programme is gratefully acknowledged.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer, Berlin/Heidelberg, Germany, 2016; pp. 21–37.
2. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 1137–1149. [CrossRef] [PubMed]

3. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In proceedings of 2017 IEEE Transactions on Pattern Analysis and Machine Intelligence, Venice, Italy, 22–29 October 2017; pp. 386–397.
4. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. Rangenet++: Fast and accurate lidar semantic segmentation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 4213–4220.
5. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017.
6. Caltagirone, L.; Bellone, M.; Svensson, L.; Wahde, M. LIDAR-Camera fusion for road detection using fully convolutional neural networks. *Robot. Auton. Syst.* **2019**, *111*, 125–131. [[CrossRef](#)]
7. Pang, S.; Morris, D.; Radha, H. CLOCs: Camera-LiDAR object candidates fusion for 3D object detection. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2021; pp. 10386–10393.
8. Van Gansbeke, W.; Neven, D.; De Brabandere, B.; Van Gool, L. Sparse and noisy lidar completion with rgb guidance and uncertainty. In Proceedings of the 2019 16th International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 27–31 May 2019; pp. 1–6.
9. Caltagirone, L.; Bellone, M.; Svensson, L.; Wahde, M.; Sell, R. LiDAR-Camera Semi-Supervised Learning for Semantic Segmentation. *Sensors* **2021**, *21*, 4813. [[CrossRef](#)] [[PubMed](#)]
10. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
11. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in perception for autonomous driving: Waymo open dataset. In proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–18 June 2020; pp. 2443–2451.
12. Chang, M.F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. Argoverse: 3D Tracking and Forecasting with Rich Maps. In proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 8740–8749.
13. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. Nusences: A multimodal dataset for autonomous driving. In proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–18 June 2020; pp. 11621–11631.
14. Sell, R.; Leier, M.; Rassölkin, A.; Ernits, J.P. Self-driving car ISEAUTO for research and education. In Proceedings of the 2018 19th International Conference on Research and Education in Mechatronics (REM), Delft, The Netherlands, 7–8 June 2018; pp. 111–116.
15. Rassölkin, A.; Gevorkov, L.; Vaimann, T.; Kallaste, A.; Sell, R. Calculation of the traction effort of ISEAUTO self-driving vehicle. In Proceedings of the 2018 25th International Workshop on Electric Drives: Optimization in Control of Electric Drives (IWED), Moscow, Russia, 31 January–2 February 2018; pp. 1–5.
16. Sell, R.; Rassölkin, A.; Wang, R.; Otto, T. Integration of autonomous vehicles and Industry 4.0. *Proc. Est. Acad. Sci.* **2019**, *68*, 389–394. [[CrossRef](#)]
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Bellone, M.; Ismailogullari, A.; Mütür, J.; Nissin, O.; Sell, R.; Soe, R.M. Autonomous driving in the real-world: The weather challenge in the Sohjoa Baltic project. In *Towards Connected and Autonomous Vehicle Highways*; Springer; Berlin/Heidelberg, Germany, 2021; pp. 229–255.
19. Brostow, G.J.; Fauqueur, J.; Cipolla, R. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognit. Lett.* **2009**, *30*, 88–97. [[CrossRef](#)]
20. Geyer, J.; Kassahun, Y.; Mahmudi, M.; Ricou, X.; Durgesh, R.; Chung, A.S.; Hauswald, L.; Pham, V.H.; Mühlegg, M.; Dorn, S.; et al. A2d2: Audi autonomous driving dataset. *arXiv* **2020**, arXiv:2004.06320.
21. Jeong, J.; Cho, Y.; Shin, Y.S.; Roh, H.; Kim, A. Complex urban dataset with multi-level sensors from highly diverse urban environments. *Int. J. Robot. Res.* **2019**, *38*, 642–657. [[CrossRef](#)]
22. Behrendt, K.; Soussan, R. Unsupervised labeled lane markers using maps. In proceedings of 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea, 27 October–2 November, 2019; pp. 832–839.
23. Huang, X.; Wang, P.; Cheng, X.; Zhou, D.; Geng, Q.; Yang, R. The apolloscape open dataset for autonomous driving and its application. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 2702–2719. [[CrossRef](#)] [[PubMed](#)]
24. Van Engelen, J.E.; Hoos, H.H. A survey on semi-supervised learning. *Mach. Learn.* **2020**, *109*, 373–440. [[CrossRef](#)]
25. Yang, X.; Song, Z.; King, I.; Xu, Z. A survey on deep semi-supervised learning. *arXiv* **2021**, arXiv:2103.00550.
26. Miller, D.J.; Uyar, H. A mixture of experts classifier with learning based on both labelled and unlabelled data. In proceedings of the 9th International Conference on Neural Information Processing Systems, Cambridge, MA, USA, 3–5 December 1996.
27. Shahshahani, B.M.; Landgrebe, D.A. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Trans. Geosci. Remote Sens.* **1994**, *32*, 1087–1095. [[CrossRef](#)]
28. Joachims, T. Transductive inference for text classification using support vector machines. In Proceedings of the Sixteenth International Conference on Machine Learning (ICML), Bled, Slovenia, 27–30 June 1999; Volume 99, pp. 200–209.

29. Belkin, M.; Niyogi, P.; Sindhvani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.* **2006**, *7*, 2399–2434.
30. Blum, A.; Mitchell, T. Combining labeled and unlabeled data with co-training. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, Madison, WI, USA, 24–26 July 1998; pp. 92–100.
31. Agrawala, A. Learning with a probabilistic teacher. *IEEE Trans. Inf. Theory* **1970**, *16*, 373–379. [[CrossRef](#)]
32. Zhu, Y.; Zhang, Z.; Wu, C.; Zhang, Z.; He, T.; Zhang, H.; Manmatha, R.; Li, M.; Smola, A. Improving semantic segmentation via self-training. *arXiv* **2020**, arXiv:2004.14960.
33. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
34. Brostow, G.J.; Shotton, J.; Fauqueur, J.; Cipolla, R. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 44–57.
35. Xie, Q.; Luong, M.T.; Hovy, E.; Le, Q.V. Self-training with noisy student improves imagenet classification. In proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–18 June 2020; pp. 10684–10695.
36. Zhai, X.; Oliver, A.; Kolesnikov, A.; Beyer, L. S4L: Self-supervised semi-supervised learning. In proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 1476–1485.
37. Ma, F.; Cavalheiro, G.V.; Karaman, S. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3288–3295.
38. Park, K.; Kim, S.; Sohn, K. High-precision depth estimation using uncalibrated LiDAR and stereo fusion. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 321–335. [[CrossRef](#)]
39. Xu, D.; Anguelov, D.; Jain, A. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 244–253.
40. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017; pp. 77–85.
41. Liang, M.; Yang, B.; Chen, Y.; Hu, R.; Urtasun, R. Multi-task multi-sensor fusion for 3d object detection. In proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 7337–7345.
42. Su, H.; Jampani, V.; Sun, D.; Maji, S.; Kalogerakis, E.; Yang, M.H.; Kautz, J. Splatnet: Sparse lattice networks for point cloud processing. In proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2530–2539.
43. Bai, M.; Mattyus, G.; Homayounfar, N.; Wang, S.; Lakshmikanth, S.K.; Urtasun, R. Deep multi-sensor lane detection. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 3102–3109.
44. Chen, Z.; Zhang, J.; Tao, D. Progressive lidar adaptation for road detection. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 693–702. [[CrossRef](#)]
45. Gu, J.; Chhetri, T.R. Range Sensor Overview and Blind-Zone Reduction of Autonomous Vehicle Shuttles. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2021; Volume 1140, p. 012006.
46. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
47. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. Available online: <http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html> (accessed on 17 December 2021).
48. Jiang, C.; Xu, H.; Zhang, W.; Liang, X.; Li, Z. Sp-nas: Serial-to-parallel backbone search for object detection. In proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–18 June 2020; pp. 11860–11869.
49. Zhang, Y.; Song, X.; Bai, B.; Xing, T.; Liu, C.; Gao, X.; Wang, Z.; Wen, Y.; Liao, H.; Zhang, G.; et al. 2nd Place Solution for Waymo Open Dataset Challenge–Real-time 2D Object Detection. *arXiv* **2021**, arXiv:2106.08713.


Appendix 3

Article III

Junyi Gu, Artjom Lind, Tek Raj Chhetri, Mauro Bellone, and Raivo Sell. End-to-end multimodal sensor dataset collection framework for autonomous vehicles. *Sensors*, 23(15), 2023

Article

End-to-End Multimodal Sensor Dataset Collection Framework for Autonomous Vehicles

Junyi Gu ^{1,*} , Artjom Lind ² , Tek Raj Chhetri ^{3,4} , Mauro Bellone ⁵  and Raivo Sell ¹ 

- ¹ Department of Mechanical and Industrial Engineering, Tallinn University of Technology Tallinn, 12616 Tallinn, Estonia; raivo.sell@taltech.ee
 - ² Intelligent Transportation Systems Lab, Institute of Computer Science, University of Tartu, 51009 Tartu, Estonia; artjom.lind@ut.ee
 - ³ Semantic Technology Institute (STI) Innsbruck, Department of Computer Science, Universität Innsbruck, 6020 Innsbruck, Austria; Tek-Raj.Chhetri@uibk.ac.at
 - ⁴ Center for Artificial Intelligence (AI) Research Nepal, Sundarharaincha 56604, Nepal
 - ⁵ FinEst Centre for Smart Cities, Tallinn University of Technology, 19086 Tallinn, Estonia; mauro.bellone@taltech.ee
- * Correspondence: junyi.gu@taltech.ee

Abstract: Autonomous driving vehicles rely on sensors for the robust perception of their surroundings. Such vehicles are equipped with multiple perceptive sensors with a high level of redundancy to ensure safety and reliability in any driving condition. However, multi-sensor, such as camera, LiDAR, and radar systems raise requirements related to sensor calibration and synchronization, which are the fundamental blocks of any autonomous system. On the other hand, sensor fusion and integration have become important aspects of autonomous driving research and directly determine the efficiency and accuracy of advanced functions such as object detection and path planning. Classical model-based estimation and data-driven models are two mainstream approaches to achieving such integration. Most recent research is shifting to the latter, showing high robustness in real-world applications but requiring large quantities of data to be collected, synchronized, and properly categorized. However, there are two major research gaps in existing works: (i) they lack fusion (and synchronization) of multi-sensors, camera, LiDAR and radar; and (ii) generic scalable, and user-friendly end-to-end implementation. To generalize the implementation of the multi-sensor perceptive system, we introduce an end-to-end generic sensor dataset collection framework that includes both hardware deploying solutions and sensor fusion algorithms. The framework prototype integrates a diverse set of sensors, such as camera, LiDAR, and radar. Furthermore, we present a universal toolbox to calibrate and synchronize three types of sensors based on their characteristics. The framework also includes the fusion algorithms, which utilize the merits of three sensors, namely, camera, LiDAR, and radar, and fuse their sensory information in a manner that is helpful for object detection and tracking research. The generality of this framework makes it applicable in any robotic or autonomous applications and suitable for quick and large-scale practical deployment.



Citation: Gu, J.; Lind, A.; Chhetri, T.R.; Bellone, M.; Sell, R. End-to-End Multimodal Sensor Dataset Collection Framework for Autonomous Vehicles. *Sensors* **2023**, *23*, 6783. <https://doi.org/10.3390/s23156783>

Academic Editors: Arturo de la Escalera Hueso and Felipe Jiménez

Received: 18 May 2023
Revised: 13 July 2023
Accepted: 20 July 2023
Published: 29 July 2023

Keywords: multimodal sensors; autonomous driving; dataset collection framework; sensor calibration and synchronization; sensor fusion



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, technological advancements such as deep learning and the introduction of autonomous vehicles (AVs) have altered every aspect of our lives and become an integral part of our economy. According to the Boston Consulting Group, the value of the AV industry in 2035 is projected to be \$77 billion [1]. In addition, the Brookings Institution and IHS predict that by 2050, almost all users will possess AVs [2]. As AVs such as Tesla self-driving cars and AuVe Tech autonomous shuttles become more prevalent in our daily lives and an alternative to conventional vehicles, the safety and security concerns of AVs

are growing [2]. Refining the current techniques can address concerns regarding AV safety and security. For instance, by enhancing object detection, we can enhance perception and reduce the probability of accidents.

Improving AV, particularly techniques such as object detection and path planning, requires field-collected AV data because field-collected AV data provide important insights, for example, human–machine interactive situations like merges, and unprotected turns [3], which are otherwise difficult to obtain from any simulated environment.

Moreover, diverse field-collected data can help AV technology to mature faster [4]. This is also why the amount of field-collected data for AVs is growing despite the availability of simulation tools such as CARLA [5] and SUMO (Simulation of Urban Mobility) [6]. Waymo’s open motion [3] and perception [7] dataset and the nuScenes [8] dataset are two examples.

However, collecting AV field data is a complex and time-consuming task. The difficulty stems from the multi-sensory (e.g., using multiple sensors such as camera, light detection and ranging (LiDAR), and radar) nature of AV environments, which are used to overcome the limitations of individual sensors. For example, the camera input can correct the abnormalities of inertial sensors [9]. However, the challenge lies in the fact that different sensors, such as LiDAR and radar sensors, have different sensing rates and resolutions and require the fusion of multimodal sensory data [10], thereby making the task of data collection even more difficult. For example, the LiDAR sensor can capture more than a million three-dimensional (3D) points per second, while the radar sensor has poor 3D resolution [11], which needs to be synchronized before use in other AV tasks such as object detection. Moreover, the data collection task is often performed alongside other regular duties, making it even more time-consuming and prone to error, which we conclude from our experience of iseAuto dataset collection [12].

With respect to the advantages of real-world field data, studies such as those by Jacob et al. [4] (see Section 2 for more) have focused on data collection frameworks for AVs. However, the work by Jacob et al. [4] does not consider the radar sensor; therefore, extra effort is required when the data are collected from a vehicle equipped with the radar sensor. Additional limitations of the work include the multi-sensor fusion of the camera, LiDAR, and radar data to provide rich contextual information. Muller et al. [13] leverage sensor fusion to provide rich contextual information like velocity, as in our work. However, the work of Muller et al. [13] does not include the radar sensor, and it is based on the CARLA simulator; hence, its effectiveness with real-world physical AVs is still being determined. Therefore, we present our work, an end-to-end general-purpose AV data collection framework featuring algorithms for sensor calibration, information fusion, and data space to collect hours of robot-related application that can generate data-driven models. The novelty of our dataset collection framework is that it covers the aspects from sensor hardware to the developed dataset that can be easily accessed and used for other autonomous-driving-related research. We provide detailed hardware specifications and the procedures to build the data acquisition and processing systems. Our dataset collection framework has backend data processing algorithms to fuse the camera, LiDAR, and radar sensing modalities together.

In summary, the contributions of this work are given below.

- We present a general purpose scalable end-to-end AV data collection framework for collecting high-quality multi-sensor radar, LiDAR, and camera data.
- The implementation and demonstration of the framework’s prototype, whose source code is available at: https://github.com/Claud1234/distributed_sensor_data_collector (accessed on 14 May 2023).
- The dataset collection framework contains backend data processing and multimodal sensor fusion algorithms.

The remainder of the paper is as follows. Section 2 reviews the autonomous driving dataset, the existing data collection frameworks, and the mainstream multimodal sensor systems related to autonomous data acquisition. Section 3 introduces the prior and

post-processing of our dataset collection framework, including sensor calibration, synchronization, and fusion. Section 4 presents the prototype mount used for testing and demonstrating the dataset collection framework. Specifically, there are detailed descriptions of the hardware and software setups of the prototype mount and the architecture configurations of the system operating, data communication, and cloud storage. Section 5 evaluates the performance of our dataset collection framework based on the hardware of prototype we built for testing. Finally, Section 6 provides a summary and conclusion.

2. Related Work

Given the scope of this work, we present relevant studies distinguishing dataset collection frameworks for autonomous driving research from multimodal sensor systems for data acquisition. The reason is that many studies typically focus on one aspect or the other, while we intend to merge these concepts in a general-purpose framework.

2.1. Dataset Collection Framework for Autonomous Driving

Recently, data have been regarded as valuable property. For autonomous driving research, collecting enough data covering different weather and illumination conditions requires a lot of investment. Therefore, most research groups use open datasets for the experiments. For example, KITTI [14] has been one of the most successful open datasets for a long time. Because of the development of sensor technology and the increasing requirements for datasets to cover more weather and traffic conditions, the latest datasets, such as Waymo [7] and nuScenes [8], have adopted modern perceptive sensors and covered various scenarios. Other similar datasets include PandaSet [15], Pixset [16], and CAD-C [17]. Although public datasets offer researchers the convenience of obtaining data, their limitations in practical and engineering applications must be addressed. Most open datasets aim to provide well-synchronized, denoised, and ready-to-use data but are reckless in publishing the details of their hardware configurations and open sourcing the developing tools, which causes problems for other researchers to create the dataset they need. As a result, dataset collection frameworks are proposed. These frameworks focus on analyzing the feasibility of modern sensors and improving the system's versatility on different platforms. Yan et al. [18] introduced a multi-sensor platform for vehicles to perceive their surroundings. Details of all the sensors, such as brand, model, and specifications, were listed in the paper. The robot operating system (ROS) was used for calibrating the sensors. Lakshminarayana et al. [19] focused on the protocols and standards for autonomous driving datasets. The author proposed an open-source framework to regularize datasets' collection, evaluation, and maintenance, especially for their usage in deep learning. By contrast, the hardware cost was discussed in [4] as the budget is always critical for the large-scale deployment of a framework. Therefore, some researchers build the dataset pipelines by simulated vehicles and sensors to avoid the heavy investment of hardware purchase and repeated human-labor work, for example, manual object labeling. Moreover, simulation-based data generation frameworks can be used in applications that are difficult to demonstrate in the real world. For example, Beck et al. [20] developed a framework to generate camera, LiDAR, and radar data in the CARLA [21] simulator to reconstruct the autonomous-vehicles-involved accidents. Muller et al. [13] used the same CARLA platform to build a data collection framework to produce data with accurate object labels and contextual information. In summary, very few works provide a comprehensive end-to-end framework from hardware deployment to sensor calibration and synchronization, then to the backend camera-LiDAR-radar fusion that can be easily implemented into the end applications such as motion planning and object segmentation.

2.2. Multimodal Sensor System for Data Acquisition

The data acquisition of the modern autonomous and assisted driving system relies on the paradigm in which multiple sensors are equipped [22]. For autonomous vehicles, most of the onboard sensors serve the purposes of proprioception (i.e., inertia, positioning) and

exteroception (i.e., distance measurement, light density). As our work concerns only the perceptive dataset collection, the review of multimodal data acquisition systems focuses on the exteroceptive sensors system for object detection and environment perception.

From the hardware perspective, exteroceptive sensors such as camera and LiDAR, and ultrasonic sensors, have to be installed in the exterior of the vehicles as they require a clear view field and less interference. For independent autonomous driving platforms, the typical solution is to install the sensors around the vehicles separately to avoid the body frame's dramatic changes. The testing vehicle [23] has 15 sensors installed on the front, top, and rear sides to ensure the performance and appearance of the vehicles are not much affected. Other autonomous driving platforms with similar sensor layouts include [24,25]. Furthermore, shuttle-like autonomous vehicles such as Navya [26] and iseAuto [27] also adopt the same principle to fulfill the legal requirements for the real-traffic-deployed shuttle bus. In contrast, another sensor installation pattern integrates all perceptive sensors as an individual mount from the vehicle, which is often seen in the works related to dataset collection and experimental platform validation. The authors of [28,29] showcase the popular datasets in which all sensors are integrated. The experimental platforms examples that have detachable mounts onto the vehicles are given by the authors of [30,31].

The multimodal sensor systems' software mainly involves the sensors' calibration and fusion. Extrinsic and temporal calibration are two primary categories for multi-sensor systems. Extrinsic calibration concerns the transformation information between different sensor frames, and temporal calibration focuses on the synchronicity of multiple sensors operating at various frequencies and latencies. The literature on extrinsic calibration methodologies is rich. For example, An et al. [32] proposed a geometric calibration framework that combines the planar chessboard and auxiliary 2D calibration object to enhance the correspondences of 3D-2D transformation. Similarly, Domhof et al. [33] replaced the 2D auxiliary object with a metallic trihedral corner to provide strong radar reflection, which aims to reduce the calibration noise for radar sensors. In contrast to the calibration methods that employ specific targets, there are approaches dedicated to calibrating sensors without a target. Jeong et al. [34] utilized road markings to estimate sensor motions and then determined the extrinsic information of sensors. In [35], the authors trained a convolutional neural network to substitute humans to calibrate camera and radar sensors. The model automatically pairs radar point clouds with image features to estimate challenging rotational information between sensors. The studies of multimodal sensor fusion for autonomous driving perception and data acquisition were reviewed in [36,37]. Recent breakthroughs in deep learning have significantly inspired researchers to fuse the multimodal data streams in the level of feature and context [38,39]. On the other hand, neural-network-based fusion approaches require a significant amount of computing power. Remarkably, Pollach et al. [40] proposed fusing the camera and LiDAR data at a probabilistic low level; the simple mathematical computation consumes less power and causes low latency. The authors of [41] focused on the implementation feasibility of the multi-sensor fusion. Like our work, the authors developed a real-time hybrid fusion pipeline composed of a fully convolutional neural network and an extended Kalman filter to fuse the camera, LiDAR, and radar data. Cost efficiency is the crucial point in [42]; the study resulted in a method that relies on Microsoft Kinect to produce color images and 3D point clouds. However, this data acquisition and fusion system mainly works for road surface monitoring.

3. Methodology

Our dataset collection framework primarily focuses on exteroceptive sensors mainly used in robotics for perception purposes, in contrast to sensors such as GPS and wheel-encoder that record the status information of the vehicle itself. Currently, one of the primary usages of the perceptive sensor data in the autonomous driving field is the obstacle-type-objects (cars, humans, and bicycles) [43] and traffic-type-objects (traffic signs and road surface) [44] detection and segmentation. The mainstream research in this field is fusing different sensory data to compensate sensors for each other limitations. There is already

a large amount of research focusing on the fusion of camera and LiDAR sensors [45], but more attention should be given to the integration of radar data. Although LiDAR sensors outperform radar sensors from the perspective of point-cloud density and object texture, radar sensors have advantages in terms of moving object detection, speed estimation, and high reliability in harsh environments such as fog and dust. Therefore, this framework innovatively exploits the characteristics of radar sensors to highlight moving objects in LiDAR point clouds and calculate their relative velocity. The radar and LiDAR fusion result is then projected onto the camera image to achieve the final radar–LiDAR–camera fusion. Figure 1 presents the framework architecture and data flow overview. In summary, the framework is composed of three modules: sensors, processing units, and cloud server. The radar, LiDAR, and camera sensors used in the framework’s prototype are TI mmwave AWR1843BOOST, Velodyne VLP-32C, and Raspberry Pi V2, respectively. Sensor drivers are the ROS nodes and forward data to the connected computing unit. The main computer (ROS master) of the prototype is the Intel® NUC 11 with the Core™ i7-1165G7 Processor, and the supporting computer (ROS slave) is the ROCK PI N10. The ROS master and slave computers are physically connected by an Ethernet cable, and the ROS slave simply sends sensory data coming from the camera and the radar to the ROS master for post processing. The communication between the cloud server and the ROS master relies on the 4G network.

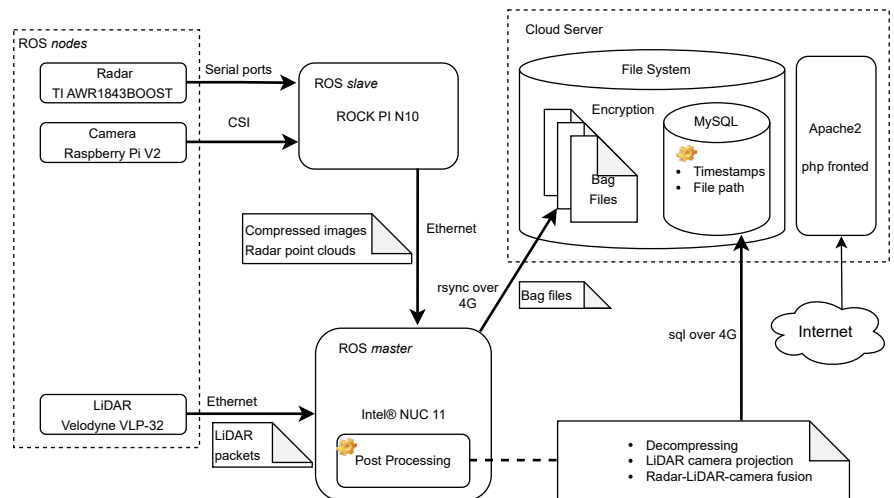


Figure 1. Overview of the framework architecture and data flow. The bold arrow-pointers denote data flow directions and corresponding protocols. Detailed descriptions of the sensor and other framework prototype hardware are in Section 4.

3.1. Sensor Calibration

For any autonomous vehicle’s perceptive system equipped with both passive (camera) and active (LiDAR, radar) sensors, referring to their capacity to measure the natural electromagnetic radiation of objects or the reflected energy emitted by the sensor. The sensor calibration is the calculation of the transformation matrices to bring all measurements in the same reference frame in order to associate different readings of the same objects coming from different sensors. A reliable calibration requires one to retrieve the intrinsic and extrinsic parameters.

3.1.1. Intrinsic Calibration

The intrinsic calibration refers to the position and orientation of the sensor in real-world coordinates by which the relative coordinate for the features is detected by the sensor. Among all popular perceptive sensors in the autonomous driving field, there is already a significant amount of work related to the intrinsic calibration of the camera and LiDAR

sensors [46,47]. LiDAR and camera are the primary sensors in this work to perceive the surrounding environment; therefore, they are the subject of intrinsic calibrations. Raspberry Pi V2 is a pinhole camera that is a well-known and widely used model [48,49]. The intrinsic calibration for the pinhole camera estimates the sensor's internal parameters, such as focal length and distortion coefficients, that comprise the camera matrix. Referring to the classification in [50], we use the photogrammetric method to calibrate the Raspberry Pi V2 camera. This method relies on planner patterns with precise geometric information in the 3D real world. For example, using a checkerboard with known square dimensions, the interior vertex points of the squares are used during the calibration. In addition, a wide-angle lens (160°) was attached to the Raspberry Pi V2 camera, resulting in significant image distortion. Therefore, rectifying the images before implementing them into any post-processing is critical. The open-source ROS 'camera_calibration' package was used in this work to calibrate the camera sensor. The 'camera_calibration' package is built upon the OpenCV camera calibration and 3D reconstruction modules. It provides the graphic interface for parameter tuning and gives the results of the distortion coefficients, camera matrix, and projection matrix. Figure 2 compares the distorted image obtained directly from the camera sensor and the processed rectified image based on the camera's intrinsic calibration results.



Figure 2. Comparing images obtained directly from the sensor to those that have been processed. (a) Raw distorted image obtained directly from the camera. (b) Rectified image.

As a highly industrialized and intact-sealed product, Velodyne VLP-32C LiDAR sensors are usually factory calibrated before shipment. Referring to the Velodyne VLP-32C's user manual, the range accuracy is claimed to be up to ± 3 cm [51]. In addition, research works such as proposed by Glennie et al. [52] and Atanacio-Jiménez et al. [53] used photogrammetry or planar structures to further calibrate the LiDAR sensors to determine the error connection. However, considering the sparsity of the LiDAR points from spatial perspective, factory calibration of the Velodyne LiDAR sensors is sufficient for most of the autonomous driving scenarios. Therefore, no extra calibration work was conducted on the LiDAR sensors in our framework.

Due to the characteristics of radar sensors in sampling frequency and spatial location, the calibration of radar sensors usually concentrates on the coordinate calibration to match the radar points and image objects [54]; points filtering to dismiss the noise and faulty detection results [55]; and error correction to compensate the mathematical errors in measurement [56]. The post-processing towards radar data in our work is overlaying radar points with the LiDAR point clouds. Therefore, the intrinsic calibration for radar sensors focuses on filtering out undesirable detection results and noise. A sophisticated method for noise and ineffective target filtering was proposed by [57], which developed intra-frame clustering and tracking algorithms to classify the valid objects signal from original radar data. The straightforward approach to calibrate the radar sensors is given in [55], which

filtered the point clouds by the speed and angular velocity information; thus, the impact of stationary objects can be reduced in radar detection results. Our work implements a similar direct method to calibrate the TI mmwave AWR1843BOOST radar sensor. The parameters and thresholds related to the resolution, velocity, and Doppler shift were fine-tuned in the environments where autonomous vehicles are operated. Most of the points for static objects were filtered out in radar data (although the noise is inevitable in detection results). As a result, there is a reduction in the number of points representing the dynamic objects in each detection frame (shown in Figure 3). This issue could be addressed by locating and clustering the objects' LiDAR points through the corresponding radar detection result. This part of the work will be detailed in Section 3.3.2.

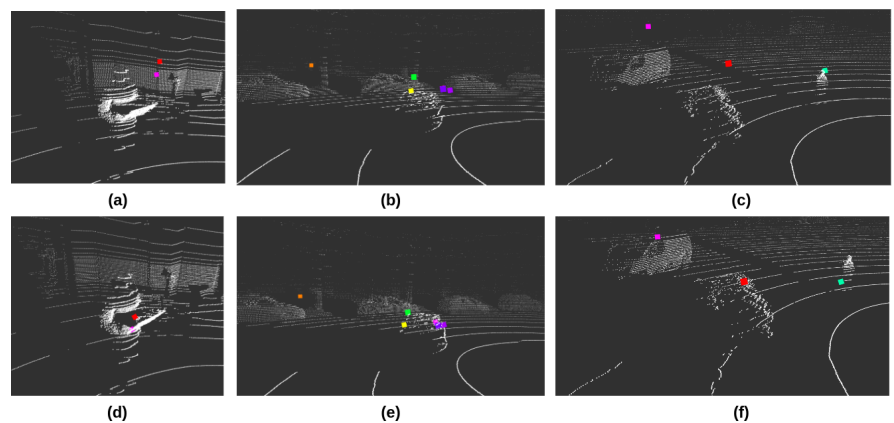


Figure 3. Performance of the LiDAR and radar extrinsic calibration in visualization. Color dots are radar points; white dots are LiDAR point clouds. The first row shows the relative locations of LiDAR and radar points without the extrinsic calibration. The second row is the results after applying the radar-LiDAR extrinsic calibration. The scene of (a,d) is indoor laboratory. (b,e) were captured in city's urban area. (c,f) are in the open area outside the city.

3.1.2. Extrinsic Calibration

For multimodal sensor systems, extrinsic calibration refers to the rigid transformation of the feature from one coordinate system to another, for example, the transformation of LiDAR points from the LiDAR coordinate frame to the camera coordinate frame. The extrinsic calibration estimates transformation parameters between the different sensor coordinates. The transformation parameters are represented as a 3×4 matrix containing the rotation (R) and translation (t) information. Extrinsic calibration is critical for sensor fusion post-processing in any multi-sensor system. One of the most important contributions of our work is the backend fusion of the camera, LiDAR, and radar sensors; thus, the extrinsic calibration was carried out between these three sensors. The principle of sensor fusion in our work is filtering out the moving objects' LiDAR points by applying the radar points, augmenting the LiDAR point data with the object's velocity readings from the radar, and then projecting the enhanced LiDAR point clouds data (that contain the location and speed information of the moving objects) onto camera images. Therefore, there is a need to extract the Euclidean transformation between the radar and LiDAR sensors and between the LiDAR and camera sensors. The standard solution is to extract the peculiar and sensitive features from the different sensors in the calibration environment. The targets used in extrinsic calibration usually have specific patterns such as planar, circular, and checkerboard for simplicity to match the features between point clouds and images.

Pairwise extrinsic calibration between the LiDAR and camera sensors in our work was inspired by the work [58]. The target for the calibration is a checkerboard with 9 and 7 squares in two directions. In practical calibration, several issues were raised and need to be noted:

- Before the extrinsic calibration, individual sensors were intrinsically calibrated and published the processed data as ROS messages. However, to have efficient and reliable data transmission and save bandwidth, ROS drivers for the LiDAR and camera sensors were programmed to publish only Velodyne packets and compressed images. Therefore, additional scripts and operations were required to handle the sensor data to match the ROS message types for the extrinsic calibration tools. Table 1 illustrates the message types of the sensors and other post-processing.
- The calibration relies on humans to match the LiDAR point and corresponding image pixel. Therefore, it is recommended to pick the noticeable features, such as the intersection of the black and white squares or the corner of the checkerboard.
- The point-pixel matches should be picked from the checkerboard in different locations covering all sensors' full field of view (FOV). For camera sensors, ensure that the pixels from the image edges were selected. Depth varieties (the distance between the checkerboard and the sensor) are critical for LiDAR sensors.
- It is a matter of fact that human errors are inevitable when pairing points and pixels. Therefore, it is suggested to select as many pairs as possible and repeat the calibration to ensure high accuracy.

Table 1. ROS message types for sensor drivers and calibration processes.

Sensor	Message Type of Topic Published by Driver	Message Type of Topic Subscribed by Calibration Processes
LiDAR Velodyne VLP-32C	velodyne_msgs/VelodyneScan	sensor_msgs/PointCloud2 (LiDAR-camera extrinsic) velodyne_msgs/VelodyneScan (radar-LiDAR extrinsic)
CameraRaspberry Pi V2	sensor_msgs/CompressedImage	sensor_msgs/Image (camera intrinsic) sensor_msgs/Image (LiDAR-camera extrinsic)
RadarTI AWR1843BOOST	sensor_msgs/PointCloud2	sensor_msgs/PointCloud2 (radar intrinsic) sensor_msgs/PointCloud2 (radar-LiDAR extrinsic)

Compared with the abundant resource for pairwise LiDAR and camera extrinsic calibration, relatively little research addressed the multimodal extrinsic calibration that includes the radar sensors. Radar sensors usually have smaller FoV than the camera and LiDAR sensors, while they also lack elevation resolution and sparse point clouds. Therefore, poor informativeness is the primary challenge for radar's extrinsic calibration. To address this problem, one of the latest references [59] proposed a two-step optimization method in which the radar data was reused in the second step to refine the extrinsic information gained from the first step calibration. However, the pursuit of our work is a universal pipeline that can be easily adapted to different autonomous platforms. Therefore, a toolbox bound with the standard ROS middleware is necessary to quickly deploy the pipeline system and execute the calibrations on autonomous vehicles. In our work, radar sensors were intrinsically calibrated to filter out most of the points for static objects. A minimum number of points were kept in each frame to represent the moving objects. An ROS-based tool was developed to compute the rotation and translation information between the LiDAR and radar coordinate frames. The calibration is based on the visualization of the Euclidean distance-based clusters of the point clouds data from two sensors. Corresponding parameters of the extrinsic calibration, such as Euler angles and displacement in X, Y, and Z directions, were manually tuned until the point cloud clusters overlapped. Please note that to properly calibrate the radar sensor, a specific calibration environment with minimal interference is required. Moreover, since the radar sensors are calibrated primarily to react

to dynamic objects, the unique object in the calibration environment should move steadily and ensure the preferable reflective capability (TI mmwave radar sensors showed higher sensitivity to metallic surfaces than others during our practical tests). Figure 3 compares the result of LiDAR and radar extrinsic calibration in visualization. Each pair of figures in the column was captured from a specific environment related to the research and real-traffic deployment of our autonomous shuttles. The first column (Figure 3a,d) shows an indoor laboratory featuring a deficient interference; the only moving object is a human with a checkerboard. The second and third columns represent the outdoor environment where the shuttles were deployed. These two pairs also represent the different traffic scenarios. The second column (Figure 3b,e) is the city's urban area, which has more vehicles and other objects (trees, street lamps, and traffic posts). The distance between the vehicles and sensors is relatively small; in this condition, radar sensors can produce more points. The third column (Figure 3c,f) is in the open area outside the city, which the vehicles run at a relatively high speed and far away from the sensors. The color dots represent the radar points, and the white dots are LiDAR point clouds data. The pictures in the first row illustrate the Euclidean distance between the LiDAR and radar point clouds before implementing the extrinsic calibration. The pictures in the second row show the results after the extrinsic calibration. By comparing the pictures in rows and columns, it is possible to see that the radar sensors produce less-noisy points data after the specific intrinsic calibration was implemented onto them. They are also more reactive to the metal surface and objects at a close distance. Moreover, after the extrinsic calibration of LiDAR and radar sensors, the alignment of the two types of sensors' point clouds data was obviously improved, which is helpful for the further processing to identify and filter out the moving objects in LiDAR sensor's point clouds data by the detection results of the radar sensor.

3.2. Sensor Synchronization

For autonomous vehicles that involve multi-sensor systems and sensor fusion applications, it is critical to address the synchronization of multiple sensors with different acquisition rates. The perceptive sensors' operating frequencies are usually limited by their own characteristics. For example, as the solid-state sensor, cameras operate at high frequencies; on the contrary, LiDAR sensors usually scan at a rate of no more than 20 Hz because of the internal rotating mechanisms. Although it is possible to set the sensors to work at the same frequencies from the hardware perspective, the latency of the sensor data streams is also a problem for matching the measurements.

In practical situations, it is not recommended to set all of the sensor frequencies identically. For example, reducing the frame rate of the camera sensors to match the frequencies of the LiDAR sensors means fewer images are produced. However, it is possible to optimize the hardware and communication setup to minimize the latency caused by the data transfer and pre-processing delays. The typical software solution to synchronize sensors matches the message headers' closest timestamps at the end-processing unit. One of the most popular open-source approaches, ROS `message_filter` [60] developed an adaptive algorithm that first finds the latest message as a reference point among the heads of all topics (a term in ROS represents the information of sensing modality). The reference point was defined as the *pivot*; based on the *pivot* and a given time threshold, messages were selected out of all topics in the queues. The whole message-pairing process was shifted along the time domain. Therefore, the messages that cannot be paired (the difference of timestamps relative to other messages exceeds the threshold) would be discarded. One of the characteristics of this adaptive algorithm is that the selection of the reference message was not fixed into one sensor modality stream (shown in Figure 4a). For the systems with multiple data streams, the number of synchronized message sets are always reconciled to the frequency of the slowest sensor.

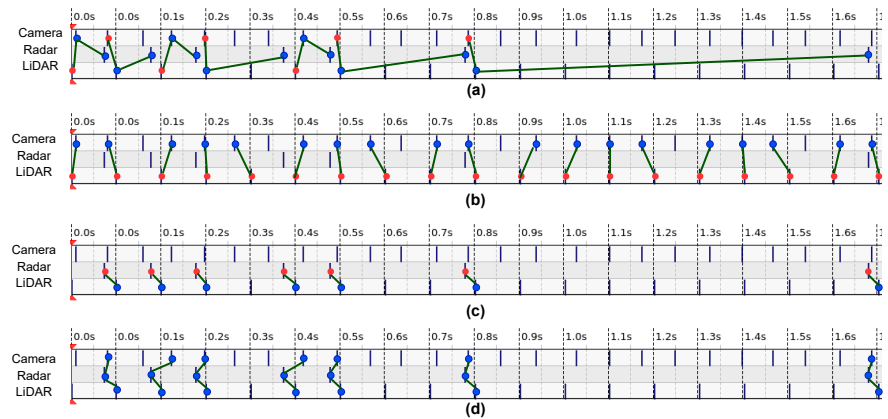


Figure 4. Illustrations of the sensor synchronization logic for `message_filter` and our algorithms. These illustrations are based on the real field data collected by our prototype. (a) shows the manner that `message_filter` carry out the multi-sensor synchronization, (b,c) show the individual LiDAR-camera and radar-LiDAR synchronization in our work, respectively. (d) is our final synchronized radar-LiDAR-camera message triplet. The messages of each sensor modality were represented by the blue points, the reference messages used for synchronization was highlighted in red. Green lines indicate the synchronized message sets. Please refer to Section 3.2 for more details.

For any multi-sensor perceptive system, the sensor synchronization principle should correspond to the hardware configuration and post-processing of the sensor fusion. As discussed in Section 4.1.2 about the sensor configurations of our work, the camera sensor has the highest rate of 15 FPS, and the LiDAR sensor operates at 10 Hz. Both camera and LiDAR sensors work at a homogeneous rate, contrary to the heterogeneous radar sensors that only produce data when moving objects are in the detection zone. Therefore, as shown in Figure 4, depending on the practical scenarios, radar data can be sparser than the camera and LiDAR data and also can scatter unevenly along the time domain. In this case, the direct implementation of the synchronization algorithm [60] will cause significant data loss of the camera and LiDAR sensors. For the generic radar-LiDAR-camera sensor fusion in our work, we divide the whole process into three modules based on the frequencies of the sensors. The first module is the fusion of the LiDAR and camera data because these two sensors have constant rates. The second module is the fusion of the radar and LiDAR sensors as they both produce the point clouds data. Finally, the last module is the fusion of the result of the second module and the camera data, achieving the thorough fusion of all three sensory modalities.

To address the issues of the hardware setup and fulfill the requirement of fusion principles in our work, we develop a specific algorithm to synchronize the data of all sensors. Inspired by the work [60], our algorithm also relies on the timestamps to synchronize the messages. Instead of the *absolute timestamp* used in [60], we used the *relative timestamp* to synchronize the message sets. The definitions of two types of timestamps are:

- *Absolute timestamp* is the time when data were produced in sensors. It was usually created by the ROS drivers of the sensors and was written in the header of each message.
- *Relative timestamp* Relative timestamp represents the time data arrive at the central processing unit. It is the Intel® NUC 11 in our prototype.

Theoretically, the *absolute timestamp* should be the basis of the sensor synchronization as it represents the exact moment in which the data was created. However, *absolute timestamp* is not always applicable and has certain drawbacks in practical scenarios. First of all, it can be effectively implemented only if all sensors are capable of assigning the timestamp to each message on the fly, which is not always possible because of the computational capacity of the hardware, and software limitations. Regarding the cost consideration, some basic

perceptive sensors are not integrated with the complex processing ability. For example, our prototype's Raspberry Pi V2 camera has no additional computing unit to capture the timestamp. However, because it is a modular Raspberry camera sensor and is directly connected with the ROCK Pi computer through the CSI socket, the *absolute timestamp* is available in the header of each image message with the assistance of the ROCK Pi computer. On the other hand, the radar sensors used in the prototype have only serial communications with the computer, and there are no *absolute timestamps* for point clouds messages.

The second requirement for implementing the *absolute timestamp* is the clock synchronization between all of the computers in the data collection framework. There are two computers in our prototype; one serves as the primary computer performing all fundamental operations, and the second is the auxiliary computer used simply for launching the sensor and forwarding data messages to the primary computer. There is a need to synchronize the clock of all computers and sensor-embedded computing units to the precision of millisecond if using the *absolute timestamps* for sensor synchronization. An important aspect to be underlined in the specific field of autonomous driving is that sensor synchronization becomes even more important as the speed of the vehicle increases, causing distortion in sensors' readings.

To simplify the deployment procedures of this data collection framework, our sensor synchronization algorithms trade off simplicity with accuracy by using the *relative timestamps*, which is the clock time of the primary computer when it receives the sensor data. Consequently, the algorithm is sensitive to the delay and bandwidth of the local area network (LAN). As mentioned in Section 4.1.1, all sensors and computers of the prototype are physically connected by internet cables and in the same Gigabyte LAN. In practical tests, before any payload was applied in the communication network, the average delay times between the primary computer and LiDAR sensor, as well as the secondary computer (camera and radar sensors), are 0.662 ms and 0.441 ms, respectively. By contrast, the corresponding delay times were 0.703 ms and 0.49 ms when data were transferred from the sensors to the primary computer. Therefore, the increasing time delay caused by transferring data in LAN is acceptable in practical scenarios. For example, the camera and LiDAR sensors' time synchronization error of the Waymo dataset is mostly bounded from -6 to 8 ms [7].

The reference frame selection is another essential issue for sensor synchronization, especially for the acquisition systems with various types of sensors. The essential difference between `message_filter` and our algorithms is that the ROS-implemented `message_filter` selects the nearest upcoming message as a reference, while our algorithms fix the reference onto the same modality stream (compare the red dot locations in Figure 4a–c). Camera and LiDAR sensors have constant frame rates, but radar sensors produce data at a variable frequency, e.g., in the presence of a dynamic object. Therefore, in this case, the single reference frame is not applicable to synchronize all of the sensors. To address this problem, we divide the synchronization process in two steps. The first step is the synchronization of the LiDAR and camera data, as shown in Figure 4b. The LiDAR sensor was chosen as the reference; thus, the frequency of the LiDAR-camera synchronized message set is the same as the LiDAR sensor's frame rate. The LiDAR-camera synchronization is continuous until the radar sensors capture the dynamic objects; in that case, the radar-LiDAR synchronization step begins, see Figure 4c. The radar sensor is the reference frame in the second synchronization step, which means that every radar message has a corresponding matched LiDAR message. As all LiDAR messages are also synchronized with the unique camera image, for every radar message, there is a thorough synchronized radar-LiDAR-camera message set (Figure 4d). The novelty of our synchronization method is separating the LiDAR and camera synchronization process from the whole procedure. As a result, we fully exploit the characteristics of density and consistency of the LiDAR and camera sensors while also keeping the possibility of synchronizing the sparse and variable information coming from radar sensors.

3.3. Sensor Fusion

Sensor fusion is critical for most autonomous-based systems as it integrates acquisition data from multiple sensors to reduce detection errors and uncertainties. Nowadays, most perceptive sensors have advantages in specific perspectives but also suffer drawbacks when working individually. For example, camera sensors may provide texture-dense information but are susceptible to changes in illumination; radar sensors can detect the reliable relative velocities of objects but struggle to produce dense point clouds; and state-of-the-art LiDAR sensors are supposed to address the limitations of camera and radar sensors but lack color and texture information. Relying on LiDAR data only makes object segmentation systems more challenging to carry out. Therefore, the common solution is combining the sensors to overcome the shortcomings of the independent sensor operation.

Camera, LiDAR, and radar sensors are considered the most popular perceptive sensors for autonomous vehicles. Presently, there are three mainstream fusion strategies: camera–LiDAR, camera–radar, and camera–LiDAR–radar. The fusion of camera and radar sensors has been widely utilized in industry. Car manufacturers combine cameras, radar, and ultrasonic sensors to perceive the vehicles' surroundings. Camera–LiDAR fusion has often been used in deep learning in recent years. The reliable X-Y-Z coordinates of LiDAR data can be projected as three-channel images. The fusion of the coordinate-projected images and the camera's RGB images can be carried out in different layers of the neural networks. Finally, the camera–LiDAR–radar fusion combines the characteristics of all three sensors to provide the excellent resolution of color and texture, precise 3D understanding of the environment, and velocity information.

In this work, we provide the radar–LiDAR–camera fusion as the backend of the dataset collection framework. Notably, we divide the whole fusion process into three steps. The first step is the fusion of the camera and LiDAR sensor because they work at constant frequencies. The second step is the fusion of the LiDAR and radar point clouds data. The last step combines the fusion result of the first two steps to achieve the complete fusion of the camera, LiDAR, and camera sensors. The advantages of our fusion approach are as follows:

- In the first step, camera–LiDAR fusion can have a maximum number of fusion results. Only a few messages were discarded during the sensor synchronization because the camera and LiDAR sensors have close and homogeneous frame rates. Therefore, the projection of the LiDAR point clouds to the camera images can be easily adapted to the input data of the neural networks.
- The second step fusion of the LiDAR and radar points grants the dataset the capability to filter out moving objects from dense LiDAR point clouds and be aware of objects' relative velocity.
- The thorough camera–LiDAR–radar fusion is the combination of the first two fusion stage results, which consume little computing power and cause minor delays.

3.3.1. LiDAR Camera Fusion

Camera sensors perceive the real world by projecting the objects onto the 2D image planes, while LiDAR point clouds data contain direct 3D geometric information. The study of [61] classified the fusion of 2D and 3D sensing modalities into three categories: high-level fusion, mid-level fusion, and low-level fusion. The high-level fusion first requires independent post-processing, such as object segmentation or tracking for each modality, then fuses the post-processing results; the low-level fusion is the integration of the basic information such as 2D/3D geometric coordinates and image pixel values in raw data, and the mid-level is an abstraction between high-level and low-level fusion, which is also known as feature-level fusion.

Our framework's low-level backend LiDAR-camera fusion focuses on the spatial coordinate matching of two sensing modalities. Instead of deep learning sensor fusion techniques, we use traditional fusion algorithms for LiDAR-camera fusion, which means the input of the fusion process is the raw data, while the output is the enhanced data [62].

One of the standard solutions for low-level LiDAR-camera fusion is converting 3D point clouds to 2D occupancy grids within the FoV of the camera sensor. There are two steps of LiDAR-camera fusion in our dataset collection framework. The first step is transforming the LiDAR data to the camera coordinate system based on the sensors' extrinsic calibration results; the process follows the equation:

$$\begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & \sin(\theta_x) \\ 0 & -\sin(\theta_x) & \cos(\theta_x) \end{bmatrix} \begin{bmatrix} \cos(\theta_y) & 0 & -\sin(\theta_y) \\ 0 & 1 & 0 \\ \sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \begin{bmatrix} \cos(\theta_z) & \sin(\theta_z) & 0 \\ -\sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} - \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix} \quad (1)$$

where a_x, a_y , and a_z are the 3D point coordinates as seen from the original frame (before the transformation); c_x, c_y , and c_z are the camera frame location coordinates; θ_x, θ_y , and θ_z are the Euler angles of the corresponding rotation of the camera frame; and d_x, d_y , and d_z are the resulting 3D point coordinates as seen from camera frame (after transformation). The following step is the projection of the 3D points to 2D image pixels as seen from the camera frame; under assumption, the camera focal length and the image resolution are known, and the following equation performs the projection:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & \frac{W}{2} \\ 0 & f_y & \frac{H}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \quad (2)$$

where d_x, d_y , and d_z are the 3D point coordinates as seen from the camera frame; f_x and f_y are camera horizontal and vertical focal length (which is known from the camera specification or discovered during the camera calibration routine); $\frac{W}{2}$ and $\frac{H}{2}$ here are the coordinates of a principal point (the image center) derived from image resolution W and H ; finally, u and v are the resulting 2D pixel coordinates. After transforming and projecting the 3D points into a 2D image, the filtering step removes all of the points that fall outside the camera view.

The fusion results of each frame are saved as two files. The first is an RGB image with projected point clouds, as shown in Figure 5a. The 2D coordinate of LiDAR points was used to pick out the corresponding pixels in the image. The assignment of the pixel color is based on the depth information of the point, and the HSV colormap was used to colorize the image. The RGB image is the visualization of the projection result, which helps evaluate the alignment of the point clouds and image pixels. The second file contains the projected 2D coordinates and X, Y, and Z axis values of the LiDAR points within the camera view. All the information was dumped as a pickle file, which can be quickly loaded and adapted to other formats, such as array and tensor. The visual demonstrations of the information in the second file are shown in Figure 5b–d, which represents the LiDAR footprint projections in XY, YZ and XZ planes, respectively. The color of pixels in each plane is proportionally scaled based on the numerical 3D axes value of the corresponding LiDAR points.

The three LiDAR footprint projections are effectively formatted by, first, projecting the LiDAR points onto the camera plane and, second, assigning the value of the LiDAR axis to a projected point. The overall algorithm can be seen in the following subsequent steps:

1. LiDAR point clouds are stored in sparse triplet format $\mathbb{L}^{3 \times N}$, where N is the number of points in LiDAR data.
2. The transformation of LiDAR point clouds to the camera reference frame occurs through the multiplication of the LiDAR matrix \mathbb{L} with the *LiDAR-to-camera* transformation matrix T_{LC} .
3. The transformed LiDAR points are projected to the camera plane, preserving the structure of the original triplet structure; in essence, the transformed LiDAR matrix L_T is multiplied by the camera projection matrix P_C ; as a result, the projected LiDAR matrix L_{pc} now contains the LiDAR point coordinates on the camera plane (pixel coordinates).
4. The camera frame width W and height H are used to cut off all the LiDAR points that fall outside the camera view. In consideration of the projected LiDAR matrix L_{pc}

from the previous step, we calculate the matrix row indices where the values satisfy the following:

- $0 \leq X_{pc} < W$
- $0 \leq Y_{pc} < H$
- $0 \leq Z_{pc}$

The row indices where L_{pc} satisfies the expressions are stored in an index array L_{idx} ; the shapes of the L_T and L_{pc} are the same, therefore it is secure to apply the derived indices L_{idx} to both the camera-frame-transformed LiDAR matrix L_T and the camera-projected matrix L_{pc} .

5. The resulting footprint images XY , YZ , and XZ are initialized following the camera frame resolution $W \times H$ and subsequently populated with black pixels (zero value).
6. Zero-value footprint images are populated as follows:
 - $XY[L_{idx}] = L[L_{idx}, 0]$
 - $YZ[L_{idx}] = L[L_{idx}, 1]$
 - $XZ[L_{idx}] = L[L_{idx}, 2]$

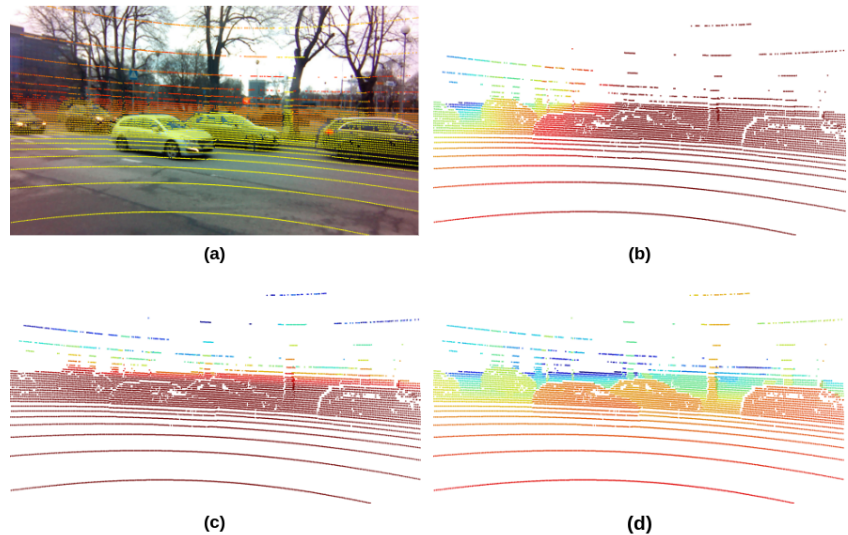


Figure 5. The projection of the LiDAR point clouds onto the camera plane in X, Y, and Z channels. (a) is RGB image, (b) is X channel projection, (c) is Y channel projection, and (d) is Z channel footprint. The color map of (a) is HSV, and (a–c) is JET.

The Algorithm 1 illustrates the procedures described above.

Algorithm 1 LiDAR transposition, projection populating the images

- 1: $L[3 \times N] \leftarrow \text{nextFrame}$
 - 2: $T_{1c} \leftarrow \text{conf}$
 - 3: $P_c \leftarrow \text{conf}$
 - 4: $L_{pr} = L * T_{1c} * P_c$
 - 5: $L_{idx} = \text{argwhere}(L_{pr} \geq \{0, 0, 0\} \ \& \ L_{pr} < \{W, H, +\infty\})$
 - 6: $XY[W \times H] \leftarrow 0$
 - 7: $YZ[W \times H] \leftarrow 0$
 - 8: $XZ[W \times H] \leftarrow 0$
 - 9: $XY[L_{idx}] = L[L_{idx}, 0]$
 - 10: $YZ[L_{idx}] = L[L_{idx}, 1]$
 - 11: $XZ[L_{idx}] = L[L_{idx}, 2]$
-

3.3.2. Radar LiDAR and Camera Fusion

This study uses millimeter wave (mmwave) radar sensors installed on the prototype mount. The motivations of equipping mmwave radar sensors on autonomous vehicles are to robustify perception against adverse weather; to prevent individual sensor failures; and, most importantly, to measure the target's relative velocity based on the Doppler effect. Currently, mmwave radar and vision fusion can be seen as a promising approach to improve object detection [63]. However, most research relies on advanced image processing methods to extract the features from the data. Therefore, an extra process is needed to process the radar points into an image-like data format. Moreover, data conversion and deep-learning-based feature extraction consume a great amount of computing power and require noise-free sensing streams. As radar and LiDAR data are both represented as 3D Cartesian coordinates, the most common solution for data fusion is simply applying a Kalman Filter [64]. Another example work [65] first converted the 3D LiDAR point clouds to virtual 2D scans and then converted the 2D radar scans to 2D obstacle maps. However, their radar sensor is the mechanical pivoting radar, which differs from our mmwave radar sensors.

In our work, the entire radar–LiDAR–camera fusion operation is divided into two steps. The first step is the fusion of radar and LiDAR sensors. The second step uses the algorithms proposed in Section 3.3.1 to fuse the first step's results and camera images. As discussed in Section 3.1, we calibrate the radar sensors primarily reactive to the dynamic objects. As a result, the principle of the radar-LiDAR fusion in our work is selecting the LiDAR point clouds of the moving objects based on the radar detection results. Figure 6 illustrates four subsequent procedures of the radar-LiDAR fusion. The first involves transforming the radar points from the radar frame coordinate to the LiDAR frame coordinate. Corresponding transformation matrices are attained from the extrinsic sensor calibration. The second involves applying the density-based spatial clustering of applications with noise (DBSCAN) algorithm to the LiDAR point clouds to cluster out the points that potentially represent the objects [66]. The third involves looking up the nearest LiDAR point clusters for the radar points that were transformed into the LiDAR frame coordinate. The fourth involves marking out the selected LiDAR point clusters in raw data (arrays contain the X, Y, and Z coordinate values) and appending the radar's velocity readings as an extra channel for selected LiDAR point clusters (or $-\infty$ in case a LiDAR point belongs to no cluster).

Figure 7 demonstrates the relative locations of the original and coordinate-transformed radar points, and the results of the radar-LiDAR fusion in our work (LiDAR point clusters of the moving objects). The reference frame for the point-cloud scattering is the one positioned at the center of the LiDAR sensor. Green dots symbolize the original radar points, whereas red dots stand for the radar points transformed to the LiDAR frame coordinate, which are the result of the first subsequent of our radar-LiDAR fusion. Blue dots are the LiDAR point of the moving objects. The selection of the LiDAR point clusters, representing the detected moving object, relies on the nearest neighbor lookup based on the Euclidean distance metric that takes coordinate-transformed radar points as the reference. Due to inherent characteristics and post-intrinsic calibration, radar sensors in our prototype only produce a handful of points for moving objects in each frame, which means the computation of the whole radar-LiDAR fusion operation is computationally efficient and can be executed on the fly.

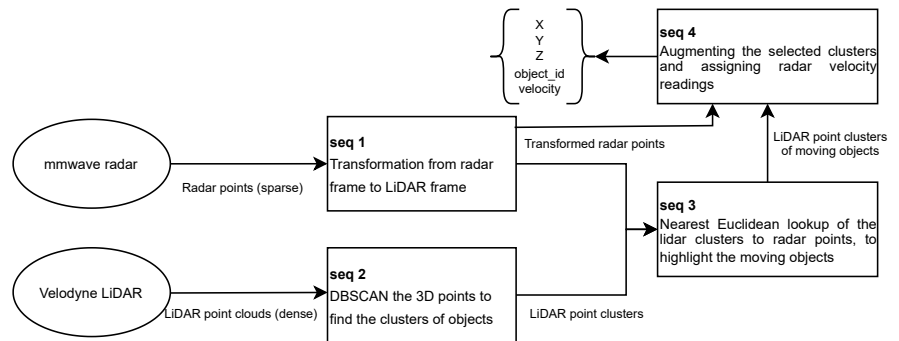


Figure 6. The workflow of radar-LiDAR fusion procedures.

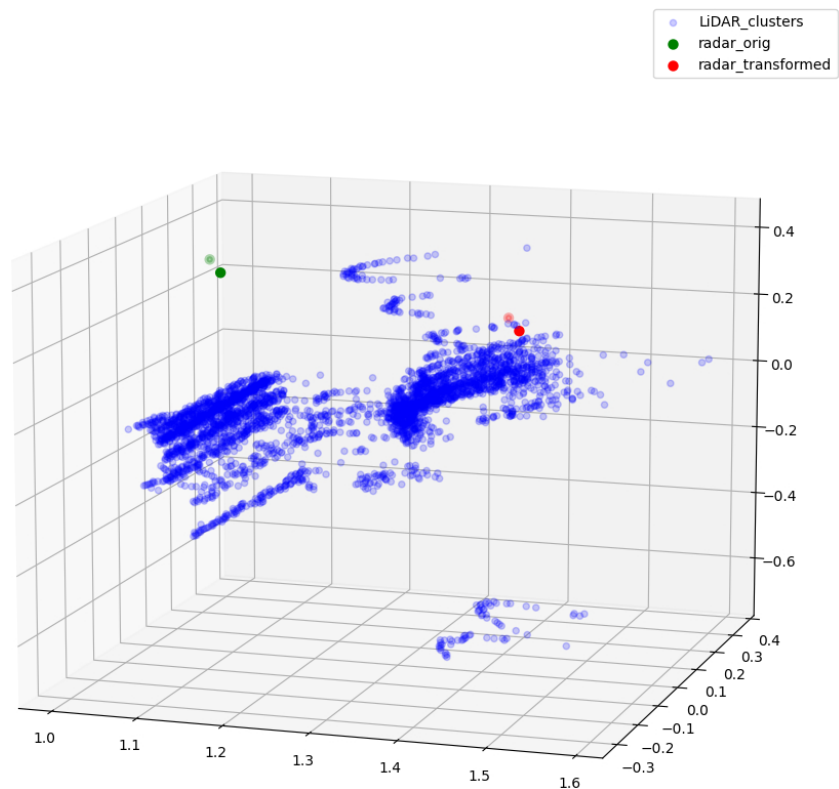


Figure 7. Relative locations of the original radar points (green), transformed radar points (red), and LiDAR point clusters of the moving object (blue). Scenario taken from a sequence similar to Figure 8.

The second step of the radar-LiDAR-camera fusion is the continuous process toward the results of the first step of radar-LiDAR fusion. The LiDAR point clusters that belong to the moving objects will be projected onto the camera plane. Figure 8a visualizes the final outcome of the radar-LiDAR-camera fusion in our dataset collection framework. LiDAR point clouds representing moving objects were filtered from the raw LiDAR data and projected onto the camera images. For each frame, moving objects' LiDAR point clusters were dumped as a pickle file containing 3D-space and 2D-projection coordinates of the points and the relative velocity information. Because of the sparsity of the radar points data, the direct projection of the radar points onto camera images has very little practical

significance (see Figure 8b). In fact, only two radar points are shown in this frame, and for this reason the significant result is the LiDAR point cluster in Figure 8a.

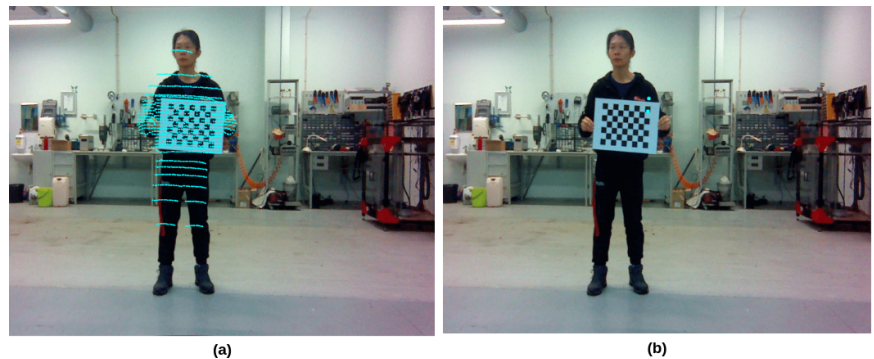


Figure 8. Illustration of the radar-LiDAR-camera. (a) Overposed LiDAR point cluster as extracted using the radar point as a reference, and (b) projection of the radar data onto the camera image.

4. Prototype Setup

This section presents our prototype for demonstrating and testing the dataset collection framework. In addition, we provide detailed introductions of the hardware installation, framework operating system, data transferring protocols, and architecture of cloud services.

4.1. Hardware Configurations

This work aims to develop a general framework for autonomous vehicles to collect sensory data when performing regular duties. In addition, process the data in formats that can be used in other autonomous-driving-related technologies, such as sensor-fusion-based object detection and real-time environment mapping. A Mitsubishi i-MiEV car was equipped with a mount on the top (shown in Figure 9b), and all the sensors were attached to the mount. To increase the hardware compatibility, two processing units were used for the prototype mount to initiate the sensors and collect data. The main processing unit, which initiates the LiDAR sensor and handles the post-processing of the data, is located inside the car. Another supporting processing unit connected to the camera and radar sensors stays on the mount (outside the car and protected by water-dust-proof shells). The dataset collection framework was operated upon by the ROS; all sensory data were captured in corresponding ROS formats.

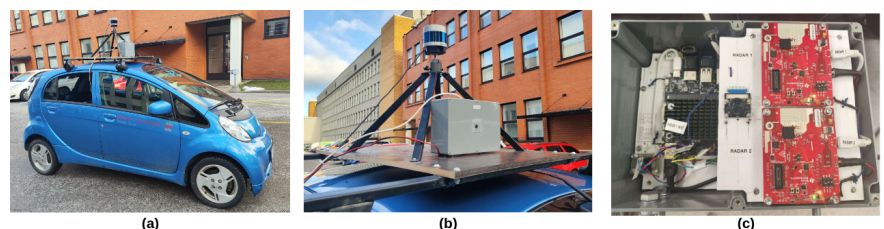


Figure 9. The prototype of the dataset collection framework. (a) is the Mitsubishi i-MiEV testing vehicle with the sensors mounted on the top. (b) shows the locations of sensors and other hardware. (c) shows the inside of the waterproof shell, which has one supporting computer, one camera, and two radar sensors.

4.1.1. Processing Unit Configurations

Three requirements have to be satisfied for the processing units and sensor components for the prototype:

- All the sensors must be modular, in a manner that they can work independently and can be easily interchanged. Therefore, there is a need for independent and modular processing units to initiate sensors and transfer the data.
- Some sensors have hardware limitations. For example, our radar sensors rely on serial ports for communication, and the cable's length affects the communication performance in practical tests. A corresponding computer for radar sensors has to stay nearby.
- The main processing unit hardware must provide enough computation resources to support complex operations such as real-time data decompression and database writing.

The main computer for the prototype is an Intel® NUC 11 with a Core™ i7-1165G7 Processor, and the supporting computer is a ROCK PI N10 with four Cortex-A53 processors. The main computer is connected to the LiDAR sensor and 4G router, subscribes to data streams of the camera and radar sensors (published by supporting processing unit), carries out the data post-processing, and then sends corresponding information to the remote database server. The supporting computer is connected to the camera and radar sensors and stays inside the water-dust-proof shell that protects other electronic devices outside the vehicle (shown in Figure 9c). The communication between the two computers relies on the LAN.

4.1.2. Sensor Installation

All the sensors installed in the prototype have been used and tested by other autonomous-driving-related projects [67,68] in the autonomous driving lab. Four perceptive sensors are installed on the prototype mount: one LiDAR, one camera, and two radars.

Currently, LiDAR and camera sensors are the mainstream in the autonomous driving field. Although it is a relatively new technology, LiDAR has become an essential sensor for many open datasets [28,69] and autonomous driving platforms [23,70]. The trend in the research community towards LiDAR sensors is using high-resolution models to produce the dense point clouds data; the maximum number of the vertical channels of the LiDAR sensors can be 128, and the range can reach 240 m. Correspondingly, dense point clouds data requires a large amount of bandwidth transference and processing power. To explicitly demonstrate our dataset collection framework and simplify the hardware implementation process, the LiDAR sensor used on the prototype is the Velodyne VLP-32C, which has 32 laser beams and vertically 40° FoV. The LiDAR sensor was connected to the main computer (NUC 11) by ethernet cable.

Camera sensors have a long developing history and are still important in modern autonomous driving technologies because of their advantages, such as reliability and cost-effectiveness. Moreover, the recent breakthrough of vision-based deep learning algorithms for object detection and segmentation has brought the researchers' focus back to the camera sensor. Therefore, it is critical for our framework to have the capability to produce and process the camera data. Since the supporting computer (Rock Pi) has the specific camera serial interface (CSI) socket, the choice of the camera sensor for the prototype mount is the Raspberry Pi V2 camera with a wide angle (160° diagonal FoV). The camera can capture 3280 × 2464 pixel static images and up to 90 Hz video mode in resolution 640 × 480.

Radar sensors have been comprehensively used on commercial cars for driving assistance. However, most of the radar-based assistant functions, such as collision warning and distance control, simply use the character of reflectivity of the radar sensors. Another iconic characteristic of the mmwave radar sensors is their capability to detect moving objects. The velocity of the moving objects can be derived based on the Doppler effect. In addition, compared with the LiDAR sensors' point clouds data that homogeneously project to all surrounding objects and whose total number of points are counted in millions, radar

sensors can only focus on moving objects and produce much more sparse point clouds data that is friendly to the data transfer and storage. As mentioned in Section 3, one of the contributions of our work is using the mmwave radar sensors to detect moving objects and enhance them in LiDAR and camera data. The testing mmwave radar sensor used for our data collection framework is Texas Instruments mmwave AWR1843BOOST with 76 to 81 GHz frequency coverage and 4 GHz available bandwidth.

Figure 9 and Table 2 show all sensors' aspects and detailed specifications. Please note that the parameters in Table 2 are the maximum values sensors can manage under the firmware and developing kit versions used in our experiments. In practical terms, the resolution and frame rate were reduced to meet the bandwidth and computation power limits. The LiDAR sensor operates at 10 Hz, and the camera runs at 15 Hz with a resolution of 1920×1080 . Moreover, the maximum unambiguous range of the radar sensor was set as 30 m, and the maximum radial velocity is 15.37 m/s. The corresponding resolution of range and radial velocity is 0.586 and 0.25 m, respectively. To address the common problems of the radar sensors, such as sparse and heterogeneous point clouds, and a high level of uncertainty and noise for moving object detection, there are two radars installed next to each other in the box, as shown in Figure 9c. Camera and radar sensors are in close proximity, so the image and points data are consistent with each other and produce accurate perceptive results. Unlike the camera and radar sensors with limited horizontal FoV, LiDAR sensors have 360° horizontal views. To fully utilize this characteristic of the LiDAR sensors, one of the most popular methods is installing multiple camera and radar sensors in all directions. For example, the acquisition system of Apolloscape [29] has up to six video cameras around the vehicle; multiple LiDAR and radar sensors were installed in pairs in [23] to cover most of the blind spots. It is a fact that the prototype mount in this work only records camera and radar data in front view. However, the scope of this work is demonstrating a generic framework for data collection and enhancement. Future work will include setting more camera–radar modules in different directions.

Table 2. Specifications of the sensors ion prototype mount.

	FoV (°)	Range (m)/Resolution	Update Rate (Hz)
Velodyne VLP-32	40 (vertical)	200	20
Raspberry Pi V2	160 (D)	3280 × 2464	90 in 640 × 480
TI mmwave AWR1843BOOST	100 (H) 40 (V)	4 cm (range resolution) 0.3 m/s (velocity resolution)	10–100

4.2. Software System

The software infrastructure of the dataset collection framework was adapted from the iseAuto, the first autonomous shuttle deployed in real-traffic scenarios in Estonia. Based on the ROS and Autoware [71], the software infrastructure of the iseAuto shuttle is a generic solution for autonomous vehicles for sensor launching, behavior making, motion planning, and artificial intelligence-related tasks. The infrastructure contains a set of modules, including human interface, process management, data logging, and transferring. Like the iseAuto shuttle, the pipeline of the dataset collection framework was operated upon the ROS and captures all the sensory data in the corresponding ROS formats. As ROS is designed with distributed computing capability, multiple computers can run the same ROS system with only one master; thus, the ROS data from different slaves is visible to the whole network. In this work, the supporting computer connected to the camera and radar sensors works as an ROS slave, and the main computer hosts the ROS master. Complete and bi-directional connectivity exists between the main and supporting computers on all ports. In addition, the Gigabyte Ethernet connection guarantees low latency to transfer the camera and radar data from the supporting computer to the main computer.

4.3. Cloud Server

In our work, the cloud server is another important component because it hosts the database module, which stores the post-processing data. The private cloud server plays a critical role in the processes of data storage and public service requests. For the iseAuto shuttle, multiple database architectures were used in the cloud server to store all kinds of data produced by the vehicle. Log data related to the low-level control system, such as braking, steering, and throttle, were stored in a PostgreSQL database. Perceptive data from the sensors were stored in a MySQL database set in parallel in the cloud server. We deploy a similar MySQL database in a remote server to store original sensory and post-processed data collected by prototype, such as camera-frame-projected and radar-enhanced LiDAR data. The database module communicates with the main computer through 4G routers. Moreover, we develop the database in a manner to be able to adapt to other autonomous platforms quickly. There is an interface that allows users to modify the database structure for different sensors and their corresponding configurations. The data that were stored in the database have the labels of the timestamps and path in file systems, which will be useful for the database query tasks. We also deploy this data collection framework onto our autonomous shuttle and publish the data collected by the shuttles when they are on real-traffic duty. The web page interface to access the data is <https://www.roboticlab.eu/fineest-mobility> (accessed on 14 May 2023).

5. Performance Evaluation

We developed this dataset collection framework primarily for the purpose of deploying on low-speed urban autonomous vehicles such as autonomous shuttles and food-delivery robots. Perceptive data were collected while autonomous vehicles were performing routine duties. Post-processing such as data decompression, sensor synchronization, and fusion were supposed to be carried out on board. Considering the computational limit of vehicles' in-built computers, it is critical to evaluate the efficiency of dataset collection framework regarding time and storage space consumption. Please note that the scope of our work is to build a generic practical solution for autonomous vehicles to collect and process perceptive data. The potential usages of the published dataset include scooter speed monitoring, and traffic-sign enhancement, which serve as transportation management for smart cities [72]. Benchmarks for other kinds of autonomous-driving-related research such as object segmentation, tracking, and path completion might benefit from the implementation of this framework, but remain out of the scope of this work.

Tables 3 and 4 evaluate the performance of this dataset collection framework in our prototype. Table 3 shows the storage occupation and time consumption of the framework's different modules to process the whole data sequence. The raw data collected from the sensors are stored as ROS bag files. There are two examples listed in this table: the first sequence is the filed-test data collected at the scene where our autonomous shuttles were deployed in Tallinn urban area. The second sequence was recorded at the indoor laboratory. The duration of our tests is 301 and 144 s, corresponding to the size of 3.7 and 0.78 GB. The output of the decompression and fusion operations in our framework are portable network graphics (PNG) images and binary pickle files for each frame, which are explained in detail in Section 3. The final output of our dataset collection framework for these two example sequences is available at https://www.roboticlab.eu/claude/fineest_framework/ (accessed on 14 May 2023). As there are two radar sensors installed in our prototype, the 'radar-LiDAR-camera Fusion' in Table 3 indicates the time consumption and data size for two radar streams. Please note that the post-processing in our framework was executed in parallel using multiple threads; therefore, the time consumption of the decompression and fusion might vary for different hardware setups and conditions. The data in Table 3 were computed by the main onboard computer of our prototype, which is Intel® NUC 11 with Core™ i7-1165G7 featuring 8 processing threads.

Table 4 shows our evaluation on the framework's per-frame performance. The first row shows the size of the RGB image and binary LiDAR points per frame. The second row

is the sum of the time consumption to produce one image, and one point-cloud binary file since the camera and LiDAR data were synchronized to the same frequency before being forwarded to the post-processing modules. The input of the ‘LiDAR Projection’ process is all of the LiDAR point clouds; therefore, this process takes the longest time compared with the other processes.

Table 3. Data size and time duration of framework’s modules to process the data sequence. The unit of the data size is gigabyte (GB), and the unit of the time is second (s).

	Sequence 1 City Urban	Sequence 2 Indoor Lab
Sequence Duration (s)	301	144
Raw Bag File Size (GB)	3.7	0.78
Synchronization (s)	4.28	1.24
Raw Data Decompressing (s)	0.36	0.09
Raw Data Writing (s)/(GB)	116.63/16.4	54.74/7.4
LiDAR-Camera Fusion (s)/(GB)	510.94/9.2	261.34/4.6
radar-LiDAR-Camera Fusion (s)/(GB)	61.97/5.8	39.38/3.3

Table 4. Data size and average time consumption of the framework’s post-processing for each frame. The output of each post-processing is an RGB image with resolution of 1920×1080 , and the binary pickle file contains the coordinates and the velocity information of the points in each corresponding frame. The unit of the data size is megabyte (MB), and the unit of the time is millisecond (ms).

	Raw Data Decompressing and Writing	LiDAR Projection	Radar-LiDAR Clustering
Size per frame			
RGB image in 1920×1080	3 MB	3 MB	3 MB
LiDAR points in binary	1.2 MB	0.9 MB	<0.1 MB
Average time per frame (RGB image in 1920×1080 + LiDAR points in binary)	79.7 ms	647.7 ms	108.44 ms

6. Conclusions

In conclusion, this study successfully presents a comprehensive end-to-end generic sensor dataset collection framework for autonomous driving vehicles. The framework includes hardware deploying solutions; sensor fusion algorithms; and a universal toolbox for calibrating and synchronizing camera, LiDAR, and radar sensors. The generality of this framework allows for its application in various robotic or autonomous systems, making it suitable for rapid, large-scale practical deployment. The promising results demonstrate the effectiveness of the proposed framework, which not only addresses the challenges of sensor calibration, synchronization, and fusion, but also paves the way for further advancements in autonomous driving research. Specifically, we showcase a streamlined and robust hardware configuration that maintains ample room for customization while preserving a generic interface for data gathering. Aiming to simplify cross-sensor data processing, we introduce a framework that efficiently handles message synchronization, and low-level data fusion. In addition, we develop a server-side platform allowing for the redundancy of connections from the recording of multiple in-field operational vehicles and the uploading of sensors data. Finally, we feature the framework with the basic web interface allowing one to overview and download the collected data (both raw and processed). Moreover, the framework has the potential for expansion through the incorporation of high-level sensor data fusion, which would enable one to track dynamic objects more effectively. This

enhancement can be achieved by integrating LiDAR-camera deep fusion techniques that not only facilitate the fusion of data from these sensors, but also tackle the calibration challenges between LiDAR and camera devices. By integrating these advanced methods, the framework can offer even more comprehensive and efficient solutions for autonomous vehicles, and other applications, requiring the robust and precise tracking of objects in their surroundings. In addition, we view comprehensive evaluations, such as the image quality assessment described by Zhai and Min [73] and the real-traffic object detection benchmark [74] of the results, as future work.

Author Contributions: Conceptualization, J.G.; Methodology, J.G. and A.L.; Software, J.G. and A.L.; Validation, J.G. and A.L.; Formal analysis, J.G.; Investigation, J.G. and T.R.C.; Resources, J.G.; Data curation, J.G. and A.L.; Writing—original draft, J.G.; Writing—review & editing, J.G., A.L., T.R.C. and M.B.; Visualization, J.G. and A.L.; Supervision, R.S.; Project administration, R.S.; Funding acquisition, R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research has received funding from two grants: the European Union’s Horizon 2020 Research and Innovation Programme, under the grant agreement No. 856602, and the European Regional Development Fund, co-funded by the Estonian Ministry of Education and Research, under grant agreement No. 2014-2020.4.01.20-0289.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code of the implementation and demonstration of our framework’s prototype is available at https://github.com/Claud1234/distributed_sensor_data_collector (accessed on 14 May 2023). The web interface to access the data collected by the framework that was deployed on real-traffic autonomous shuttle is at <https://www.roboticlab.eu/finest-mobility> (accessed on 14 May 2023). The final output of our dataset collection framework for two example sequences in Section 5 is available at https://www.roboticlab.eu/claude/finest_framework/ (accessed on 14 May 2023).

Acknowledgments: The financial support from the Estonian Ministry of Education and Research and the Horizon 2020 Research and Innovation Programme is gratefully acknowledged.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Le Mero, L.; Yi, D.; Dianati, M.; Mouzakitis, A. A Survey on Imitation Learning Techniques for End-to-End Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 14128–14147. [CrossRef]
2. Bathla, G.; Bhadane, K.; Singh, R.K.; Kumar, R.; Aluvalu, R.; Krishnamurthi, R.; Kumar, A.; Thakur, R.N.; Basheer, S. Autonomous Vehicles and Intelligent Automation: Applications, Challenges, and Opportunities. *Mob. Inf. Syst.* **2022**, *2022*, 7632892. [CrossRef]
3. Ettinger, S.; Cheng, S.; Caine, B.; Liu, C.; Zhao, H.; Pradhan, S.; Chai, Y.; Sapp, B.; Qi, C.R.; Zhou, Y.; et al. Large Scale Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 9710–9719.
4. Jacob, J.; Rabha, P. Driving data collection framework using low cost hardware. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
5. de Gelder, E.; Paardekooper, J.P.; den Camp, O.O.; Schutter, B.D. Safety assessment of automated vehicles: How to determine whether we have collected enough field data? *Traffic Inj. Prev.* **2019**, *20*, S162–S170. [CrossRef]
6. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; Wießner, E. Microscopic traffic simulation using sumo. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 2575–2582.
7. Sun, P.; Kretschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
8. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A Multimodal Dataset for Autonomous Driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.

9. Alatise, M.B.; Hancke, G.P. A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. *IEEE Access* **2020**, *8*, 39830–39846. [CrossRef]
10. Blasch, E.; Pham, T.; Chong, C.Y.; Koch, W.; Leung, H.; Braines, D.; Abdelzaher, T. Machine Learning/Artificial Intelligence for Sensor Data Fusion—Opportunities and Challenges. *IEEE Aerosp. Electron. Syst. Mag.* **2021**, *36*, 80–93. [CrossRef]
11. Wallace, A.M.; Mukherjee, S.; Toh, B.; Ahrabian, A. Combining automotive radar and LiDAR for surface detection in adverse conditions. *IET Radar Sonar Navig.* **2021**, *15*, 359–369. [CrossRef]
12. Gu, J.; Bellone, M.; Sell, R.; Lind, A. Object segmentation for autonomous driving using iseAuto data. *Electronics* **2022**, *11*, 1119. [CrossRef]
13. Muller, R.; Man, Y.; Celik, Z.B.; Li, M.; Gerdes, R. Drivetruth: Automated autonomous driving dataset generation for security applications. In Proceedings of the International Workshop on Automotive and Autonomous Vehicle Security (AutoSec), San Diego, CA, USA, 24 April 2022.
14. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
15. Xiao, P.; Shao, Z.; Hao, S.; Zhang, Z.; Chai, X.; Jiao, J.; Li, Z.; Wu, J.; Sun, K.; Jiang, K.; et al. PandaSet: Advanced Sensor Suite Dataset for Autonomous Driving. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 3095–3101. [CrossRef]
16. Déziel, J.L.; Merriaux, P.; Tremblay, F.; Lessard, D.; Plourde, D.; Stanguennec, J.; Goulet, P.; Olivier, P. PixSet: An Opportunity for 3D Computer Vision to Go beyond Point Clouds with a Full-Waveform LiDAR Dataset. *arXiv* **2021**, arXiv:2102.12010.
17. Pitropov, M.; Garcia, D.E.; Rebello, J.; Smart, M.; Wang, C.; Czarnecki, K.; Waslander, S. Canadian adverse driving conditions dataset. *Int. J. Robot. Res.* **2021**, *40*, 681–690. [CrossRef]
18. Yan, Z.; Sun, L.; Krajník, T.; Ruichek, Y. EU Long-term Dataset with Multiple Sensors for Autonomous Driving. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 10697–10704. [CrossRef]
19. Lakshminarayana, N. Large scale multimodal data capture, evaluation and maintenance framework for autonomous driving datasets. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
20. Beck, J.; Arvin, R.; Lee, S.; Khattak, A.; Chakraborty, S. Automated vehicle data pipeline for accident reconstruction: New insights from LiDAR, camera, and radar data. *Accid. Anal. Prev.* **2023**, *180*, 106923. [CrossRef]
21. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the Conference on Robot Learning, PMLR, Mountain View, CA, USA, 13–15 November 2017; pp.1–16.
22. Xiao, Y.; Codevilla, F.; Gurram, A.; Urfalioglu, O.; López, A.M. Multimodal end-to-end autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 537–547. [CrossRef]
23. Wei, J.; Snider, J.M.; Kim, J.; Dolan, J.M.; Rajkumar, R.; Litkouhi, B. Towards a viable autonomous driving research platform. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, Australia, 23–26 June 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 763–770.
24. Grisleri, P.; Fedriga, I. The brave autonomous ground vehicle platform. *IFAC Proc. Vol.* **2010**, *43*, 497–502. [CrossRef]
25. Bertozzi, M.; Bombini, L.; Broggi, A.; Buzzoni, M.; Cardarelli, E.; Cattani, S.; Cerri, P.; Coati, A.; Debattisti, S.; Falzoni, A.; et al. VIAC: An out of ordinary experiment. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 175–180. [CrossRef]
26. Self-Driving Made Real—NAVYA. Available online: <https://navya.tech/fr> (accessed on 2 May 2023).
27. Gu, J.; Chhetri, T.R. Range Sensor Overview and Blind-Zone Reduction of Autonomous Vehicle Shuttles. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1140*, 012006. [CrossRef]
28. Chang, M.F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. Argoverse: 3D Tracking and Forecasting with Rich Maps. *arXiv* **2019**, arXiv:1911.02620.
29. Wang, P.; Huang, X.; Cheng, X.; Zhou, D.; Geng, Q.; Yang, R. The apolloSCOPE open dataset for autonomous driving and its application. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *1*, 2702–2719.
30. Thrun, S.; Montemerlo, M.; Dahlkamp, H.; Stavens, D.; Aron, A.; Diebel, J.; Fong, P.; Gale, J.; Halpenny, M.; Hoffmann, G.; et al. Stanley: The robot that won the DARPA Grand Challenge. *J. Field Robot.* **2006**, *23*, 661–692. [CrossRef]
31. Zhang, J.; Singh, S. Laser-visual-inertial odometry and mapping with high robustness and low drift. *J. Field Robot.* **2018**, *35*, 1242–1264. [CrossRef]
32. An, P.; Ma, T.; Yu, K.; Fang, B.; Zhang, J.; Fu, W.; Ma, J. Geometric calibration for LiDAR-camera system fusing 3D-2D and 3D-3D point correspondences. *Opt. Express* **2020**, *28*, 2122–2141. [CrossRef]
33. Domhof, J.; Kooij, J.F.; Gavrilu, D.M. An extrinsic calibration tool for radar, camera and lidar. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 8107–8113.
34. Jeong, J.; Cho, Y.; Kim, A. The road is enough! Extrinsic calibration of non-overlapping stereo camera and LiDAR using road information. *IEEE Robot. Autom. Lett.* **2019**, *4*, 2831–2838. [CrossRef]
35. Schöller, C.; Schnettler, M.; Krämmer, A.; Hinz, G.; Bakovic, M.; Güzet, M.; Knoll, A. Targetless rotational auto-calibration of radar and camera for intelligent transportation systems. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3934–3941.

36. Huang, K.; Shi, B.; Li, X.; Li, X.; Huang, S.; Li, Y. Multi-modal sensor fusion for auto driving perception: A survey. *arXiv* **2022**, arXiv:2202.02703.
37. Cui, Y.; Chen, R.; Chu, W.; Chen, L.; Tian, D.; Li, Y.; Cao, D. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 722–739. [[CrossRef](#)]
38. Caltagirone, L.; Bellone, M.; Svensson, L.; Wahde, M. LIDAR–camera fusion for road detection using fully convolutional neural networks. *Robot. Auton. Syst.* **2019**, *111*, 125–131. [[CrossRef](#)]
39. Caltagirone, L.; Bellone, M.; Svensson, L.; Wahde, M.; Sell, R. LiDAR–camera semi-supervised learning for semantic segmentation. *Sensors* **2021**, *21*, 4813. [[CrossRef](#)]
40. Pollach, M.; Schiegg, F.; Knoll, A. Low latency and low-level sensor fusion for automotive use-cases. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 6780–6786.
41. Shahian Jahromi, B.; Tulabandhula, T.; Cetin, S. Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles. *Sensors* **2019**, *19*, 4357. [[CrossRef](#)] [[PubMed](#)]
42. Chen, Y.L.; Jahanshahi, M.R.; Manjunatha, P.; Gan, W.; Abdelbarr, M.; Masri, S.F.; Becerik-Gerber, B.; Caffrey, J.P. Inexpensive multimodal sensor fusion system for autonomous data acquisition of road surface conditions. *IEEE Sensors J.* **2016**, *16*, 7731–7743. [[CrossRef](#)]
43. Meyer, G.P.; Charland, J.; Hegde, D.; Laddha, A.; Vallespi-Gonzalez, C. Sensor fusion for joint 3d object detection and semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019.
44. Guan, H.; Yan, W.; Yu, Y.; Zhong, L.; Li, D. Robust traffic-sign detection and classification using mobile LiDAR data with digital images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2018**, *11*, 1715–1724. [[CrossRef](#)]
45. Yeong, D.J.; Velasco-Hernandez, G.; Barry, J.; Walsh, J. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. *Sensors* **2021**, *21*, 2140. [[CrossRef](#)]
46. Liu, Z.; Wu, Q.; Wu, S.; Pan, X. Flexible and accurate camera calibration using grid spherical images. *Opt. Express* **2017**, *25*, 15269–15285. [[CrossRef](#)]
47. Vel’as, M.; Španěl, M.; Materna, Z.; Herout, A. Calibration of RGB Camera with Velodyne Lidar. In Proceedings of the 22nd International Conference in Central European Computer Graphics, Visualization and Computer Vision in Co-Operation with EUROGRAPHICS Association, Plzen, Czech Republic, 2–5 June 2014; pp. 135–144.
48. Pannu, G.S.; Ansari, M.D.; Gupta, P. Design and implementation of autonomous car using Raspberry Pi. *Int. J. Comput. Appl.* **2015**, *113*, 22–29.
49. Jain, A.K. Working model of self-driving car using convolutional neural network, Raspberry Pi and Arduino. In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1630–1635.
50. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [[CrossRef](#)]
51. Velodyne-VLP32C Datasheet. Available online: https://www.mapix.com/wp-content/uploads/2018/07/63-9378_Rev-D_ULTRA-Puck_VLP-32C_Datasheet_Web.pdf (accessed on 6 June 2023).
52. Glennie, C.; Lichti, D.D. Static calibration and analysis of the Velodyne HDL-64E S2 for high accuracy mobile scanning. *Remote Sens.* **2010**, *2*, 1610–1624. [[CrossRef](#)]
53. Atanacio-Jiménez, G.; González-Barbosa, J.J.; Hurtado-Ramos, J.B.; Ornelas-Rodríguez, F.J.; Jiménez-Hernández, H.; García-Ramírez, T.; González-Barbosa, R. Lidar velodyne hdl-64e calibration using pattern planes. *Int. J. Adv. Robot. Syst.* **2011**, *8*, 59. [[CrossRef](#)]
54. Milch, S.; Behrens, M. Pedestrian detection with radar and computer vision. In Proceedings of the PAL 2001—Progress in Automobile Lighting, Laboratory of Lighting Technology, 25–26 September 2001; Herbert utzverlag GMBH: Munchen, Germany, 2001; Volume 9.
55. Huang, W.; Zhang, Z.; Li, W.; Tian, J. Moving object tracking based on millimeter-wave radar and vision sensor. *J. Appl. Sci. Eng.* **2018**, *21*, 609–614.
56. Liu, F.; Sparbert, J.; Stiller, C. IMMPDA vehicle tracking system using asynchronous sensor fusion of radar and vision. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 168–173.
57. Guo, X.-p.; Du, J.-s.; Gao, J.; Wang, W. Pedestrian detection based on fusion of millimeter wave radar and vision. In Proceedings of the 2018 International Conference on Artificial Intelligence and Pattern Recognition, Beijing, China, 18–20 August 2018; pp. 38–42.
58. Yin, L.; Luo, B.; Wang, W.; Yu, H.; Wang, C.; Li, C. CoMask: Corresponding Mask-Based End-to-End Extrinsic Calibration of the Camera and LiDAR. *Remote Sens.* **2020**, *12*, 1925. [[CrossRef](#)]
59. Peršić, J.; Marković, I.; Petrović, I. Extrinsic 6dof calibration of a radar–lidar–camera system enhanced by radar cross section estimates evaluation. *Robot. Auton. Syst.* **2019**, *114*, 217–230. [[CrossRef](#)]
60. Message_Filters—ROS Wiki. Available online: https://wiki.ros.org/message_filters (accessed on 7 March 2023).
61. Banerjee, K.; Notz, D.; Windelen, J.; Gavarraja, S.; He, M. Online camera lidar fusion and object detection on hybrid data for autonomous driving. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1632–1638.

62. Fayyad, J.; Jaradat, M.A.; Gruyer, D.; Najjaran, H. Deep learning sensor fusion for autonomous vehicle perception and localization: A review. *Sensors* **2020**, *20*, 4220. [[CrossRef](#)]
63. Wei, Z.; Zhang, F.; Chang, S.; Liu, Y.; Wu, H.; Feng, Z. Mmwave radar and vision fusion for object detection in autonomous driving: A review. *Sensors* **2022**, *22*, 2542. [[CrossRef](#)]
64. Hajri, H.; Rahal, M.C. Real time lidar and radar high-level fusion for obstacle detection and tracking with evaluation on a ground truth. *arXiv* **2018**, arXiv:1807.11264.
65. Fritsche, P.; Zeise, B.; Hemme, P.; Wagner, B. Fusion of radar, LiDAR and thermal information for hazard detection in low visibility environments. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 96–101.
66. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
67. Pikner, H.; Karjust, K. Multi-layer cyber-physical low-level control solution for mobile robots. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1140*, 012048. [[CrossRef](#)]
68. Sell, R.; Leier, M.; Rassölkin, A.; Ernits, J.P. Self-driving car ISEAUTO for research and education. In Proceedings of the 2018 19th International Conference on Research and Education in Mechatronics (REM), Delft, The Netherlands, 7–8 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 111–116.
69. Geyer, J.; Kassahun, Y.; Mahmudi, M.; Ricou, X.; Durgesh, R.; Chung, A.S.; Hauswald, L.; Pham, V.H.; Mühlegg, M.; Dorn, S.; et al. A2d2: Audi autonomous driving dataset. *arXiv* **2020**, arXiv:2004.06320.
70. Broggi, A.; Buzzoni, M.; Debattisti, S.; Grisleri, P.; Laghi, M.C.; Medici, P.; Versari, P. Extensive tests of autonomous driving technologies. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1403–1415. [[CrossRef](#)]
71. Kato, S.; Tokunaga, S.; Maruyama, Y.; Maeda, S.; Hirabayashi, M.; Kitsukawa, Y.; Monrroy, A.; Ando, T.; Fujii, Y.; Azumi, T. Autoware on board: Enabling autonomous vehicles with embedded systems. In Proceedings of the 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), Porto, Portugal, 11–13 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 287–296.
72. A Conceptual Ecosystem Solution to Transport System Management. Available online: <https://www.finestcentre.eu/mobility> (accessed on 23 June 2023).
73. Zhai, G.; Min, X. Perceptual image quality assessment: A survey. *Sci. China Inf. Sci.* **2020**, *63*, 211301. [[CrossRef](#)]
74. Fritsch, J.; Kühnl, T.; Geiger, A. A new performance measure and evaluation benchmark for road detection algorithms. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), Hague, The Netherlands, 6–9 October 2013; pp. 1693–1700. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Appendix 4

Article IV

Junyi Gu, Mauro Bellone, Tomáš Pivoňka, and Raivo Sell. CLFT: Camera-LiDAR Fusion Transformer for Semantic Segmentation in Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, 2024

CLFT: Camera-LiDAR Fusion Transformer for Semantic Segmentation in Autonomous Driving

Junyi Gu , Mauro Bellone , Tomáš Pivoňka , and Raivo Sell 

Abstract—Critical research about camera-and-LiDAR-based semantic object segmentation for autonomous driving significantly benefited from the recent development of deep learning. Specifically, the vision transformer is the novel ground-breaker that successfully brought the multi-head-attention mechanism to computer vision applications. Therefore, we propose a vision-transformer-based network to carry out camera-LiDAR fusion for semantic segmentation applied to autonomous driving. Our proposal uses the novel progressive-assemble strategy of vision transformers on a double-direction network and then integrates the results in a cross-fusion strategy over the transformer decoder layers. Unlike other works in the literature, our camera-LiDAR fusion transformers have been evaluated in challenging conditions like rain and low illumination, showing robust performance. The paper reports the segmentation results over the vehicle and human classes in different modalities: camera-only, LiDAR-only, and camera-LiDAR fusion. We perform coherent controlled benchmark experiments of the camera-LiDAR fusion transformer (CLFT) against other networks that are also designed for semantic segmentation. The experiments aim to evaluate the performance of CLFT independently from two perspectives: multimodal sensor fusion and backbone architectures. The quantitative assessments show our CLFT networks yield an improvement of up to 10% for challenging dark-wet conditions when comparing with Fully-Convolutional-Neural-Network-based (FCN) camera-LiDAR fusion neural network. Contrasting to the network with transformer backbone but using single modality input, the all-around improvement is 5-10%.

Our full code is available online for an interactive demonstration and application¹.

Index Terms—Camera-LiDAR fusion, Transformer, Semantic Segmentation, Autonomous driving.

I. INTRODUCTION

Semantic segmentation of the surrounding environment is a challenging topic in autonomous driving and plays a critical role in various intelligent-vehicle-related research-tasks such as maneuvering, path planning [1] [2], and scene understanding [3]. The field of semantic segmentation has greatly advanced due to the evolution of deep neural networks, particularly Convolutional Neural Networks (CNN), along with the availability of open datasets. Early studies [4] took camera RGB images as input and tested them with

datasets that had relatively monotonous scenarios [5]. In recent years, the blooming of perceptive sensor industries and strict safety requirements motivated semantic segmentation research related to different sensors and comprehensive scenarios. LiDAR sensors are involved the most in all kinds of research. Examples of the popular LiDAR-only methods include VoxNet [6], PointNet [7], and RotationNet [8]. However, multimodal sensor fusion is perceived as a promising technique to solve the problem of autonomous driving and has become the mainstream option for semantic segmentation [9].

As an applied research, the advancement of semantic segmentation is driven by the proposals of neural network backbones. One of the most popular neural networks recently proposed is the transformer [10], which implemented the multi-head attention mechanism [11] into the Natural Language Processing (NLP) application. The proposal of the Vision Transformer (ViT) [12] inspired researchers to explore its potential in environment perception for autonomous driving. In this work, we introduce the camera-LiDAR fusion transformer (CLFT). CLFT maintains the generic encoder-decoder architecture of a transformer-based network but uses a novel progressive-assemble strategy of vision transformers on a double-direction network. The results of the two network directions are then integrated using a cross-fusion strategy over the transformer decoder layers.

The CLFT aims to address the following issues that are challenging and less explored in the autonomous driving community.

(i) **Unbalanced sample distribution.** In real-traffic scenarios, dealing with an unbalanced sample distribution poses a significant challenge for autonomous vehicles. For instance, while vehicle lanes consistently have more cars than humans (primarily encountered at crossings or sidewalks), achieving precise perception of human entities remains paramount for the optimal functioning of any autonomous vehicle. Our previous camera-LiDAR FCN-based fusion model (CLFCN) [13] achieved more than 90% accuracy in vehicle classification. However, its accuracy in the human class is limited, reaching only 50%. Due to the under-representation of the human class in the dataset, CNNs face challenges in effectively learning knowledge during explicit down-sampling processes. In contrast, vision transformers maintain a consistent resolution for representations across all stages. Furthermore, their incorporation of a multi-head self-attention mechanism inherently provides an advantage in handling global context, making them more adept at addressing challenges associated with imbalanced class distributions.

(ii) **The consistency of multimodal input data formats.**

Corresponding Author: Mauro Bellone
J. Gu (junyi.gu@taltech.ee) and R. Sell (raivo.sell@taltech.ee) are with the Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Estonia.

M. Bellone (mauro.bellone@taltech.ee) is with FinEst Centre for Smart Cities, Tallinn University of Technology, Estonia.

Tomáš Pivoňka (tomas.pivonka@cvut.cz) is with Czech Institute of Informatics, Robotics, and Cybernetics and Department of Cybernetics, Czech Technical University in Prague, Czech Republic.

¹<https://github.com/Claud1234/CLFT>

LiDAR sensors have attracted broad interest from autonomous driving community and there are different strategies to process the LiDAR's point clouds data [14]. Unlike previous works in this field that integrate a voxel view of the LiDAR with the camera view [15] [16], our work uses the strategy to project the LiDAR point clouds along XY , YZ , and XZ plane views; thus, the camera and LiDAR inputs are amalgamated into a unified data representation for subsequent operations, encompassing feature extraction, assembly, and fusion. Although our CLFT models require the pre-processing of LiDAR point clouds such as calibration, filtering, and projection, we have verified that it is possible to carry out all these operations on the fly based on the current hardware specifications on autonomous vehicles [17] without significant overhead. Together with the inference time analysis in Section V, it is possible to claim the practical potential applicability of our models.

The niche of our work compared to other state-of-the-art transformer-based multimodal fusion techniques is detailed in Section II. The contribution of this work can be summarily outlined as follows:

- We introduce a new network architecture named CLFT, employing an innovative progressive-assemble strategy of vision transformers within a double-direction network.
- To the best of our knowledge [18] [19], CLFT is the first open-source transformer-based network that directly uses camera and LiDAR sensory input for object semantic segmentation tasks.
- We divide datasets based on illumination and weather conditions. This approach allows us to compare and highlight the robustness and efficacy of different models in challenging real-world situations.
- We prove the advancement and prospect of multimodal transformer-based models in the autonomous driving perception field, especially the segmentation of under-represented traffic objects.

The remainder of the paper is as follows. Section II reviews the state-of-the-art literature on camera-LiDAR deep fusion and transformer usage in autonomous driving. We analyze the gap in current research and explain how our work contributes to the field. Section III introduces the CLFT architecture details. Section IV presents the pre-processing and configurations of the dataset we used in this work. Section V reports the experiment results and discussion. Finally, a conclusion is conducted in Section VI.

II. RELATED WORK

Given the scope of this work, we revisit relevant literature on two aspects of semantic object segmentation for autonomous driving. The first part reviews the popular camera-LiDAR fusion-based deep learning proposals. The second part presents the recent usage of transformers in autonomous driving research.

A. Camera-LiDAR fusion-based deep learning

The fusion of camera and LiDAR data stands out as one of the extensively investigated topics in multimodal fusion,

particularly in the context of traffic object detection and segmentation. Various taxonomies are employed to categorize deep fusion algorithms that integrate camera and LiDAR information. To distinguish different fusion principles we adopt the patterns suggested in [9], namely *signal-level*, *feature-level*, *result-level*, and *multi-level* fusion. This systematic categorization aids in better understanding and comparing the diverse approaches employed in the fusion of camera and LiDAR data for enhanced performance in traffic-related applications.

(i) The *signal-level* fusion is expressed as early-stage fusion as it relies on spatial coordinate matching and raw data (e.g. 2D/3D geometric coordinates, image pixel values) integration to achieve the fusion of two sensing modalities. Depth completion [20] [21] is an iconic application which is instinctively suitable for *signal-level* fusion. Work [22] [23], and [24] explored the possibility of using *signal-level* fusion in road/lane detection scenarios and its performance-computation trade-off. There are relatively few works that implement *signal-level* fusion for traffic object detection and segmentation [25] [26] because texture information loss is inevitable in sparse mapping and projection process.

(ii) On the other hand, the literature of *feature-level* fusion is rich. In general, the LiDAR data is involved in fusion as either a voxel grid or 2D projection, and the feature map is the most common format for image input. VoxelNet [27] is the leading work to sample raw point clouds as sparse voxels before the fusion with camera data. The examples of the fusion of LiDAR's 2D projections and camera images are [28] [29] [30].

(iii) The intuition of *result-level* fusion is using the weight-based logical operations to combine the prediction results from different modalities, which is adopted in work [31] [32].

(iv) The *multi-level* fusion combines the other three fusion approaches mentioned above to overcome the shortcomings of the respective method. Van Gansbeke et al. [33] combined *signal-level* and *feature-level* fusion in a network for depth prediction. PointFusion [34] explored the *result-level* and *feature-level* fusion combination by first generating 2D bounding boxes, then filtering the LiDAR points based on these 2D boxes, at last, using a ResNet [35] and PointNet [7] network to integrate image and point clouds features to 3D object predictions. Other *multi-level* fusion research includes [36] [37].

During the literature review, we observe that the transition from *signal/result-level* to *multi-level* fusion is the general trend of camera-LiDAR deep fusion. To mitigate some limitations such as computational complexity, early works usually extract geometric information directly from LiDAR data to leverage the existing ready-to-use image processing networks. The recent research tends to carry out the fusion in a *multi-level* format, that adopts various fusion strategies and context encoding processes. Our work contributes in the line of a *multi-level* fusion architecture which uses a transformer head to encode the input and then execute the cross-fusion of camera and LiDAR data.

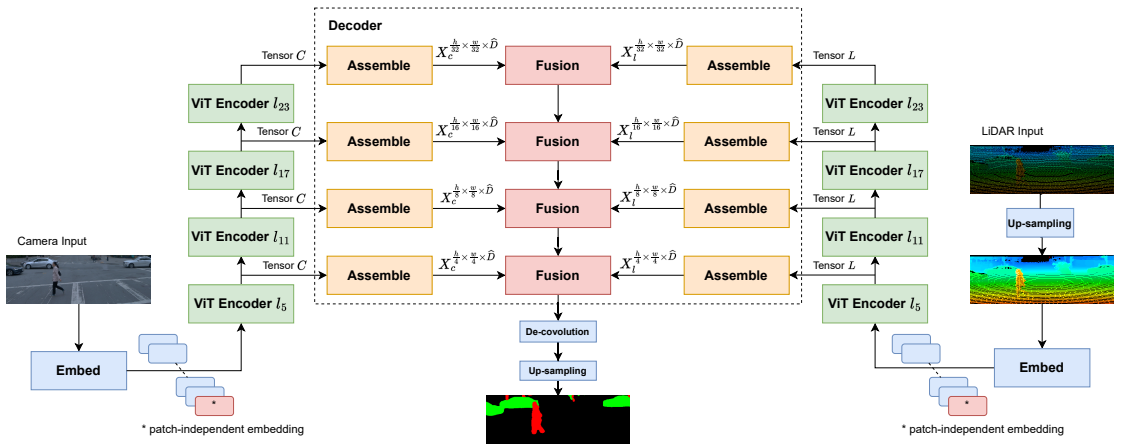


Fig. 1. The overall architecture of our double-direction network shows camera data flowing from the left side into the ViT encoder, while LiDAR data flows from the right. The camera input is individual RGB channels, and the LiDAR input stands as XY, YZ, and XZ projection planes. The cross-fusion strategy is shown in the center and highlighted using a dashed rectangle.

B. Transformers in autonomous driving research

The attention mechanism [11] has garnered significant attention from researchers across diverse fields since its introduction by Vaswani et al. in the transformer architecture for natural language processing (NLP) tasks [10]. Among the most notable transformer variants is the Vision Transformer (ViT) [12], showcasing its capabilities in computer vision with direct applications in autonomous driving. Specifically, the autonomous driving perception tasks benefit the most from the attention-mechanism's strengths in global context and long-range dependencies handling. In this section, we review the state-of-the-art transformer-based works for 2D and 3D general perception in autonomous driving.

The 2D perception applications of autonomous driving extract the information from camera images. Lane detection is the most prevalent task among 2D perception research. Peng et al. [38] proposed a bird's eye view transformer-based architecture for road surface segmentation. Work [39] adopted a lightweight transformer structure for lane shape prediction, first modeled lane markings as regressive polynomials, then optimized the polynomial parameters by a transformer query and Hungarian fitting loss algorithms. Other transformer deep networks for road/lane segmentation include [15] [40]. There are relatively fewer works of 2D segmentation because the multimodal fusion is the trend for semantic segmentation in recent. Panoptic SegFormer [41] proposed a panoptic segmentation framework utilizing a supervised mask decoder and a query decoupling method to execute the semantic and instance segmentation.

The research of transformer-based 3D object detection and segmentation is abundant. DETR3D [42] is a variant of the popular DETR [43] model but extended its 2D object detection potential to 3D detection scenarios. DETR3D relied on multi-view images to recover 3D information and used backward geometric projection to combine 2D feature extraction and

3D prediction. FUTR3D [44] is a counterpart network to DETR3D, featuring a modality-agnostic feature sampler designed to accommodate multimodal sensory input for precise 3D bounding box predictions. PETR [45] embedded 3D coordinate information into image to produce 3D position-aware features. BEVFormer [46] employed spatial and temporal attention layers for bird's eye view features to improve the performance of 3D object detection and map segmentation. Work [47] and [48] focused on the 3D segmentation. TPVFormer [47] reduced the computational requirement by transforming the volume to three bird's eye view planes. VoxFormer [48] generated 3D voxels from 2D images, then performed cross and self attention mechanisms to 3D voxel queries to compute semantic segmentation results.

With reference to our review, there are relatively few research works on the semantic object segmentation, let alone the multimodal fusion of camera and LiDAR sensors. Work [44] and [16] directly used LiDAR input, but their focus are 3D detection and occupancy prediction. Moreover, other latest works [47] and [48] produced the voxel and pseudo-point-clouds from the camera input, then carried out the semantic occupancy prediction. While our CLFT models directly take LiDAR data as input, and adopt another strategy to process the LiDAR point clouds as image views in camera plane to achieve 2D semantic object segmentation. Foremost, our work plays a crucial role in bridging the gap in multimodal semantic object segmentation within the realm of autonomous driving research.

III. METHODOLOGY

There are two aims of our CLFT models in this work; first is to outperform the existing state-of-the-art single modality transformer-based models; second is to compete with the recent CNN-based models in terms of traffic object segmentation by fusing the camera and LiDAR data. We maintain the overall structure of the transformer network for dense

prediction (DPT) [49] but invoke a late fusion strategy in its convolutional decoder, which first assemble the LiDAR and camera data in parallel and then integrate their feature map representations. We explore the capability of transformer-based networks in semantic segmentation with the advantages of LiDAR sensors, prove transformer networks' potential to classify the less represented samples in contrast with CNNs, at last, provide a late fusion strategy for transformer-related sensor fusion research.

The encoder-decoder structure has been widely implemented in image analysis transformers. We closely follow the protocol of ViT [12] to establish the encoders in our network to create the multi-layer perceptron (MLP) heads for camera and LiDAR data separately. For the decoders, we refer, but leverage proposals in work [49] to assemble and integrate the feature representations from camera and LiDAR sensors to create the object segmentation that is more precise than single modality. Figure 1 shows the overall architecture of our network.

1) *Encoder*: ViT innovatively proposed an encoder to convert an image into multiple tokens that can be treated in the same way as words in a sentence; consequently, transferred the standard transformer from NLP to computer vision applications. The ViT encoder uses two different procedures to transfer the images into tokens. The first approach divides an image into fixed-size non-overlapping patches, followed by linear projection of their flattened vector representations. The second approach extracts feature patches from a CNN feature map and then feeds them into the transformer as tokens. We retain the ViT's conventions to define the encoder variants in our work, namely, 'CLFT-base', 'CLFT-large', 'CLFT-huge', and 'CLFT-hybrid'. The 'base', 'large', and 'huge' indicate the encoder's configuration such as layer, size, and amount of parameters. The 'hybrid' means other neural network backbones are integrated in the model. The 'CLFT-base', 'CLFT-large', and 'CLFT-huge' architectures use patch-based embedding methods, have 12, 24, and 32 transformer layers, and the feature dimension D of each token are 768, 1024, and 1280, respectively. The 'CLFT-hybrid' encoder employs a ResNet50 network to extract pixel features as image embeddings, followed by 12 transformer layers. The patch size p of all our experiments is 16. The resolution of the input camera and LiDAR image (h, w) is $(384, 384)$, which means the total amount of pixels for each patch $\frac{h \times w}{p^2} = 576$ is smaller than feature dimensions D of all variants; thus, the knowledge can be retrieved from input in pixel-wise. For the 'CLFT-hybrid' encoder, it extracts the features from the input patch of $384 \div 16 = 24$ resolution. All the encoders are pretrained using ImageNet [50]. Following work in ViT, we concatenate position embeddings with image embeddings to retain positional information. Moreover, there is an individual learnable token in sequence for classification purposes. This classification token is represented as red block with the asterisk in Figure 1. It is similar to BERT's 'class' token [51], independent from all image patches and positionally embedded. Please refer to the original work [12] for the details of these encoder architectures.

2) *Decoder*: The transformer networks designed for computer vision usually modify the decoder by implementing convolutional layers at different stages. Ranftl et al. [49] proposed a transformer network for dense prediction (DPT) that progressively assembles tokens from various encoder layers into image-like representations to achieve final dense prediction. Inspired by DPT's decoder architecture, we construct a decoder to process the LiDAR and camera tokens in parallel.

As illustrated in Figure 1, we pick four transformer encoder layers denoted as t ($t = \{2, 5, 8, 11\}$ for 'CLFT-base' and 'CLFT-hybrid', $t = \{5, 11, 17, 23\}$ for 'CLFT-large'), then assemble the tokens from each layer to an image-like representation of feature maps. The feature map representations at the initial layers of the network are up-sampled to a high resolution, whereas representations from deep layers were down-sampled to a low resolution. The resolutions are anchored to input image size (h, w) , and the sampling coefficients corresponding to encoder layers t are $s = \{4, 8, 16, 32\}$. In detail, there are two steps in the assembly process. As illustrated in Algorithm 1, the first step replicates and concatenates the patch-independent 'classification token' with all other tokens individually, then forwards the concatenated representations to an MLP process with GELU non-linear activation [52]. The number of individual tokens is denoted as k .

Algorithm 1 The projection of the 'classification token'.

Input: Input tensor T , representing either the camera or LiDAR channels containing the 'classification token' and patch tokens.

Output: Concatenated tensor representations X_T

- 1: $T_{cls} = \text{replicate}\{T[:, 0]\}$
 - 2: $T_{concat} = T[:, i] \parallel T_{cls} \quad \forall i = 1, \dots, k$
 - 3: $X_T = \text{GELU}(W \cdot T_{concat} + b)$
-

Equation 1 shows the second step, which first concatenates the tokens from the first step based on their initial positional order to yield an image-like representation, then passes this representation to two convolution operations. The first convolution projects the representation from dimension D to \hat{D} (\hat{D} is set as 256 in our experiments). The second convolution applies up-sampling and down-sampling toward representation concerning the different layers of transformer encoders. X_c and X_l are the concatenated camera and LiDAR representations, N represents the total amount of patches. The generic workflows of these two steps are shown in Figure 2.

$$X_t^{N \times D} \Rightarrow X_t^{\frac{h}{s} \times \frac{w}{s} \times \hat{D}} \quad (1)$$

$$X_t = \{X_c, X_l\} \quad s = \{4, 8, 16, 32\}$$

$$t = \{2, 5, 8, 11\} \text{ or } \{5, 11, 17, 23\}$$

The last process of our decoder is the cross-fusion of camera and LiDAR feature maps, which is progressively illustrated in Figure 3. We refer to the feature fusion strategy from RefineNet [53] that forwards the camera and LiDAR representations through two residual convolution units (RCU) in sequence. The camera and LiDAR's representations are

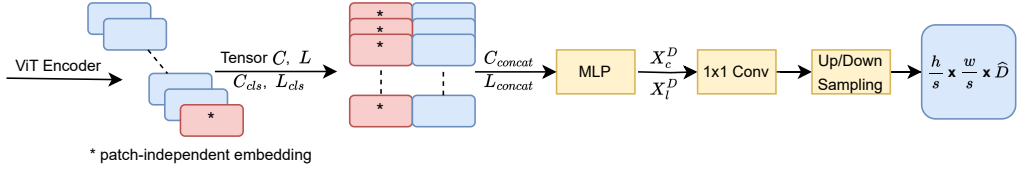


Fig. 2. Assemble architecture for each transformer decoder block, tokens of each layers are assembled to image-like representations of feature maps.

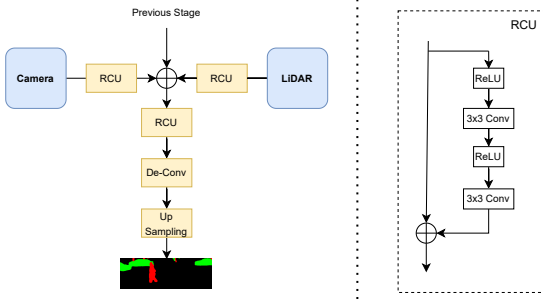


Fig. 3. Each fusion block receives data from the previous stage and integrates camera-LiDAR data coming from the ViT encoder. Each of this block has residual units, de-convolution, and up-sampling.

summed with the results from the previous fusion operation and then went through one additional RCU. We pass the output of the last fusion layer to a deconvolutional and up-sampling module to compute the final predicted segmentation. The fusion of the information coming from the LiDAR and the camera can happen in any of the fusion block as the connection weights are automatically learned in the network through error back-propagation. The idea of our multiple fusion blocks is to integrate the concept of late-fusion (as each fusion blocks is placed after each assemble block) and the concept of cross-fusion [24] as the connection with each feature map can happen in any of the fusion blocks with different weights. The network automatically learns to weight the best block to integrate tensor information coming from different sensors.

IV. DATASET CONFIGURATION

The primary purpose of this work is to compare the performance of the vision transformer and CNN backbones for semantic segmentation. Our previous work [13] successfully modeled and evaluated a ResNet50-based FCN to carry out camera-LiDAR fusion. In order to maintain an accordant experiment environment, we construct the input data based on Waymo dataset [54] to evaluate CLFT and other models.

Waymo dataset is recorded by multiple high-quality cameras and LiDAR sensors. The scenes of Waymo dataset span various illumination levels, weather conditions, and traffic scenarios. Therefore, as shown in Table I, we manually partitioned the data sequences into four subsets: light-dry, light-wet, dark-dry, and dark-wet. The ‘light’ and ‘dark’ indicate the relative

illumination conditions. The ‘dry’ and ‘wet’ represent the weather difference in precipitation.

TABLE I
AMOUNT OF THE FRAMES IN FOUR BROAD SUBSETS FOR WAYMO OPEN DATASET.

Light-Dry	Dark-Dry	Light-Wet	Dark-Wet
14940	1640	4520	900

We provide intersection over union (IoU) as the primary indication of model evaluation, with precision and recall values as supplementary information. Please note that the IoU is primarily used in object detection applications, in which the output is the bounding box around the object. Therefore, We modify the ordinary IoU algorithm to fit the multi-class pixel-wise semantic segmentation. The essential change is related to the ambiguous pixels (pixels have no valid labels, details in Section IV-B) that fall out of the class list. We assign these pixels as void and exclude them from the evaluation. The performance of networks is measured by the statistics of the number of pixels that have identical classes indicated in prediction and ground truth.

A. LiDAR Data Processing

The LiDAR readings reflect the object’s 3D geometric information in the real world. Coordinate values in three spatial channels contain features that can be exploited by neural networks. As a result, regarding camera-LiDAR fusion, it is common to extract and fuse multi-target features such as images’ color textures and point clouds’ location information, which is an approach namely as feature-level fusion [55].

We adopt feature-level fusion in this work. Thus, we project 3D LiDAR point clouds into the camera plane to create 2D occupancy grids in XY , YZ , and XZ planes. All the points in LiDAR point clouds are transformed and projected following Equation 2 and 3, respectively.

$$[x_t, y_t, z_t]^T = (r \ p \ y) \left([x_i, y_i, z_i]^T - [x_c, y_c, z_c]^T \right) \quad (2)$$

$$r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\rho) & \sin(\rho) \\ 0 & -\sin(\rho) & \cos(\rho) \end{bmatrix} \quad p = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad y = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In Equation 2, x_t , y_t , and z_t are the 3D point coordinates after transformation (in camera frame); r , p , and y represent the Euler rotation matrices to the camera frame with (ρ, θ, ϕ) representing the corresponding Euler angles. x_i , y_i , and z_i are the 3D point coordinates before transformation (in LiDAR

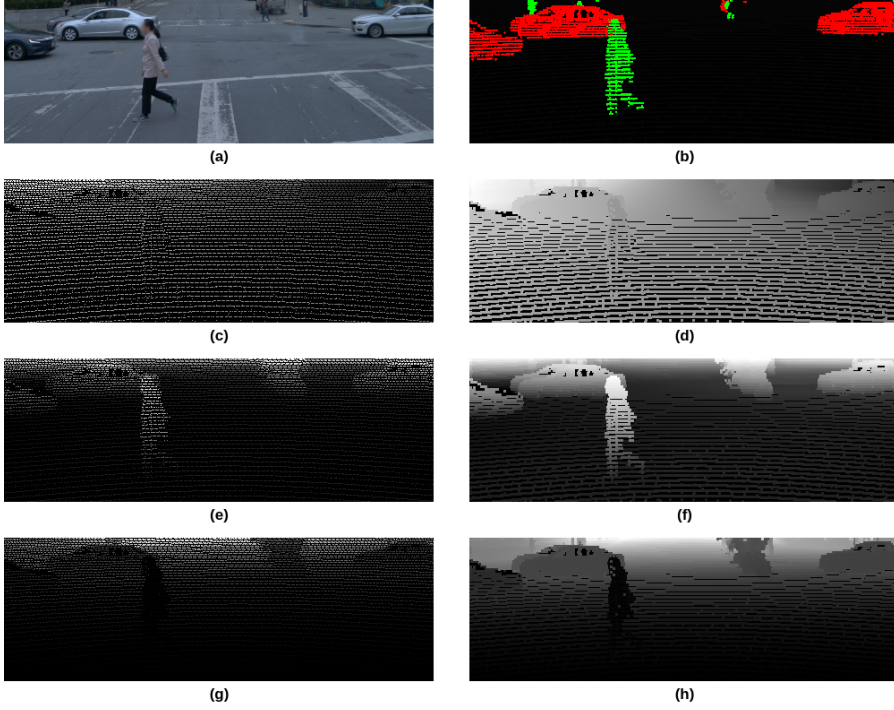


Fig. 4. Examples of camera image, semantic annotation mask, and pre-processing of LiDAR data. (a) is the RGB image. (b) illustrates the object semantic masks obtained from LiDAR ground truth bounding boxes. (c) (e) (g) are LiDAR projection images in X, Y, Z channels, respectively, while (d) (f) (h) are corresponding up-sampled dense images. Please note that for visualization purposes, the grayscale intensity in (c)-(h) is proportionally scaled based on the numerical 3D coordinate values of the LiDAR points.

frame); x_c , y_c , and z_c denote the camera frame location coordinates.

$$(u, v, 1)^T = \begin{bmatrix} f_x & 0 & \frac{w}{2} \\ 0 & f_y & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix} (x, y, z)^T \quad (3)$$

In Equation 3, u and v are column and row positions of the point in 2D image plane; f_x and f_y denote camera's horizontal and vertical focal length; w and h represent image resolution; x , y , and z are transformed 3D point coordinates (same as x_t , y_t , and z_t in Equation 2).

Algorithm 2 LiDAR points filtering and image pixel values population

Input: LiDAR point 3D coordinates L , projected LiDAR point coordinates P , image resolution w and h .

Output: LiDAR projection footprints XY , YZ , and ZX .

- 1: $idx = \text{argwhere}(P < \{w, h, +\infty\} \ \& \ P >= \{0, 0, 0\})$
- 2: $XY[w \times h] \leftarrow 0$
- 3: $YZ[w \times h] \leftarrow 0$
- 4: $XZ[w \times h] \leftarrow 0$
- 5: $XY[idx] = L[idx, 0]$
- 6: $YZ[idx] = L[idx, 1]$
- 7: $XZ[idx] = L[idx, 2]$

The operation after transforming and projecting the 3D point clouds into 2D images is filtering, which aims to discard all the points that fall out of the camera view. Waymo Open dataset is collected using five LiDAR and five camera sensors covering all vehicle directions. This work uses the top LiDAR's point clouds and the front camera's image data. As shown in Algorithm 2, three projection footprint images denoted as XY , YZ , and ZX are generated. The pixels corresponding to 3D points are assigned with x , y , and z coordinates, while the rest are populated with zero. At last, we up-sample the LiDAR images before feeding them to machine learning algorithms, as it is a common practice in LiDAR-based object detection research [56] [57]. Figure 4 (c)-(g) show the results of the procedure described in this subsection.

B. Object Semantic Masks

Ground truth annotations in Waymo dataset are represented by 2D and 3D bounding boxes, which correspond to camera and LiDAR data separately. There are three classes in image annotations: vehicles, pedestrians, and cyclists. Point clouds annotations have an extra class which is traffic signs. There are two obstacles when using Waymo's ground truth annotations in our networks.

Firstly, vision-transformer-based networks are well-known for requiring vast samples [12]. However, the cyclists and

TABLE II
PERFORMANCE COMPARISON OF CLFT-HYBRID VARIANT, CLFCN AND PANOPTIC SEGFORMER. BOLD INDICATES THE BEST VALUES IN EACH ROW PER CLASS. (IN PERCENTAGE UNIT) (C, L, AND C+L INDICATE CAMERA-ONLY, LIDAR-ONLY, AND FUSION MODALITIES, RESPECTIVELY)

	CLFT-Hybrid (C+L)		CLFCN (C)		CLFCN (L)		CLFCN (C+L)		Panoptic SegFormer (C)		Panoptic SegFormer (L)	
	Vehicle	Human	vehicle	Human	Vehicle	Human	Vehicle	Human	Vehicle	Human	Vehicle	Human
Light-Dry	91.35	66.04	88.08	55.57	88.58	53.04	91.07	62.50	85.89	61.02	66.41	40.78
Light-Wet	91.72	66.03	88.54	52.13	89.47	50.06	92.77	64.66	83.58	49.70	63.07	29.87
Dark-Dry	90.62	65.66	81.16	42.87	86.16	48.83	89.41	60.33	81.45	44.67	70.25	38.69
Dark-Wet	90.18	53.51	74.49	43.14	87.51	46.68	89.90	56.70	70.50	14.68	54.40	39.00

traffic signs are relatively rare-represented in the Waymo dataset. We notice our CLFT models struggle to learn and predict these two classes in experimental setting as they are less represented in the dataset. We assume that with additional data also traffic signs and cyclists can be properly classified. Therefore, we discard the traffic signs in this work and merge the cyclists and pedestrians as a new class of so-called human.

Secondly, our research aims for semantic segmentation, which requires annotations denoted as object contours. Since Waymo dataset labeled the object in LiDAR sensor readings as a 3D upright bounding box, we project all the points in the bounding box into the image plane by the same procedure described in Section IV-A. Figure 4 (b) shows an example of semantic masks for vehicle and human classes. Please note that a limitation of this approach is that some object pixels have no valid labels because there are no corresponding LiDAR points.

V. RESULTS

As mentioned in Section I, our CLFT is the first transformer-based model fusing the camera and LiDAR sensory data for semantic segmentation. The experiments in this work focus on the controlled benchmark comparisons in two aspects: i) neural network architecture, ii) input modality.

The FCN is believed to be the recent generation of deep learning methods with remarkable performance improvements and has become the mainstream for semantic segmentation [58]. Therefore, we choose the CLFCN [13], an FCN-based network that fuses camera and LiDAR data for semantic segmentation, as the reference to explore the advantages of transformer backbone. Since the transformer is well-known for its strengths in capturing global context and solving long-range dependencies, we expect our transformer-based model to outperform the FCN-based model in scenarios such as unevenly distributed datasets and underrepresented samples.

Only a few existing deep learning methods process the LiDAR input using the same principle as in this work: representing the 3D point clouds as 2D grid-based feature maps [14]. We compare the CLFT with the Panoptic SegFormer [41] that is also transformer-backbone to evaluate the significance of various input modalities. However, the Panoptic SegFormer is purely vision-based. We follow the procedures in Section IV to produce the point clouds projection images as LiDAR modality input for Panoptic SegFormer, but the camera-LiDAR-fusion mode is not directly applicable to Panoptic SegFormer. It is critical to maintain the same input data splits and configurations in experiments for all models.

A. Experimental setup

The details of the input dataset configuration are described in Section IV. The dataset splits for training, validation, and testing are 60%, 20%, and 20% of the total number of frames, respectively. The four data subsets, light-dry, light-wet, dark-dry, and dark-wet, are shuffled and mixed for training and validation but tested individually. We adopt the default hyperparameter configurations for CLFCN and Panoptic SegFormer in training. Please refer to authors' original work for details [41]. We employ weighted cross-entropy loss function and Adam optimization [59] for CLFT networks training. The transformer encoder of CLFT is initiated from ImageNet pre-trained weights. The transformer decoder and CLFCN's ResNet backbone are initiated randomly. The learning rate decay of CLFT networks training follows $l_i = l_0(\alpha^i)$, where l_0 is the initial learning rate, and α is 0.99. The batch size of CLFT networks training is set as 32 by default, but set as 24 for several experiments that exceed the memory limit, for example, the fusion mode of CLFT-large variant. Other hyperparameter settings can be found in the code we public. The transformer-based networks are trained using an NVIDIA A100 80GB GPU due to the large memory requirement of transformer networks. Relatively low-memory-required FCN training is executed on a desktop equipped NVIDIA RTX2070 Super GPU. The software environment of all experiments is Python3.9 and CUDA11.2. Please refer to our GitHub link for more details about the environment. Data normalization, augmentation and early stopping are also used to generate the models as in all most recent state-of-the-art methods.

B. Network performance and comparison

The main result of this work is reported in Table II and Table III. Values are shown as the IoU for the two interest classes, vehicle and human, in different modalities and weather scenarios. The modalities are indicated as C, L, and C+L, referring to the camera, LiDAR, and fusion, respectively.

As shown in Table II, the CLFT-hybrid variant outperforms the CLFCN and Panoptic SegFormer in all scenarios, demonstrating high segmentation capabilities over the same data. Specifically, in dry environmental conditions, CLFT-hybrid fusion modality archives 91% IoU for vehicles and 66% for humans, while CLFCN fusion modality has 90% for vehicles and 61% for humans. For single modality, Panoptic SegFormer achieves a similar performance of CLFCN for vehicle class but outperforms for human class with less fine-tuned works (61.02% against 55.57% in light-dry environment),

TABLE III
PERFORMANCE COMPARISON OF ALL CLFT VARIANTS, CLFCN, AND PANOPTIC SEGFORMER. (IN PERCENTAGE UNIT)(C, L, AND C+L INDICATE CAMERA-ONLY, LiDAR-ONLY, AND FUSION MODALITIES, RESPECTIVELY)

	VEHICLE			HUMAN		
	Precision	Recall	IoU	Precision	Recall	IoU
CLFT-Base (C+L)	93.63	95.95	90.12	71.97	79.47	60.68
CLFT-Large (C+L)	93.81	96.14	90.46	72.27	77.76	60.56
CLFT-Hybrid (C+L)	94.15	96.69	91.26	75.76	82.75	65.46
CLFCN (C+L)	93.17	97.67	91.19	65.63	92.89	62.51
Panoptic SegFormer (C)	94.82	88.43	84.40	81.11	63.78	55.55
Panoptic SegFormer (L)	89.57	70.85	65.48	67.84	46.85	38.29

which reinforces the transformer's strength regarding under-represented samples. The difference between our CLFT and other models is even more evident in challenging conditions such as dark and wet, where CLFT-hybrid performance drops by 1-2 percentage points while CLFCN and Panoptic SegFormer in single modalities drop by 5-10 percentage points. In these cases, fusion seems to play a pivotal role in CLFCN while showing only slight improvements in CLFT-hybrid, demonstrating the robustness of CLFT-hybrid in performing data fusion in all types of conditions.

The Panoptic SegFormer has obvious weak performance in LiDAR modality. This is because it is designed to process RGB visual input. We carry out the LiDAR processing separately to produce the camera-plane maps with 3D coordinate information; then we feed the maps to Panoptic SegFormer. The experiment results prove the necessity to integrate the LiDAR processing into the neural networks' architecture. Though CLFT-hybrid outperforms the CLFCN in fusion in most cases, it is essential to see that CLFCN models benefit more from the fusion, as the improvement from individual modalities seems to be higher, particularly in night conditions. On the other hand, our CLFT models already show high performance in challenging conditions with the fusion of camera and LiDAR data.

Table III summarizes the performance of CLFT variants, CLFCN, and Panoptic SegFormer. We present the precision, recall, and IoU for all models. In order to have a straightforward comparison, we combine four weather scenarios for performance evaluation. In all cases, the CLFT-hybrid variant performs better than the base and huge variants. This result is consistent with what Dosovitskiy et al. [12] reported in their ablation experiments, in which ResNet-based transformer variants outperform the variants that use patch-based embedding procedures. Though the CLFT-hybrid achieves the highest IoU score, CLFCN and Panoptic SegFormer have higher recall and precision results, respectively.

C. Ablation study

Table IV reports our results using camera (C), LiDAR (L), and fusion (C+L). According to our ablation study in Table IV, it is possible to conclude that fusion provides an improvement over single-modality networks.

One might note that results for the individual modalities, particularly LiDAR, show already performance over 90% (before fusion); this result is also in line with many other studies in the field, for instance, in [60] the authors reached

TABLE IV
ABLATION STUDY BASED ON CLFT-HYBRID VARIANT. (IN PERCENTAGE UNIT)

(C, L, and C+L indicate camera-only, LiDAR-only, and fusion modalities, respectively)

C	L	IoU		Precision		Recall	
		Vehicle	Human	Vehicle	Human	Vehicle	Human
All weather							
✓		91.16	64.38	93.86	73.33	96.88	84.05
	✓	91.19	65.17	93.93	72.89	96.85	84.19
✓	✓	91.26	65.46	94.15	75.76	96.69	82.75
Light-Dry							
✓		91.23	64.87	93.83	72.63	97.05	85.86
	✓	91.32	64.92	93.96	72.68	97.02	85.88
✓	✓	91.35	66.04	94.14	75.31	96.86	84.29
Light-Wet							
✓		91.67	64.87	94.52	76.49	96.82	81.36
	✓	91.52	64.28	94.40	74.43	96.78	82.49
✓	✓	91.72	66.03	94.69	78.27	96.96	80.84
Dark-Dry							
✓		90.51	65.62	93.15	74.30	96.96	84.66
	✓	90.47	65.18	93.27	74.30	96.96	84.16
✓	✓	90.62	65.66	93.38	77.39	96.68	81.25
Dark-Wet							
✓		89.62	52.46	93.60	70.00	95.70	67.69
	✓	89.74	49.95	93.69	67.28	95.51	65.97
✓	✓	90.18	53.51	94.40	68.68	95.29	70.79

over 90% IoU in the car class on the SemanticKitti dataset [61].

Inspecting the analysis on all-weather, one can see that CLFT-hybrid provides a small improvement (less than one percentage point in both classes). However, as by construction, the dataset split is strongly unbalanced (see Table I) toward light-dry scenario (roughly 68% of the total). The amount of light scenarios covers over 88% of the total number of frames. Clearly, the class that is better represented in the dataset affects the overall result the most.

To better appreciate the improvement in our studies, Table IV is also divided according to the data split in Table I. Under these conditions, it is possible to assert that fusion has a higher impact in dark scenarios, covering roughly 12% of the total number of frames in our dataset.

The unbalance of the dataset has an impact on both environment conditions and object classes, thus the vehicle class (with already over 90% accuracy) is less affected, while the human class shows better improvements, reaching around 2-4% in rainy conditions.

D. Inference time analysis

Table V presents an additional study on the inference time. In the experiments, we make the statistic of CUDA event time on NVIDIA A100 GPU for fusion modality of all models. All the models are set in evaluation mode for inference time calculation. We use the image in Figure 4 as input, first warm up the GPU with 2000 iterations, then calculate the mean time of the event stream for another 2000 iterations. The CPU and GPU are synchronized when recording timestamps. In general, FCN-based models have obvious advantages against the transformer-based models in terms of computational efficiency. The Panoptic SegFormer has the highest inference time among all models in experiments. It appears that the CLFCN is faster than our best-performing model, the CLFT-hybrid. However, this difference is only about 10ms per frame, which can be considered reasonable in a trade-off between performance and speed. For autonomous driving, where safety comes first, classification performance should always be considered a crucial parameter in the network design.

TABLE V
INFERENCE TIME COMPARISON OF ALL CLFT VARIANTS, CLFCN AND PANOPTIC SEGFORMER (IN MILLISECONDS UNIT)(C, L, AND C+L INDICATE CAMERA-ONLY, LiDAR-ONLY, AND FUSION MODALITIES, RESPECTIVELY)

NETWORK	MODALITY	TIME
CLFT-base	C+L	16.23
CLFT-Large		36.75
CLFT-Hybrid		25.69
CLFCN		15.94
Panoptic SegFormer	C	93.52
	L	93.45

E. Qualitative results

Figure 5 presents examples of segmented images from the Waymo dataset to appreciate the results of this work from a qualitative point of view. Following the above mentioned contribution of this work, the qualitative evaluation is also divided by network structure, weather and illumination conditions. The three CLFT variants, 'Base', 'Large', and 'Hybrid', are compared with the Panoptic SegFormer and CLFCN modalities. The segmentation results from models are overlaid to the camera images for comparison. The first row is the ground truth segmentation provided by the dataset. Please note that the annotations of the Waymo dataset are based on the LiDAR point clouds data, which is a common labeling strategy adopted by many famous multi-modal datasets for autonomous driving, including SemanticKitti and nuScenes [62] datasets. The LiDAR-points-based labeling strategy results the 2D semantic masks contain the pixels without valid label. Waymo dataset claimed to have the highest per-frame point clouds density among the SemanticKitti, nuScenes, and Argoverse [63] datasets, which is the reason why the Waymo dataset better fits for the evaluation of CLFT networks for 2D semantic segmentation tasks.

The qualitative results generally follow the same consistency as in numerical benchmarks. The CLFT-Hybrid variant discloses the most contextual details and its segmentation

results are more identical to ground truth than other networks, especially in challenging and under-represented environments. For example, the vehicles in night-dry (the third column) scenario, the CLFCN networks detect less details even with fine-tuning efforts, proves that the transformer is more effective than FCN in specific situations. Moreover, the single-modality segmentation results from Panoptic SegFormer and CLFCN networks show the necessities and advancements of multi-modal sensor fusion in autonomous driving.

VI. CONCLUSION

In this paper, we propose a transformer-based multimodal fusion method for semantic segmentation. Based on all the above cases, it is possible to say our CLFT model is one of the cutting-edge neural networks for 2D traffic object semantic segmentation. Specifically, the CLFT models benefit from the multimodal sensor fusion and transformer's multi-attention mechanism, make a significant improvement for under-represented samples (maximum 10 percent IoU increase for human class). However, it is worth mentioning that transformer networks intuitively require a large amount of data for training. In our experiments, light-wet and dark-wet subsets only take into account 12% of the total input data, which explains that the CLFCN model outperforms the CLFT-hybrid model in some cases in Table II.

This work proposes the adoption of a vision transformer's strategy to divide the input image into non-overlapping patches or extract feature patches from CNN feature maps. Intuitively, we project and up-sample LiDAR data to dense point clouds images, then design a double-direction network to assemble and cross-fuse the camera and LiDAR representations to achieve final segmentation. We maintain the same input dataset splits and configurations in all our experiments and successfully demonstrate the transformer's merit against the FCN regarding object segmentation tasks. Specifically, we classify the input data into sub-categories of different illumination and weather conditions dedicated to comprehensively evaluating the models. Similar to prior transformer works, we prove its potential on uneven-distributed datasets and under-represented samples. At last, we want to highlight that the initiation of CLFT lies on the progress to extend our framework that aims to cover all aspects of low-speed autonomous shuttles, including hardware configuration, dataset collection and post-processing for perception [17], validation [64], and path planning [65]. We develop the CLFT to be compatible with other systems in terms of environment, data formats, and operating platforms, which grants our work the advantages in scalability and practical application on real autonomous shuttles.

ACKNOWLEDGMENT

This research was funded by the European Union's Horizon 2020 Research and Innovation Programme, under the grant agreement No. 856602. This research was co-funded by the European Union under the project Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590).

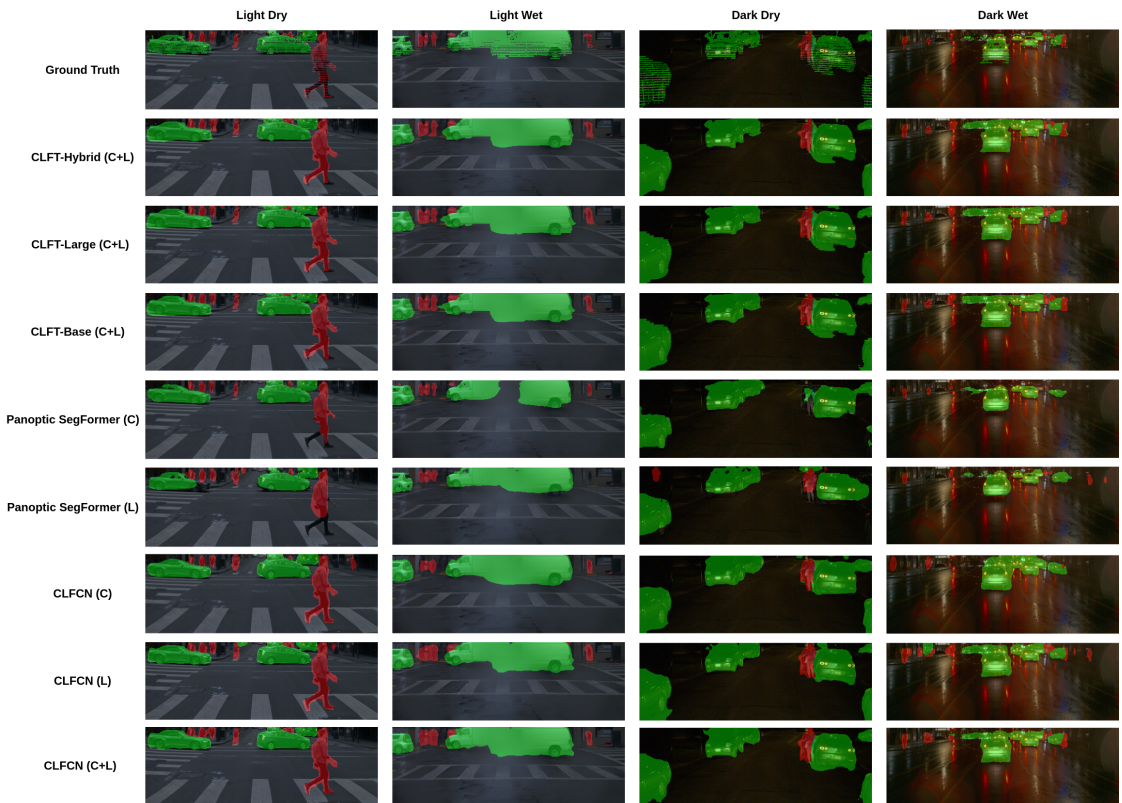


Fig. 5. Qualitative comparison of segmentation results between different models.

REFERENCES

- [1] L. Bartolomei, L. Teixeira, and M. Chli, "Perception-aware path planning for uavs using semantic segmentation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5808–5815.
- [2] D. K. Dewangan and S. P. Sahu, "Driving behavior analysis of intelligent vehicle system for lane detection using vision-sensor," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6367–6375, 2021.
- [3] J. Fritsch, T. Kühnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 1693–1700.
- [4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.
- [5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [6] D. Maturana and S. Scherer, "Voxelnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928.
- [7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.
- [8] A. Kanezaki, Y. Matsushita, and Y. Nishida, "Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5010–5019.
- [9] Y. Cui, R. Chen, W. Chu, L. Chen, D. Tian, Y. Li, and D. Cao, "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 722–739, 2021.
- [10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [11] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.
- [12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [13] J. Gu, M. Bellone, R. Sell, and A. Lind, "Object segmentation for autonomous driving using iseauto data," *Electronics*, vol. 11, no. 7, p. 1119, 2022.
- [14] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2020.
- [15] L. Chen, C. Sima, Y. Li, Z. Zheng, J. Xu, X. Geng, H. Li, C. He, J. Shi, Y. Qiao *et al.*, "Persformer: 3d lane detection via perspective transformer and the openlane benchmark," in *European Conference on Computer Vision*. Springer, 2022, pp. 550–567.
- [16] Y. Li, Y. Chen, X. Qi, Z. Li, J. Sun, and J. Jia, "Unifying voxel-based representation with transformer for 3d object detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18 442–18 455, 2022.
- [17] J. Gu, A. Lind, T. R. Chhetri, M. Bellone, and R. Sell, "End-to-

- end multimodal sensor dataset collection framework for autonomous vehicles," *Sensors*, vol. 23, no. 15, 2023.
- [18] J. Zhong, Z. Liu, and X. Chen, "Transformer-based models and hardware acceleration analysis in autonomous driving: A survey," 2023.
- [19] H. Thisanke, C. Deshan, K. Chamith, S. Seneviratne, R. Vidanaarachchi, and D. Herath, "Semantic segmentation using vision transformers: A survey," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 106669, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0952197623008539>
- [20] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3288–3295.
- [21] X. Cheng, P. Wang, C. Guan, and R. Yang, "Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 10615–10622.
- [22] F. Wulf, B. Schäufele, O. Sawade, D. Becker, B. Henke, and I. Radusch, "Early fusion of camera and lidar for robust road detection based on u-net fcn," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1426–1431.
- [23] J.-S. Lee and T.-H. Park, "Fast road detection by cnn-based camera–lidar fusion and spherical coordinate transformation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5802–5810, 2021.
- [24] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "Lidar–camera fusion for road detection using fully convolutional neural networks," *Robotics and Autonomous Systems*, vol. 111, pp. 125–131, 2019.
- [25] J. Dou, J. Xue, and J. Fang, "Seg-voxelnet for 3d vehicle detection from rgb and lidar data," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4362–4368.
- [26] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4604–4612.
- [27] 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*, 2018, pp. 4490–4499.
- [28] L. Caltagirone, M. Bellone, L. Svensson, M. Wahde, and R. Sell, "Lidar–camera semi-supervised learning for semantic segmentation," *Sensors*, vol. 21, no. 14, p. 4813, 2021.
- [29] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, and Z. Tang, "Befusion: A simple and robust lidar-camera fusion framework," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10421–10434, 2022.
- [30] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *International Journal of Computer Vision*, vol. 128, no. 5, pp. 1239–1285, 2020.
- [31] M. Jaritz, T.-H. Vu, R. d. Charette, E. Wirbel, and P. Pérez, "xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12605–12614.
- [32] S. Gu, T. Lu, Y. Zhang, J. M. Alvarez, J. Yang, and H. Kong, "3-d lidar+ monocular camera: An inverse-depth-induced fusion framework for urban road detection," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 351–360, 2018.
- [33] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy lidar completion with rgb guidance and uncertainty," in *2019 16th International Conference on Machine Vision Applications (MVA)*, 2019, pp. 1–6.
- [34] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 244–253.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [36] K. El Madawi, H. Rashed, A. El Sallab, O. Nasr, H. Kamel, and S. Yogamani, "Rgb and lidar fusion based 3d semantic segmentation for autonomous driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 7–12.
- [37] X. Zhao, Z. Liu, R. Hu, and K. Huang, "3d object detection using scale invariant and feature reweighting networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 9267–9274.
- [38] L. Peng, Z. Chen, Z. Fu, P. Liang, and E. Cheng, "Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 5935–5943.
- [39] R. Liu, Z. Yuan, T. Liu, and Z. Xiong, "End-to-end lane shape prediction with transformers," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3694–3702.
- [40] Y. Bai, Z. Chen, Z. Fu, L. Peng, P. Liang, and E. Cheng, "Curveformer: 3d lane detection by curve propagation with curve queries and attention," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7062–7068.
- [41] Z. Li, W. Wang, E. Xie, Z. Yu, A. Anandkumar, J. M. Alvarez, P. Luo, and T. Lu, "Panoptic segformer: Delving deeper into panoptic segmentation with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1280–1289.
- [42] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," in *Conference on Robot Learning*. PMLR, 2022, pp. 180–191.
- [43] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.
- [44] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, "Futr3d: A unified sensor fusion framework for 3d detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 172–181.
- [45] Y. Liu, T. Wang, X. Zhang, and J. Sun, "Petr: Position embedding transformation for multi-view 3d object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 531–548.
- [46] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Befformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *European conference on computer vision*. Springer, 2022, pp. 1–18.
- [47] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-perspective view for vision-based 3d semantic occupancy prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9223–9232.
- [48] Y. Li, Z. Yu, C. Choy, C. Xiao, J. M. Alvarez, S. Fidler, C. Feng, and A. Anandkumar, "Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9087–9098.
- [49] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12179–12188.
- [50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [51] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [52] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.
- [53] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.
- [54] 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*, 2020, pp. 2446–2454.
- [55] K. Banerjee, D. Notz, J. Windelen, S. Gavarraju, and M. He, "Online camera lidar fusion and object detection on hybrid data for autonomous driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1632–1638.
- [56] C. Premebida, J. Carreira, J. Batista, and U. Nunes, "Pedestrian detection combining rgb and dense lidar data," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4112–4117.
- [57] J. Schlosser, C. K. Chow, and Z. Kira, "Fusing lidar and images for pedestrian detection using convolutional neural networks," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2198–2205.
- [58] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing*, vol. 493, pp. 626–646, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S09525231222000054>
- [59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:6628106>

- [60] X. Zhu, H. Zhou, T. Wang, F. Hong, W. Li, Y. Ma, H. Li, R. Yang, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar-based perception," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6807–6822, 2021.
- [61] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.
- [62] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.
- [63] 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*, 2019, pp. 8748–8757.
- [64] M. Malayjerdi, Q. A. Goss, M. İ. Akbaş, R. Sell, and M. Bellone, "A two-layered approach for the validation of an operational autonomous shuttle," *IEEE Access*, 2023.
- [65] E. Malayjerdi, R. Sell, M. Malayjerdi, A. Udal, and M. Bellone, "Practical path planning techniques in overtaking for autonomous shuttles," *Journal of Field Robotics*, vol. 39, no. 4, pp. 410–425, 2022.



Raivo Sell received his Ph.D. degree in Product Development from Tallinn University of Technology in 2007 and currently working as a professor of robotics at TalTech. His research interest covers mobile robotics and self-driving vehicles, smart city, and early design issues of mechatronic system design. He is running the Autonomous Vehicles research group at TalTech as a research group leader with a strong experience and research background in mobile robotics and self-driving vehicles. Raivo Sell has been a visiting researcher at ETH Zürich, Aalto University, and most recently at Florida Polytechnic University in the US, awarded as a Chart Engineer and International Engineering Educator.



Gu Junyi received the B.S. degree in School of Optical-Electrical and Computer Engineering from the University of Shanghai for Science and Technology, Shanghai, China, in 2017. He received the M.S. degree in the Institute of Technology from the University of Tartu, Tartu, Estonia, in 2020. He is currently pursuing the Ph.D. degree at the Department of Mechanical and Industrial Engineering, Tallinn University of Technology, Tallinn, Estonia. His research interests include multi-sensor fusion, semantic segmentation, artificial intelligence, and

autonomous driving.



Mauro Bellone received his M.S. degree in Automation Engineering from the University of Salento, Lecce, Italy, where he received his Ph.D. in Mechanical and Industrial Engineering in 2014. His interests comprise mobile robotics, autonomous vehicles, energy, computer vision, and control systems. His research focuses on advanced sensory perception for mobile robotics and artificial intelligence. From 2015 to 2020, he worked with the applied artificial intelligence research group of Chalmers University of Technology, where he actively contributed to

several autonomous driving projects. In 2021, he was appointed as an adjunct professor at Tallinn University of technology, supporting the research team in the area of smart transportation systems.



Tomáš Pivoňka has received his master's degree in robotics at Faculty of Electrical Engineering of Czech Technical University in Prague (CTU) in 2018, where he continues in Ph.D. study program Artificial Intelligence and Biocybernetics. He works at the Intelligent and Mobile Robotics Group of Czech Institute of Informatics, Robotics and Cybernetics, CTU. His main research interests are visual localization, navigation, and computer vision.

Curriculum Vitae

1. Personal data

Name	Junyi Gu
Date and place of birth	29 August 1995 Shanxi, China
Nationality	Chinese

2. Contact information

Phone	+372 56785073
E-mail	claude.gujunyi@gmail.com

3. Education

2020–2024 Tallinn University of Technology,
Department of Mechanical and Industrial Engineering,
PhD

2018–2020 University of Tartu,
Institute of Technology,
Robotics and Computer Engineering, MSc

2013–2017 University of Shanghai for Science and Technology,
School of Optical-Electrical and Computer Engineering,
Measurement and Control technology, BSc

4. Language competence

Chinese	Native
English	Bilingual Proficiency

5. Defended theses

- 2020, Towards Faster Masking of Dynamic Objects for Visual Simultaneous Localization and Mapping, M.Sc., supervisor Prof. Amnir Hadachi, co-supervisor Artjom Lind, University of Tartu, Institute of Technology.
- 2017, The Design of the Control System of Automatic Robot based on PIC18, BEng, supervisor Prof. Yuming Shen, University of Shanghai for Science and Technology, School of Optical-Electrical and Computer Engineering.

6. Field of research

- ETIS RESEARCH FIELD: 4. Natural Sciences and Engineering; 4.13. Mechanical Engineering, Automation Technology, and Manufacturing Technology

- CERCS RESEARCH FIELD: T125 Automation, robotics, control engineering
- SPECIFICATION: Autonomous systems and self-driving vehicles; Techniques and methods for the sensor fusion

7. Scientific work

Journal Articles

1. J. Gu, M. Bellone, T. Pivoňka and R. Sell, "CLFT: Camera-LiDAR Fusion Transformer for Semantic Segmentation in Autonomous Driving," in IEEE Transactions on Intelligent Vehicles IF:14, doi: 10.1109/TIV.2024.3454971.
2. J. Gu, A. Lind, T. R. Chhetri, M. Bellone, and R. Sell, "End-to-end multimodal sensor dataset collection framework for autonomous vehicles," Sensors IF:3.9, vol. 23, no. 15, 2023. doi: 10.3390/s23156783
3. J. Gu, M. Bellone, R. Sell, and A. Lind, "Object segmentation for autonomous driving using iseauto data," Electronics IF:2.9, vol. 11, no. 7, 2022, issn: 2079-9292. doi: 10.3390/electronics11071119

Conference Proceedings

1. H. Pikner, R. Sell, and J. Gu, "Robot bus low-level control system transformation to an open-source solution," AIP Conference Proceedings, May 2023. doi: <https://doi.org/10.1063/5.0189277>
2. J. Gu and T. R. Chhetri, "Range sensor overview and blind-zone reduction of autonomous vehicle shuttles," 1, vol. 1140, IOP Publishing, May 2021, p. 012 006. doi: 10.1088/1757-899X/1140/1/012006

Elulookirjeldus

1. Isikuandmed

Nimi	Junyi Gu
Sünniaeg ja -koht	29.08.1995, Shanxi, Hiina
Kodakondsus	Hiina

2. Kontaktandmed

Aadress	Tallinna Tehnikaülikool, Mehaanika ja tööstustehnika instituut, Ehitajate tee 5, 19086 Tallinn, Estonia
Telefon	+372 56785073
E-post	claude.gujunyi@gmail.com

3. Haridus

2020–2024 Tallinna Tehnikaülikool, Inseneriteaduskond, Tootearendus ja robotika, doktoriõpe

2018–2020 Tartu ülikool, Tehnoloogiainstituut, Arvutitehnika ja robotika, MSc

2013–2017 Shanghai Teadus ja tehnikaülikool, Optika-elektrotehnika ja arvutitehnika teaduskond, Mõõtmis- ja juhtimistehnoloogia, BSc

4. Keelteoskus

hiina keel	emakeel
inglise keel	kõrgtase

5. Kaitstud lõputööd

- 2020, Towards Faster Masking of Dynamic Objects for Visual Simultaneous Localization and Mapping, magistratöö, juhendaja prof. Amnir Hadachi, kaasjuhendaja Artjom Lind, Tartu ülikool.
- 2017, The Design of the Control System of Automatic Robot based on PIC18, bakalaureusetöö, juhendaja prof. Yuming Shen, Shanghai Teaduse ja Tehnoloogia ülikool.

6. Teadustöö põhisuunad

- ETIS VALDKOND: 4. Loodusteadused ja tehnika; 4.13. Mehhanotehnika, automaatika, tööstustehnoloogia
- CERCS VALDKOND: T125 Automatiseerimine, robotika, juhtimistehnika

- TÄPSUSTUS: Autonoomsed süsteemid ja isejuhtivad sõidukid; Andurite andmetöötlus ja objektituvastus

7. Teadustegevus

Teadusartiklite, konverentsiteeside ja konverentsiettekannete loetelu on toodud ingliskeelse elulookirjelduse juures.

ISSN 2585-6901 (PDF)
ISBN 978-9916-80-211-3 (PDF)